密级：飞创秘密



飞创X-One2.0极速交易系统

交易API接口文档

V0.1.1

2019-07-08

大连飞创信息技术有限公司

（版权所有，翻版必究）

**文档信息**

|  |  |
| --- | --- |
| 项目名称 | 飞创X-One 2.0极速交易系统项目 |
| 文档名称 | <飞创X-One 2.0极速交易系统应用程序接口说明文档> |
| 版本号 | <v0.1> |
| 作者 | 飞创X-One 2.0极速交易系统项目组 |

**文档修订历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **日期** | **修订内容** | **修订者** | **备注** |
| 0.1 | 2019/7/8 | 接口整理 | 飞创X-One 2.0极速交易系统开发组 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1. 简介 5](#_Toc17221603)

[2. 极速交易系统与Xone的差异点 5](#_Toc17221604)

[3. 接口类库文件说明 5](#_Toc17221605)

[4. 系统架构 6](#_Toc17221606)

[4.1 接口与其他系统关系 6](#_Toc17221611)

[4.2 通讯模式 6](#_Toc17221612)

[4.3 接口模式 6](#_Toc17221613)

[5. 接口开发规范 8](#_Toc17221614)

[5.1 命名空间 8](#_Toc17221616)

[5.2 开发流程 8](#_Toc17221617)

[5.3 CX1FtdcTraderSpi接口 8](#_Toc17221618)

[5.4 RequestID字段 8](#_Toc17221619)

[5.5 LocalOrderID本地委托号字段 9](#_Toc17221620)

[5.6 X1OrderID柜台委托号字段 9](#_Toc17221621)

[5.7 连接断开与重连 9](#_Toc17221622)

[5.8 pErrorInfo异常信息数据结构 9](#_Toc17221623)

[6. 业务与接口对照表 10](#_Toc17221624)

[7. CX1FtdcTraderAPI使用参考手册 10](#_Toc17221625)

[7.1. CX1FtdcTraderApi接口 10](#_Toc17221633)

[7.1.1. CreateCX1FtdcTraderApi方法 11](#_Toc17221634)

[7.1.2. Init方法 11](#_Toc17221635)

[7.1.3. SetLogLevel方法 11](#_Toc17221636)

[7.1.4. Join 方法 12](#_Toc17221638)

[7.1.5. Release方法 12](#_Toc17221639)

[7.1.6. SubscribePrivateTopic方法 12](#_Toc17221640)

[7.1.7. ReqUserLogin方法 13](#_Toc17221641)

[7.1.8. ReqUserLogout方法 13](#_Toc17221642)

[7.1.9. ReqInsertOrder方法 14](#_Toc17221643)

[7.1.10. ReqCancelOrder方法 16](#_Toc17221644)

[7.1.11. ReqQryOrderInfo方法 16](#_Toc17221645)

[7.1.12. ReqQryPosition方法 17](#_Toc17221646)

[7.1.13. ReqQryPositionDetail方法 18](#_Toc17221647)

[7.1.14. ReqQryMatchInfo方法 19](#_Toc17221648)

[7.1.15. ReqQryCustomerCapital方法 20](#_Toc17221649)

[7.1.16. ReqQryExchangeStatus方法 20](#_Toc17221650)

[7.1.17. ReqQryExchangeInstrument方法 21](#_Toc17221651)

[7.1.18. ReqWarmInsertOrder方法 21](#_Toc17221652)

[7.2. CX1FtdcTraderSpi接口 22](#_Toc17221653)

[7.2.1. OnFrontConnected方法 22](#_Toc17221654)

[7.2.2. OnFrontDisconnected方法 22](#_Toc17221655)

[7.2.3. OnRspUserLogin方法 22](#_Toc17221656)

[7.2.4. OnRspUserLogout方法 23](#_Toc17221657)

[7.2.5. OnRspInsertOrder方法 24](#_Toc17221658)

[7.2.6. OnRspCancelOrder方法 25](#_Toc17221660)

[7.2.7. OnRtnErrorMsg方法 26](#_Toc17221661)

[7.2.8. OnRtnMatchedInfo方法 26](#_Toc17221662)

[7.2.9. OnRtnOrder方法 27](#_Toc17221664)

[7.2.10. OnRtnCancelOrder方法 28](#_Toc17221665)

[7.2.11. OnRspQryOrderInfo方法 29](#_Toc17221666)

[7.2.12. OnRspQryPosition方法 31](#_Toc17221667)

[7.2.13. OnRspQryPositionDetail方法 32](#_Toc17221668)

[7.2.14. OnRspQryMatchInfo方法 33](#_Toc17221669)

[7.2.15. OnRspCustomerCapital方法 34](#_Toc17221670)

[7.2.16. OnRspQryExchangeStatus方法 35](#_Toc17221671)

[7.2.17. OnRspQryExchangeInstrument方法 36](#_Toc17221674)

[8. 附录一(错误码表) 37](#_Toc17221675)

# 简介

飞创X-One 2.0极速交易系统应用程序接口（API）是一个基于C++的类库，通过使用和扩展类库提供的功能来实现相关的交易功能，包括资金账户登录，报单、撤单、等功能。

本文档的主要内容包括：

* 接口类库文件说明
* 接口的系统架构
* 接口开发规范
* 业务与接口的对照表
* 接口定义

# 极速交易系统与Xone的差异点

1、本系统不支持【本地委托号LocalOrderID + 会话号session\_id】撤单方式

2、本系统的所有查询接口的回调函数有三个参数：第一个参数为响应的具体数据，第二个参数为处理结果，表明本次请求的处理结果是成功还是失败，第三个参数为标志位，标志位为1表示最后一条数据。当失败时，第二个参数不为nullptr，第一个参数为nullptr；当成功时，第二个参数为nullptr，第一个参数不为nullptr，有具体的返回结构。

详情请参考7.2节中查询接口回调函数

3、本系统的回调接口实现与xone存在一定的差异，该版本中的结构体中有些字段没有实现，详情请参考第7点接口使用参考手册

4、本系统API与交易系统直连，不需要通过前置

5、本系统有新增错误码

# 接口类库文件说明

接口类库包含如下文件：

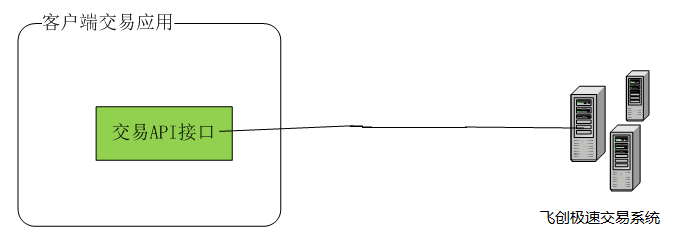
|  |  |  |
| --- | --- | --- |
| **文件名** | **版本** | **文件描述** |
| X1FtdcApiDataType.h | V0.1 | 定义接口所需的数据类型的头文件 |
| X1FtdcApiStruct.h | V0.1 | 定义接口所需的数据接口的头文件 |
| X1FtdcTraderApi.h | V0.1 | 定义交易接口的头文件 |

最低环境要求：Centos 7.2、编译器版本gcc4.8.5、c++11

# 系统架构



## 接口与其他系统关系



**说明：**

* 本API接口在客户端交易应用（简称客户端）与飞创X-One 2.0极速交易系统的通信时使用；
* 客户端可以是程序化交易系统；
* 交易API接口负责与飞创X-One 2.0极速交易系统连接；

## 通讯模式

API与飞创X-One 2.0极速交易系统的通讯协议是建立在TCP协议上的通讯协议，一旦建立TCP连接，双方将保持该连接（长连接）

* 对话通讯模式：客户端主动发起请求，飞创X-One 2.0极速交易系统接收请求，并立即将应答返回给客户端；
* 私有通讯模式：由飞创X-One 2.0极速交易系统主动发起，通过TCP长连接向客户端发送特定信息，比如委托回报，成交回报等等；

## 接口模式

API封装了两个接口，分别为CX1FtdcTraderApi和CX1FtdcTraderSpi，两个接口对API与飞创X-One 2.0极速交易系统的通信及通信报文协议进行了封装，方便客户端应用程序的开发。客户端应用程序可以通过CX1FtdcTraderApi的接口发出操作请求，通过继承CX1FtdcTraderSpi并重载回调函数来处理后台服务的响应。

#### 5.3.1 对话流和查询流编程接口通常如下：

请求：int CX1FtdcTraderApi::ReqXXX(CX1FtdcxxxField \* pReqXXX)

响应：void CX1FtdcTraderSpi::OnRspXXX(CX1FtdcxxxField \* pRspXXX,

CX1FtdcRspErrorField \* pErrorInfo,

bool bIsLast);

其中请求接口的参数内容不能为空，每次请求时，需要检查接口的返回值是否为0，每个结构体里包含了一个RequestID字段，当请求查询信息返回时，可以通过该字段将请求与响应对应起来。

当API收到后台服务应答时，CX1FtdcTraderSpi的回调函数将被调用，即会调用客户端继承并实现的Spi函数。如果响应数据不止一个，则回调函数会被多次调用。

登录登出、下单撤单接口的回调函数有两个参数：第一个参数为响应的具体数据，第二个参数为处理结果，表明本次请求的处理结果是成功还是失败。当失败时该值不为nullptr，所以当Spi函数被调用时，首先应该检测该值是否为nullptr，并从中获取错误ID和错误信息。

查询接口的回调函数有三个参数：第一个参数为响应的具体数据，第二个参数为处理结果，表明本次请求的处理结果是成功还是失败，第三个参数为标志位，标志位为1表示最后一条数据。当失败时，第二个参数不为nullptr，第一个参数为nullptr；当成功时，第二个参数为nullptr，第一个参数不为nullptr，有具体的返回结构。

#### 5.3.2私有流编程接口

私有流中的数据中有客户的私有信息，如委托回报、成交回报等

通常私有流接收回报的编程如下：

void CX1FtdcTraderSpi::OnRtnXXX(CX1FtdcxxxField \* pRtnXXX);

void CX1FtdcTraderSpi::OnRtnErrorMsg(CX1FtdcErrorRtnField \* pErrorInfo);

当收到交易所的确认时，该类接口将被调用，如委托回报，回调参数为具体内容。

#### 5.3.3 线程说明

与用户有关的API线程说明如下：

1. 发包线程：实例化CX1FtdcTraderApi(用户在自己的逻辑线程中进行)
2. 收包线程：CX1FtdcTraderApi对象初始化后(调用Init函数)，API后台会自动创建回报处理线程，当收到用户请求对应的回报时，该线程会调用CX1FtdcTraderSpi子类对象中重载的回调函数。
3. 该版本不建议多线程调用接口

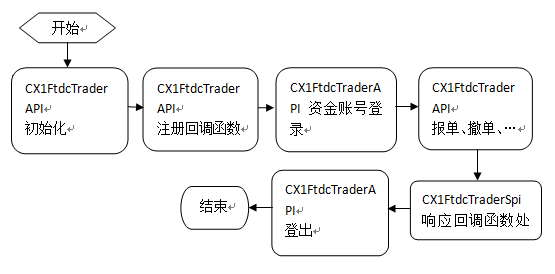
# 接口开发规范



## 命名空间

客户端交易API方法的命名空间为“x1ftdcapi”，使用该接口时，请添加命名空间的引用：using namespace x1ftdcapi;

## 开发流程



## CX1FtdcTraderSpi接口

* CX1FtdcTraderSpi接口定义了事件通知接口，开发人员必须正确继承并实现CX1FtdcTraderSpi接口，编写对应的事件处理方法。注意，在Spi函数接口中有CX1FtdcRspErrorField的参数时，若该参数不为nullptr表示有错误产生。

## RequestID字段

* 接口定义了每次请求与响应报文的唯一标识：RequestID字段，客户端应用通过该字段，将请求报文与响应报文对应起来，该id由客户应用产生并维护。

## LocalOrderID本地委托号字段

* 本地委托号唯一标识了一个会话中的每次报单，以及关联该报单的后续委托、成交、撤单等回报信息。在一个资金账号的一次登录会话范围内，本地委托号不能重复，且必须是大于等于1的数字，可选的处理方式：每次报单在上一本地委托号基础上加1。注意，本地委托号应该每次递增,如果输入负数会被强转为无符号4字节整数。

## X1OrderID柜台委托号字段

* 柜台委托号在飞创X-One 2.0极速交易系统里唯一标识了一笔报单，该笔报单在每个交易日里不会重复，且一般总是从1开始递增的（0为无效的柜台委托号）。该委托号在委托回报成功时，会返回。

## 连接断开与重连

* 当TCP连接断开时，将使用OnFrontDisconnected（）回调方法通知客户端；此时客户端API也会自动检测与飞创X-One 2.0极速交易系统之间的连接，当网络可用，将自动建立连接，并使用OnFrontConnected（）方法通知客户端，客户端可以在该方法中完成登录请求任务。

## pErrorInfo异常信息数据结构

* pErrorInfo:返回异常信息的地址。在Spi函数进入时，若该指针不为nullptr，表示有错误，此时应处理错误信息，对于正确的调用，该变量值为nullptr。一般情况下，程序可先判断该变量是否为nullptr，再根据情况进行后续处理。

|  |
| --- |
| struct CX1FtdcRspErrorField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcSessionIDType SessionID; //会话标识  TX1FtdcAccountIDType AccountID; //资金账号  TX1FtdcErrorIDType ErrorID; //错误ID  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcErrorMsgInfoType ErrorMsg; //错误信息  TX1FtdcInstrumentIDType InstrumentID; //合约代码  }; |

* 错误ID和错误信息对应关系参见[附录](#_附录一(错误码表))

