

---

# 交易 API 和行情 API 接口规范

Version: 2.00 版本  
发布日期: 2025 年 09 月 12 日

## I. 修订记录、核准记录和审核记录

### 修订记录

| 版本编号     | 修订日期     | 主要修订摘要 |
|----------|----------|--------|
| v2.00 版本 | 20250829 |        |

### 核准记录

| 核准人员 | 属于部门（单位） | 核准日期 |
|------|----------|------|
|      |          |      |
|      |          |      |

### 审核记录

| 审核人员 | 属于部门（单位） | 审核日期 |
|------|----------|------|
|      |          |      |
|      |          |      |

# 目 录

|                                      |    |
|--------------------------------------|----|
| 第一部分 交易系统接口介绍 .....                  | 1  |
| 1. 介绍 .....                          | 2  |
| 1.1. 背景 .....                        | 2  |
| 1.2. TraderAPI 简介 .....              | 2  |
| 1.3. MduserAPI 简介 .....              | 3  |
| 1.4. TraderAPI/MduserAPI 支持的平台 ..... | 4  |
| 1.5. 联系方式 .....                      | 4  |
| 1.6. 版本历史 .....                      | 4  |
| 1.6.1. v2.00 版本 .....                | 4  |
| 2. FTD 体系结构 .....                    | 7  |
| 2.1. 通讯模式 .....                      | 7  |
| 2.2. 数据流 .....                       | 8  |
| 3. 接口模式 .....                        | 10 |
| 3.1. TraderAPI 接口 .....              | 10 |
| 3.1.1. 对话流和查询流编程接口 .....             | 10 |
| 3.1.2. 私有流编程接口 .....                 | 11 |
| 3.1.3. 公共流编程接口 .....                 | 11 |
| 3.2. MduserAPI 接口 .....              | 12 |
| 3.2.1. 对话流编程接口 .....                 | 12 |
| 3.2.2. 行情流编程接口 .....                 | 13 |
| 4. 运行模式 .....                        | 14 |
| 4.1. 工作流程 .....                      | 14 |
| 4.1.1. 初始化阶段 .....                   | 14 |
| 4.1.2. 功能调用阶段 .....                  | 14 |
| 4.2. 工作线程 .....                      | 15 |
| 4.3. 与交易系统的连接 .....                  | 16 |
| 4.4. TraderAPI 与交易前置的交互 .....        | 16 |
| 4.5. MduserAPI 与行情前置的交互 .....        | 18 |
| 4.6. 本地文件 .....                      | 20 |
| 4.7. 请求和应答日志文件 .....                 | 20 |
| 4.8. 可靠数据流的订阅方式 .....                | 21 |
| 4.8.1. API 维护重传报文的序号 .....           | 21 |
| 4.8.2. 会员端系统维护重传报文的序号 .....          | 22 |
| 4.9. 心跳机制 (Heartbeat) .....          | 22 |
| 4.10. 灾备接口 .....                     | 23 |
| 第二部分 TraderAPI 参考手册 .....            | 25 |
| 1. TraderAPI 接口分类 .....              | 26 |
| 1.1. 管理接口 .....                      | 26 |
| 1.2. 业务接口 .....                      | 26 |
| 2. TraderAPI 接口说明 .....              | 30 |
| 2.1. CShfeFtdcTraderSpi 接口 .....     | 30 |

|         |                                   |    |
|---------|-----------------------------------|----|
| 2.1.1.  | OnFrontConnected 方法 .....         | 30 |
| 2.1.2.  | OnFrontDisconnected 方法 .....      | 30 |
| 2.1.3.  | OnHeartBeatWarning 方法 .....       | 31 |
| 2.1.4.  | OnPackageStart 方法 .....           | 31 |
| 2.1.5.  | OnPackageEnd 方法 .....             | 32 |
| 2.1.6.  | OnRspUserLogin 方法 .....           | 32 |
| 2.1.7.  | OnRspUserLogout 方法 .....          | 34 |
| 2.1.8.  | OnRspUserPasswordUpdate 方法 .....  | 35 |
| 2.1.9.  | OnRspSubscribeTopic 方法 .....      | 36 |
| 2.1.10. | OnRspQryTopic 方法 .....            | 37 |
| 2.1.11. | OnRspError 方法 .....               | 38 |
| 2.1.12. | OnRspOrderInsert 方法 .....         | 38 |
| 2.1.13. | OnRspOrderAction 方法 .....         | 42 |
| 2.1.14. | OnRspQuoteInsert 方法 .....         | 44 |
| 2.1.15. | OnRspQuoteAction 方法 .....         | 47 |
| 2.1.16. | OnRspExecOrderInsert 方法 .....     | 49 |
| 2.1.17. | OnRspExecOrderAction 方法 .....     | 51 |
| 2.1.18. | OnRspQryPartAccount 方法 .....      | 53 |
| 2.1.19. | OnRspQryOrder 方法 .....            | 54 |
| 2.1.20. | OnRspQryQuote 方法 .....            | 57 |
| 2.1.21. | OnRspQryTrade 方法 .....            | 59 |
| 2.1.22. | OnRspQryClient 方法 .....           | 61 |
| 2.1.23. | OnRspQryPartPosition 方法 .....     | 63 |
| 2.1.24. | OnRspQryClientPosition 方法 .....   | 64 |
| 2.1.25. | OnRspQryInstrument 方法 .....       | 66 |
| 2.1.26. | OnRspQryInstrumentStatus 方法 ..... | 68 |
| 2.1.27. | OnRspQryBulletin 方法 .....         | 69 |
| 2.1.28. | OnRspQryMarketData 方法 .....       | 71 |
| 2.1.29. | OnRspQryHedgeVolume 方法 .....      | 73 |
| 2.1.30. | OnRtnTrade 方法 .....               | 74 |
| 2.1.31. | OnRtnOrder 方法 .....               | 76 |
| 2.1.32. | OnRtnQuote 方法 .....               | 78 |
| 2.1.33. | OnRtnExecOrder 方法 .....           | 80 |
| 2.1.34. | OnRtnInstrumentStatus 方法 .....    | 81 |
| 2.1.35. | OnRtnInsInstrument 方法 .....       | 82 |
| 2.1.36. | OnRtnBulletin 方法 .....            | 83 |
| 2.1.37. | OnRtnFlowMessageCancel 方法 .....   | 84 |
| 2.1.38. | OnErrRtnOrderInsert 方法 .....      | 85 |
| 2.1.39. | OnErrRtnOrderAction 方法 .....      | 88 |
| 2.1.40. | OnErrRtnQuoteInsert 方法 .....      | 90 |
| 2.1.41. | OnErrRtnQuoteAction 方法 .....      | 92 |
| 2.1.42. | OnErrRtnExecOrderInsert 方法 .....  | 94 |
| 2.1.43. | OnErrRtnExecOrderAction 方法 .....  | 96 |
| 2.1.44. | OnRspQryExecOrder 方法 .....        | 98 |

|         |                                         |     |
|---------|-----------------------------------------|-----|
| 2.1.45. | OnRspQryExchangeRate 方法 .....           | 100 |
| 2.1.46. | OnRspAbandonExecOrderInsert 方法 .....    | 101 |
| 2.1.47. | OnRspAbandonExecOrderAction 方法 .....    | 103 |
| 2.1.48. | OnRspQryAbandonExecOrder 方法 .....       | 105 |
| 2.1.49. | OnRtnAbandonExecOrder 方法 .....          | 107 |
| 2.1.50. | OnErrRtnAbandonExecOrderInsert 方法 ..... | 108 |
| 2.1.51. | OnErrRtnAbandonExecOrderAction 方法 ..... | 110 |
| 2.1.52. | OnRspQuoteDemand 方法 .....               | 112 |
| 2.1.53. | OnRtnQuoteDemandNotify 方法 .....         | 113 |
| 2.1.54. | OnRspOptionSelfCloseUpdate 方法 .....     | 114 |
| 2.1.55. | OnErrRtnOptionSelfCloseUpdate 方法 .....  | 116 |
| 2.1.56. | OnRtnOptionSelfCloseUpdate 方法 .....     | 118 |
| 2.1.57. | OnRspOptionSelfCloseAction 方法 .....     | 119 |
| 2.1.58. | OnErrRtnOptionSelfCloseAction 方法 .....  | 121 |
| 2.1.59. | OnRspQryOptionSelfClose 方法 .....        | 122 |
| 2.1.60. | OnRspAuthenticate 方法 .....              | 124 |
| 2.2.    | CShfeFtdcTraderApi 接口 .....             | 125 |
| 2.2.1.  | CreateFtdcTraderApi 方法 .....            | 126 |
| 2.2.2.  | GetVersion 方法 .....                     | 126 |
| 2.2.3.  | Release 方法 .....                        | 126 |
| 2.2.4.  | Init 方法 .....                           | 127 |
| 2.2.5.  | Join 方法 .....                           | 127 |
| 2.2.6.  | GetTradingDay 方法 .....                  | 127 |
| 2.2.7.  | RegisterSpi 方法 .....                    | 128 |
| 2.2.8.  | RegisterFront 方法 .....                  | 128 |
| 2.2.9.  | RegisterNameServer 方法 .....             | 128 |
| 2.2.10. | SetHeartbeatTimeout 方法 .....            | 129 |
| 2.2.11. | OpenRequestLog 方法 .....                 | 130 |
| 2.2.12. | OpenResponseLog 方法 .....                | 130 |
| 2.2.13. | SubscribePrivateTopic 方法 .....          | 131 |
| 2.2.14. | SubscribePublicTopic 方法 .....           | 131 |
| 2.2.15. | SubscribeUserTopic 方法 .....             | 132 |
| 2.2.16. | ReqUserLogin 方法 .....                   | 132 |
| 2.2.17. | ReqUserLogout 方法 .....                  | 134 |
| 2.2.18. | ReqUserPasswordUpdate 方法 .....          | 134 |
| 2.2.19. | ReqSubscribeTopic 方法 .....              | 135 |
| 2.2.20. | ReqQryTopic 方法 .....                    | 136 |
| 2.2.21. | ReqOrderInsert 方法 .....                 | 137 |
| 2.2.22. | ReqOrderAction 方法 .....                 | 139 |
| 2.2.23. | ReqQuoteInsert 方法 .....                 | 140 |
| 2.2.24. | ReqQuoteAction 方法 .....                 | 142 |
| 2.2.25. | ReqExecOrderInsert 方法 .....             | 143 |
| 2.2.26. | ReqExecOrderAction 方法 .....             | 145 |
| 2.2.27. | ReqQryPartAccount 方法 .....              | 146 |

|                           |                                    |     |
|---------------------------|------------------------------------|-----|
| 2.2.28.                   | ReqQryOrder 方法 .....               | 147 |
| 2.2.29.                   | ReqQryQuote 方法 .....               | 148 |
| 2.2.30.                   | ReqQryTrade 方法 .....               | 149 |
| 2.2.31.                   | ReqQryClient 方法 .....              | 150 |
| 2.2.32.                   | ReqQryPartPosition 方法 .....        | 150 |
| 2.2.33.                   | ReqQryClientPosition 方法 .....      | 151 |
| 2.2.34.                   | ReqQryInstrument 方法 .....          | 152 |
| 2.2.35.                   | ReqQryInstrumentStatus 方法 .....    | 153 |
| 2.2.36.                   | ReqQryMarketData 方法 .....          | 154 |
| 2.2.37.                   | ReqQryBulletin 方法 .....            | 154 |
| 2.2.38.                   | ReqQryHedgeVolume 方法 .....         | 155 |
| 2.2.39.                   | ReqQryExecOrder 方法 .....           | 156 |
| 2.2.40.                   | ReqQryExchangeRate 方法 .....        | 157 |
| 2.2.41.                   | ReqAbandonExecOrderInsert 方法 ..... | 158 |
| 2.2.42.                   | ReqAbandonExecOrderAction 方法 ..... | 159 |
| 2.2.43.                   | ReqQryAbandonExecOrder 方法 .....    | 161 |
| 2.2.44.                   | ReqQuoteDemand 方法 .....            | 162 |
| 2.2.45.                   | ReqOptionSelfCloseUpdate 方法 .....  | 163 |
| 2.2.46.                   | ReqOptionSelfCloseAction 方法 .....  | 164 |
| 2.2.47.                   | ReqQryOptionSelfClose 方法 .....     | 165 |
| 2.2.48.                   | ReqAuthenticate 方法 .....           | 166 |
| 3.                        | TraderAPI 接口开发示例 .....             | 168 |
| 第三部分 MduserAPI 参考手册 ..... |                                    | 172 |
| 1.                        | MduserAPI 接口分类 .....               | 173 |
| 1.1.                      | 管理接口 .....                         | 173 |
| 1.2.                      | 业务接口 .....                         | 173 |
| 2.                        | MduserAPI 接口说明 .....               | 175 |
| 2.1.                      | CShfeFtdcMduserSpi 接口 .....        | 175 |
| 2.1.1.                    | OnFrontConnected 方法 .....          | 175 |
| 2.1.2.                    | OnFrontDisconnected 方法 .....       | 175 |
| 2.1.3.                    | OnHeartBeatWarning 方法 .....        | 176 |
| 2.1.4.                    | OnPackageStart 方法 .....            | 176 |
| 2.1.5.                    | OnPackageEnd 方法 .....              | 177 |
| 2.1.6.                    | OnRspUserLogin 方法 .....            | 177 |
| 2.1.7.                    | OnRspUserLogout 方法 .....           | 178 |
| 2.1.8.                    | OnRspSubscribeTopic 方法 .....       | 179 |
| 2.1.9.                    | OnRspQryTopic 方法 .....             | 180 |
| 2.1.10.                   | OnRspError 方法 .....                | 181 |
| 2.1.11.                   | OnRtnDepthMarketData 方法 .....      | 182 |
| 2.1.12.                   | OnRtnFlowMessageCancel 方法 .....    | 184 |
| 2.1.13.                   | OnRspUserPasswordUpdate 方法 .....   | 185 |
| 2.2.                      | CShfeFtdcMduserApi 接口 .....        | 186 |
| 2.2.1.                    | CreateFtdcMduserApi 方法 .....       | 186 |
| 2.2.2.                    | GetVersion 方法 .....                | 187 |

|               |                                   |     |
|---------------|-----------------------------------|-----|
| 2.2.3.        | Release 方法 .....                  | 187 |
| 2.2.4.        | Init 方法 .....                     | 188 |
| 2.2.5.        | Join 方法 .....                     | 188 |
| 2.2.6.        | GetTradingDay 方法 .....            | 188 |
| 2.2.7.        | RegisterSpi 方法 .....              | 188 |
| 2.2.8.        | RegisterFront 方法 .....            | 189 |
| 2.2.9.        | RegisterNameServer 方法 .....       | 189 |
| 2.2.10.       | SetHeartbeatTimeout 方法 .....      | 190 |
| 2.2.11.       | OpenRequestLog 方法 .....           | 190 |
| 2.2.12.       | OpenResponseLog 方法 .....          | 191 |
| 2.2.13.       | SubscribeMarketDataTopic 方法 ..... | 191 |
| 2.2.14.       | ReqUserLogin 方法 .....             | 192 |
| 2.2.15.       | ReqUserLogout 方法 .....            | 193 |
| 2.2.16.       | ReqSubscribeTopic 方法 .....        | 194 |
| 2.2.17.       | ReqQryTopic 方法 .....              | 195 |
| 2.2.18.       | ReqUserPasswordUpdate 方法 .....    | 195 |
| 3.            | MduserAPI 接口开发示例 .....            | 197 |
| 第四部分 附录 ..... |                                   | 200 |
| 1.            | 错误编码列表 .....                      | 201 |
| 2.            | 枚举值列表 .....                       | 206 |
| 3.            | 数据类型列表 .....                      | 210 |
| 4.            | API 返回值列表 .....                   | 213 |