# 业务与接口对照表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **业务类型** | **业务** | **请求接口** | **响应接口** | **数据流** |
| 登录 | 登录 | CX1FtdcTraderApi::ReqUserLogin | CX1FtdcTraderSpi::OnRspUserLogin | 对话流 |
| 登出 | CX1FtdcTraderApi::ReqUserLogout | CX1FtdcTraderSpi::OnRspUserLogout | 对话流 |
| 交易 | 下单 | CX1FtdcTraderApi:: ReqInsertOrder | CX1FtdcTraderSpi::OnRspInsertOrder | 对话流 |
|  | 撤单 | CX1FtdcTraderApi::ReqCancelOrder | CX1FtdcTraderSpi:: OnRspCancelOrder | 对话流 |
|  | Warm包 | CX1FtdcTraderApi::ReqWarmInsertOrder | N/A |  |
| 私有回报 | 错误回报 | N/A | CX1FtdcTraderSpi::OnRtnErrorMsg | 私有流 |
| 成交回报 | N/A | CX1FtdcTraderSpi::OnRtnMatchedInfo | 私有流 |
| 委托回报 | N/A | CX1FtdcTraderSpi:: OnRtnOrder | 私有流 |
| 撤单回报 | N/A | CX1FtdcTraderSpi::OnRtnCancelOrder | 私有流 |
| 查询 | 委托查询 | CX1FtdcTraderApi::ReqQryOrderInfo | CX1FtdcTraderSpi::OnRspQryOrderInfo | 查询流 |
| 成交查询 | CX1FtdcTraderApi::ReqQryMatchInfo | CX1FtdcTraderSpi::OnRspQryMatchInfo | 查询流 |
| 持仓查询 | CX1FtdcTraderApi:: ReqQryPosition | CX1FtdcTraderSpi::OnRspQryPosition | 查询流 |
| 持仓明细查询 | CX1FtdcTraderApi::ReqQryPositionDetail | CX1FtdcTraderSpi::OnRspQryPositionDetail | 查询流 |
| 资金查询 | CX1FtdcTraderApi:: ReqQryCustomerCapital | CX1FtdcTraderSpi::OnRspCustomerCapital | 查询流 |
| 交易所合约查询 | CX1FtdcTraderApi:: ReqQryExchangeInstrument | CX1FtdcTraderSpi::OnRspQryExchangeInstrument | 查询流 |
|  | 交易所状态查询 | CX1FtdcTraderApi::ReqQryExchangeStatus | CX1FtdcTraderSpi::OnRspQryExchangeStatus | 查询流 |
|  | 指定合约查询 | CX1FtdcTraderApi:: ReqQrySpecifyInstrument | CX1FtdcTraderSpi::OnRspQrySpecifyInstrument | 查询流 |
| 其他 | 设置日志等级 | SetLogLevel | N/A |  |
| 密码修改 | CX1FtdcTraderApiImp::ReqResetPassword | CX1FtdcTraderSpi:: OnRspResetPassword | 对话流 |

# CX1FtdcTraderAPI使用参考手册

约定： m/M 必填

NM 不需要必填

BLK/BLANK 填空格

ZERO 填0

N/A 不关心，不修改，不使用



## CX1FtdcTraderApi接口

**注意事项：**

1）用户调用接口前，需要进行memset清空结构体内容，再填写需要的字段信息。

3）买卖，开平、投保、订单类型、订单属性这些字段按照头文件指定数值输入，输入其他值也会判错或者被转换为不确定数值。

注：本节所示接口的委托响应均有可能晚于委托回报返回。

### CreateCX1FtdcTraderApi方法

产生一个CX1FtdcTraderApi实例

**函数原型：**

|  |
| --- |
| static CX1FtdcTraderApi \*CreateCX1FtdcTraderApi(); |

**返回值：**

返回一个指向CX1FtdcTraderApi实例的指针。

### Init方法

该方法会和交易服务器建立连接，并启动一个接收线程，同时该方法注册一个回调函数集。

**函数原型：**

|  |
| --- |
| int Init( char \*pszFrontAddr, CX1FtdcTraderSpi \*pSpi, int output\_core, int input\_core); |

**参数：**

pszFrontAddr：交易系统交易链接地址。

采用如下的格式："tcp://172.16.0.31:5556"的形式。

pSpi：指向回调函数集的指针。

output\_core：输出线程绑定的cpu core id，-1表示不绑定

input\_core：输入线程绑定的cpu core id，-1表示不绑定

**返回值：**

参见附录一(错误码表)

**说明：**

**Init失败原因可能：**

1. 填写的addr格式不正确或者addr中的ip地址及端口不正确。
2. 网络问题，可telnet连接ip及port，检查是否畅通

### SetLogLevel方法

在init成功后，可以调用该接口设置api的log输出等级和刷新级别（异步日志）

**函数原型：**

|  |
| --- |
| void SetLogLevel(int log\_level = 3, int flush\_level = 3); |

**参数：**

log\_level：设置log的输出等级，默认值为3-WARN

flush\_level：设置日志立即写入文件等级，默认值为3-WARN。

建议使用默认值。

**返回值：**

### 无Join 方法

等待接口线程结束运行

**函数原型：**

|  |
| --- |
| int Join(void); |

**返回值：**

参见附录一(错误码表)

### Release方法

退出API各线程，释放API的各项资源

**函数原型：**

|  |
| --- |
| voidRelease(void); |

**返回值：**

无

**说明：**

当调用CreateCX1FtdcTraderApi生成一个API实例后，退出时必须调用Release接口，否则会造成资源泄漏。就像C中malloc了一块内存，需要free一样。

### SubscribePrivateTopic方法

订阅私有流，在Init方法调用之后，ReqUserLogin方法调用之前调用。若不调用该接口，则收不到私有流

**函数原型：**

|  |
| --- |
| void SubscribePrivateTopic(int priflow\_req\_flag, unsigned int pri\_no); |

**返回值：**

无

**说明：**

priflow\_req\_flag：私有流拉取方式，取值范围如下：

X1\_PrivateFlow\_Req\_Quick 0：只传送登录后私有流的内容

X1\_PrivateFlow\_Req\_Restart 1：从本交易日开始重传

X1\_PrivateFlow\_Req\_Resume 2：从上次收到的续传

X1\_PrivateFlow\_Req\_Specify 3：从指定编号开始续传

pri\_no：指定的私有流编号。当priflow\_req\_flag取值3时，该参数才有效，其他情况，传入0值即可。

**注意：**

当用户在同一个会话中拉取私有流过程中，如果收到OnFrontDisconnect断开连接的消息，Reason为查询连接断开，错误码为165。即此时查询连接断开，用户在该会话中可以正常进行交易报单，但是收不到回报。

建议用户检查网络，重新登录。

### ReqUserLogin方法

用户发出请求登录。

**函数原型：**

|  |
| --- |
| int ReqUserLogin(struct CX1FtdcUserLoginField \*pUserLoginData); |

**参数：**

pUserLoginData：指向用户登录请求结构的地址。用户请求登录结构：

|  |
| --- |
| struct CX1FtdcUserLoginField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcPasswdType Passwd; //密码  TX1FtdcCompanyIDType　　　　 CompanyID; //厂商ID (N/A)  TX1FtdcAppIDType AppID; // AppID  TX1FtdcAuthCodeType AuthCode; // 授权码  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | **不约束** | 请求ID |
| AccountID | M |  | 资金帐号ID |
| Passwd | M |  | 密码 |
| CompanyID | BLK |  | 厂商ID |
| AppID | M |  | AppID |
| AuthCode | M |  | 授权码 |

**返回值：**

参见附录一(错误码表)

**说明：**

AccountID为用户的登录号.登录账号长度不超过12字节。

### ReqUserLogout方法

用户发出退出请求。

**函数原型：**

|  |
| --- |
| int ReqUserLogout( struct CX1FtdcUserLogoutField \*pUserLogoutData ); |

**参数：**

pUserLogoutData:指向用户退出请求结构的地址。用户请求退出结构：

|  |
| --- |
| struct CX1FtdcUserLogoutField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金帐号ID  TX1FtdcSessionIDType SessionID; //会话ID  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | **不约束** | 请求ID |
| AccountID | M |  | 资金帐号ID |
| SessionID | N/A |  | 会话ID |

**返回值：**

参见附录一(错误码表)

注：AccountID为用户的登录号：需要传入正确的登录号。

### ReqInsertOrder方法

用户发出下单请求。

**函数原型：**

|  |
| --- |
| int ReqInsertOrder( struct CX1FtdcInsertOrderField \*pInsertOrderData ); |

**参数：**

pInsertOrderData:指向用户请求报单结构地址。用户请求报单结构：

|  |
| --- |
| struct CX1FtdcInsertOrderField  {  TX1FtdcAccountIDType AccountID; //资金账户  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcPriceType InsertPrice; //报单价格  TX1FtdcAmountType OrderAmount; //报单数量  TX1FtdcBuySellTypeType BuySellType; //买卖标志  TX1FtdcOpenCloseTypeType OpenCloseType; //开平标志  TX1FtdcSpeculatorType Speculator; //投保类型  TX1FtdcInsertType InsertType; //自动单类别  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcOrderPropertyType OrderProperty; //报单附加属性  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcAmountType MinMatchAmount; //最小成交量  TX1FtdcReservedType ReservedType2; //预留字段2  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcCustomCategoryType CustomCategory; //自定义类别  TX1FtdcPriceType ProfitLossPrice; //止盈止损价格(N/A)  TX1FtdcSeatID SeatID; //席位号  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **段名称** | **是否必填** | **取值范围** | **字段描述** |
| AccountID | M | **不约束** | 资金帐号ID |
| LocalOrderID | M | **大于0，不重复，递增** | 本地委托号 |
| InstrumentID | M | **交易所系统支持的合约** | 合约代码 |
| InsertPrice | M | **不约束** | 报单价格 |
| OrderAmount | M | **不约束** | 报单数量 |
| BuySellType | M | **1买；2卖** | 买卖标志 |
| OpenCloseType | M | **1开仓；2平仓；** | 开平标志 |
| Speculator | M | **0投机; 1套保; 2套利（投机** | 投保类型 |
| InsertType | M | **1普通单;** | 委托类别 |
| OrderType | M | **1限价; 2市价; 4套利;8 互换[此版本暂不支持4和8]** | 报单类型 |
| OrderProperty | M | **‘0’无属性订单; ‘1’FAK订单; ‘2’FOK** | 报单附加属性 |
| InstrumentType | M | **0期货；1期权（普通单）** | 合约类型 |
| MinMatchAmount | NM | 0:大商所强制为0 | 最小成交量 |
| ReservedType2 | BLK |  | 预留字段 |
| RequestID | NM | 不约束 | 请求ID |
| CustomCategory | NM |  | 自定义 |
| ProfitLossPrice | NM |  | 止盈止损 |
| SeatID | NM |  | 席位号 |

**返回值：**

参见附录一(错误码表)

**说明：**

OrderType目前只支持输入：

#define DFITC\_LIMITORDER 1 ///<限价委托

#define DFITC\_MKORDER 2 ///<市价委托

注意：用户输入的资金账号AccountID必须保证不超过12字节。

SeatID 席位号：

用户可以指定席位下单

用户也可以默认填0，报单会轮巡所有的配置席位。

### ReqCancelOrder方法

用户发出撤单请求。

**函数原型：**

|  |
| --- |
| int ReqCancelOrder( struct CX1FtdcCancelOrderField \*pCancelOrderData ); |

**参数：**

pCancelOrderData:指向请求撤单结构地址。用户请求撤单结构：

|  |
| --- |
| struct CX1FtdcCancelOrderField  {  TX1FtdcAccountIDType AccountID; //登录账户ID  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcOrderSysIDType OrderSysID; //系统委托号  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcSessionIDType SessionID; //会话ID  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| AccountID | M |  | 登录账户ID |
| X1OrderID | NM | **大于等于0** | 柜台委托号 |
| LocalOrderID | NM | **大于等于0** | 本地委托号 |
| OrderSysID | NM | **大于等于0** | 系统委托号 |
| InstrumentID | M | **交易所系统支持的合约** | 合约号 |
| RequestID | NM | 不约束 | 请求ID |
| SessionID | NM | **大于等于0** | 会话ID |

**返回值：**

参见附录一(错误码表)

**说明：**

支持两种方式撤单：1. X1OrderID；2. OrderSysID优先级同序号；

当柜台异常重启后，可以使用X1OrderID或者OrderSysID进行撤单。

本系统不支持LocalOrderID + SessionID的撤单方式

### ReqQryOrderInfo方法

用户发出当日委托查询请求。

**函数原型：**

|  |
| --- |
| int ReqQryOrderInfo (struct CX1FtdcQryOrderField \*pOrderData); |

**参数：**

pOrderData：指向请求当日委托查询结构地址。用户请求当日委托查询结构：

|  |
| --- |
| struct CX1FtdcQryOrderField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcCustomCategoryType CustomCategory; //自定义类别  TX1FtdcOrderAnswerStatusType OrderStatus; //委托状态  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcInstrumentIDType InstrumentID; //合约代码  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| AccountID | M |  | 登录账户ID |
| InstrumentType | M | 0期货；1期权 | 合约类型 |
| CustomCategory | N/A |  | 自定义类别 |
| OrderStatus | NM |  | 委托状态 |
| OrderType | NM | 1限价; 2市价; 4套利;8 互换 | 报单类型 |
| X1OrderID | NM |  | 柜台委托号 |
| LocalOrderID | NM |  | 本地委托号 |
| InstrumentID | NM | 交易所系统支持的合约 | 合约号 |

**返回值：**

参见附录一(错误码表)

**说明：**

如果想查询所有委托记录，则只需要输入AccountID和InstrumentType即可。

查询指定报单类型的单子，如限价单或市价单等。

废单仅支持交易所废单的委托查询。

不支持用户使用LocalOrderID进行查询。

### ReqQryPosition方法

用户发出持仓查询请求。

**函数原型：**

|  |
| --- |
| int ReqQryPosition( struct CX1FtdcQryPositionField\*pPositionData ); |

**参数：**

pPositionData：指向请求持仓查询结构地址。用户请求持仓查询结构：

|  |
| --- |
| struct CX1FtdcQryPositionField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| AccountID | M |  | 登录账户ID |
| InstrumentType | M | 0期货；1期权 | 合约类型 |
| InstrumentID | NM | 交易所系统支持的合约 | 合约号 |

**返回值：**

参见附录一(错误码表)