# 第一部分 交易系统接口介绍

第一章介绍了交易系统（简称交易系统）的两个 API 接口，TraderAPI 用于会员系统发送交易和查询指令，接收私有流、公共流、对话流和查询流；MduserAPI 用于会员系统和行情转发商系统接收行情流。

第二章介绍了两个 API 使用的通信协议 FTD(Futures Trading Data Exchange Protocol)，重点说明了数据流。

第三章介绍了两个 API 对应不同类型应用的编程接口。

第四章介绍了 API 的运行模式，包括线程之间的通信、心跳机制和可靠数据流的传输机制。



# 1. 介绍

## 1.1. 背景

在中国证监会的统一领导下，上海期货交易所、郑州商品交易所和大连商品交易所联合研究了期货交易数据交换协议（Futures Trading Data Exchange Protocol，简称 FTD 或者 FTD 协议）。中国证监会于 2005 年 3 月 25 日发布了 FTD（JR/T 0016-2004），并于同日作为行业标准实施。

交易系统本质上使用 FTD 作为会员远程交易的接入协议，FTD 协议较为复杂，为简化会员端远程交易系统的开发难度，同时也为提高交易系统运行的可靠性，上海期货交易所发布了交易系统的交易 API 接口（简称 TraderAPI）和行情 API 接口（简称 MduserAPI）。

会员系统调用 TraderAPI，连接交易系统，发送请求指令，接收响应或回报后，TraderAPI 回调会员系统；会员系统或者行情转发商系统调用 MduserAPI，对接交易系统，接收行情数据后，MduserAPI 回调行情接收系统。

会员和行情转发商接收交易所行情的系统统称为行情接收系统，会员系统和行情接收系统统称为会员端或会员端系统。

TraderAPI 封装了会员系统与交易系统之间复杂的协议转换、数据同步和网络通信。TraderAPI 基于 TCP 协议与交易系统的交易前置建立虚拟链路通信通道，实现会员系统的交易和查询。通过 TraderAPI 建立的连接通道具有多地址注册、断线自动重连和交易数据自动重传等特点。

MduserAPI 与 TraderAPI 类似，建立一个基于 TCP 的虚链路通道与交易系统行情前置连接，实现行情的订阅和接收。

## 1.2. TraderAPI 简介

TraderAPI 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交易功能，包括报单与报价的录入、报单与报价的撤销、执行宣告与放弃执行宣告的录入、执行宣告与放弃执行宣告的撤销、询价、资金查询、报单与报价

的查询、成交单查询、客户查询、会员持仓查询、客户持仓查询、合约查询、合约交易状态查询等功能。

Windows 平台类库支持 Windows32, 包含以下 5 个文件:

| 文件名                   | 文件描述       |
|-----------------------|------------|
| FtdcTraderApi.h       | 交易接口头文件    |
| FtdcUserApiStruct.h   | 数据结构头文件    |
| FtdcUserApiDataType.h | 数据类型头文件    |
| ftdtraderapi.dll      | 动态链接库二进制文件 |
| ftdtraderapi.lib      | 导入库文件      |

建议使用 Visual Studio2017 及以上版本编译器。

类 Linux 平台类库支持红帽企业 Linux (Red Hat Enterprise Linux, RHEL)

7 和银河麒麟高级服务器操作系统 (Kylin) V10, 包含以下 4 个文件:

| 文件名                   | 文件描述       |
|-----------------------|------------|
| FtdcTraderApi.h       | 交易接口头文件    |
| FtdcUserApiStruct.h   | 数据结构头文件    |
| FtdcUserApiDataType.h | 数据类型头文件    |
| libftdtraderapi.so    | 动态链接库二进制文件 |

### 1.3. MduserAPI 简介

MduserAPI 也是一个基于 C++ 的类库, 通过使用和扩展类库提供的接口来实现行情订阅和接收功能。

Windows 平台类库支持 Windows32, 包含以下 5 个文件:

| 文件名                   | 文件描述       |
|-----------------------|------------|
| FtdcMduserApi.h       | 行情接口头文件    |
| FtdcUserApiStruct.h   | 数据结构头文件    |
| FtdcUserApiDataType.h | 数据类型头文件    |
| ftdmdapi.dll          | 动态链接库二进制文件 |
| ftdmdapi.lib          | 导入库文件      |

建议使用 VS2017 及以上版本编译器。

类 Linux 平台类库支持 RHEL7 和 Kylin V10，包含以下 4 个文件：

| 文件名                   | 文件描述       |
|-----------------------|------------|
| FtdcMduserApi.h       | 行情接口头文件    |
| FtdcUserApiStruct.h   | 数据结构头文件    |
| FtdcUserApiDataType.h | 数据类型头文件    |
| libftdmdapi.so        | 动态链接库二进制文件 |

## 1.4. TraderAPI/MduserAPI 支持的平台

目前支持以下平台：

- X86-64/RHEL7：包括.h 文件和.so 文件
- X86/Windows32：包括.h 文件、.dll 文件和.lib 文件
- ARM64/KylinV10：包括.h 文件和.so 文件
- X86-64/KylinV10：包括.h 文件和.so 文件

## 1.5. 联系方式

电话：+86-021-68400802

邮箱：tech@shfe.com.cn

## 1.6. 版本历史

### 1.6.1. v2.00 版本

相对于 API 1.0，本版本主要有以下变化：

- 本版本删除了以下函数接口：
  - TraderAPI 删除 ReqAdminOrderInsert、OnRspAdminOrderInsert 接口。
  - TraderAPI 删除 ReqQryMBLMarketData、OnRspQryMBLMarketData 接口。
  - TraderAPI 删除 RegisterCertificateFile 接口。

- TraderAPI 删除 RegisterGMCertificateFile 接口。
- TraderAPI 删除 OnRtnAliasDefine 接口。
- TraderAPI 删除 ReqCombOrderInsert、OnRspCombOrderInsert、ReqQryCombOrder、OnRspQryCombOrder、OnRtnCombOrder、OnErrRtnCombOrderInsert、OnRtnInsCombinationLeg、OnRtnDelCombinationLeg 接口。
- TraderAPI 删除 ReqQryInformation、OnRspQryInformation 接口
- TraderAPI 删除 OnRtnDelInstrument 接口。
- TraderAPI 删除 ReqQryCreditLimit、OnRspQryCreditLimit 接口。
- TraderAPI 删除 RegisterCryptAlgorithm 接口。
- MduserAPI 删除 RegisterCertificateFile 接口。
- MduserAPI 删除 RegisterGMCertificateFile 接口。
- MduserAPI 删除 RegisterCryptAlgorithm 接口。
- 本版本新增了以下函数接口：
  - MduserAPI 新增“用户密码修改请求”接口，见【第三部分 2.2.18 ReqUserPasswordUpdate 方法】。
  - MduserAPI 新增“打开请求日志文件”和“打开应答日志文件”接口，见【第三部分 2.2.11 OpenRequestLog 方法、2.2.12 OpenResponseLog 方法】。
- 本版本修改了以下函数接口：
  - TraderAPI 修改 RegisterFront、RegisterNameServer 接口，函数参数类型由 char\*修改为 const char\*，见【第二部分 2.2.8 RegisterFront 方法、2.2.9 RegisterNameServer 方法】。
  - MduserAPI 修改 RegisterFront、RegisterNameServer 接口，函数参数类型由 char\*修改为 const char\*，见【第三部分 2.2.8 RegisterFront 方法、2.2.9 RegisterNameServer 方法】。
- 本版本新增了以下函数返回值和错误原因：
  - TraderAPI、MduserAPI 新增请求接口的返回值，见【第四部分“API 返回值列表”】。

- TraderAPI 修改并新增 API 断线原因，见【第二部分 2.1.2 OnFrontDisconnected 方法】。
- MduserAPI 修改并新增 API 断线原因，见【第三部分 2.1.2 OnFrontDisconnected 方法】。
- 本版本增加了以下枚举值：
  - ProductClass 产品类型增加枚举值：SHFE\_FTDC\_PC\_Spread(套利)；
  - HedgeFlag 套保标志类型增加枚举值：SHFE\_FTDC\_HF\_None(无)；
  - OffsetFlag 开平标志类型增加枚举值：SHFE\_FTDC\_OF\_None(无)；
  - TradeType 成交类型增加枚举值：SHFE\_FTDC\_TRDT\_SpreadDerived(套利衍生成交)；
  - PriceSource 成交价类型增加枚举值：SHFE\_FTDC\_PSRC\_Imply(指导价)。

## 2. FTD 体系结构

### 2.1. 通讯模式

FTD 协议中，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。

FTD 涉及的通讯模式共有三种：

- 对话通讯模式
- 私有通讯模式
- 公共通讯模式（即 API 1.0 的广播通讯模式）

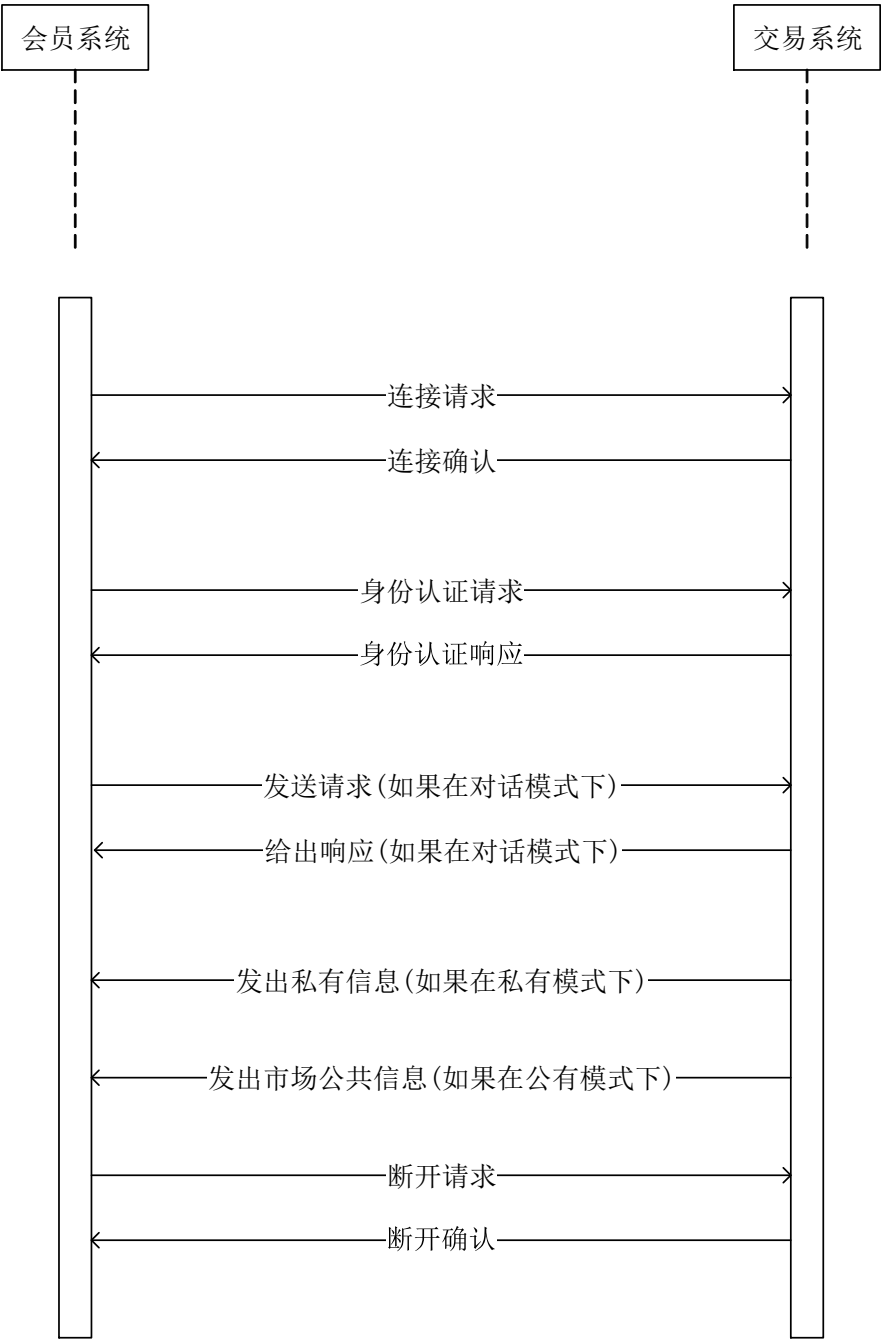
对话通讯模式是指由会员端系统主动发起的通讯请求。该请求被交易系统接收和处理，并给予响应。例如报单、查询等。这种通讯模式与普通的客户/服务器模式相同。

私有通讯模式是指交易系统基于会员端系统的订阅请求，主动向特定的会员或者特定会员的特定交易员发出的信息。例如成交回报、行情通知等。

公共通讯模式是指交易系统主动向市场中的所有会员都发出相同的信息。例如公告、市场公共信息等。

通讯模式和网络连接并非简单的一一对应关系。具体而言，一个网络连接可能传输多种不同通讯模式的报文，而一种通讯模式的报文也能够多个不同的连接中进行传输。

无论哪种通讯模式，其通讯过程都如图 1 所示：



图表 1 各通讯模式的工作过程

## 2.2. 数据流

交易前置支持对话通讯模式、私有通讯模式、公共通讯模式。行情前置支持对话通讯模式、私有通讯模式。

### 1、对话通讯模式

对话通讯模式是双向的，支持对话数据流（简称对话流）和查询数据流（简

称查询流)。

会员端系统发送交易请求或查询请求，交易系统反馈应答。交易系统不维护对话流和查询流的状态。系统故障时，对话流和查询流均会重置，在途的数据可能会丢失。

## 2、私有通讯模式

私有通讯模式是单向的，支持会员私有流、交易员私有流和行情主题流（简称行情流）。

在私有通讯模式下，数据流是可靠的，交易系统维护整个系统的私有流或行情流。在一个交易日内，会员端系统断线后恢复连接时，可以请求交易系统发送指定序号之后的私有流数据或行情流数据。私有流向会员系统提供报单回报、成交回报等信息，行情流向行情接收系统提供行情信息。私有流分为会员私有流和交易员私有流两种。

交易系统维护每个会员的私有流，与会员相关的回报信息，如报单回报、成交回报等都通过会员私有流下发。交易员需有相应的权限才可以订阅会员私有流。

交易员私有流与会员私有流类似，但只包含该交易员发起的交易产生的回报信息。每个交易员都可以订阅自己的交易员私有流。

交易系统提供的行情内容是按照主题组织的。每个主题包括一组合约的行情。交易所会设定每个行情用户所能订阅的行情主题。每个行情主题对应着一个行情流。

要获得行情通知，行情接收系统必须在连接行情前置后，订阅一个或多个行情主题。

## 3、公共通讯模式

公共通讯模式是单向的，支持公共数据流（简称公共流）。

交易系统向会员系统发送市场公共信息。公共流是一个可靠的数据流，交易系统维护整个系统的公共流，在一个交易日内，会员系统断线恢复连接时，可以请求交易系统发送指定序号之后的公共流数据。



## 3. 接口模式

### 3.1. TraderAPI 接口

TraderAPI 提供了两个接口类，分别为 CShfeFtdcTraderApi 和 CShfeFtdcTraderSpi。这两个接口类是对 FTD 协议的封装。

会员系统可以通过 CShfeFtdcTraderApi 发出操作请求，通过继承 CShfeFtdcTraderSpi 并重载回调函数来处理交易系统的回报或响应。

#### 3.1.1. 对话流和查询流编程接口

通过对话流进行通讯的编程接口通常如下：

```
///  
///请求：  
  
int CShfeFtdcTraderApi::ReqXXX(  
  
    CShfeFtdcXXXField* pReqXXX,  
  
    int nRequestID);  
  
///  
///响应：  
  
void CShfeFtdcTraderSpi::OnRspXXX(  
  
    CShfeFtdcXXXField* pRspXXX,  
  
    CShfeFtdcRspInfoField* pRspInfo,  
  
    int nRequestID,  
  
    bool bIsLast);
```

请求接口包含两个参数。其中：

第一个参数为请求的内容，不能为空。该参数根据请求命令的不同使用不同的类，该类的成员变量的类型和合法的数值请参阅附录中的“[枚举值列表](#)”和“[数据类型列表](#)”。

第二个参数为请求号。请求号由会员系统负责维护，交易所建议每个请求的请求号不重复。在接收交易系统的响应时，可以得到当时发出请求时填写的请求号，从而可以将响应与请求对应起来。

当收到交易系统应答时，CShfeFtdcTraderSpi 的回调函数会被调用。如果响应数据不止一个，则回调函数会被多次调用。

回调函数包含四个参数。其中：

第一个参数为响应的具体数据，如果出错或没有结果可能为 NULL。

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。在发生多次回调时，除了第一次回调，其它的回调该参数均为 NULL。

第三个参数为请求号，即原来发出请求时填写的请求号。

第四个参数为响应结束标志，表明是否是本次响应的最后一次回调。

### 3.1.2. 私有流编程接口

私有流中的数据为会员或者交易员的私有信息，包括报单回报、成交回报、报价回报、执行宣告回报等。

通过私有流接收回报的编程接口通常如下：

```
void CShfeFtdcTraderSpi::OnRtnXXX(CShfeFtdcXXXField* pXXX);  
  
////或者  
  
void CShfeFtdcTraderSpi::OnErrRtnXXX(  
    CShfeFtdcXXXField* pXXX,  
    CShfeFtdcRspInfoField* pRspInfo);
```

当收到交易系统通过私有流发布的回报数据时，CShfeFtdcTraderSpi 的回调函数会被调用。回调函数的参数为回报的具体内容。

### 3.1.3. 公共流编程接口

公共流数据包含交易所合约、公告等公共信息。

通过公共流接收回报的编程接口通常如下：

```
void CShfeFtdcTraderSpi::OnRtnXXX(CShfeFtdcXXXField* pXXX);
```

当收到交易系统通过公共流发布的回报数据时，CShfeFtdcTraderSpi 的回调函数会被调用。回调函数的参数为通知的具体内容。

## 3.2. MduserAPI 接口

与 TraderAPI 类似，MduserAPI 提供了两个接口类，分别为 CShfeFtdcMduserApi 和 CShfeFtdcMduserSpi。这两个接口类是对 FTD 协议的封装。

行情接收系统可以通过 CShfeFtdcMduserApi 发出操作请求，通过继承 CShfeFtdcMduserSpi 并重载回调函数来处理交易系统的回报或响应。

### 3.2.1. 对话流编程接口

通过对话流进行通讯的编程接口通常如下：

```
///请求：  
  
int CShfeFtdcMduserApi::ReqXXX(  
  
    CShfeFtdcXXXField* pReqXXX,  
  
    int nRequestID);  
  
///响应：  
  
void CShfeFtdcMduserSpi::OnRspXXX(  
  
    CShfeFtdcXXXField* pRspXXX,  
  
    CShfeFtdcRspInfoField* pRspInfo,  
  
    int nRequestID,  
  
    bool bIsLast);
```

请求接口包含两个参数。其中：

第一个参数为请求的内容，不能为空。

第二个参数为请求号。请求号由行情接收系统负责维护，交易所建议每个请求的请求号不重复。在接收交易系统的响应时，可以得到当时发出请求时填写的请求号，从而可以将响应与请求对应起来。

当收到交易系统应答时，CShfeFtdcMduserSpi 的回调函数会被调用。如果响应数据不止一个，则回调函数会被多次调用。

回调函数包含四个参数。其中：

第一个参数为响应的具体数据，如果出错或没有结果可能为 NULL。

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。在发生多次回调时，除了第一次回调，其它的回调该参数均为 NULL。

第三个参数为请求号，即原来发出请求时填写的请求号。

第四个参数为响应结束标志，表明是否是本次响应的最后一次回调。

### 3.2.2. 行情流编程接口

行情流中的数据包含了交易系统推送的行情信息。

通过行情流接收回报的编程接口通常如下：

```
void CShfeFtdcMduserSpi::OnRtnXXX(CShfeFtdcXXXField* pXXX);
```

当收到行情数据时，CShfeFtdcMduserSpi 的回调函数会被调用。回调函数的参数为通知的具体内容。

## 4. 运行模式

### 4.1. 工作流程

会员端系统与交易系统的交互过程分为两个阶段：初始化阶段和功能调用阶段。

#### 4.1.1. 初始化阶段

在初始化阶段，会员端系统的程序必须完成如下步骤（具体代码请参考开发实例）：

| 序号 | 会员系统                       | 行情接收系统                     |
|----|----------------------------|----------------------------|
| 1  | 生成一个 CShfeFtdcTraderApi 实例 | 生成一个 CShfeFtdcMduserApi 实例 |
| 2  | 生成一个事件处理的实例                | 生成一个事件处理的实例                |
| 3  | 注册一个事件处理的实例                | 注册一个事件处理的实例                |
| 4  | 订阅私有流<br>订阅公共流             | 订阅行情流                      |
| 5  | 注册交易前置 NameServer 的网络地址    | 注册行情前置 NameServer 的网络地址    |
| 6  | 初始化                        | 初始化                        |

#### 4.1.2. 功能调用阶段

在功能调用阶段，会员端系统可以任意调用交易或行情接口中的请求方法，如 ReqUserLogin、ReqOrderInsert 等，同时提供回调函数以接收响应和回报信息。  
注意事项：

1. API 请求的输入参数不能为 NULL。
2. API 请求的返回参数，0 表示正确，其他表示错误，详细错误编码请参阅附录中的“[返回值列表](#)”。
3. 会员端系统发送请求指令时会受到流控机制的限制，若超出流控则当前

请求发送失败。

4. 流控包括通讯流控和在途流控，通讯流控指 1 秒内发出的请求数不能超过设定值，在途流控指已发送但未收到应答的请求数不能超过设定值。

4.2. 工作线程

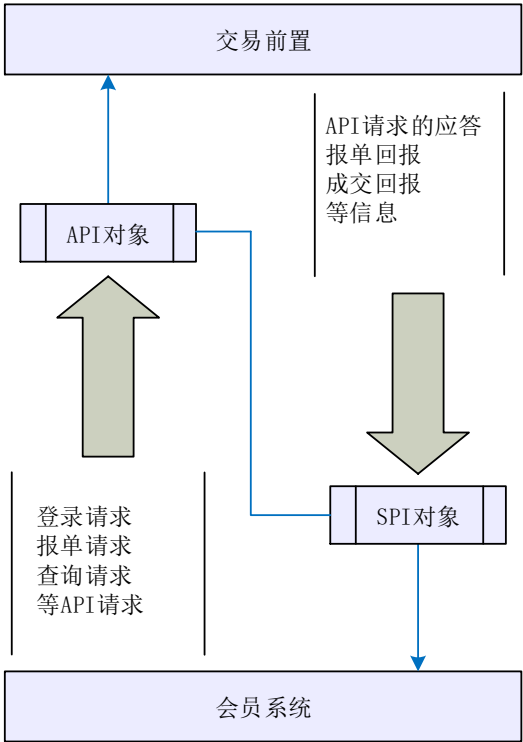
会员端系统至少由两个线程组成，一个是系统应用程序主线程，一个是 API 工作线程（TraderAPI 或者 MduserAPI）。应用程序与交易前置或行情前置的通信是由 API 工作线程驱动的。

CShfeFtdcTraderApi 和 CShfeFtdcMduserApi 提供的接口是线程安全的，可以由多个线程同时调用。

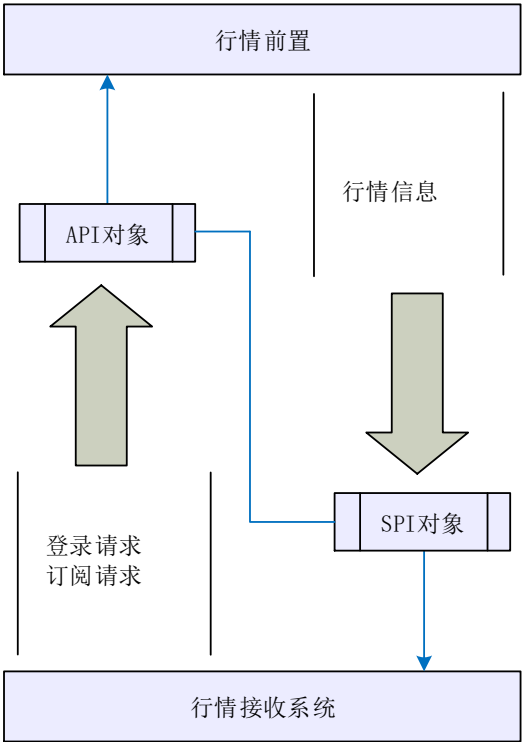
CShfeFtdcTraderSpi 提供的回调接口是由 TraderAPI 工作线程驱动，通过实现 SPI 中的接口方法，从交易前置接收数据。

CShfeFtdcMduserSpi 提供的回调接口是由 MduserAPI 工作线程驱动，通过实现 SPI 中的接口方法，从行情前置接收数据。

如果重载的某个回调函数阻塞，则等于阻塞了 TraderAPI 或者 MduserAPI 工作线程，API 与交易前置或行情前置的通信会停止，因此通常应该迅速返回。



图表 2 TraderAPI 工作线程



图表 3 MduserAPI 工作线程

### 4.3. 与交易系统的连接

TraderAPI 和 MduserAPI 使用 FTD 协议分别与交易前置和行情前置进行通信。TraderAPI 和 MduserAPI 使用 RegisterNameServer 方法注册 FENS(Front-End Name Server)地址，连接交易前置和行情前置。

交易所部署多个交易前置和行情前置，用于负载均衡且互为备份，从而提高系统的性能和可靠性。为保证交易通信的可靠性，TraderAPI 和 MduserAPI 可以注册多个 FENS 地址。API 在初始化后，从已注册的 FENS 地址中选择一个 FENS 地址，尝试建立网络连接，如果不成功，则依次逐个尝试其它 FENS 地址，直到连接成功为止。