**说明：**

如果没有提供合约代码，则查询全部持仓信息

### ReqQryPositionDetail方法

用户发出持仓明细查询请求。

**函数原型：**

|  |
| --- |
| int ReqQryPositionDetail ( struct CX1FtdcQryPositionDetailField\*pPositionDetailData ); |

**参数：**

pPositionDetailData：指向请求持仓明细查询结构地址。用户请求持仓明细查询结构：

|  |
| --- |
| struct CX1FtdcQryPositionDetailField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| AccountID | M |  | 登录账户ID |
| InstrumentType | M | 0期货；1期权 | 合约类型 |
| InstrumentID | NM | 交易所系统支持的合约 | 合约号 |

**返回值：**

参见附录一(错误码表)

**说明：**

如果没有提供合约代码，则查询所有合约的持仓明细信息

### ReqQryMatchInfo方法

用户发出当日成交查询请求。

**函数原型：**

|  |
| --- |
| int ReqQryMatchInfo (struct CX1FtdcQryMatchField\* pMatchData); |

**参数：**

pMatchData：指向请求当日成交查询结构地址。用户请求当日成交查询结构：

|  |
| --- |
| struct CX1FtdcQryMatchField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcCustomCategoryType CustomCategory; //自定义类别  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcInstrumentIDType InstrumentID; //合约代码  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| AccountID | M |  | 登录账户ID |
| InstrumentType | M | 0期货；1期权 | 合约类型 |
| CustomCategory | N/A |  | 自定义类别 |
| OrderType | NM | 1限价; 2市价; 4套利;8 互换 | 报单类型 |
| X1OrderID | NM |  | 柜台委托号 |
| InstrumentID | NM | 交易所系统支持的合约 | 合约号 |

**返回值：**

参见附录一(错误码表)

**说明：**

如果想查询所有委托记录，则只需要输入AccountID和InstrumentType即可。

### ReqQryCustomerCapital方法

用户发出资金查询请求。

**函数原型：**

|  |
| --- |
| int ReqQryCustomerCapital( struct CX1FtdcQryCapitalField\*pCapitalData ); |

**参数：**

pCapitalData：指向请求资金查询结构地址。用户请求资金查询结构：

|  |
| --- |
| struct CX1FtdcQryCapitalField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| AccountID | M |  | 登录账户ID |

**返回值：**

参见附录一(错误码表)

### ReqQryExchangeStatus方法

用户发出交易所状态查询请求

**函数原型：**

|  |
| --- |
| int ReqQryExchangeStatus(struct CX1FtdcQryExchangeStatusField\*pQryExchangeStatusData) |

**参数：**

pQryExchangeStatusData：指向交易所状态查询请求结构的地址。

|  |
| --- |
| struct CX1FtdcQryExchangeStatusField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcExchangeIDType ExchangeID; //交易所编码  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| ExchangeID | M |  | 交易所编码 |

**返回值：**

参见附录一(错误码表)

### ReqQryExchangeInstrument方法

查询交易所合约列表。

**函数原型**：

|  |
| --- |
| IntReqQryExchangeInstrument(  struct CX1FtdcQryExchangeInstrumentField\*pExchangeInstrumentData ); |

**参数：**

pExchangeInstrumentData：指向交易所合约查询结构的地址。

|  |
| --- |
| struct CX1FtdcQryExchangeInstrumentField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcExchangeIDType ExchangeID; //交易所编码  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | 不约束 | 请求ID |
| AccountID | M |  | 资金帐号ID |
| ExchangeID | NM |  | 交易所编码 |
| InstrumentType | M | 0 期货 1期权 | 合约类型 |

**返回值：**

参见附录一(错误码表)

**说明：**

当ExchangeID为空时，表示查询各交易所的所有合约代码，也可指定查询某交易所的所有合约代码信息，但该接口不能查询到套利合约代码。

### ReqWarmInsertOrder方法

用户发送warm包。

**函数原型：**

|  |
| --- |
| int ReqWarmInsertOrder(); |

**返回值：**

参见附录一(错误码表)

**说明：**

在登录成功后，用户可以调用该接口主动warm下单流，来提高之后正常下单时的下单速度。该接口存在流控，当前流控间隔为100ms

### ReqResetPassword方法

用户密码修改请求接口。

**函数原型：**

|  |
| --- |
| int ReqResetPassword(struct CX1FtdcReqResetPasswordField \* pResetPasswordData); |

**参数：**

pResetPasswordData：指向用户修改密码请求结构的地址。用户修改密码请求结构：

|  |
| --- |
| struct CX1FtdcReqResetPasswordField  {  TX1FtdcRequestIDType RequestID; // 请求ID  TX1FtdcAccountIDType AccountID; // 资金账户ID  TX1FtdcPasswordType OldPassword; // 旧密码  TX1FtdcPasswordType NewPassword; // 新密码  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | **不约束** | 请求ID |
| AccountID | M |  | 资金帐号ID |
| OldPassword | M |  | 旧密码 |
| NewPassword | M |  | 新密码 |

**返回值：**

参见附录一(错误码表)

**说明：**

在登录成功后，用户可以调用该接口进行资金账号密码修改。

### ReqQrySpecifyInstrument方法

用户发出指定合约信息查询请求。

**函数原型：**

|  |
| --- |
| int ReqQrySpecifyInstrument (struct CX1FtdcQrySpecificInstrumentField\* pInstrument); |

**参数：**

pInstrument：指向请求指定合约信息查询结构地址。用户请求指定合约信息查询结构：

|  |
| --- |
| struct CX1FtdcQrySpecificInstrumentField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcExchangeIDType ExchangeID; //交易所ID  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  }; |

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **是否必填** | **取值范围** | **字段描述** |
| RequestID | NM | **不约束** | 请求ID |
| AccountID | M |  | 资金帐号ID |
| InstrumentType | M | 0期货；1期权 | 合约类型 |
| ExchangeID | NM |  | 交易所ID |
| InstrumentID | M | 交易所系统支持的合约 | 合约号 |

**返回值：**

参见附录一(错误码表)

**说明：**

ExchangeID并非必须输入选项，因为目前期货市场上，一个合约代码在各交易所是唯一存在的。

## CX1FtdcTraderSpi接口

CX1FtdcTraderSpi实现了事件通知接口，用户需要实现此类接口，编写事件处理方法来处理用户感兴趣的事件。

报单响应，报单回报，撤单响应，撤单回报中的SessionID，LocalOrderID是一致的。

报单回报，撤单响应，撤单回报中的X1OrderID是一致的。

### OnFrontConnected方法

该方法是在Api和飞创X-One 2.0极速交易系统建立连接后被调用，该调用仅仅是说明tcp连接已经建立成功。用户需要自行登录才能进行后续的业务操作，当然也可以在该函数内进行登录请求。连接失败则此方法不会被调用。

**函数原型：**

|  |
| --- |
| void OnFrontConnected(); |

### OnFrontDisconnected方法

该方法是在Api和飞创X-One 2.0极速交易系统连接断开后被调用。

**函数原型：**

|  |
| --- |
| void OnFrontDisconnected(int nReason); |

### OnRspUserLogin方法

当用户发出登录请求后，飞创X-One 2.0极速交易系统返回响应时此方法会被调用，通知用户登录是否成功。

**函数原型：**

|  |
| --- |
| VoidOnRspUserLogin(  struct CX1FtdcRspUserLoginField\*pUserLoginInfoRtn,  struct CX1FtdcRspErrorField \*pErrorInfo ) |

**参数：**

pUserLoginInfoRtn：返回用户登录信息结构地址：

|  |
| --- |
| struct CX1FtdcRspUserLoginField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金帐号ID  TX1FtdcAccountLoginResultType LoginResult; //登录结果  TX1FtdcLocalOrderIDType InitLocalOrderID; //初始本地委托号(N/A)  TX1FtdcSessionIDType SessionID; //会话ID  TX1FtdcErrorIDType ErrorID; //错误ID  TX1FtdcErrorMsgInfoType ErrorMsg; //错误信息  TX1FtdcTimeType DCEtime; //大商所时间(N/A)  TX1FtdcTimeType SHFETime; //上期所时间(N/A)  TX1FtdcTimeType CFFEXTime; //中金所时间(N/A)  TX1FtdcTimeType CZCETime; //郑商所时间(N/A)  TX1FtdcTimeType INETime; //上能所时间(N/A)  }; |

**说明：**

1、当loginResult为0时表示登录成功。登录失败时，pErrorInfo中将包含错误ID和错误信息。成功时，用户将获取一个会话ID

2、当登录成功ErrorID为0时，返回的pErrorInfo为一个空指针；当登录失败，ErrorID不为0时，pUserLoginInfoRtn和pErrorInfo均不为空

### OnRspUserLogout方法

当用户发出退出请求后，返回响应此方法会被调用，通知用户退出状态。

**函数原型：**

|  |
| --- |
| void OnRspUserLogout(  struct CX1FtdcRspUserLogoutInfoField\*pUserLogoutInfoRtn,  struct CX1FtdcRspErrorField \*pErrorInfo ); |

**参数：**

pUserLogoutInfoRtn:返回用户退出信息结构地址：

|  |
| --- |
| struct CX1FtdcRspUserLogoutInfoField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcAccountLogoutResultType LogoutResult; //退出结果  TX1FtdcErrorIDType ErrorID; //错误ID  TX1FtdcErrorMsgInfoType ErrorMsg; //错误信息  }; |

说明：

1、登出失败时，pErrorInfo中包含了错误ID及错误信息。

2、当登出成功ErrorID为0时，返回的pErrorInfo为一个空指针；当登出失败，ErrorID不为0时，pUserLogoutInfoRtn和pErrorInfo均不为空

### OnRspInsertOrder方法

下单应答。当用户录入报单后，返回响应时，该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspInsertOrder(  struct CX1FtdcRspOperOrderField\*pOrderRtn,  struct CX1FtdcRspErrorField\*pErrorInfo  ); |

**参数：**

pOrderRtn：返回下单响应信息结构地址。下单响应信息结构：

|  |
| --- |
| struct CX1FtdcRspOperOrderField  {  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcOrderAnswerStatusType OrderStatus; //委托状态  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcPriceType Margin; //冻结保证金  TX1FtdcSessionIDType SessionID; //会话ID  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcPriceType Fee; //手续费  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcExchangeIDType ExchangeID; //交易所  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcOpenCloseTypeType OpenCloseType; //开平  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcSpeculatorType Speculator; //投资类别  TX1FtdcPriceType InsertPrice; //委托价  TX1FtdcPriceType ProfitLossPrice; //止盈止损价格(N/A)  TX1FtdcAmountType MinMatchAmount; //最小成交量  TX1FtdcAmountType OrderAmount; //委托数量  TX1FtdcInsertType InsertType; //自动单类别(N/A)  TX1FtdcOrderTypeType OrderType; //订单类型  TX1FtdcOrderPropertyType OrderProperty; //订单属性  TX1FtdcClientIDType ClientID; //交易编码  }; |

**说明：**

1、报单失败时，pErrorInfo中包含了错误ID及错误信息。

2、报单成功时，是表示柜台系统确认了该笔报单，但报单未到交易所。

3、MinMatchAmount字段，大商所设置为0，用户输入无效。

4、InsertType字段，只支持普通委托单，不支持自动单和条件单。

5、订单类型为市价单时，委托价返回请求时所填数值。

6、当报单成功(错误码为0)时，返回的pErrorInfo为一个空指针；当报单失败 (ErrorID不为0)时，pOrderRtn和pErrorInfo均不为空

### OnRspCancelOrder方法

撤单应答。当用户撤单后，返回响应时该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspCancelOrder(struct CX1FtdcRspOperOrderField\* pOrderCanceledRtn,  struct CX1FtdcRspErrorField \*pErrorInfo ); |

**参数：**

pOrderCanceledRtn：返回撤单响应信息结构。撤单响应信息结构：

|  |
| --- |
| struct CX1FtdcRspOperOrderField  {  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcOrderAnswerStatusType OrderStatus; //委托状态  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcPriceType Margin; //冻结保证金  TX1FtdcSessionIDType SessionID; //会话ID  TX1FtdcPriceType Fee; //手续费  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcAccountIDType AccountID; //资金账户ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcExchangeIDType ExchangeID; //交易所  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcOpenCloseTypeType OpenCloseType; //开平  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcSpeculatorType Speculator; //投资类别  TX1FtdcPriceType InsertPrice; //委托价  TX1FtdcPriceType ProfitLossPrice; //止盈止损价格(N/A)  TX1FtdcAmountType MinMatchAmount; //最小成交量(N/A)  TX1FtdcAmountType OrderAmount; //委托数量  TX1FtdcInsertType InsertType; //自动单类别(N/A)  TX1FtdcOrderTypeType OrderType; //订单类型  TX1FtdcOrderPropertyType OrderProperty; //订单属性  TX1FtdcClientIDType ClientID; //交易编码  }; |

**说明：**

1、撤单发生错误时，需要检查pErrorInfo是否为NULLPTR。且客户端收到该响应时，只表示柜台确认了这笔撤单请求。

2、Margin字段值为0；

3、Fee字段值为0；

4、MinMatchAmount字段值为0；

5、当撤单成功(错误码为0)时，返回的pErrorInfo为一个空指针；当撤单失败 (ErrorID不为0)时，pOrderCanceledRtn和pErrorInfo均不为空

### OnRtnErrorMsg方法

API内部或者柜台内部发生异常错误，通过此方法通知用户。

API内部错误：包含API内存数据异常、系统资源申请异常等。

柜台内存错误：包含交易所API内部抛出异常，柜台内部数据异常，系统资源申请异常等。

**函数原型：**

|  |
| --- |
| void OnRtnErrorMsg( struct CX1FtdcRspErrorField\*pErrorInfo ); |

### OnRtnMatchedInfo方法

成交回报，当委托成功交易后此方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRtnMatchedInfo( struct CX1FtdcRspPriMatchInfoField\*pRtnMatchData )； |