交易所将公布至少两个 FENS 地址，因此会员端系统应该至少注册两个 FENS 地址防止所连 FENS 服务发生故障从而引发单点隐患。

### 4.4. TraderAPI 与交易前置的交互

会员系统通过 TraderAPI 与交易前置交互。会员系统的请求通过 TraderAPI 发送到交易前置；交易前置的应答和回报通过 TraderAPI 返回给会员系统。

TraderAPI 的交易接口和私有流接口相互关联，例如用户（即交易员）发送报单录入请求 ReqOrderInsert，会收到报单响应 OnRspOrderInsert，说明交易系统已经收到报单。报单进入交易系统后，如果报单的状态发生变化，就会收到报单回报 OnRtnOrder。如果报单被撮合(包括全部成交和部分成交)，就会收到成交回报 OnRtnTrade。其中，一个用户的报单回报和成交回报也会被所属会员下其他交易员（该交易员具有订阅会员私有流的权限并且订阅该流）接收到。

以交易员日常交易为例，有两个会员系统 A 和 B，交互过程如下：

1. 交易员 A 报单，cu2511，买，20 手，74000 元；

- CShfeFtdcTraderApi::ReqOrderInsert：报单录入请求。本函数由会员系统调用，通过对话流发送到交易前置。
- 交易系统报单处理：报单系统编号为 1；由于此时撮合队列中无对手，报单状态为“未成交还在队列中”。交易前置发出报单响应给交易员 A 的对话流；发出报单回报给交易员 A 的私有流和交易员 A 所属会员的私有流。响应和回报的报文由 TraderAPI 工作线程处理并调

用 SPI 对象的方法。

- CShfeFtdcTraderSpi::OnRspOrderInsert: 交易前置给出请求的应答, 内容为: 录入成功, 本地编号为 1 的报单的系统编号为 1。本函数由 TraderAPI 工作线程在收到交易前置的报单录入请求应答后调用。
- CShfeFtdcTraderSpi::OnRtnOrder: 交易前置在交易员 A 所属会员的私有流和交易员 A 的私有流中给出报单回报。本函数由 TraderAPI 工作线程在收到交易前置的报单回报后调用。如果会员 A 还有其他交易员连接并登录到交易系统, 并且订阅会员私有流, 将收到相同的报单回报 (以下同)。

## 2. 交易员 B 报单, cu2511, 卖, 10 手, 74000 元;

- CShfeFtdcTraderApi::ReqOrderInsert: 报单录入请求。
- 交易系统报单处理: 报单系统编号为 2; 尝试撮合并能成交, 因此报单状态为“完全成交”。交易前置发出报单响应给交易员 B 的对话流; 发出报单回报给交易员 B 的私有流和交易员 B 所属的会员私有流; 发出成交回报给交易员 B 的私有流和交易员 B 所属的会员私有流; 发出报单回报给交易员 A 的私有流和交易员 A 所属会员的私有流, 提示系统编号为 1 的报单状态交易系统已经修改为“部分成交还在队列中”, 报单的“剩余数量”调整为 10 手; 发出成交回报给交易员 A 的私有流和交易员 A 所属的会员私有流。**交易系统将确保同一笔报单的报单回报在成交回报之前送达会员系统, 并且报单回报中的“剩余数量”字段已经反映交易系统报单簿中的最新数量。**
- CShfeFtdcTraderSpi::OnRspOrderInsert: 交易前置给出请求的应答, 内容为: 录入成功, 本地编号为 1 的报单的系统编号为 2。
- CShfeFtdcTraderSpi::OnRtnOrder: 交易前置在交易员 B 所属会员的私有流和交易员 B 的私有流中给出报单回报, 报单状态为“完全成交”。
- CShfeFtdcTraderSpi::OnRtnTrade: 交易前置在交易员 B 所属会员的私有流和交易员 B 的私有流中给出成交回报。
- CShfeFtdcTraderSpi::OnRtnOrder: 交易前置在交易员 A 所属会员的私有流和交易员 A 的私有流中给出报单回报, 报单状态为“部分成交还在队列中”, 剩余数量为 10 手。
- CShfeFtdcTraderSpi::OnRtnTrade: 交易前置在交易员 A 所属会员的私有流和交易员 A 的私有流中给出成交回报。

## 3. 交易员 A 撤单;

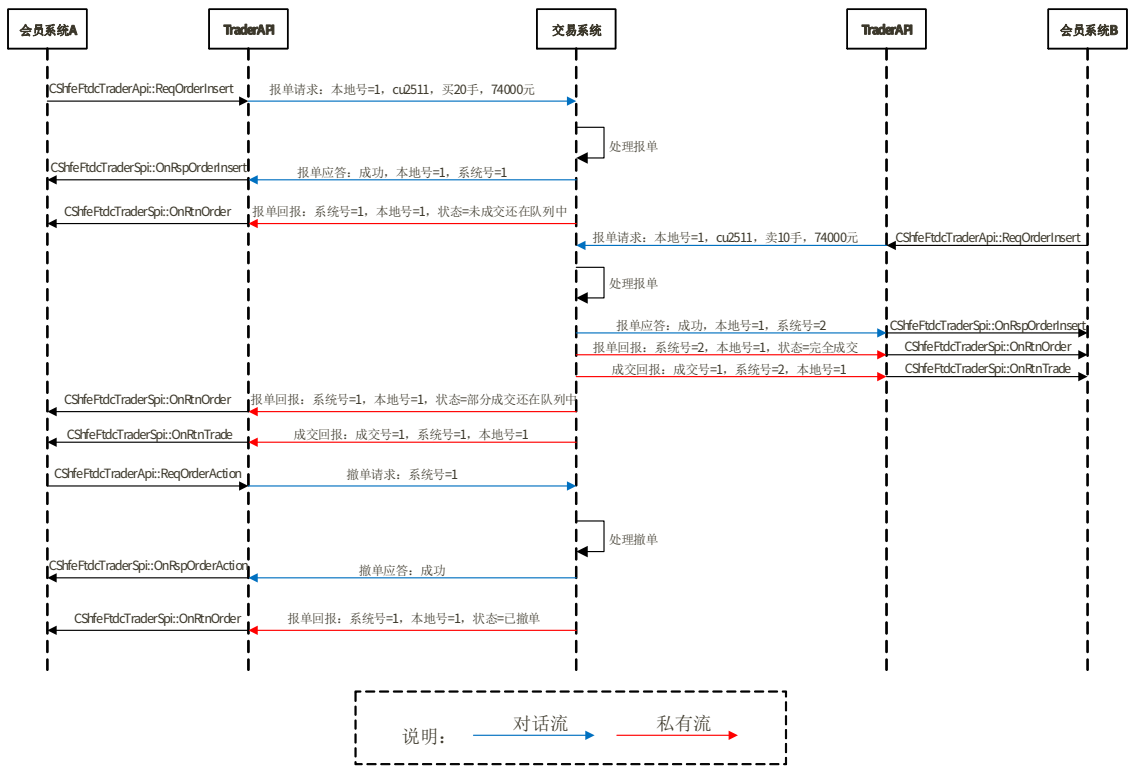
- CShfeFtdcTraderApi::ReqOrderAction: 报单操作请求。本函数由会员系统调用, 通过对话流



发送到交易前置。

- 交易系统撤单处理：撤销系统编号为 1 的剩余报单。交易前置发出撤单响应给交易员 A 的对话流；发出报单回报给交易员 A 的私有流和交易员 A 所属会员的私有流。响应和回报的报文由 TraderAPI 工作线程处理并调用 SPI 对象的方法。
- CShfeFtdcTraderSpi::OnRspOrderAction：交易前置给出请求的应答，内容为：撤单成功。本函数由 TraderAPI 工作线程在收到交易前置应答后调用。
- CShfeFtdcTraderSpi::OnRtnOrder：本函数由 TraderAPI 工作线程在收到交易前置的报单回报后调用。如果会员 A 还有其他交易员连接并登录到交易系统，并且订阅会员私有流，将收到相同的报单回报。

下图说明了会员系统、TraderAPI 和交易系统之间的 UML 交互图。



图表 4 会员系统与交易系统信息交互示例

4.5. MduserAPI 与行情前置的交互

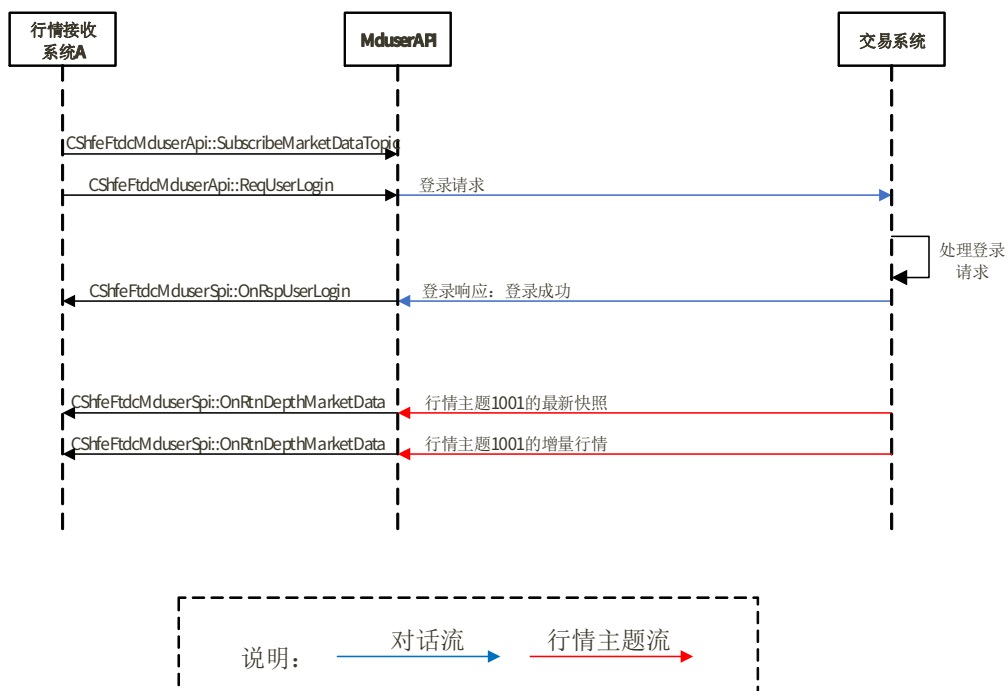
行情接收系统通过 MduserAPI 与行情前置交互。行情接收系统的请求通过 MduserAPI 发送到行情前置；行情前置的应答和回报通过 MduserAPI 返回给行情

接收系统。

以行情转发商订阅行情为例，行情转发商以快照模式订阅行情主题 1001，有一个行情接收系统 A，交互过程如下：

- `CShfeFtdcMduserApi::SubscribeMarketDataTopic`：订阅行情主题。本函数由行情接收系统调用。
- `CShfeFtdcMduserApi::ReqUserLogin`：登录请求。本函数由行情接收系统调用，通过对话流发送到行情前置。
- 交易系统请求处理：该登录请求合法，则行情前置通过对话流发送登录请求响应给行情转发商；通过行情流发送行情通知给行情转发商。响应和行情通知的报文由 `MduserAPI` 工作线程处理。
- `CShfeFtdcMduserSpi::OnRspUserLogin`：行情前置给出登录请求的应答，本函数由 `MduserAPI` 工作线程在收到行情前置应答后调用。
- `CShfeFtdcMduserSpi::OnRtnDepthMarketData`：行情前置在行情转发商的行情流中给出 1001 主题的最新快照信息。本函数由 `MduserAPI` 工作线程在收到行情前置的行情通知后调用。
- `CShfeFtdcMduserApi::OnRtnDepthMarketData`：行情前置在行情转发商的行情流中给出 1001 主题的增量行情信息。本函数由 `MduserAPI` 工作线程在收到行情前置的行情通知后调用。

下图说明了行情接收系统、MduserAPI 和交易系统之间的 UML 交互图。



图表 5 行情接收系统与交易系统信息交互示例

## 4.6. 本地文件

TraderAPI 在运行过程中，会将一些数据写入本地文件中。调用 CreateFtdcTraderApi 函数，可以传递一个参数，指明存贮本地文件的路径。该路径必须在运行前已创建好。本地文件名称是“trade.con”。

MduserAPI 也与此类似，函数名称为：CreateFtdcMduserApi。本地文件名称是“md.con”。

## 4.7. 请求和应答日志文件

TraderAPI 和 MduserAPI 提供了两个日志接口，用于记录通信日志。OpenRequestLog 用于打开请求日志，OpenResponseLog 用于打开应答日志。日志打开后，所有的业务请求将记入请求日志，所有的业务应答和回报将记入应答日志。登录请求、密码修改请求、密码修改应答里的密码字段以及终端认证请求里的终端认证授权码在日志里为空。

请求格式为：

时戳,请求名称,请求参数名称=”请求参数内容”

应答格式为：

时戳,应答名称,响应代码,响应信息,应答参数名称=”应答参数内容”

回报格式为：

时戳,回报名称,回报参数名称=”回报参数内容”

## 4.8. 可靠数据流的订阅方式

FTD 协议中私有流、公共流、行情流等可以保证数据可靠、有序地传送到对方，称为可靠数据流。可靠数据流对于保证会员端系统的数据正确性和完整性非常重要。例如，会员系统通过会员私有流中的各类回报，可以得到足够的信息，从而完成会员端系统的业务处理。为了保证会员端系统业务的正确性，需要可靠、有序、唯一地接收私有流中的报文。

可靠数据流是依靠重传机制保证数据可靠和有序传送。会员端系统负责管理数据流的序号，如果出现传输中断，可以从指定序号开始，重新订阅数据流，来保证数据的完整性。

对话流和查询流不支持重传，是不可靠的数据流。

API 提供了两种管理可靠数据流的方式：API 维护重传报文的序号和会员端系统维护重传报文的序号。

### 4.8.1. API 维护重传报文的序号

API 将收到的可靠数据流的报文序号定期写入本地文件 `trade.con` 和 `md.con`。如果会员端系统退出后再重新订阅数据流，则可以使用本地文件中记录的报文序号来订阅数据流。在会员端系统异常断开时，可能对应报文序号尚未写入本地文件，重连后会造成同一个报文两次回调到会员端系统，此时会员端系统应做去重处理。