**参数：**

CX1FtdcRspPriMatchInfoField：指向成交回报的结构。成交回报数据结构：

|  |
| --- |
| struct CX1FtdcMatchRtnField  {  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcOrderSysIDType OrderSysID; //报单编号(交易所报单编号)  TX1FtdcMatchIDType MatchID; //成交编号  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcOpenCloseTypeType OpenCloseType; //开平标志  TX1FtdcPriceType MatchedPrice; //成交价格  TX1FtdcAmountType OrderAmount; //委托数量  TX1FtdcAmountType MatchedAmount; //成交数量  TX1FtdcDateType MatchedTime; //成交时间  TX1FtdcPriceType InsertPrice; //报价  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcMatchType MatchType; //成交类型  TX1FtdcSpeculatorType Speculator; //投保  TX1FtdcExchangeIDType ExchangeID; //交易所ID  TX1FtdcFeeType Fee; //手续费(N/A)  TX1FtdcSessionIDType SessionID; //会话标识  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcAccountIDType AccountID; //资金账号  TX1FtdcOrderAnswerStatusType OrderStatus; //申报结果  TX1FtdcPriceType Margin; //开仓为保证金,平仓为解冻保证金(N/A)  TX1FtdcPriceType FrozenCapita; //成交解冻委托冻结的资金(N/A)  TX1FtdcAdjustmentInfoType AdjustmentInfo; //组合或对锁的保证金调整信息,格式:[合约代码,买卖标志,投资类别,调整金额;] (N/A)  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcPriceType Turnover; //成交金额  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcInsertType InsertType; //自动单类别(N/A)  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcLocalOrderIDType PriFlowNo; //私有流编号  } |

说明：

Fee字段值为0；

Margin字段值为0；

### OnRtnOrder方法

下单委托成功后，或交易所拒绝报单时此方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRtnOrder( struct CX1FtdcRspPriOrderField\*pRtnOrderData )； |

**参数：**

pRtnOrderData：指向委托回报地址。委托回报数据结构：

|  |
| --- |
| struct CX1FtdcRspPriOrderField  {  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcOrderSysIDType OrderSysID; //报单编号  TX1FtdcOrderAnswerStatusType OrderStatus; //委托状态  TX1FtdcSessionIDType SessionID; //会话ID  TX1FtdcDateType SuspendTime; //挂起时间(N/A)  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcExchangeIDType ExchangeID; //交易所  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcOpenCloseTypeType OpenCloseType; //开平  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcSpeculatorType Speculator; //投资类别  TX1FtdcPriceType InsertPrice; //委托价  TX1FtdcPriceType ProfitLossPrice; //止盈止损价格(N/A)  TX1FtdcAccountIDType AccountID; //资金账号  TX1FtdcAmountType CancelAmount; //撤单数量(N/A)  TX1FtdcAmountType OrderAmount; //委托数量  TX1FtdcInsertType InsertType; //自动单类别(N/A)  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcX1OrderIDType ExtX1OrderID; //算法单编号(N/A)  TX1FtdcReservedType ReservedType2; //预留字段2(N/A)  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcOrderPropertyType OrderProperty; //订单属性  TX1FtdcAmountType MinMatchAmount; //最小成交量  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcErrorIDType ErrorID; //错误ID  TX1FtdcErrorMsgInfoType StatusMsg; //状态信息  TX1FtdcExtOrderType ExtOrderType; //条件单类型(N/A)  TX1FtdcLocalOrderIDTypePriFlowNo //私有流编号  }; |

**说明：**

1、LocalOrderID和SessionID在系统重启后，不进行恢复，默认为0。

2、OrderStatus一律返回未成交还在队列中3

3、CancelAmount字段值为0；

4、MinMatchAmount字段值为0【大商所】；

### OnRtnCancelOrder方法

当撤单成功后或撤单被交易所拒绝时该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRtnCancelOrder(struct CX1FtdcRspPriCancelOrderField\* pCancelOrderData) |

**参数：**

pCancelOrderData：指向撤单回报结构。撤单回报结构

|  |
| --- |
| struct CX1FtdcRspPriCancelOrderField  {  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcOrderSysIDType OrderSysID; //报单编号  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcPriceType InsertPrice; //报单价格  TX1FtdcBuySellTypeType BuySellType; //买卖类型  TX1FtdcOpenCloseTypeType OpenCloseType; //开平标志  TX1FtdcAmountType CancelAmount; //撤单数量  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcSpeculatorType Speculator //投保  TX1FtdcExchangeIDType ExchangeID; //交易所ID  TX1FtdcDateType CanceledTime; //撤单时间  TX1FtdcSessionIDType SessionID; //会话标识  TX1FtdcOrderAnswerStatusType OrderStatus; //申报结果  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcAccountIDType AccountID; //资金账号  TX1FtdcAmountType OrderAmount; //委托数量  TX1FtdcPriceType Margin; //保证金  TX1FtdcPriceType Fee; //手续  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcPriceType ProfitLossPrice; //止盈止损价格(N/A)  TX1FtdcAmountType MinMatchAmount; //最小成交量  TX1FtdcInsertType InsertType; //自动单类别(N/A)  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcErrorIDType ErrorID; //错误ID  TX1FtdcErrorMsgInfoType StatusMsg; //状态信息  TX1FtdcOrderPropertyType OrderProperty; //报单附加属性  TX1FtdcLocalOrderIDTypePriFlowNo //私有流编号  }; |

**说明：**

Margin、Fee、MinMatchAmount字段值为0；

### OnRspQryOrderInfo方法

查询当日委托响应，当用户发出委托查询后，该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspQryOrderInfo(  struct CX1FtdcRspOrderField\* pRtnOrderData,  struct CX1FtdcRspErrorField \* pErrorInfo,  bool bIsLast  ) |

**参数：**

bIsLast：表明是否是最后一条响应信息

pRtnOrderData：指向委托回报结构。委托回报数据结构

|  |
| --- |
| struct CX1FtdcRspOrderField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcOrderAnswerStatusType OrderStatus; //委托状态  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcOpenCloseTypeType OpenClose; //开平标志  TX1FtdcPriceType InsertPrice; //委托价  TX1FtdcAmountType OrderAmount; //委托数量  TX1FtdcPriceType MatchedPrice; //成交价格  TX1FtdcAmountType MatchedAmount; //成交数量  TX1FtdcAmountType CancelAmount; //撤单数量  TX1FtdcInsertType InsertType; //自动单类别(N/A)  TX1FtdcSpeculatorType Speculator; //投保  TX1FtdcDateType CommTime; //委托时间  TX1FtdcDateType SubmitTime; //申报时间(N/A)  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcExchangeIDType  ExchangeID; //交易所ID  TX1FtdcFrontAddrType OperStation; //委托地址(N/A)  TX1FtdcAccountIDType AccountID; //客户号  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcSessionIDType SessionId; //会话ID  TX1FtdcReservedType ReservedType2; //预留字段2(N/A)  TX1FtdcOrderSysIDType OrderSysID; //主场单号  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcPriceType Margin; //保证金  TX1FtdcPriceType Fee; //手续费  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcPriceType ProfitLossPrice; //止损止盈价(N/A)  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcOrderPropertyType OrderProperty; //订单属性  }; |

**说明：**

1、LocalOrderID也是下单时该单子的本地委托号，如果同一账号从多个客户端下单，则查询返回的LocalOrderID可能是重复的。

2、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pRtnOrderData非空；当查询失败 (ErrorID不为0)时，返回的pRtnOrderData为一个空指针，pErrorInfo非空

3、Margin字段在委托为期权买时，显示的为权利金冻结

### OnRspQryPosition方法

查询持仓响应。当用户发出持仓查询指令后，系统返回响应时该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspQryPosition(  struct CX1FtdcRspPositionField\*pPositionInfoRtn,  struct CX1FtdcRspErrorField \*pErrorInfo,  bool bIsLast  ); |

**参数：**

bIsLast:表示是否是最后一条消息

pPositionInfoRtn：返回持仓信息结构地址。持仓信息结构：

|  |
| --- |
| struct CX1FtdcRspPositionField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金帐号ID  TX1FtdcExchangeIDType ExchangeID; //交易所代码  TX1FtdcInstrumentIDType InstrumentID; //合约号  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcPriceType OpenAvgPrice; //开仓均价  TX1FtdcPriceType PositionAvgPrice; //持仓均价  TX1FtdcAmountType PositionAmount; //持仓量  TX1FtdcAmountType TotalAvaiAmount; //总可用  TX1FtdcAmountType TodayAvaiAmount; //今可用  TX1FtdcAmountType LastAvaiAmount; //昨可用  TX1FtdcAmountType TodayAmount; //今仓  TX1FtdcAmountType LastAmount; //昨仓  TX1FtdcAmountType TradingAmount; //平今挂单量  TX1FtdcProfitLossType DatePositionProfitLoss; //盯市持仓盈亏(N/A)  TX1FtdcProfitLossType DateCloseProfitLoss; //盯市平仓盈亏(N/A)  TX1FtdcProfitLossType Premium; //权利金  TX1FtdcProfitLossType ProfitLoss; //浮动盈亏(N/A)  TX1FtdcProfitLossType Margin; //占用保证金  TX1FtdcSpeculatorType Speculator; //投保类别  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcPriceType PreSettlementPrice; //昨结算价  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcAmountType YesterdayTradingAmount; //平昨挂单量  TX1FtdcProfitLossType OptionValue; // 期权市值  }; |

说明：

1、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pPositionInfoRtn非空；当查询失败 (ErrorID不为0)时，返回的pPositionInfoRtn为一个空指针，pErrorInfo非空。

### OnRspQryPositionDetail方法

查询持仓明细响应。当用户发出持仓明细查询指令后，系统返回响应时该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspQryPositionDetail(  struct CX1FtdcRspPositionDetailField\* pPositionDetailRtn,  struct CX1FtdcRspErrorField \*pErrorInfo,  bool bIsLast  ); |

**参数：**

bIsLast:表示是否是最后一条消息

pPositionDetailRtn：返回持仓明细信息结构地址。持仓明细信息结构：

|  |
| --- |
| struct CX1FtdcRspPositionDetailField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金帐号ID  TX1FtdcExchangeIDType ExchangeID; //交易所代码  TX1FtdcInstrumentIDType InstrumentID; //合约号  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcPriceType OpenPrice; //开仓价  TX1FtdcAmountType Volume; //手数  TX1FtdcMatchIDType MatchID; //成交编号  TX1FtdcDateType MatchedDate; //成交日期 TX1FtdcProfitLossType DatePositionProfitLoss; //盯市持仓盈亏(N/A) TX1FtdcProfitLossType DateCloseProfitLoss; //盯市平仓盈亏(N/A) TX1FtdcProfitLossType ProfitLoss; //浮动盈亏(N/A)  TX1FtdcProfitLossType Margin; //占用保证金  TX1FtdcSpeculatorType Speculator; //投保类别  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcPriceType PreSettlementPrice; //昨结算价  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcAmountType CloseOrderVol; //平仓委托数量  TX1FtdcAmountType CloseMatchVol; //平仓成交数量  TX1FtdcPositionDateType PositionDateType; //持仓日期类型  }; |

说明：

1、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pPositionDetailRtn非空；当查询失败 (ErrorID不为0)时，返回的pPositionDetailRtn为一个空指针，pErrorInfo非空。

### OnRspQryMatchInfo方法

查询当日成交信息响应，当用户发出成交查询后该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspQryMatchInfo(  struct CX1FtdcRspMatchField\* pRtnMatchData,  struct CX1FtdcRspErrorField \* pErrorInfo,  bool bIsLast  ); |

**参数：**

bIsLast：表明是否是最后一条响应信息。

pRtnMatchData：指向成交回报地址。成交回报结构：

SessionId此版本不支持

|  |
| --- |
| struct CX1FtdcRspMatchField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcX1OrderIDType X1OrderID; //柜台委托号  TX1FtdcAccountIDType AccountID; //资金账号  TX1FtdcExchangeIDType ExchangeID; //交易所ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcBuySellTypeType BuySellType; //买卖  TX1FtdcOpenCloseTypeType OpenClose; //开平  TX1FtdcPriceType MatchedPrice; //成交价格  TX1FtdcAmountType MatchedAmount; //成交数量  TX1FtdcPriceType MatchedMort; //成交金额  TX1FtdcSpeculatorType Speculator; //投保类别  TX1FtdcDateType MatchedTime; //成交时间  TX1FtdcMatchIDType MatchedID; //成交编号  TX1FtdcLocalOrderIDType LocalOrderID; //本地委托号  TX1FtdcClientIDType ClientID; //交易编码  TX1FtdcMatchType  MatchType; //成交类型  TX1FtdcInstrumentTypeType  InstrumentType; //合约类型  TX1FtdcSessionIDType SessionId; //会话ID  TX1FtdcReservedType ReservedType2; //预留字段2(N/A)  TX1FtdcCustomCategoryType CustomCategory; //自定义类别(N/A)  TX1FtdcPriceType Fee; //手续费  TX1FtdcOrderTypeType OrderType; //报单类型  TX1FtdcOrderSysIDType OrderSysID; //主场单号  }; |

**说明：**

1、成交信息响应的会话号和本地号均为0，自定义类别为空。

2、飞创X-One 2.0极速交易系统反演后，原始委托单中的LocalOrderID和SessionId失效。

3、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pRtnMatchData非空；当查询失败 (ErrorID不为0)时，返回的pRtnMatchData为一个空指针，pErrorInfo非空。。

### OnRspCustomerCapital方法

资金查询应答。当用户发出资金查询指令后，系统返回响应时该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspCustomerCapital(  struct CX1FtdcRspCapitalField\*pCapitalInfoRtn,  struct CX1FtdcRspErrorField \*pErrorInfo,  bool bIsLast  ); |