CShfeFtdcTraderApi 的 `SubscribePrivateTopic` 方法、`SubscribePublicTopic` 方法、`SubscribeUserTopic` 方法和 CShfeFtdcMduserApi 的 `SubscribeMarketDataTopic` 方法用于订阅可靠数据流。

通过接口参数可以指定订阅方式，订阅方式有三种：重发（RESTART）、续传（RESUME）和快照（QUICK）。

- 重发（RESTART）方式从数据流中的第一个报文开始传输，该方式下 API 将忽略本地文件中记录的数据流报文序号。
- 续传（RESUME）方式从本地文件中记录的数据流报文序号之后开始传输。如果是行情流，则先传输那一时刻的主题行情快照，再开始从

指定的序号开始传输行情。为保持会员交易数据的完整性，对于会员或者交易员私有流，交易所建议使用 RESUME 方式。

- 快照（QUICK）方式从订阅时刻数据流最大序号开始传输，如果是行情流，则先传输该主题的最新行情快照。快照方式主要用于不需要保证数据完整性的场合，例如通讯或者软件故障后迅速接收和恢复行情，对于会员或者交易员私有流，交易所不建议使用 QUICK 方式。

注意：若可靠数据流报文已通过 SPI 的回调函数通知到会员端系统，但对应报文序号尚未写入文件，则会造成同一个报文两次回调到会员端系统。

#### 4.8.2. 会员端系统维护重传报文的序号

每当 API 收到一个可靠数据流的报文，先调用 SPI 的 OnPackageStart 函数通知会员端系统收到一个报文，再调用 SPI 的回调函数通知会员端系统处理业务数据，最后调用 SPI 的 OnPackageEnd 函数通知会员端系统该报文的回调结束。在 OnPackageStart 和 OnPackageEnd 接口中，会员端系统可以得到当前回调的报文的序号。会员端系统可以记录该序号，在重传可靠数据流时，将该序号作为 ReqSubscribeTopic 方法的参数（与 RESUME 方式类似）。

通过 ReqSubscribeTopic 方法，会员端系统可以指定数据流重传报文的序号。如果指定序号为 0，则重发整个数据流（与 RESTART 方式类似）；如果指定序号为-1，则从订阅时刻数据流的最大序号开始传输（与 QUICK 方式类似）。

对于行情流的订阅，如果指定的重传序号不为 0，则先传输该序号报文产生之前的行情快照。

#### 4.9. 心跳机制（Heartbeat）

心跳报文用于检测连接是否有效。连接的一端如果在规定超时时间（timeout）内未收到对方的任何报文，则认为 TCP 虚链路失效，应主动断开；如果一方在一定的时间间隔（interval）内未向对方发送任何报文，则应向对方发送心跳报文，以维持 TCP 虚链路的正常状态。

API 提供了 SetHeartbeatTimeout 方法，用于设定会员端系统检测 TCP 虚链路有效性的超时时间。交易系统定时向 API 发送心跳报文，若 API 超过 timeout/2

秒未收到交易系统的任何报文时，将触发回调 `OnHeartBeatWarning`，若 API 超过 `timeout` 秒未收到交易系统的任何报文时，TCP 连接将会中断，并且触发回调 `OnFrontDisconnected`。

例如，会员端系统将心跳超时设置为 16 秒，如果 API 在 8 秒内未收到交易系统的任何报文，则触发回调 `OnHeartBeatWarning`。如果超过 16 秒未收到交易系统的任何报文，API 会主动断开网络连接，并且触发回调 `OnFrontDisconnected`。

交易前置和行情前置通过心跳方式监测会员端系统的 TCP 连接，`timeout` 参数也同时用于交易前置和行情前置对会员端系统的监测。`timeout` 参数默认设置为 10 秒；`timeout` 参数允许的最小值为 4 秒，最大值为 181 秒。

`timeout` 参数设置偏大，发生链路中断情况下会员端系统切换时间可能增加；设置偏小，可能发生非预期切换。因此，需要综合考虑会员端系统应用情况和网络情况合理设置。

**交易所建议会员端系统将 `timeout` 参数值设置为 10 至 30 秒之间。**

## 4.10. 灾备接口

交易所建有张江、期货大厦和北京三个数据中心，三个中心使用高速线路实现系统的互联互通。张江数据中心为现主用数据中心。交易系统同时在三个数据中心运行，主中心进行业务处理，备中心实时接收主中心的数据。

当发生数据中心切换时，备数据中心接管主数据中心工作，继续进行业务处理。数据中心切换时，可能出现少量业务数据丢失。会员端系统需要通过 API 接口获知要回退的数据流序号。

- 一、API 用户登录请求报文中的“数据中心代码”（`DataCenterID`）字段，用于标识上次登录的数据中心。交易系统在用户登录应答报文中返回当前使用的数据中心。
- 二、会员端系统可根据 API“数据流回退”（`OnRtnFlowMessageCancel`）接口，得到回退的数据流序号。
- 三、回退的数据流序号包括回退起始序列号和回退结束序列号，两序号之间的数据属于已作废数据，会员端系统需根据回退的数据流序号对已

接收数据做回退处理。比如当前会员端系统请求从序号 100 开始订阅数据流，而交易系统返回的数据流回退通知中回退起始序列号为 95，回退结束序列号为 100，则会员端系统需将序号 96-100 的数据执行回退操作。若会员端系统有订阅后续数据的需求，交易所建议从回退起始序列号 95 开始重新订阅。

## 第二部分 TraderAPI 参考手册

本部分主要提供给会员系统开发商阅读，内容包括：

第一章为 TraderAPI 接口分类。

第二章为 TraderAPI 接口说明。

第三章为 TraderAPI 接口开发示例。



# 1. TraderAPI 接口分类

## 1.1. 管理接口

TraderAPI 的管理接口用于对 API 的生命周期和运行参数进行控制。

| 接口类型     | 接口名称                                       | 说明                   |
|----------|--------------------------------------------|----------------------|
| 生命周期管理接口 | CShfeFtdcTraderApi::CreateFtdcTraderApi    | 创建 TraderApi 实例      |
|          | CShfeFtdcTraderApi::GetVersion             | 获取 API 版本            |
|          | CShfeFtdcTraderApi::Release                | 删除接口实例               |
|          | CShfeFtdcTraderApi::Init                   | 初始化                  |
|          | CShfeFtdcTraderApi::Join                   | 等待接口线程结束运行           |
|          | CShfeFtdcTraderApi::GetTradingDay          | 获得当前交易日              |
| 参数管理接口   | CShfeFtdcTraderApi::RegisterSpi            | 注册回调接口               |
|          | CShfeFtdcTraderApi::RegisterFront          | 注册前置网络地址             |
|          | CShfeFtdcTraderApi::RegisterNameServer     | 注册 FENS 地址           |
|          | CShfeFtdcTraderApi::SetHeartbeatTimeout    | 设置心跳超时时间             |
| 订阅接口     | CShfeFtdcTraderApi::SubscribePrivateTopic  | 订阅会员私有流              |
|          | CShfeFtdcTraderApi::SubscribePublicTopic   | 订阅公共流                |
|          | CShfeFtdcTraderApi::SubscribeUserTopic     | 订阅交易员私有流             |
| 日志接口     | CShfeFtdcTraderApi::OpenRequestLog         | 打开请求日志文件             |
|          | CShfeFtdcTraderApi::OpenResponseLog        | 打开应答日志文件             |
| 通信状态接口   | CShfeFtdcTraderSpi::OnFrontConnected       | 与交易系统建立起通信连接时，该方法被调用 |
|          | CShfeFtdcTraderSpi::OnFrontDisconnected    | 与交易系统通信连接断开时，该方法被调用  |
|          | CShfeFtdcTraderSpi::OnHeartBeatWarning     | 当长时间未收到报文时，该方法被调用    |
|          | CShfeFtdcTraderSpi::OnPackageStart         | 报文回调开始通知             |
|          | CShfeFtdcTraderSpi::OnPackageEnd           | 报文回调结束通知             |
| 灾备接口     | CShfeFtdcTraderSpi::OnRtnFlowMessageCancel | 数据流回退通知              |

## 1.2. 业务接口

| 业务类型 | 业务 | 请求接口 / 响应接口                                                              | 数据流 |
|------|----|--------------------------------------------------------------------------|-----|
| 登录   | 登录 | CShfeFtdcTraderApi::ReqUserLogin<br>CShfeFtdcTraderSpi::OnRspUserLogin   | 对话流 |
|      | 登出 | CShfeFtdcTraderApi::ReqUserLogout<br>CShfeFtdcTraderSpi::OnRspUserLogout | 对话流 |

| 业务类型     | 业务           | 请求接口 / 响应接口                                                                                                              | 数据流 |
|----------|--------------|--------------------------------------------------------------------------------------------------------------------------|-----|
|          | 修改用户<br>口令   | CShfeFtdcTraderApi::ReqUserPasswordUpdate<br>CShfeFtdcTraderSpi::OnRspUserPasswordUpdate                                 | 对话流 |
|          | 终端认证         | CShfeFtdcTraderApi::ReqAuthenticate<br>CShfeFtdcTraderSpi::OnRspAuthenticate                                             | 对话流 |
| 订阅       | 订阅主题         | CShfeFtdcTraderApi::ReqSubscribeTopic<br>CShfeFtdcTraderSpi::OnRspSubscribeTopic                                         | 对话流 |
|          | 查询主题         | CShfeFtdcTraderApi::ReqQryTopic<br>CShfeFtdcTraderSpi::OnRspQryTopic                                                     | 查询流 |
| 交易       | 报单录入         | CShfeFtdcTraderApi::ReqOrderInsert<br>CShfeFtdcTraderSpi::OnRspOrderInsert                                               | 对话流 |
|          | 报单操作         | CShfeFtdcTraderApi::ReqOrderAction<br>CShfeFtdcTraderSpi::OnRspOrderAction                                               | 对话流 |
|          | 报价录入         | CShfeFtdcTraderApi::ReqQuoteInsert<br>CShfeFtdcTraderSpi::OnRspQuoteInsert                                               | 对话流 |
|          | 报价操作         | CShfeFtdcTraderApi::ReqQuoteAction<br>CShfeFtdcTraderSpi::OnRspQuoteAction                                               | 对话流 |
|          | 执行宣告<br>录入   | CShfeFtdcTraderApi::ReqExecOrderInsert<br>CShfeFtdcTraderSpi::OnRspExecOrderInsert                                       | 对话流 |
|          | 执行宣告<br>操作   | CShfeFtdcTraderApi::ReqExecOrderAction<br>CShfeFtdcTraderSpi::OnRspExecOrderAction                                       | 对话流 |
|          | 放弃执行<br>宣告录入 | CShfeFtdcTraderApi::ReqAbandonExecOrderInsert<br>CShfeFtdcTraderSpi::OnRspAbandonExecOrderInsert                         | 对话流 |
|          | 放弃执行<br>宣告操作 | CShfeFtdcTraderApi::ReqAbandonExecOrderAction<br>CShfeFtdcTraderSpi::OnRspAbandonExecOrderAction                         | 对话流 |
|          | 询价请求<br>录入   | CShfeFtdcTraderApi::ReqQuoteDemand<br>CShfeFtdcTraderSpi::OnRspQuoteDemand<br>CShfeFtdcTraderSpi::OnRtnQuoteDemandNotify | 对话流 |
|          | 期权自对<br>冲更新  | CShfeFtdcTraderApi::ReqOptionSelfCloseUpdate<br>CShfeFtdcTraderSpi::OnRspOptionSelfCloseUpdate                           | 对话流 |
|          | 期权自对<br>冲操作  | CShfeFtdcTraderApi::ReqOptionSelfCloseAction<br>CShfeFtdcTraderSpi::OnRspOptionSelfCloseAction                           | 对话流 |
| 私有回<br>报 | 成交回报         | CShfeFtdcTraderSpi::OnRtnTrade                                                                                           | 私有流 |
|          | 报单回报         | CShfeFtdcTraderSpi::OnRtnOrder                                                                                           | 私有流 |
|          | 报价回报         | CShfeFtdcTraderSpi::OnRtnQuote                                                                                           | 私有流 |
|          | 执行宣告<br>回报   | CShfeFtdcTraderSpi::OnRtnExecOrder                                                                                       | 私有流 |
|          | 报单录入<br>错误回报 | CShfeFtdcTraderSpi::OnErrRtnOrderInsert                                                                                  | 私有流 |
|          | 报单操作<br>错误回报 | CShfeFtdcTraderSpi::OnErrRtnOrderAction                                                                                  | 私有流 |
|          | 报价录入         | CShfeFtdcTraderSpi::OnErrRtnQuoteInsert                                                                                  | 私有流 |