**参数：**

pCapitalInfoRtn：返回资金信息结构地址。资金信息结构：

|  |
| --- |
| struct CX1FtdcRspCapitalField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcAccountIDType AccountID; //资金帐号  TX1FtdcEquityType PreEquity; //上日权益  TX1FtdcEquityType TodayEquity; //当日客户权益  TX1FtdcProfitLossType CloseProfitLoss; //平仓盈亏  TX1FtdcProfitLossType PositionProfitLoss; //持仓盈亏  TX1FtdcProfitLossType FrozenMargin; //冻结资金  TX1FtdcProfitLossType Margin; //持仓保证金  TX1FtdcProfitLossType Fee; //当日手续费  TX1FtdcProfitLossType Available; //可用资金  TX1FtdcProfitLossType Withdraw; //可取资金(N/A)  TX1FtdcRiskDegreeType RiskDegree; //风险度(N/A)  TX1FtdcPremiumType TodayPremiumIncome; //本日权利金收入  TX1FtdcPremiumType TodayPremiumPay; //本日权利金付出  TX1FtdcPremiumType YesterdayPremium; //昨权利金收付  TX1FtdcMarketValueType OptMarketValue; //期权市值  TX1FtdcProfitLossType ProfitLoss; //浮动盈亏  TX1FtdcProfitLossType TotalFundOut; //总出金  TX1FtdcProfitLossType TotalFundIn; //总入金  TX1FtdcRatioType UpFloathingRatio; //浮动比例上限（N/A）  TX1FtdcRatioType DownFloathingRatio; //浮动比例下限（N/A）  TX1FtdcRatioType AllocateRatio //分配比例（N/A）  TX1FtdcCurrencyTypeCurrencyID; //币种代码  TX1FtdcProfitLossType Mortgage; //质押金额(N/A)  TX1FtdcProfitLossType FundMortgageIn; //货币质入金额（N/A）  TX1FtdcProfitLossType FundMortgageOut; //货币质出金额（N/A）  TX1FtdcProfitLossType FundMortgageAvailable; //货币质押余额（N/A）  TX1FtdcProfitLossType YesterdayMargin; //昨保证金(N/A)  }; |

**说明：**

1、质押金额此版本不用。

2、CurrencyID字段只支持人民币。

3、YesterdayMargin字段赋值为0。

4、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pCapitalInfoRtn非空；当查询失败 (ErrorID不为0)时，返回的pCapitalInfoRtn为一个空指针，pErrorInfo非空。

### OnRspQryExchangeStatus方法

交易所状态查询响应，当用户发出交易所状态查询指令后该方法会被调用。

**函数参数：**

|  |
| --- |
| void OnRspQryExchangeStatus(  struct CX1FtdcRspExchangeStatusField\* pRspExchangeStatusData,  struct CX1FtdcRspErrorField \* pErrorInfo,  bool bIsLast  ) |

**参数：**

pRspExchangeStatusData：返回交易所状态结构的地址，该结构如下：

|  |
| --- |
| struct CX1FtdcRspExchangeStatusField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcExchangeStatusType  ExchangeStatus; //交易所状态  TX1FtdcExchangeIDType ExchangeID; //交易所编码  }; |

**说明：**

1、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pRspExchangeStatusData非空；当查询失败 (ErrorID不为0)时，返回的pRspExchangeStatusData为一个空指针，pErrorInfo非空。

### OnRspQryExchangeInstrument方法

合约查询请求应答。当用户发出合约查询指令后，返回响应时该方法会被调用。

**函数原型：**

|  |
| --- |
| void OnRspQryExchangeInstrument(  struct CX1FtdcRspExchangeInstrumentField\* pInstrumentData,  struct CX1FtdcRspErrorField\* pErrorInfo,  bool bIsLast  ); |

**参数：**

bIsLast:表示是否是最后一条消息

pInstrumentData：返回合约信息结构地址。合约信息结构：

|  |
| --- |
| struct CX1FtdcRspExchangeInstrumentField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcExchangeIDType ExchangeID; //交易所编码  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcVarietyNameType VarietyName; //品种名称  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcInstrumentTypeType OrderTopLimit; //委托上限  TX1FtdcAmountType MktOrderTopLimit; //市价委托上限  TX1FtdcPriceType ContractMultiplier; //合约乘数  TX1FtdcPriceType  MinPriceFluctuation; //最小变动价位  TX1FtdcInstrumentMaturityType InstrumentMaturity; //合约最后交易日  TX1FtdcPriceType UpperLimitPrice; //涨停板价  TX1FtdcPriceType LowerLimitPrice; //跌停板价  TX1FtdcPriceType PreClosePrice; //昨收盘  TX1FtdcPriceType PreSettlementPrice; //昨结算价  TX1FtdcPriceType SettlementPrice; //结算价  TX1FtdcAmountType PreOpenInterest; //昨持仓量(N/A)  TX1FtdcInstrumentPrefixType InstrumentPrefix; //合约前缀  TX1FtdcInstrumenExpirationDateType InstrumenExpiration; //合约到期日  TX1FtdcOptionTypeType OptionType; //期权类型  TX1FtdcInstrumentIDType Underlying; //期权对应的标的合约代码  TX1FtdcOptionTypeType OptionType; //期权类型  TX1FtdcPriceType StrikePrice; //执行价格  TX1FtdcRiskDegreeType ExchangeRiskDegree; //交易所期权最低保障风险系数(N/A)  TX1FtdcPriceType MinMargin; //单位(手)期权合约最小保证金(N/A)  TX1FtdcAmountType TradeSize; //期权开仓单位(N/A)  }; |

**说明：**

1、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pInstrumentData非空；当查询失败 (ErrorID不为0)时，返回的pInstrumentData为一个空指针，pErrorInfo非空。

### OnRspResetPassword方法

修改用户密码响应，当用户请求修改密码后，系统返回响应时该方法会被调用。

**函数参数：**

|  |
| --- |
| void OnRspResetPassword (  struct CX1FtdcRspResetPasswordField \* pResetPasswordData,  struct CX1FtdcRspErrorField\* pErrorInfo  ) |

**参数：**

pResetPasswordData：返回修改密码的结果地址，该结构如下：

|  |
| --- |
| struct CX1FtdcRspResetPasswordField  {  TX1FtdcRequestIDType RequestID; // 请求ID  TX1FtdcAccountIDType AccountID; // 资金帐号ID  TX1FtdcResetPasswordType ResetPasswordResult; // 修改密码结果 |

**说明：**

当密码修改成功(错误码为0)时，返回的pErrorInfo为一个空指针，pResetPasswordData非空；当查询失败 (ErrorID不为0)时，返回的pResetPasswordData和pErrorInfo均为非空结构体。

### OnRspQrySpecifyInstrument方法

查询指定合约信息响应，当用户发出查询制定合约指令后该方法会被调用。

**函数参数：**

|  |
| --- |
| void OnRspQrySpecifyInstrument(  struct CX1FtdcRspSpecificInstrumentField\* pInstrument,  struct CX1FtdcRspErrorField\* pErrorInfo,  bool bIsLast  ) |

**参数：**

pInstrument：指向返回的合约信息结构的地址，该结构如下：

|  |
| --- |
| Struct CX1FtdcRspSpecificInstrumentField  {  TX1FtdcRequestIDType RequestID; //请求ID  TX1FtdcInstrumentIDType InstrumentID; //合约代码  TX1FtdcRatioType LongMarginRatio; //多头保证金率  TX1FtdcRatioType ShortMarginRatio; //空头保证金率  TX1FtdcPriceType LongMarginRatioByVolume; //多头保证金费(定额)  TX1FtdcPriceType ShortMarginRatioByVolume; //空头保证金费(定额)  TX1FtdcRatioType OpenFeeVolRatio; //开仓手续费按手数计算  TX1FtdcRatioType CloseFeeVolRatio; //平仓手续费按手数计算  TX1FtdcRatioType CloseTodayFeeVolRatio; //平今手续费按手数计算  TX1FtdcRatioType OpenFeeAmtRatio; //开仓手续费率按金额计算  TX1FtdcRatioType CloseFeeAmtRatio; //平仓手续费率按金额计算  TX1FtdcRatioType CloseTodayFeeAmtRatio;//平今手续费率按金额计算  TX1FtdcInstrumentTypeType OrderTopLimit; //委托上限  TX1FtdcPriceType ContractMultiplier; //合约乘数  TX1FtdcPriceType  MinimumPriceChange; //最小变动价位  TX1FtdcInstrumentTypeType InstrumentType; //合约类型  TX1FtdcInstrumentMaturityType InstrumentMaturity; //合约最后交易日  TX1FtdcComputeModeType ComputeMode; //计算方式(N/A)  TX1FtdcPriceType AtMoneyNorm; //平值按定额(卖期权)(N/A)  TX1FtdcPriceType UpperLimitPrice; //涨停板价  TX1FtdcPriceType LowerLimitPrice; //跌停板价  TX1FtdcPriceType PreClosePrice; //昨收盘  TX1FtdcPriceType PreSettlementPrice; //昨结算价  TX1FtdcPriceType SettlementPrice; //结算价  TX1FtdcAmountType PreOpenInterest; //昨持仓量(N/A)  TX1FtdcRatioType OptExecRatio; //期权：行权按比例，期货：交割按比例(N/A)  TX1FtdcRatioType OptExecRatioPerVol; //期权：行权按定额，期货：交割按定额(N/A)  TX1FtdcCalculateMode CalculateMode; // 冻结资金计算方式(N/A)  TX1FtdcPriceType ReferencePrice; // 参考价格  }; |

**说明：**

1、当查询成功(错误码为0)时，返回的pErrorInfo为一个空指针，pInstrument非空；当查询失败 (ErrorID不为0)时，返回的pInstrument为一个空指针，pErrorInfo非空飞创极速交易系统。

# 附录一(错误码表)

|  |  |  |
| --- | --- | --- |
| **错误码** | **错误信息** | **说明** |
| 0 | Success | 成功 |
| 1 | User name invalid | 无效的账号 |
| 2 | Password incorrect | 密码错误 |
| 3 | Insert order failed | 下单失败 |
| 4 | Insert order rtn failed | 下单响应失败 |
| 5 | Cancel order failed | 撤单失败 |
| 6 | Internal error | 内部错误 |
| 7 | Client ptr is nullptr | 交易编码信息未找到 |
| 8 | Account in client ptr is nullptr | 客户信息未找到 |
| 9 | Client is invalid | 交易编码无效 |
| 10 | Trade system is disabled | 系统未达到稳态 |
| 11 | Can not find contract id | 合约代码未找到 |
| 12 | Local number is invalid | 本地编号无效 |
| 13 | Investor id is invalid | 客户号无效 |
| 14 | Trade system internal error | 系统内部错误 |
| 15 | Check money error | 资金校验错误 |
| 16 | Check open buy error | 买开校验错误 |
| 17 | Check open sell error | 卖开校验错误 |
| 18 | Check close sell error | 卖平校验错误 |
| 19 | Check close buy error | 买平校验错误 |
| 20 | Input bs error | 无效的买卖标志 |
| 21 | Input oc error | 无效的开平标志 |
| 22 | Can not find contract info or contract is forbidden | 合约信息未找到或者该合约被禁止交易 |
| 23 | Local order id invalid | 本地报单编号无效 |
| 24 | Local order info invalid | 本地委托信息无效 |
| 25 | Check Trade system state failed | 交易系统状态检查失败 |
| 26 | Flow control failed | 流控失败 |
| 27 | Cancel order failed | 撤单失败 |
| 28 | Insert order failed | 下单失败 |
| 29 | Strategy error | 策略失败 |
| 30 | Login already | 重复登录 |
| 31 | Vio put data error when order match | 委托成交时虚拟输入输出数据错误 |
| 32 | Vio put data number error when order match | 委托成交时虚拟输入输出数据编号错误 |
| 33 | The user is not login | 客户未登录 |
| 34 | Exceed Trade systemmax order | 超过交易系统最大委托数量 |
| 35 | Query server connect failed | 连接查询前置服务器失败 |
| 36 | Query server login failed | 登录查询前置服务器失败 |
| 37 | Query server logout failed | 登出查询前置服务器失败 |
| 38 | Invalid buy sell flag | 无效的买卖标志 |
| 39 | Invalid open close flag | 无效的开平标志 |
| 40 | Invalid speculate hedge flag | 无效的投资类别 |
| 41 | No trade code | 交易编码不存在 |
| 42 | The trade code has been frozen,cannot trade | 该交易编码已冻结,不能交易 |
| 43 | Invalid contract | 无效的合约 |
| 44 | The order price mismatches the float-price of the contract | 合约价格最小变动单位不符合 |
| 45 | Invalid order amount | 委托数量无效 |
| 46 | The trade code was forbidden trading this kind of contract | 该交易编码禁止交易该品种 |
| 47 | Invalid order price is out of price cap or floor,cannot trade | 价格超出涨跌停 |
| 48 | Placed close-position order failed since no enough positions | 平仓数量超过可平数量 |
| 49 | Not suport market price order in the exchange | 该品种不支持市价单 |
| 50 | Invalid combination contract | 套利合约不存在 |
| 51 | The local order ID already used | 本地报单号重复 |
| 52 | Invalid order type | 无效的订单类型 |
| 53 | Invalid contract type | 无效的合约类型 |
| 54 | Invalid order property | 无效的订单属性 |
| 55 | The profit(loss)price must be less than the principal price | 买止盈(损)价格必须小于委托价格 |
| 56 | Sell the surplus(loss) than the entrusted price price | 卖止盈(损)价格必须大于委托价格 |
| 57 | The order already canceled,cannot cancel it again | 该笔订单已撤单，不能重复撤单 |
| 58 | The force-close order does not allow to cancel | 强平单不可以撤单 |
| 59 | Max\_margin value is invalid | 合约的每手最大保证金设置不合法 |
| 60 | Invalid exchange member\_no or trade\_no | 无效的交易编码 |
| 61 | Trader address invalid | 交易地址无效 |
| 62 | Param error | 参数错误 |
| 63 | Query api ret failed | 查询api返回失败 |
| 64 | Order rtn,but not find contract id | 报单响应，未找到合约代码 |
| 65 | Order rtn,but not find exchange id | 报单响应，未找到交易所代码 |
| 66 | Open position amount reach upper limit | 开仓数量已达上限 |
| 67 | Open position amount reach proprietary trading upper limit | 开仓数量已达交易所上限 |
| 68 | Cancel order failed,x1\_order\_id and local\_order\_id are invalid | 撤单失败，无效的报单编号和本地报单编号 |
| 69 | Trader client check failed | 会员客户检查不通过 |
| 70 | Client without position | 客户无持仓 |
| 71 | Trader money deficiency | 交易资金不足 |
| 72 | Order status checked failed | 交易状态检查失败 |
| 73 | Sorry, Trade systemhasno server | 对不起，找不到交易系统服务器 |
| 74 | Open position exceed member agent position limit | 开仓数量超出会员上限 |
| 75 | Invalid speculator type | 无效的投资类别 |
| 76 | Invalid openclose type | 无效的开平标志 |
| 77 | Invalid buysell type | 无效的买卖标志 |
| 78 | Invalid order property | 无效的报单属性 |
| 79 | Invalid min match cnt | 无效的最小成交量 |
| 80 | Invalid instrument type | 无效的合约类型 |
| 81 | User account invalid | 无效的账号 |
| 82 | Order is wait to be canceled. | 订单等待撤单 |
| 83 | Session id invalid. | 无效的会话id |
| 84 | order info invalid. | 无效的委托信息 |
| 85 | Order cnt less min open cnt. | 报单数量小于最小开仓数量 |
| 86 | Order cnt less min close cnt. | 报单数量小于最小平仓数量 |
| 87 | Contract type invalid. | 合约类型无效 |
| 88 | Limit order cnt max limit each | 限价单超过限制 |
| 89 | Market order cnt max limit each | 市价单超过限制 |
| 90 | Api flowmeter limit | api流量限制 |
| 91 | server flowmeter limit | 服务器流量限制 |
| 92 | Send cancel all orders to exchange error | 发送全部撤单指令到交易所发生错误 |
| 93 | The buy price must be less than the sell price | 买价必须小于卖价 |
| 94 | Send for quote to exchange error | 报价发送错误 |
| 95 | Api version does not match. | api版本号不匹配 |
| 96 | Exchange non tradable | 交易所不可交易 |
| 97 | Not allow cancel order in this moment. | 当前状态不能撤单 |
| 98 | Have reasonable quotation,cannot for quote. | 有合理的报价，非做市商的 |
| 99 | Market maker not allow for quote. | 做市商不允许报价 |
| 100 | Server config error,member\_no and trader\_no are not match. | 服务器配置错误，member\_no和trader\_no不匹配 |
| 101 | Variety status is not in trading state. | 交易所非交易状态 |
| 102 | Cancel order failed,x1\_order\_id invalid | 报单编号无效，撤单失败 |
| 103 | Cancel order failed,spd\_order\_id invalid | 柜台委托号无效，撤单失败 |
| 104 | Invalid quote interval. | 报价区间无效 |
| 105 | Invalid quote contract or no market maker. | 无效的报价合约或者没有做市商 |
| 106 | Full matched,can not cancel. | 全部成交，不能撤单 |
| 107 | Exchange is not supported. | 交易所不支持 |
| 108 | Api duplicate init is forbidden. | api重复初始化被禁止 |
| 109 | Mmap failed | 磁盘映射失败 |
| 110 | Conf of cpu core id is invalid | cpu核心配置id无效 |
| 111 | Login too often,please try again later... | 登录太频繁，请稍后再试... |
| 112 | Exceed trade system auto max order. | 超过系统自动最大委托数 |
| 113 | Access control forbidden. | 禁止访问控制 |
| 114 | X1 order id invalid. | 无效的报单编号 |
| 115 | Too many session | 会话数量超过限制 |
| 116 | The exchange trading channel not stable | 交易所交易渠道不稳定 |
| 117 | Init parameter invalid,port is empty | 初始化参数无效，端口为空 |
| 118 | IOC need to be in continuous trading phase | IOC需要在连续交易阶段 |
| 119 | The order already matched | 订单已成交 |
| 120 | Member or client have no permission to trade the specified contract. | 会员或客户没有权限交易此品种 |
| 121 | Member or client have only close position permission to trade the specified contract. | 会员或客户对此品种只有平仓权限 |
| 122 | Member have no trade role. | 会员没有交易规则 |
| 123 | Hedge volume not enough. | 套保额度不足 |
| 124 | Invalid insert type. | 无效的自动单类别 |
| 125 | Already for quote. | 已经报过价 |
| 126 | Unmatched error code in system. | 系统中未匹配到错误号 |
| 127 | Cancel all order end. | 取消所有委托结束 |
| 128 | Trade code is forbidden to open position. | 交易编码禁止开仓 |
| 129 | Futures auto order not support. | 不支持期货开市自动单 |
| 130 | Option auto order not support. | 不支持期权开市自动单 |
| 131 | Invalid trade unit connection. | 无效的交易单元连接 |
| 132 | Invalid OrderSysID. | 无效的系统委托号 |
| 133 | This time period forbid option execution. | 此时间段禁止期权执行相关操作 |
| 134 | Return of order not received. | 交易单元没有收到报单回报 |
| 135 | The order is not market maker quote order. | 订单不是做市商应价单 |
| 136 | Offsetflag invalid | 无效的对冲标志 |
| 137 | Comstattus invalid | 无效的组合状态 |
| 138 | Abandflag invalid | 无效的放弃标识 |
| 139 | Check version timeout. | 版本校验超时 |
| 140 | Exchange contract does not match. | 交易所与合约不匹配 |
| 141 | Appid invalid | 无效的APPID |
| 142 | Appid communication mode invalid | 无效的APPID通讯模式 |
| 143 | Authcode invalid | 无效的授权码 |
| 144 | Authcode no access | 授权码禁止准入 |
| 145 | Encrypt version mismatched | 密钥版本不匹配 |
| 146 | Getsysteminfo error | 采集信息错误 |
| 147 | Getsysteminfo empty | 采集信息为空 |
| 148 | Getsysteminfo size exceed | 采集信息超过最大长度 |
| 149 | Getsysteminfo no access | 信息采集权限不足 |
| 150 | Getsysteminfo parameter invalid | 信息采集参数错误 |
| 151 | Getsysteminfo decrypt failed | 信息采集解密失败 |
| 152 | Getsysteminfo format error | 信息采集格式错误 |
| 153 | Getsysteminfo except flag invalid | 采集信息无效的异常标志 |
| 154 | Getsysteminfoip format error | 采集信息IP格式错误 |
| 155 | Getsysteminfologintiome format error | 采集信息登录时间格式错误 |
| 156 | Price type invalid | 无效的价格类型 |
| 157 | Account\_id trade right forbidden | 资金账号无交易权限 |
| 158 | Quot address invalid | 无效的行情服务器地址 |
| 159 | Quot server connect failed | 行情服务器连接失败 |
| 160 | Not subscribed or already unsubscribed the contract | 未订阅或已退订该合约 |
| 161 | Trade\_unit is not ready | 交易单元状态不可用 |
| 162 | Query\_unit is not ready | 查询单元状态不可用 |
| 163 | Replay fail | 反演失败 |
| 164 | Failed to send packet | 数据包发送失败 |
| 165 | Query Disconnect | 查询未连接 |
| 166 | Trade Disconnect | 交易未连接 |
| 1001 | Unknown error | 未知错误 |
| 1002 | The length of the account ID is longer than 12, please input again | 输入的资金账号超过12位,请重新输入 |
| 1003 | Contains invalid charactor | 包含非法字符 |
| 1004 | Invalid price value | 无效的价格 |
| 1009 | Invalid command | 无效的指令 |
| 1011 | Query frequency is too fast, this query operation is failed | 查询的频率过快,本次查询操作不成功 |
| 1012 | The frequency of placing or canceling order is too fast, this order request operation is failed | 报单的频率过快,本次报单操作不成功 |
| 1013 | Receive multiple invalid command requests, front will close this connection | 非法请求多次，系统将断开连接 |
| 1014 | The API version is too low, please upgrade | 您使用的API版本号过低，请升级 |
| 1015 | The version of the connected front is too low, please use correct API version | 您连接的前置版本过低，请使用正确的API版本 |
| 1016 | Try to connect to md front using trade api | 尝试连接到非交易交易子系统 |
| 1017 | Try to connect to trade front using md api | 尝试连接到非行情交易子系统 |
| 1018 | The md front doesnot configure multicasting market data | 行情前置未配置组播方式发送行情 |
| 1019 | The connected md front does not support custom market data | 您连接的行情前置暂不支持组合行情功能 |
| 1024 | Invalid custom category | 无效的自定义类别 |
| 1030 | Max session number limit by this account:%d | 用户在线会话超出上限 |
| 1031 | Not login | 未登录 |
| 1032 | This user already login | 该账户已登录 |
| 1033 | This kind user has approached max login limit, please use other IDs | 该账号已达到最大登录次数限制,请使用其它账号 |
| 1034 | Query exchange time failed, please login again to get exchange time | 获取交易所时间失败，请尝试重新登录获取时间 |
| 1035 | Not login or already logout | 未登录或已登出 |
| 1036 | Logout failed or already logout, please check | 登出失败或已登出,请检查 |
| 1037 | The trade server performance reaches to upper limit, please try again later | 服务器性能已达限制,请稍后再试 |
| 1038 | Invalid contract | 无效的合约 |
| 1039 | Empty contract | 合约为空 |
| 1040 | Not subscribed or already unsubscribed the contract. | 未订阅或已退订该合约 |
| 1041 | Invalid price. | 无效的价格 |
| 1050 | Cannot find the order ID corresponding to the account ID | 找不到该资金账号所对应的撤单委托号 |
| 1051 | The contract is incorrect to the order ID | 该合约和柜台委托号不匹配 |
| 1052 | Cannot find the order ID for the contract | 找不到该合约对应的柜台委托号 |
| 1053 | Cannot find the order info by the local order ID | 找不到本地委托号对应的委托信息 |
| 1054 | The local order ID is incorrect to the contract | 该合约和本地委托号不匹配 |
| 1055 | Both order ID and local ID are empty | 本地委托号和柜台委托号都为空 |
| 1056 | Cannot find any valid trade code | 找不到可以交易的交易编码 |
| 1057 | Please check the condition order configure file | 请检查条件单配置文件 |
| 1058 | Cannot find the order info by the algorithm order ID | 找不到该算法单编号对应的委托信息 |
| 1059 | The algorithm order ID is incorrect to the contract | 该合约与算法单编号不匹配 |
| 1060 | The order is in placing, please try to cancel it again later | 申报中，请稍后再撤 |
| 1061 | The exchange does not support batch cancellation orders | 该交易所不支持批量撤单指令 |
| 1062 | Invalid transaction time, banned mass cancellation | 非交易时间,禁止批量撤单 |
| 1080 | The local order ID already used | 本地委托号已经存在 |
| 1081 | Invalid order type | 无效的报单类型 |
| 1082 | Invalid contract type | 无效的合约类型 |
| 1083 | Arbitrage or roll over exchange order doesnot support the order within property | 套利或展期互换委托单不支持带属性的定单 |
| 1084 | Invalid investment type | 无效的投保类型 |
| 1085 | Invalid buy-sell flag | 无效的买卖标志 |
| 1086 | Invalid open-close flag | 无效的开平标志 |
| 1087 | Invalid auto order flag | 无效的自动单的标志 |
| 1088 | Invalid order property | 无效的订单属性 |
| 1089 | The local order ID must be positive number | 本地委托号不能为负值 |
| 1090 | The order type does not match the given contract | 下单类型和合约不匹配 |
| 1091 | The contract type is incorrect to this contract | 合约与合约类型不匹配 |