| 业务类型     | 业务                   | 请求接口 / 响应接口                                                                      | 数据流 |
|----------|----------------------|----------------------------------------------------------------------------------|-----|
|          | 错误回报                 |                                                                                  |     |
|          | 报价操作<br>错误回报         | CShfeFtdcTraderSpi::OnErrRtnQuoteAction                                          | 私有流 |
|          | 执行宣告<br>录入错误<br>回报   | CShfeFtdcTraderSpi::OnErrRtnExecOrderInsert                                      | 私有流 |
|          | 执行宣告<br>操作错误<br>回报   | CShfeFtdcTraderSpi::OnErrRtnExecOrderAction                                      | 私有流 |
|          | 放弃执行<br>宣告回报         | CShfeFtdcTraderSpi::OnRtnAbandonExecOrder                                        | 私有流 |
|          | 放弃执行<br>宣告录入<br>错误回报 | CShfeFtdcTraderSpi::OnErrRtnAbandonExecOrderInsert                               | 私有流 |
|          | 放弃执行<br>宣告操作<br>错误回报 | CShfeFtdcTraderSpi::OnErrRtnAbandonExecOrderAction                               | 私有流 |
|          | 期权自对<br>冲更新回<br>报    | CShfeFtdcTraderSpi::OnRtnOptionSelfCloseUpdate                                   | 私有流 |
|          | 期权自对<br>冲更新错<br>误回报  | CShfeFtdcTraderSpi::OnErrRtnOptionSelfCloseUpdate                                | 私有流 |
|          | 期权自对<br>冲操作错<br>误回报  | CShfeFtdcTraderSpi::OnErrRtnOptionSelfCloseAction                                | 私有流 |
| 公共通<br>知 | 合约交易<br>状态通知         | CShfeFtdcTraderSpi::OnRtnInstrumentStatus                                        | 公共流 |
|          | 增加合约<br>通知           | CShfeFtdcTraderSpi::OnRtnInsInstrument                                           | 公共流 |
|          | 公告通知                 | CShfeFtdcTraderSpi::OnRtnBulletin                                                | 公共流 |
| 查询       | 资金查询                 | CShfeFtdcTraderApi::ReqQryPartAccount<br>CShfeFtdcTraderSpi::OnRspQryPartAccount | 查询流 |
|          | 报单查询                 | CShfeFtdcTraderApi::ReqQryOrder<br>CShfeFtdcTraderSpi::OnRspQryOrder             | 查询流 |
|          | 报价查询                 | CShfeFtdcTraderApi::ReqQryQuote<br>CShfeFtdcTraderSpi::OnRspQryQuote             | 查询流 |
|          | 成交查询                 | CShfeFtdcTraderApi::ReqQryTrade<br>CShfeFtdcTraderSpi::OnRspQryTrade             | 查询流 |
|          | 客户查询                 | CShfeFtdcTraderApi::ReqQryClient<br>CShfeFtdcTraderSpi::OnRspQryClient           | 查询流 |
|          | 会员持仓                 | CShfeFtdcTraderApi::ReqQryPartPosition                                           | 查询流 |

| 业务类型     | 业务           | 请求接口 / 响应接口                                                                                | 数据流        |
|----------|--------------|--------------------------------------------------------------------------------------------|------------|
|          | 查询           | CShfeFtdcTraderSpi::OnRspQryPartPosition                                                   |            |
|          | 客户持仓<br>查询   | CShfeFtdcTraderApi::ReqQryClientPosition<br>CShfeFtdcTraderSpi::OnRspQryClientPosition     | 查询流        |
|          | 合约查询         | CShfeFtdcTraderApi::ReqQryInstrument<br>CShfeFtdcTraderSpi::OnRspQryInstrument             | 查询流        |
|          | 合约交易<br>状态查询 | CShfeFtdcTraderApi::ReqQryInstrumentStatus<br>CShfeFtdcTraderSpi::OnRspQryInstrumentStatus | 查询流        |
|          | 套保额度<br>查询   | CShfeFtdcTraderApi::ReqQryHedgeVolume<br>CShfeFtdcTraderSpi::OnRspQryHedgeVolume           | 查询流        |
|          | 行情查询         | CShfeFtdcTraderApi::ReqQryMarketData<br>CShfeFtdcTraderSpi::OnRspQryMarketData             | 查询流        |
|          | 公告查询         | CShfeFtdcTraderApi::ReqQryBulletin<br>CShfeFtdcTraderSpi::OnRspQryBulletin                 | 查询流        |
|          | 执行宣告<br>查询   | CShfeFtdcTraderApi::ReqQryExecOrder<br>CShfeFtdcTraderSpi::OnRspQryExecOrder               | 查询流        |
|          | 汇率查询         | CShfeFtdcTraderApi::ReqQryExchangeRate<br>CShfeFtdcTraderSpi::OnRspQryExchangeRate         | 查询流        |
|          | 放弃执行<br>宣告查询 | CShfeFtdcTraderApi::ReqQryAbandonExecOrder<br>CShfeFtdcTraderSpi::OnRspQryAbandonExecOrder | 查询流        |
|          | 期权自对<br>冲查询  | CShfeFtdcTraderApi::ReqQryOptionSelfClose<br>CShfeFtdcTraderSpi::OnRspQryOptionSelfClose   | 查询流        |
| 错误应<br>答 | 错误应答         | CShfeFtdcTraderSpi::OnRspError                                                             | 对话流<br>查询流 |

## 2. TraderAPI 接口说明

### 2.1. CShfeFtdcTraderSpi 接口

CShfeFtdcTraderSpi 实现了事件通知接口。用户必须派生 CShfeFtdcTraderSpi 接口，编写事件处理方法来处理需要的事件。

#### 2.1.1. OnFrontConnected 方法

当会员系统与交易前置建立起 TCP 虚链路（连接）后，该方法被调用。该连接是由 API 自动建立的。

函数原形：

```
void OnFrontConnected();
```

注意：OnFrontConnected 被调用仅说明 TCP 连接成功，会员系统必须自行登录，才能进行后续的业务操作。

#### 2.1.2. OnFrontDisconnected 方法

当会员系统与交易前置的 TCP 虚链路断开时，该方法被调用。当发生这个情况后，API 会自动重新连接。自动重连地址，可能是原来注册的地址，也可能是系统支持的其它可用的通信地址，它由程序自动选择。

函数原形：

```
void OnFrontDisconnected(int nReason);
```

参数：

**nReason:** 连接断开原因：

- 0x1001，表示网络读失败；
- 0x1002，表示网络写失败；
- 0x2001，表示接收心跳超时；

- 0x2002，表示报文加密失败；
- 0x2003，表示报文解密失败；
- 0x2004，表示收到未订阅主题的报文；
- 0x2005，表示收到的报文序号不连续；
- 0x2006，表示报文长度不合法；
- 0x2007，表示报文转换错误；
- 0x2008，表示登录错误前置服务。

### 2.1.3. OnHeartBeatWarning 方法

心跳超时警告。当长时间未收到报文时，该方法被调用。缺省超时告警时间设置为 5 秒。如果调用过 SetHeartbeatTimeout(unsigned int timeout)自行设定心跳超时，则警告时间为 timeout/2。

函数原形：

```
void OnHeartBeatWarning(int nTimeLapse);
```

参数：

**nTimeLapse:** 距离上次接收报文的时间（秒）。

### 2.1.4. OnPackageStart 方法

报文回调开始通知。当 API 收到一个报文后，如果该报文属于公共流，私有流（会员私有流，交易员私有流）则调用本方法，然后是各数据域的回调，最后是报文回调结束通知。

函数原形：

```
void OnPackageStart(int nTopicID, int nSequenceNo);
```

参数：

**nTopicID:** 主题代码（如私有流、公共流）。

**nSequenceNo:** 报文序号。

### 2.1.5. OnPackageEnd 方法

报文回调结束通知。当 API 收到一个报文后，如果该报文属于公共流，私有流（会员私有流，交易员私有流）则调用报文回调开始通知，然后是各数据域的回调，最后调用本方法。

函数原形：

```
void OnPackageEnd(int nTopicID, int nSequenceNo);
```

参数：

**nTopicID**：主题代码（如私有流、公共流）。

**nSequenceNo**：报文序号。

### 2.1.6. OnRspUserLogin 方法

当会员系统发出登录请求之后，交易系统返回响应时，该方法会被调用，通知会员系统登录是否成功。

函数原形：

```
void OnRspUserLogin(
    CShfeFtdcRspUserLoginField* pRspUserLogin,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

**pRspUserLogin**：返回用户登录信息的地址。

用户登录信息结构：

```
struct CShfeFtdcRspUserLoginField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///登录成功时间
    TShfeFtdcTimeType   LoginTime;
    ///最大本地报单号
    TShfeFtdcOrderLocalIDType   MaxOrderLocalID;
    ///交易用户代码
```

```
TShfeFtdcUserIDType UserID;
///会员代码
TShfeFtdcParticipantIDType ParticipantID;
///交易系统名称
TShfeFtdcTradingSystemNameType   TradingSystemName;
///数据中心代码
TShfeFtdcDataCenterIDType   DataCenterID;
///会员私有流当前长度
TShfeFtdcSequenceNoType PrivateFlowSize;
///交易员私有流当前长度
TShfeFtdcSequenceNoType UserFlowSize;
///业务发生日期
TShfeFtdcDateType ActionDay;

};
```

注意：取业务的发生日期时应该取 ActionDay，取交易日时应该取 TradingDay，以下同。

**pRspInfo**：返回用户响应信息的地址。错误代码为 0 时，表示操作成功，以下同。响应信息结构：

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType   ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType   ErrorMsg;
};
```

可能出现的错误：

| 错误代码 | 错误提示            | 可能的原因                 |
|------|-----------------|-----------------------|
| 3    | 会员找不到           | 登录的会员代码错误             |
| 45   | 结算组初始化状态不对      | 交易系统尚未初始化完毕，可以稍后再重试   |
| 59   | 用户重复登录          | 不同 IP 地址的相同用户重复登录     |
| 60   | 用户名或口令错误        | 登录的交易用户代码或口令错误        |
| 62   | 用户不活跃           | 交易系统不允许该用户登录          |
| 64   | 用户不属于此会员        | 登录的会员代码错误             |
| 65   | 错误的登录 IP 地址     | 发起登录的计算机没有交易所允许 IP 地址 |
| 75   | 前置不活跃           | 交易系统前置不活跃             |
| 100  | 错误的用户类型         | 非交易用户登录交易系统           |
| 106  | 重复的会话           | 相同会话多次重复登录            |
| 135  | 用户认证失败          | 用户密钥校验失败              |
| 136  | 用户没有权限直连前置      | 用户没有权限直连前置            |
| 150  | 自营会员登陆前未认证或认证失败 | 自营会员终端信息没有经过认证        |

**nRequestID**：返回用户登录请求的 ID，该 ID 由用户在登录时指定。

**IsLast**：指示该次返回是否为针对 nRequestID 的最后一次返回。



### 2.1.7.OnRspUserLogout 方法

当会员系统发出登出请求之后，交易系统返回响应时，该方法会被调用，通知会员系统登出是否成功。

函数原形：

```
void OnRspUserLogout(  
    CShfeFtdcRspUserLogoutField* pRspUserLogout,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pRspUserLogout：** 返回用户登出信息的地址。用户登出信息结构：

```
struct CShfeFtdcRspUserLogoutField {  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
};
```

**pRspInfo：** 返回用户响应信息的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误：

| 错误代码 | 错误提示      | 可能的原因        |
|------|-----------|--------------|
| 66   | 用户尚未登录    | 尚未登录         |
| 67   | 并没有以此用户登录 | 登出时的用户和登录时不同 |
| 68   | 并没有以此会员登录 | 登出时的会员和登录时不同 |

**nRequestID：** 返回用户登出请求的 ID，该 ID 由用户在登出时指定。

**bIsLast：** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.8. OnRspUserPasswordUpdate 方法

用户密码修改应答。当会员系统发出用户密码修改指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspUserPasswordUpdate(  
    CShfeFtdcUserPasswordUpdateField* pUserPasswordUpdate,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pUserPasswordUpdate：**指向用户密码修改结构的地址，包含了用户密码修改请求的输入数据。用户密码修改结构：

```
struct CShfeFtdcUserPasswordUpdateField {  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///旧密码  
    TShfeFtdcPasswordType    OldPassword;  
    ///新密码  
    TShfeFtdcPasswordType    NewPassword;  
};
```

**pRspInfo：**指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误：

| 错误代码 | 错误提示      | 可能的原因               |
|------|-----------|---------------------|
| 23   | 结算组数据没有同步 | 交易系统尚未初始化完毕，可以稍后再重试 |
| 58   | 用户不匹配     | 要修改口令的用户和登录时不同      |
| 60   | 用户名或口令错误  | 原口令错误               |
| 62   | 用户不活跃     | 用户没有登录、交易、改密码的权限    |
| 66   | 用户尚未登录    | 尚未登录                |

|     |                                        |                |
|-----|----------------------------------------|----------------|
| 68  | 并没有以此会员登录                              | 修改口令的会员号与登录时不同 |
| 147 | 用户新密码不符合要求（长度不低于 8 位，必须包含数字、大写字母和小写字母） | 用户的新密码不符合规则    |

**nRequestID:** 返回用户密码修改请求的 ID，该 ID 由用户在密码修改时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.9.OnRspSubscribeTopic 方法

订阅主题应答。当会员系统发出订阅主题指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspSubscribeTopic(  
    CShfeFtdcDisseminationField* pDissemination,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pDissemination:** 指向订阅主题结构的地址，包含了要订阅的主题和起始报文的序号。订阅主题结构：

```
struct CShfeFtdcDisseminationField {  
    ///序列系列号  
    TShfeFtdcSequenceSeriesType    SequenceSeries;  
    ///序列号  
    TShfeFtdcSequenceNoType    SequenceNo;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};  
可能出现的错误：  
错误代码 错误提示                可能的原因
```

|   |             |                      |
|---|-------------|----------------------|
| 1 | 无效的会话或主题不存在 | 订阅的主题不存在或订阅的主题数超过了上限 |
|---|-------------|----------------------|

**nRequestID:** 返回用户订阅主题请求的 ID，该 ID 由用户在订阅主题时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.10.OnRspQryTopic 方法

查询主题应答。当会员系统发出查询主题指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryTopic(  
    CShfeFtdcDisseminationField* pDissemination,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pDissemination:** 指向查询主题结构的地址，包含了要查询的主题和该主题报文个数。查询主题结构：

```
struct CShfeFtdcDisseminationField {  
    ///序列系列号  
    TShfeFtdcSequenceSeriesType    SequenceSeries;  
    ///序列号  
    TShfeFtdcSequenceNoType    SequenceNo;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};  
可能出现的错误： 无
```

**nRequestID:** 返回用户查询主题请求的 ID，该 ID 由用户在查询主题时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.11.OnRspError 方法

针对用户请求的出错通知。

函数原形:

```
void OnRspError(
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

**pRspInfo:** 返回用户响应信息的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType   ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示                            | 可能的原因          |
|------|---------------------------------|----------------|
| 1    | Not Login                       | 尚未登录           |
|      | Too High FTD Version            | FTD 版本太高       |
|      | Unrecognized ftd tid            | FTD 报文头错误      |
| 134  | API 校验未通过                       | 用户会话认证信息校验未通过  |
| 151  | 版本校验未通过                         | 交易 API 版本校验未通过 |
| 997  | api authentication failure      | 非法的 API 接入     |
|      | api crypt info failure          | API 加密信息查询失败   |
| 998  | query frequency is too high     | 发送的查询频率太高      |
| 999  | the last query result is on way | 有未发送完的查询响应数据   |

**nRequestID:** 返回用户操作请求的 ID, 该 ID 由用户在操作请求时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.12.OnRspOrderInsert 方法

报单录入应答。当会员系统发出报单录入指令后, 交易系统返回响应时, 该方法会被调用。

## 函数原形:

```
void OnRspOrderInsert(
    CShfeFtdcInputOrderField* pInputOrder,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

## 参数:

**pInputOrder:** 指向报单录入结构的地址, 包含了提交报单录入时的输入数据和交易系统返回的报单编号。输入报单结构:

```
struct CShfeFtdcInputOrderField {
    ///报单编号*, 该字段由交易系统返回。
    TShfeFtdcOrderSysIDType OrderSysID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
    ///报单价格条件
    TShfeFtdcOrderPriceTypeType OrderPriceType;
    ///买卖方向
    TShfeFtdcDirectionType Direction;
    ///组合开平标志
    TShfeFtdcCombOffsetFlagType CombOffsetFlag;
    ///组合套保标志
    TShfeFtdcCombHedgeFlagType CombHedgeFlag;
    ///价格
    TShfeFtdcPriceType LimitPrice;
    ///数量
    TShfeFtdcVolumeType VolumeTotalOriginal;
    ///有效期类型
    TShfeFtdcTimeConditionType TimeCondition;
    ///GTD 日期, 未使用
    TShfeFtdcDateType GTDDate;
    ///成交量类型
    TShfeFtdcVolumeConditionType VolumeCondition;
    ///最小成交量
    TShfeFtdcVolumeType MinVolume;
```

```
///触发条件
TShfeFtdcContingentConditionType    ContingentCondition;
///止损价，未使用
TShfeFtdcPriceType    StopPrice;
///强平原因
TShfeFtdcForceCloseReasonType ForceCloseReason;
///本地报单编号
TShfeFtdcOrderLocalIDType    OrderLocalID;
///自动挂起标志
TShfeFtdcBoolType    IsAutoSuspend;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType    BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType    IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;

};
```

\* OrderSysID: 报单编号，交易系统对普通报单及报价产生的报单采用统一编号管理，在交易系统中该编号递增。不同业务的 SysID 采用独立编号管理，如报单的 OrderSysID 与报价的 QuoteSysID，两者编号互不关联。

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示          | 可能的原因                                  |
|------|---------------|----------------------------------------|
| 2    | 合约找不到         | 找不到报单中的合约                              |
| 3    | 会员找不到         | 找不到报单中的会员                              |
| 4    | 客户找不到         | 找不到报单中的客户                              |
| 6    | 报单字段错误        | 报单中有字段值不合法（枚举值越界）或发现非强平单中填写强平原因为非整数倍强平 |
| 12   | 重复的报单         | 报单中的本地报单号重复                            |
| 15   | 客户没有在该会员开户    | 报单中的客户并没有在指定的会员中开户                     |
| 16   | IOC 需在连续交易阶段  | 在非连续交易阶段报入 IOC 单                       |
| 17   | GFA 需在集合竞价阶段  | 在非集合竞价阶段报入 GFA 的报单                     |
| 19   | 数量约束应在 IOC 单上 | 数量约束不是任意数量的报单的时间条件不是 IOC               |
| 20   | GTD 报单过期了     | GTD 报单中的 GTD 日期已经过期                    |
| 21   | 最小数量大于报单数量    | 有最小数量条件的报单的最小数量大于报单数量                  |

|      |                     |                                |
|------|---------------------|--------------------------------|
| 22   | 交易所数据没有同步           | 交易系统尚未初始化完毕，可以稍后再重试            |
| 23   | 结算组数据没有同步           | 交易系统尚未初始化完毕，可以稍后再重试            |
| 26   | 当前状态禁止此项操作          | 合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡    |
| 31   | 平仓时客户持仓不足           | 报入平仓报单时，客户的持仓不足                |
| 32   | 超出客户限仓              | 报入开仓报单时，超过了客户的一般限仓             |
| 33   | 平仓时会员持仓不足           | 报入平仓报单时，会员的持仓不足                |
| 34   | 超出会员限仓              | 报入开仓报单时，超过了会员限仓                |
| 35   | 找不到帐号               | 找不到报单使用的资金帐号                   |
| 36   | 资金不足                | 资金帐号中没有足够的资金                   |
| 37   | 不合法的数量              | 报单数量不是最小报单数量要求的正整数倍，或者超过最大报单数量 |
| 48   | 价格非最小单位的倍数          | 报单的价格不是合约的最小变动单位的整数倍           |
| 49   | 价格超出涨停板             | 报单的价格高于合约的涨停板                  |
| 50   | 价格跌破跌停板             | 报单的价格低于合约的跌停板                  |
| 51   | 没有交易权限              | 对指定合约、或者客户对指定的合约、或者用户没有交易权限    |
| 52   | 只能平仓                | 会员对指定合约、或者客户对指定的合约、或者用户只有平仓的权限 |
| 53   | 没有此项交易角色            | 会员在指定合约上不具有该客户对应的交易角色          |
| 54   | 会话找不到               | 用户未登录                          |
| 57   | 不能为其他会员操作           | 用户给非其所属的会员进行操作                 |
| 58   | 用户不匹配               | 报单中的用户和登录时的用户不匹配               |
| 72   | 不允许自然人开仓            | 交割月自然人类型的客户发起开仓请求              |
| 78   | GTD 报单没有设定日期        | GTD 报单没有指定 GTD 日期              |
| 79   | 不被支持的报单类型           | 交易所不支持此种报单类型                   |
| 83   | 止损单仅用于连续交易          | 在非连续交易阶段报入止损单                  |
| 84   | 止损单需是 IOC 或 GFD     | 止损单的时间条件既不是 IOC，也不是 GFD        |
| 95   | 止损报单需说明止损价          | 止损单没有指定止损价                     |
| 96   | 保值额度不足              | 报入套保报单时，客户的套保额度不足              |
| 98   | 强平单需由管理员使用          | 非管理员用户报了强平单                    |
| 101  | 结算会员不能做交易           | 报单的会员类型为结算会员                   |
| 102  | 无法找到对应结算会员          | 找不到报单会员对应的结算会员                 |
| 103  | 当日套保仓位不能平仓          | 套保仓位不应使用平今仓报单进行平仓              |
| 114  | 最优价单不能排队            | 最优价单的时间条件不是 IOC                |
| 131  | 超出客户合约当日开仓限额        | 客户在某个合约上的开仓量超出开仓限额             |
| 132  | 超出客户产品每秒报单限额        | 客户在某个产品上一秒以内的报单笔数超出限额          |
| 153  | 市价单必须是 GFD 或者 IOC 单 | 市价单时间条件不为 GFD 或 IOC            |
| 154  | 市价单需在连续交易阶段         | 在非连续交易阶段报入市价单                  |
| 155  | 市价单只支持期货期权          | 在非期货期权合约上报入市价单                 |
| 1005 | 无记录                 | 报单所对应的合约记录缺失                   |

**nRequestID:** 返回用户报单录入操作请求的 ID，该 ID 由用户在报单录入时指定。



**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 注意:

CShfeFtdcRspInfoField.ErrorID 为零（意味着本次报单录入成功），CShfeFtdcInputOrderField\* pInputOrder 中仅报单编号（交易系统给出的系统编号）和本地报单编号有意义，用于交易系统与会员系统之间报单的关联。该报单的具体内容需要从私有流获得。

CShfeFtdcInputOrderField 各数据字段的说明请参阅 OnRtnOrder 方法。

## 2.1.13.OnRspOrderAction 方法

报单操作应答。报单操作包括报单的撤销、报单的挂起、报单的激活、报单的修改。当会员系统发出过报单操作指令后，交易系统返回响应时，该方法会被调用。

### 函数原形:

```
void OnRspOrderAction(  
    CShfeFtdcOrderActionField* pOrderAction,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

### 参数:

**pOrderAction:** 指向报单操作结构的地址，包含了提交报单操作的输入数据和交易系统返回的报单编号。报单操作结构:

```
struct CShfeFtdcOrderActionField {  
    ///报单编号  
    TShfeFtdcOrderSysIDType OrderSysID;  
    ///本地报单编号  
    TShfeFtdcOrderLocalIDType OrderLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码
```

```
TShfeFtdcUserIDType UserID;
///价格，未使用
TShfeFtdcPriceType LimitPrice;
///数量变化，未使用
TShfeFtdcVolumeType VolumeChange;
///操作本地编号
TShfeFtdcOrderLocalIDType ActionLocalID;
///业务单元，未使用
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

**错误代码 错误提示**

|    |              |
|----|--------------|
| 2  | 合约找不到        |
| 3  | 会员找不到        |
| 4  | 客户找不到        |
| 8  | 报单操作字段错误     |
| 15 | 客户没有在该会员开户   |
| 16 | IOC 需在连续交易阶段 |
| 17 | GFA 需在集合竞价阶段 |
| 20 | GTD 报单过期了    |
| 22 | 交易所数据没有同步    |
| 23 | 结算组数据没有同步    |
| 24 | 报单找不到        |
| 26 | 当前状态禁止此项操作   |

**可能的原因**

|                                    |
|------------------------------------|
| 找不到报单操作中的合约                        |
| 报单操作中的会员找不到                        |
| 报单操作中的客户找不到                        |
| 报单操作中有字段值不合法（枚举值越界）                |
| 客户并没有在指定的会员中开户                     |
| 在非连续交易阶段企图操作 IOC 单                 |
| 在非集合竞价阶段企图操作 GFA 的报单               |
| GTD 报单中的 GTD 日期已经过期了               |
| 交易系统尚未初始化完毕，可以稍后再重试                |
| 交易系统尚未初始化完毕，可以稍后再重试                |
| 找不到要操作的报单                          |
| 对于激活操作，合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡 |
| 对于其他操作，交易状态不是连续交易或者集合竞价报单          |
| 报单已经全部成交了                          |
| 报单已经撤销了                            |
| 修改报单数量后，剩余报单数量小于 0                 |

|     |              |                                       |
|-----|--------------|---------------------------------------|
| 31  | 平仓时客户持仓不足    | 操作平仓报单时，客户的持仓不足                       |
| 32  | 超出客户限仓       | 激活报单时，超过了客户的一般限仓                      |
| 33  | 平仓时会员持仓不足    | 操作平仓报单时，会员的持仓不足                       |
| 34  | 超出会员限仓       | 激活报单时，超过了会员限仓                         |
| 35  | 找不到帐号        | 找不到应当使用的资金帐号                          |
| 36  | 资金不足         | 资金帐号中没有足够的资金                          |
| 37  | 不合法的数量       | 修改后报单的数量不是最小报单数量要求的正整数倍，或者不在有效报单数量范围内 |
| 48  | 价格非最小单位的倍数   | 修改后报单的价格不是合约的最小变动单位的整数倍               |
| 49  | 价格超出涨停板      | 修改后报单的价格高于合约的涨停板                      |
| 50  | 价格跌破跌停板      | 修改后报单的价格低于合约的跌停板                      |
| 51  | 没有交易权限       | 对指定合约、或者客户对指定的合约、或者用户没有交易权限           |
| 52  | 只能平仓         | 会员对指定合约、或者客户对指定的合约、或者用户只有平仓的权限        |
| 54  | 会话找不到        | 用户未登录                                 |
| 57  | 不能为其他会员操作    | 用户给非其所属的会员进行操作                        |
| 58  | 用户不匹配        | 报单操作中的用户和登录时的用户不匹配                    |
| 71  | 不能操作衍生报单     | 用户对衍生报单进行操作                           |
| 72  | 不允许自然人开仓     | 交割月，自然人类型的客户对开仓类型的报单做激活或修改的操作         |
| 76  | 报单已经被挂起      | 在报单挂起时，报单已经被挂起了                       |
| 77  | 报单已经被激活      | 在报单激活时，报单已经被激活了                       |
| 79  | 不被支持的报单类型    | 交易所不支持此种报单类型，例如：报单修改                  |
| 83  | 止损单仅用于连续交易   | 在非连续交易阶段操作止损单                         |
| 95  | 止损报单需说明止损价   | 止损单没有指定止损价                            |
| 96  | 保值额度不足       | 报单激活或修改后，客户的套保额度不足                    |
| 98  | 强平单需由管理员使用   | 非管理员用户操作强平单                           |
| 99  | 不能为其他用户操作    | 非授权用户操作同会员的其他用户报入的报单                  |
| 131 | 超出客户合约当日开仓限额 | 客户在某个合约上的开仓量超出开仓限额                    |
| 133 | 超出客户产品每秒撤单限额 | 客户在某个产品上一秒以内的撤单笔数超出限额                 |

**nRequestID:** 返回用户报单操作请求的 ID，该 ID 由用户在报单操作时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.14.OnRspQuoteInsert 方法

报价录入应答。当会员系统发出过报价录入指令后，交易系统返回响应时，该方法会被调用。

## 函数原形:

```
void OnRspQuoteInsert(
    CShfeFtdcInputQuoteField* pInputQuote,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