| 1092 | During the continuous trading, banned under open automatically | 连续交易期间,禁止下开市自动单 |
| 1093 | Buy only the profit (loss) price must be less than the principal price | 买止盈（损）价格必须小于委托价格 |
| 1094 | Sell the surplus (loss) than the entrusted price price | 卖止盈（损）价格必须大于委托价格 |
| 1095 | Option not surpport profit (loss) price order | 暂不支持期权止盈（损）指令 |
| 1096 | Only the CFFEX market orders to support the order property | 仅中金所市价单支持该订单属性 |
| 1098 | Inquiry instruction only support options contract | 询价指令仅支持期权合约 |
| 1102 | Send error data to trade front, please check | 发送数据错误,请检查 |
| 1110 | Not support trading order at this optimized-xspeed system | 该极速版XSpeed不支持此交易指令 |
| 1114 | This account cannot open a new position | 该客户禁止开仓 |
| 1120 | Invalid comparison flag | 无效的比较标志 |
| 1121 | Invalid price reference | 无效的价格参照 |
| 1122 | Invalid correlation property | 无效的关联属性 |
| 1123 | Invalid association number | 无效的关联编号 |
| 1124 | Invalid order amount | 委托数量无效 |
| 1125 | Invalid effective date | 无效的有效日期 |
| 1126 | Invalid freezing type | 无效冻结类型 |
| 1127 | The breakdown number is invalid | 无效的击穿次数 |
| 1128 | The order price mismatches the float-price of the contract | 该合约变动价格有误 |
| 1129 | The breakdown type is invalid | 无效的击穿属性类型 |
| 1130 | The Conditional order type is invalid | 无效的条件单类型 |
| 1131 | Invalid limit number | 无效的数量限制 |
| 1200 | Invalid exchange code | 无效的交易所代码 |
| 1201 | The contract is not from the specified exchange | 交易所和合约不匹配 |
| 1202 | Cannot find the contract info | 未找到该合约的合约信息 |
| 1203 | Invalid or empty trade code | 交易编码无效或为空 |
| 1204 | Please check the order ID | 请检查柜台委托号 |
| 1205 | The password length is longer than 16, and it must be digits or letters | 新密码长度不能超过16位,且必须为数字或字母 |
| 1206 | Queried empty data | 查询数据结果为空 |
| 1207 | Empty date when request confirm bill | 请求确认日期为空 |
| 1208 | Please input correct confirm date, form:yyyy.mm.dd | 请输入正确的日期 |
| 1209 | Invalid confirm flag | 无效的确认标志 |
| 1210 | Invalid delegate state | 无效委托状态 |
| 1211 | The query command does not support arbitrage contracts | 该查询指令暂不支持套利合约 |
| 1212 | Query frequency is too fast or the last querying was not finished | 查询频率过快或上次查询未完成 |
| 1213 | The spdOrderID is incorrect to the accountID | 资金账号和柜台委托号不匹配 |
| 1214 | Cannot find the matched order info for the specified contract and order ID | 找不到该合约和柜台委托号对应的成交信息 |
| 1215 | Cannot find the order info by the xspeed order ID | 找不到该柜台委托号对应的委托信息 |
| 1401 | Front communication exception | 应答数据异常 |
| 1900 | Order rejected by exchange | 报单被交易所拒绝 |
| 1905 | Quote order rejected by exchange | 做市商报价报单失败被交易所拒绝 |
| 1910 | Cancellation action refused by exchange | 撤单失败,被交易所拒绝 |
| 1920 | Quote cancellation action refused by exchange | 做市商报价撤单失败,被交易所拒绝 |
| 2001 | Failed to open the database connection | 数据库建立连接失败 |
| 2006 | System does not allow nullptr password entry | 系统不允许空密码登录 |
| 2102 | Customer number does not exist | 客户号不存在 |
| 2103 | Trade password error. | 交易密码有误 |
| 2601 | Invalid trade category. | 无效的交易类别 |
| 2603 | Invalid investment category, the exchange can only speculation or hedging transactions | 无效的投资类别,该交易所只能投机或套保交易 |
| 2607 | Trade number does not exist. | 交易编码不存在 |
| 2608 | Trade number error. | 交易编码对应该错误 |
| 2609 | Trade number was frozen. | 该交易编码已冻结,不能交易 |
| 2610 | Trade number was forbid to open position. | 该交易编码禁止开新仓 |
| 2102 | The customer number orresponding to the trade number c does not exist | 该交易编码所对应的客户号不存在 |
| 2612 | The customer cannot use this delegate way | 该客户不能通过此种委托方式委托 |
| 2613 | The customer status is abnormal, can not trade. | 该客户状态异常,不能交易 |
| 2614 | The customer is forbidden to open or close position. | 该客户禁止开仓/该客户禁止平仓 |
| 2619 | Invaild contract id | 合约代码无效,不能委托 |
| 2620 | This contract can not trade. | 该合约代码现在不能交易 |
| 2622 | This contract price tick is 0, can not trade. | 该合约代码参数中变动价位为0,不能交易 |
| 2623 | This contract price tick is changed. | 该合约代码的交易变动价格是 |
| 2626 | The trade number is forbidden to trade this variety. | 该交易编码禁止交易该品种 |
| 2267 | Invalid order amount. | 委托数量无效,应该是开仓单位的整数倍 |
| 2635 | Invalid order price is out of price cap or floor,cannot trade | 委托价格超过涨跌停板价格,不能交易 |
| 2636 | System not initialized | 系统未初始化 |
| 2641 | Entrust failed. | 委托失败,插入品种保证金汇总表失败 |
| 2642 | Entrusted failure, lack of customer funds. | 委托失败,客户资金不足 |
| 2661 | Trade number can not entrust hedge order. | 交易编码不能做套保 |
| 2662 | Trade number can not entrust speculate order. | 交易编码不能做投机 |
| 2709 | Invalid order price. | 您输入的委托价格无效 |
| 2710 | Exchange not support market order. | 本交易所不支持市价指令 |
| 2715 | The stop loss order of the exchange must provide a profit(loss) margin. | 交易所止损止盈指令必须提供止损止盈价 |
| 2720 | DCE V5 not support market order. | 大连V5不支持市价指令 |
| 2721 | DCE V5 not support property order. | 大连V5不支持带属性的报单 |
| 2722 | Unable to determine whether the transaction period. | 无法判断是否交易时间段 |
| 2723 | Non trading time can not entrust. | 非交易时间不能下单 |
| 2800 | Entrust failed, not support trading order. | 委托失败:目前不支持的交易指令 |
| 2801 | Invalid combine contract. | 组合代码无效 |
| 2804 | Price margin beyond limit. | 价差超出限制 |
| 2805 | Order amount beyond limit. | 委托数量超出限制 |
| 2807 | Trade number error. | 交易编码对应该错误 |
| 2811 | Customer can not entrust this kind of order. | 该客户不能通过此种委托方式委托 |
| 2610 | The customer is forbidden to open position. | 该客户禁止开仓 |
| 2814 | The customer is forbidden to close position. | 该客户禁止平仓 |
| 3698 | Invalid combine contract. | 组合合约,非法 |
| 3401 | Order does not exist | 合同号对应的委托不存在 |
| 3402 | Client information not match. | 原委托的客户信息与输入的客户信息不匹配! |
| 3404 | Can not cancel order. | 不能撤单! |
| 3406 | Not receive response from Exchange yet, cancel later. | 该委托交易所未回复结果,暂时不允许撤单,请稍后再撤 |
| 3407 | Last cancel action not be sent, cancel later. | 上笔撤单未申报,请稍后再撤 |
| 3408 | Modify blocked balances of account error. | 修改资金帐户委托冻结余额出错 |
| 3411 | Cancellation failed, cancellation number less than thaw positions | 撤单失败,持仓解冻小于撤单数量 |
| 3412 | Modify order record error, cancel again. | 修改委托记录失败,请重新再撤 |
| 3415 | Record cancel application error. | 记录撤单申请出错 |
| 11001 | No FENS server to connect. | 没有FENS服务器可以连接 |
| 11002 | FENS server connect failed. | FENS服务器连接失败 |
| 11003 | Invalid parameter. | 错误的参数 |
| 11004 | Duplicate data. | 数据重复 |
| 11005 | No front get from FENS. | 没有从FENS得来的前置可以连接 |
| 11006 | Connect front failed through FENS. | 经FENS连接前置失败 |
| 11007 | Invalid quote contract or no market maker. | 无效合约或者没有做市商做市 |
| 11008 | Invalid quote interval. | 无效报价区间 |
| 11009 | Already for quote. | 已报过价 |
| 11010 | Have reasonable quotation,cannot for quote. | 有合理报价单，不能报价 |
| 11011 | Variety status is not in trading state. | 品种不在交易状态 |
| 11012 | Market maker not allow for quote. | 做市商不允许报价 |
| 11013 | Trader client check failed. | 做市商会员客户检查不通过 |
| 11014 | Client check failed. | 会员客户检查不通过 |
| 11015 | Server config error,member\_no and trader\_no are not match. | 服务器配置错误，会员和席位不匹配 |
| 11016 | Member not match. | 会员不匹配 |
| 11017 | CA check failed. | CA校验失败 |
| 11018 | Login failed. | 登录失败 |
| 11019 | Trader have no permission. | 会员无权限 |
| 11020 | Send exchange board failed. | 发送交换机机失败 |
| 11021 | Client not matched. | 用户不符 |
| 11022 | Password change failed. | 密码修改失败 |
| 11023 | CA certification package error. | Ca认证请求包错误 |
| 11024 | CA already certified. | 已经经过CA认证 |
| 11025 | Session not certified by CA | 没有经过CA认证Session |
| 11026 | Session certification package error. | Session认证请求包错误 |
| 11027 | Login package error. | 登录包错误 |
| 11028 | Already login. | 已经登录 |
| 11029 | Trader can not super login. | 交易员不能超级登录 |
| 11030 | Order source check failed | 定单来源检查错误 |
| 11031 | Logout. | 登出 |
| 11032 | Already sent order with the same local number. | 此本地报单号对应的定单已报入 |
| 11033 | Trader number not exsited. | 席位不存在 |
| 11034 | No trade code | 交易编码不存在//交易编码不存在 |
| 11035 | Client check failed. | 客户检查错误 |
| 11036 | Invalid contract type. | 无效合约类型 |
| 11037 | Invalid order type. | 无效订单类型 |
| 11038 | Invalid open close flag. | 无效的开平标志 |
| 11039 | Invalid buy sell flag. | 无效买卖标志. |
| 11040 | Invalid speculate hedge flag. | 无效的投保标志 |
| 11041 | Invalid order property. | 无效的订单属性 |
| 11042 | This order category can only have one order. | 此定单类别只能下一个定单 |
| 11043 | Batch order can only have two orders. | 批量订单只能有两个单子 |
| 11044 | Order type of batch order is disaccord. | 批量定单两腿定单类别不一致 |
| 11045 | Trader client check failed.. | 会员客户检查不通过 |
| 11046 | Already canceled the account. | 已经销户 |
| 11047 | Seat does not exsit. | 席位不存在 |
| 11048 | Seat not belong to member. | 席位不属于会员 |
| 11049 | According to contract transaction permission check, contract can not trade. | 合约交易权限检查，合约不可以交易 |
| 11050 | According to contract transaction permission check, contract can only be closed. | 合约交易权限检查，合约只可平仓 |
| 11051 | The trade code was forbidden trading this kind of contract. | 交易编码禁止交易此合约 |
| 11052 | According to the client transaction permission check, contract can only be closed. | 客户交易权限检查，合约只可平仓 |
| 11053 | According to the trader transaction permission check, contract can not trade. | 席位交易权限检查，合约不可以交易 |
| 11054 | According to the trader transaction permission check, contract can only be closed. | 席位交易权限检查，合约只可平仓 |
| 11055 | Invalid combination contract. | 无效的组合合约 |
| 11056 | Invalid contract. | 合约无效 |
| 11057 | Batch orders can only be an option and a futures orders. | 批量订单只可能一个期权定单和一个期货定单 |
| 11058 | Unlisted contract. | 合约未上市 |
| 11059 | Invalid order amount. | 无效的委托数量. |
| 11060 | Price tick cannot be zero. | 价格最小变动单位不能为0,基础数据错 |
| 11061 | The order price mismatches the float-price of the contract. | 合约的委托价格与浮动价位不匹配 |
| 11062 | Contract market does not exist. | 合约行情不存在 |
| 11063 | Invalid order price is out of price cap or floor, cannot trade. | 委托价不在涨跌停范围内 |
| 11064 | Stop-loss price out of limit price. | 止损价位不在涨跌幅度内 |
| 11065 | Stop loss price of buy direction profit stop order, must less than the trigger price. | 买方向止损价格必须小于触发价格 |
| 11066 | Stop loss price of sell direction profit stop order, must less than the trigger price. | 卖方向止损价格必须小时触发价格 |
| 11067 | Stop profit price of buy direction profit stop order, must less than the trigger price. | 买方向止盈价格必须小于触发价格 |
| 11068 | Stop profit price of sell direction profit stop order, must less than the trigger price. | 卖方向止盈价格必须小于触发价格 |
| 11069 | Arbitrage contract makret does not exist. | 套利合约行情不存在 |
| 11070 | Order status checked failed. | 订单状态检查失败 |
| 11071 | The order already canceled, cannot cancel it again. | 订单已撤单，不能再撤单 |
| 11072 | Full matched, cannot cancel. | 全部成交，不能在撤单 |
| 11073 | Order does not exist. | 委托单不存在 |
| 11074 | This member have no settlement account. | 此会员无结算帐号 |
| 11075 | Arbitrage strategy rules table does not exist the record. | 套利策略规则表里不存在此记录 |
| 11076 | Can't get the fixed deposit. | 取不到固定保证金 |
| 11077 | Variety status is not in trading state. | 合约品种不在交易状态 |
| 11078 | Trade code is forbidden to open position. | 交易编码禁止开仓 |
| 11079 | Client does not exist. | 客户不存在 |
| 11080 | The option price must be greater than TICK. | 期权价格必须大于TICK |
| 11081 | The option stop-loss price must be greater than TICK. | 期权止损价格必须大于TICK |
| 11082 | Order source check failed. | 定单来源检查错误 |
| 11083 | There is no option combination margin ratio. | 看涨（跌）期权组合的保证金比例表无对应的组合记录 |
| 11084 | The force-close order does not allow to cancel. | 强平单不能撤单 |
| 11085 | Can't cancel others' order. | 不能撤销他人的单子 |
| 11086 | Market maker sign error. | 做市商标记错误 |
| 11087 | Bilateral quotation command must have two orders. | 双边报价指令要求两个单子 |
| 11088 | Bilateral quotation command must be a limit order. | 双边报价指令必须为限价委托 |
| 11089 | Bilateral quotation command must not have order property. | 双边报价指令必须无定单属性委托 |
| 11090 | Bilateral quotation command two legs must be market makers. | 双边报价指令两条腿必须都是做市商 |
| 11091 | Bilateral quotation command two legs must be equal amount. | 双边报价指令两腿委托数量相等 |
| 11092 | Bilateral quotation command two legs must be the same contract. | 双边报价指令两腿合约相同 |
| 11093 | One of bilateral quotation two legs must be buy direction, another must be sell direction. | 双边报价指令两腿必须一买一卖 |
| 11094 | The contract has been suspended. | 合约已经停牌 |
| 11095 | Quote order must set speculator type. | 报价委托必须设置投保类型 |
| 11096 | Only market maker can send bilateral quotation. | 只有做市商可以下双边报价指令 |
| 11097 | Bilateral quotation command two legs must be the same order sort. | 双边报价指令两腿定单类别不一致 |
| 11098 | Only allow a cancellations. | 只允许一个撤单域 |
| 11099 | Bilateral quotation command two legs must be the same client. | 双边报价指令两腿客户号不一致 |
| 11100 | Bilateral quotation command sell price must greater than buy price. | 双边报价指令卖价必须大于买价 |
| 11101 | Varieties of trading permission does not exist. | 品种交易权限不存在 |
| 11102 | The variety does not support bilateral quotations. | 该品种不支持双边报价定单 |
| 11103 | The variety only support marketmakers bilateral quotation. | 该品种只支持做市商下双边报价定单 |
| 11104 | Market makers have no bilateral quotation permission to trade the variety. | 做市商在该品种无双边报价权限 |
| 11105 | Bilateral quotation order two legs must be the same order sort. | 双边报价定单两腿定单类别不一致 |
| 11106 | Not support market price order in the exchange. | 交易所不支持市价单 |
| 11107 | Funds of settlement account does not exist. | 结算帐号对应的交易资金不存在 |
| 11108 | Arbitrage contract does not exist. | 套利合约不存在 |
| 11109 | Arbitrage strategy rules table does not exist the record. | 套利策略规则表里不存在此记录 |
| 11110 | No system number corresponding record in the order table. | 委托表没有系统号对应的记录 |
| 11111 | Order does not exist. | 委托单不存在 |
| 11112 | Contract market does not exist. | 合约行情不存在 |
| 11113 | Batch number corresponding conposition details does not exist. | 成交批次号对应的持仓明细不存在 |
| 11114 | Seat does not exist. | 席位不存在 |
| 11115 | The seat member relationship is not correct. | 席位会员关系不正确 |
| 11116 | Incorrect seat password. | 席位密码不正确！ |
| 11117 | Seat have been logged in. | 席位已经登录！ |
| 11118 | Seat have been logged out. | 席位已经登出！ |
| 11119 | Seat not login. | 席位没有登录！ |
| 11120 | Control mode change failed. | 控制模式改变失败！ |
| 11121 | Market status change failed. | 市场状态改变失败！ |
| 11122 | No trade code. | Notradecode. |
| 11123 | Incorrect user password. | 用户密码不正确！ |
| 11124 | User already login. | 用户已经登录！ |
| 11125 | User already logout. | 用户已经登出！ |
| 11126 | User not login. | 用户没有登录！ |
| 11127 | Not allowed to change status. | 不允许从当前状态直接转变到目的地状态！ |
| 11128 | Non super login cannot disable CA. | 非超级登录不能禁用CA！ |
| 11129 | The new password should be between 8 and 16 bits. | 新密码长度应该在8到16位！ |
| 11130 | Password modification difference is not enough. | 密码修改的差异度不够！ |
| 11131 | Exchange shop corresponding funds have normal storage. | 票号对应的资金已经正常入库！ |
| 11132 | Not enough funds. | 出金资金不够！ |
| 11133 | Not allowed to rollout funds during the transaction. | 场上不允许出金！ |
| 11134 | Failed to send packet. | 发送数据包失败！ |
| 11135 | Check no record. | 查无记录！ |
| 11136 | Initial login, please modify the initial password, or the next can not log on. | 初次登录，请修改初始密码，否则下次不能登录 |
| 11137 | Seat frozen. | 席位被冻结！ |
| 11138 | The trade code has been frozen,cannot trade. | 交易编码被冻结，不能交易 |
| 11139 | Password valid parameter does not exist. | 密码有效参数不存在！ |
| 11140 | Logon failure valid parameter does not exist. | 登录失败有效参数不存在！ |
| 11141 | Login password has expired, please contact the system administrator to reset the password. | 登录密码已过期，请联系系统管理员重置密码！ |
| 11142 | There is no settlement account in the account table. | 结算账号表中不存在对应的结算账户！ |
| 11143 | There is no record of the minimum balance sheet of the settlement reserve. | 结算准备金最低余额表不存在相应的记录！ |
| 11144 | Option execution request failed. | 期权执行申请失败！ |
| 11145 | Option execution request does not exist. | 期权执行申请单不存在！ |
| 11146 | User already exists. | 用户已经存在！ |
| 11147 | Contract trading authority already exists. | 合约交易权限已经存在！ |
| 11148 | Contract trading authority does not exist. | 合约交易权限不存在！ |
| 11149 | Seat trading authority already exists. | 席位交易权限已经存在！ |
| 11150 | Seat trading authority does not exist. | 席位交易权限不存在！ |
| 11151 | Client trading authority does not exist. | 客户交易权限不存在！ |
| 11152 | Client trading authority already exists. | 客户交易权限已经存在！ |
| 11153 | Incorrect trading authority. | 交易权限不正确 |
| 11154 | Password expiration prompt time does not exist. | 密码到期提示时间不存在！ |
| 11155 | Operation type error. | 操作类型错误！ |
| 11156 | Exchange code corresponding to the exchange does not exist. | 交易所代码对应的交易所不存在！ |
| 11157 | No client positions. | 没有客户持仓！ |
| 11158 | Client does not exist. | 客户不存在！ |
| 11159 | Do not need funds to compulsively close positions. | 不需要资金不够强平！ |
| 11160 | The system is not in a state of suspension and cannot to generate compulsory close positions order manually. | 系统未处于暂停状态，不能手工生成强平单！ |
| 11161 | Already generated compulsory close positions, whether or not generate again. | 本日已做过生成强平处理，是否再次生成！ |
| 11162 | Today have no compulsory close positions. | 本日还没强平！ |
| 11163 | The same transaction day does not allow the reverse time switch state. | 同一个交易日不允许倒着时间切换状态 |
| 11164 | Client check failed. | 会员客户检查不通过！ |
| 11165 | Client preference portfolio positions no record. | 客户优惠组合持仓表无相应记录！ |
| 11166 | Client arbitrage positions no record. | 客户套利持仓表无相应记录 |
| 11167 | There is no portfolio contract in the portfolio. | 组合优惠表不存在组合合约 |
| 11168 | There is no portfolio contract in the discount portfolio. | 优惠组合比例表不存在组合合约 |
| 11169 | Market seats are not allowed to modify the password in the transaction background. | 行情席位不允许在交易后台修改密码 |
| 11170 | Automatic mode now. | 当前是自动模式 |
| 11171 | Incorrect IP address. | IP地址不正确 |
| 11172 | Trader money deficiency. | 会员资金不足 |
| 11173 | The the credit of seats is not enough. | 席位信用额度不够 |
| 11174 | No trading margin. | 交易编码无交易保证金 |
| 11175 | Order sort is not option exection. | 定单类别不是期权执行申请 |
| 11176 | Failed to send packet. | 发送数据包失败 |
| 11177 | No records. | 查无记录 |
| 11178 | There is no record in the arbitrage strategy. | 套利策略规则表里不存在此记录 |
| 11179 | Funds of settlement account does not exist. | 结算帐号对应的交易资金不存在 |
| 11180 | Members are self-employed but client is not. | 会员是自营,客户不是自营 |
| 11181 | Open position amount reach proprietary trading upper limit. | 开仓数量达到交易上限 |
| 11182 | Members are brokers but client is not. | 会员是经纪,客户不是经纪 |
| 11183 | Open position exceed member agent position limit. | 开仓超过会员持仓限额 |
| 11184 | Open positions over comprehensive position limits of member. | 开仓超过会员综合限仓 |
| 11185 | Open position amount reach upper limit. | 开仓数量达到上限 |
| 11186 | Not enough position can be closed. | 持仓不足，不能平仓 |
| 11187 | Client without position. | 会员无持仓 |
| 11188 | The remaining amount of the customer's hedge not enough. | 客户剩余保值额度不足 |
| 11189 | Hedge volume not enough. | 套保额度不足 |
| 11190 | This group is not allowed to cancel the order in this transaction. | 会员超仓，不可开仓 |
| 11191 | Individual customers cannot open the contract in delivery month. | 个人客户不能在处于交割月的合约上开仓 |
| 11192 | The client of this group is not a market maker. | 本地报单号重复 |
| 11193 | Member positions exceed the limit position, can not open. | 本撮合组本交易状态下，不允许撤销定单 |
| 11194 | The end of the batch cancel action in this group. | 本撮合组该交易编码不是做市商 |
| 11195 | The local order number is repeated. | 本撮合组批量撤单结束 |
| 11196 | The relationship of quotation seat and member is not correct. | 行情席位会员关系不正确 |
| 11197 | Quotation seat not login. | 行情席位没有登录 |
| 11198 | Incorrect quotation seat password. | 行情席位密码不正确 |
| 11199 | Quotation seat new password should be between 8 and 16 bits. | 行情席位新密码长度应该在8到16位 |
| 11200 | Quotation seat password modification difference is not enough. | 行情席位密码修改的差异度不够 |
| 11201 | Quotation seat does not exist. | 行情席位不存在 |
| 11202 | No response. | 无应答 |
| 11203 | Forwarding failure. | 转发失败 |
| 11204 | Reply data error. | 应答数据错误 |
| 11205 | FTCP domain error. | FTCP域错误 |
| 11206 | Message driven thread has not been created | 消息驱动线程尚未创建 |
| 11207 | CA module has not been created | CA模块尚未创建 |
| 11208 | API Control module has not been created | APIControl模块尚未创建 |
| 11209 | Failed to create TCP connection | 创建TCP连接失败 |
| 11210 | Synchronous call timeout | 同步调用超时错误 |
| 11211 | Physical connection error | 物理连接出错 |
| 11212 | CA authentication module signature error | CA认证模块签名错误 |
| 11213 | Repeated authentication | 重复认证 |
| 11214 | Verify server signature failed | 验证服务器端签名失败 |
| 11215 | Repeated login | 重复登录 |
| 11216 | Not passed CA certification | 尚未通过CA认证 |
| 11217 | Not logged in | 尚未登录前置 |
| 11218 | CA module initialization failed | CA模块初始化失败 |
| 11219 | No front can be connected | 没有前置可以连接 |
| 11220 | Sync timeout cannot be zero | 同步超时不能为零 |
| 11221 | Repeated session request | 重复会话请求 |
| 11222 | No session request before authentication | 在认证之前尚未通过会话请求 |
| 11223 | Session and authentication seats are different | 会话和认证过程中席位号不同 |
| 11224 | CA certificate initialization error | CA证书初始化错误 |
| 11225 | Failed to authenticate CA certificate | 认证应答CA证书解密失败 |
| 11226 | Error loading dynamic library | 装载动态库出错 |
| 11227 | Read and write private key device failed | 读写私钥设备失败 |
| 11228 | Private key password error | 私钥密码错误 |
| 11229 | Initialization error | 初始化错误 |
| 11230 | Memory alloc error | 内存分配错误 |
| 11231 | Validity of the certificate error | 获取证书有效期错误 |
| 11232 | Market status not allow | 市场状态不允许 |
| 11233 | Empty member id | 会员号不能为空 |
| 11234 | Instrument not exsit | 合约不存在 |
| 11235 | Invalid member id | 会员客户检查不通过 |
| 11236 | Invalid number | 数量不合理 |
| 11237 | Record not exsit | 记录不存在 |
| 11238 | Record has been canceled | 记录已撤销 |
| 11239 | Offset set error | 对冲设置错误 |
| 11240 | Not in last trading day | 非到期日不能进行设置 |
| 11241 | Duplicate set | 重复设置 |
| 11242 | Perfshtype error | 投保标志设置错误 |
| 11243 | Req options exercise failed | 期权执行申请失败 |
| 11244 | Req perform offset failed | 履约对冲设置失败 |
| 11245 | Cancel perfrom offset failed | 履约对冲撤销失败 |
| 11246 | Req options offset failed | 期权对冲设置失败 |
| 11247 | Cancel options offset failed | 期权对冲撤销失败 |
| 11248 | Req auto\_exec\_aband failed | 放弃自动行权设置失败 |
| 11249 | Cancel auto\_exec\_aband failed | 放弃自动行权撤销失败 |
| 11250 | Contract type not options | 合约类型不是期权 |
| 11251 | Variety trading status not exsit | 品种交易状态不存在 |
| 11252 | Send data package failed | 发送数据包失败 |
| 11253 | Variety trading status not allowed | 品种交易状态不允许 |
| 11254 | CombAction must be comb or unlock | 组合状态必须是组合或解锁 |
| 11255 | Combination position not enough | 优惠组合持仓量不足 |
| 11256 | Two leg trading status is not the same | 组合合约两腿所属品种的交易状态不一致 |
| 11257 | Single leg position is not enough | 单腿持仓不足 |
| 11258 | Only allow one data field | 只允许一个数据域 |
| 11259 | The SHflag of two legs are different | 双边订单两腿投保标志不一致 |
| 11260 | The customer didn't have hedge qualification in this variety | 该客户在该品种交易类型上没有套保资格 |
| 11261 | The customer didn't have speculative qualification in this variety | 该客户在该品种交易类型上不能投机开仓 |
| 11262 | This contract is non-tradable | 实控组交易权限检查，合约不可以交易 |
| 11263 | This contract only allowed close position | 实控组交易权限检查，合约只可平仓 |
| 11264 | Can't speculate in delivery month | 个人客户不能在交割月的合约上开投机仓 |
| 11265 | Speculate volume over control group limit | 开仓数量超过实控组投机限制 |
| 11266 | Hedge volume over control group limit | 开仓数量超过实控组套保限制 |
| 11267 | Multiple volume over control group limit | 开仓数量超过实控组综合限制 |
| 11268 | Control account not exist | 无此实控关系账户 |
| 11269 | Open volume over contract limit | 开仓超过当日客户在合约 |
| 11270 | Open volume over variety limit | 开仓超过当日客户在品种上开仓限额 |
| 11271 | Control group open volume over contract limit | 开仓超过当日实控组在合约 |
| 11272 | Control group open volume over variety limit | 开仓超过当日实控组在品种上开仓限额 |
| 11273 | Overseas customer open volume over limit | 开仓数量超过境外客户组综合限制 |
| 11274 | Overseas customer account not exist | 无此境外客户组账户 |
| 11275 | Not support this contract | 本撮合主机不支持该合约 |
| 11276 | Basic order not allow multiple orders | 基本定单不允许多个定单 |
| 11277 | No this member | 无此会员 |
| 11278 | No this customer | 无此客户 |
| 11279 | Trade status does not accept this order | 本交易状态不接受该类定单 |
| 11280 | Trade status error,stop service | 交易状态错误，暂停服务 |
| 11281 | Arbitrage order is illegal | 套利定单非法 |
| 11282 | The amount of compulsory close positions is illegal | 强平单数量非法 |
| 11283 | Message is illegal | 非法报文 |
| 11284 | There is no orders to be canceled | 批量撤单请求，没有需要撤的定单 |
| 11285 | Order is canceled by system | 定单被系统撤销 |
| 11286 | There is no such cancel order | 无此撤销定单 |
| 11287 | Change order invalid | 修改定单无效 |
| 11288 | Change order amount invalid | 修改定单数量非法 |
| 11289 | Seat password out of data | 席位密码已过期，请联系管理员重置密码 |