## 参数:

**pInputQuote:** 指向输入报价结构的地址, 包含了报价录入操作的输入数据, 和交易系统返回的报价编号。输入报价结构:

```
struct CShfeFtdcInputQuoteField {
    ///报价编号
    TShfeFtdcQuoteSysIDType QuoteSysID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
    ///本地报价编号
    TShfeFtdcOrderLocalIDType QuoteLocalID;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///买方组合开平标志
    TShfeFtdcCombOffsetFlagType BidCombOffsetFlag;
    ///买方组合套保标志
    TShfeFtdcCombHedgeFlagType BidCombHedgeFlag;
    ///买方价格
    TShfeFtdcPriceType BidPrice;
    ///卖方组合开平标志
    TShfeFtdcCombOffsetFlagType AskCombOffsetFlag;
    ///卖方组合套保标志
    TShfeFtdcCombHedgeFlagType AskCombHedgeFlag;
    ///卖方价格
    TShfeFtdcPriceType AskPrice;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
```

```

    ///IP 地址
    TShfeFtdcIPAddressType  IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType  MacAddress;
    ///询价编号
    TShfeFtdcOrderSysIDType  QuoteDemandID;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType      ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType     ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|    |            |
|----|------------|
| 2  | 合约找不到      |
| 3  | 会员找不到      |
| 4  | 客户找不到      |
| 7  | 报价字段错误     |
| 13 | 重复的报价      |
| 15 | 客户没有在该会员开户 |
| 22 | 交易所数据没有同步  |
| 23 | 结算组数据没有同步  |
| 26 | 当前状态禁止此项操作 |
| 31 | 平仓时客户持仓不足  |
| 32 | 超出客户限仓     |
| 33 | 平仓时会员持仓不足  |
| 34 | 超出会员限仓     |
| 35 | 找不到帐号      |
| 36 | 资金不足       |
| 37 | 不合法的数量     |
| 48 | 价格非最小单位的倍数 |
| 49 | 价格超出涨停板    |
| 50 | 价格跌破跌停板    |
| 51 | 没有交易权限     |
| 52 | 只能平仓       |
| 53 | 没有此项交易角色   |
| 54 | 会话找不到      |

**可能的原因**

|                                |
|--------------------------------|
| 找不到报价中的合约                      |
| 找不到报价中的会员                      |
| 找不到报价中的客户                      |
| 报价中有字段值不合法（枚举值越界）              |
| 报价中的本地报价号重复                    |
| 报价中的客户并没有在指定的会员中开户             |
| 交易系统尚未初始化完毕，可以稍后再重试            |
| 交易系统尚未初始化完毕，可以稍后再重试            |
| 合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡    |
| 客户的持仓不足                        |
| 该报价导致客户的一般持仓超过了限仓              |
| 会员的持仓不足                        |
| 该报价导致会员的持仓超过了限仓                |
| 找不到报价使用的资金帐号                   |
| 资金帐号中没有足够的资金                   |
| 报价数量不是最小报单数量要求的正整数倍，或者超过最大报单数量 |
| 报价的价格不是合约的最小变动单位的整数倍           |
| 报价的价格高于合约的涨停板                  |
| 报价的价格低于合约的跌停板                  |
| 对指定合约、或者客户对指定的合约、或者用户没有交易权限    |
| 会员对指定合约、或者客户对指定的合约、或者用户只有平仓的权限 |
| 会员在指定合约上不具有该客户对应的交易角色          |
| 用户未登录                          |

|      |              |                       |
|------|--------------|-----------------------|
| 57   | 不能为其他会员操作    | 用户给非其所属的会员进行操作        |
| 58   | 用户不匹配        | 报价中的用户和登录时的用户不匹配      |
| 72   | 不允许自然人开仓     | 交割月自然人类型的客户发起开仓请求     |
| 79   | 不被支持的报单类型    | 交易所不支持此种报单类型          |
| 96   | 保值额度不足       | 报入报价时，客户的套保额度不足       |
| 98   | 强平单需由管理员使用   | 非管理员用户报了强平单           |
| 101  | 结算会员不能做交易    | 报价的会员类型为结算会员          |
| 102  | 无法找到对应结算会员   | 找不到报价会员对应的结算会员        |
| 103  | 当日套保仓位不能平仓   | 套保仓位不应使用平今仓报价进行平仓     |
| 131  | 超出客户合约当日开仓限额 | 客户在某个合约上的开仓量超出开仓限额    |
| 132  | 超出客户产品每秒报单限额 | 客户在某个产品上一秒以内的报单笔数超出限额 |
| 1005 | 无记录          | 报价所对应的合约记录缺失          |

**nRequestID:** 返回用户报价录入操作请求的 ID，该 ID 由用户在报价录入时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.15.OnRspQuoteAction 方法

报价操作应答。包括报价的撤销、报价的挂起、报价的激活、报价的修改。当会员系统发出过报价操作指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspQuoteAction(  
    CShfeFtdcQuoteActionField* pQuoteAction,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pQuoteAction:** 指向报价操作结构的地址，包含了报价操作请求的输入数据和交易系统返回的报价编号。报价操作结构：

```
struct CShfeFtdcQuoteActionField {  
    ///报价编号  
    TShfeFtdcQuoteSysIDType QuoteSysID;  
    ///本地报价编号  
    TShfeFtdcOrderLocalIDType QuoteLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码
```

```

TShfeFtdcParticipantIDType ParticipantID;
///客户代码
TShfeFtdcClientIDType ClientID;
///交易用户代码
TShfeFtdcUserIDType UserID;
///操作本地编号
TShfeFtdcOrderLocalIDType ActionLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|    |            |
|----|------------|
| 2  | 合约找不到      |
| 3  | 会员找不到      |
| 4  | 客户找不到      |
| 8  | 报单操作字段错误   |
| 9  | 报价操作字段错误   |
| 15 | 客户没有在该会员开户 |
| 22 | 交易所数据没有同步  |
| 23 | 结算组数据没有同步  |
| 25 | 报价找不到      |
| 26 | 当前状态禁止此项操作 |
| 28 | 报单已经全部成交   |
| 29 | 报单已经撤销     |
| 35 | 找不到帐号      |

**可能的原因**

|                                       |
|---------------------------------------|
| 找不到报价操作中的合约                           |
| 报价操作中的会员找不到                           |
| 报价操作中的客户找不到                           |
| 报价操作衍生的报单中有字段值不合法（例如：价格不是浮点数或不在有效范围内） |
| 报价操作中有字段值不合法（枚举值越界或操作标志为修改、激活或挂起）     |
| 客户并没有在指定的会员中开户                        |
| 交易系统尚未初始化完毕，可以稍后再重试                   |
| 交易系统尚未初始化完毕，可以稍后再重试                   |
| 找不到要操作的报价                             |
| 对于激活操作，合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡    |
| 对于其他操作，交易状态不是连续交易或者集合竞价报单             |
| 报价派生的报单已经全部成交了                        |
| 报价派生的报单已经撤销了                          |
| 找不到应当使用的资金帐号                          |

|    |           |                             |
|----|-----------|-----------------------------|
| 36 | 资金不足      | 资金帐号中没有足够的资金                |
| 51 | 没有交易权限    | 对指定合约、或者客户对指定的合约、或者用户没有交易权限 |
| 54 | 会话找不到     | 用户未登录                       |
| 57 | 不能为其他会员操作 | 用户给非其所属的会员进行操作              |
| 58 | 用户不匹配     | 报价操作中的用户和登录时的用户不匹配          |
| 70 | 报价已经被取消   | 报价已经被取消                     |
| 99 | 不能为其他用户操作 | 非授权用户操作同会员的其他用户报入的报价        |

**nRequestID:** 返回用户报价操作请求的 ID，该 ID 由用户在报价操作时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.16.OnRspExecOrderInsert 方法

执行宣告录入应答。当会员系统执行宣告录入后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspExecOrderInsert(
    CShfeFtdcInputExecOrderField* pInputExecOrder,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

**pInputExecOrder:** 指向输入执行宣告结构的地址。输入执行宣告结构：

```
struct CShfeFtdcInputExecOrderField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地执行宣告编号
    TShfeFtdcOrderLocalIDType ExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
```



```

    ///开平标志
    TShfeFtdcOffsetFlagType  OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType  HedgeFlag;
    ///申请的持仓方向，即作为买方（多头）还是卖方（空头）来提出申请
    TShfeFtdcPosiDirectionType  PosiDirection;
    ///期权行权后是否保留期货持仓的标记，未使用
    TShfeFtdcExecOrderPositionFlagType  ReservePositionFlag;
    ///期权行权后生成的期货持仓是否自对冲
    TShfeFtdcExecOrderCloseFlagType  CloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType  BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType  BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType  IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType  MacAddress;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType  ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType  ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|    |            |
|----|------------|
| 2  | 合约找不到      |
| 3  | 会员找不到      |
| 4  | 客户找不到      |
| 15 | 客户没有在该会员开户 |
| 22 | 交易所数据没有同步  |
| 23 | 结算组数据没有同步  |
| 26 | 当前状态禁止此项操作 |
| 31 | 平仓时客户持仓不足  |
| 33 | 平仓时会员持仓不足  |
| 35 | 找不到帐号      |
| 36 | 资金不足       |
| 37 | 不合法的数量     |
| 51 | 没有交易权限     |
| 54 | 会话找不到      |

**可能的原因**

|                      |
|----------------------|
| 找不到执行宣告中的合约          |
| 执行宣告中的会员找不到          |
| 执行宣告中的客户找不到          |
| 执行宣告中的客户并没有在指定的会员中开户 |
| 交易系统尚未初始化完毕，可以稍后再重试  |
| 交易系统尚未初始化完毕，可以稍后再重试  |
| 合约交易状态不是连续交易、交易业务处理  |
| 执行宣告录入时，客户的持仓额度不足    |
| 执行宣告录入时，会员的持仓额度不足    |
| 找不到应当使用的资金帐号         |
| 资金帐号中没有足够的资金         |
| 执行宣告数量不合法            |
| 对指定合约、或者客户对指定的合约、    |
| 或者用户没有交易权限           |
| 用户未登录                |

|      |                |                     |
|------|----------------|---------------------|
| 57   | 不能为其他会员操作      | 用户给非其所属的会员进行操作      |
| 58   | 用户不匹配          | 执行宣告中的用户和登录时的用户不匹配  |
| 79   | 不被支持的报单类型      | 交易所不支持此种报单类型        |
| 89   | 执行宣告字段错误       | 执行宣告中有字段值不合法（枚举值越界） |
| 91   | 重复的执行宣告        | 执行宣告中的本地执行宣告号重复     |
| 94   | 执行宣告只能用于期权     | 执行宣告中的合约是非期权合约      |
| 101  | 结算会员不能做交易      | 执行宣告的会员类型为结算会员      |
| 102  | 无法找到对应结算会员     | 找不到执行宣告会员对应的结算会员    |
| 127  | 不在宣告期内         | 不在合约交割期（可行权周期）内     |
| 129  | 执行或放弃执行宣告不能为开仓 | 执行宣告的开平标志只能限定为平仓    |
| 146  | 只有持有多头仓位才能行权   | 只有期权买方才能行权          |
| 1005 | 无记录            | 执行宣告所对应的合约记录缺失      |

**nRequestID:** 返回用户执行宣告录入请求的 ID，该 ID 由用户在执行宣告录入时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.17.OnRspExecOrderAction 方法

执行宣告操作应答。执行宣告操作包括执行宣告的撤销、执行宣告的挂起、执行宣告的激活、执行宣告的修改。当会员系统执行宣告操作后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspExecOrderAction(  
    CShfeFtdcExecOrderActionField* pExecOrderAction,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pExecOrderAction:** 指向执行宣告操作结构的地址。执行宣告操作结构：

```
struct CShfeFtdcExecOrderActionField {  
    ///执行宣告编号  
    TShfeFtdcExecOrderSysIDType  ExecOrderSysID;  
    ///本地执行宣告编号  
    TShfeFtdcOrderLocalIDType  ExecOrderLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType  ActionFlag;  
    ///会员代码
```

```
TShfeFtdcParticipantIDType ParticipantID;
///客户代码
TShfeFtdcClientIDType ClientID;
///交易用户代码
TShfeFtdcUserIDType UserID;
///操作本地编号
TShfeFtdcOrderLocalIDType ActionLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示       | 可能的原因                       |
|------|------------|-----------------------------|
| 2    | 合约找不到      | 找不到执行宣告操作中的合约               |
| 3    | 会员找不到      | 执行宣告操作中的会员找不到               |
| 4    | 客户找不到      | 执行宣告操作中的客户找不到               |
| 15   | 客户没有在该会员开户 | 客户并没有在指定的会员中开户              |
| 22   | 交易所数据没有同步  | 交易系统尚未初始化完毕，可以稍后再重试         |
| 23   | 结算组数据没有同步  | 交易系统尚未初始化完毕，可以稍后再重试         |
| 26   | 当前状态禁止此项操作 | 合约交易状态不是连续交易、交易业务处理         |
| 35   | 找不到帐号      | 找不到应当使用的资金帐号                |
| 36   | 资金不足       | 资金帐号中没有足够的资金                |
| 51   | 没有交易权限     | 对指定合约、或者客户对指定的合约、或者用户没有交易权限 |
| 54   | 会话找不到      | 用户未登录                       |
| 57   | 不能为其他会员操作  | 用户给非其所属的会员进行操作              |
| 58   | 用户不匹配      | 执行宣告操作中的用户和登录时的用户不匹配        |
| 89   | 执行宣告字段错误   | 执行宣告中有字段值不合法                |
| 90   | 执行宣告操作字段错误 | 执行宣告操作中有字段值不合法（枚举值越界）       |
| 92   | 执行宣告已经取消   | 要操作的执行宣告已经取消                |
| 93   | 执行宣告找不到    | 要操作的执行宣告没有找到                |

|      |        |                  |
|------|--------|------------------|
| 127  | 不在宣告期内 | 不在合约交割期（可行权周期）内  |
| 1005 | 无记录    | 执行宣告操作所对应的合约记录缺失 |

**nRequestID:** 返回用户执行宣告操作请求的 ID，该 ID 由用户在执行宣告操作时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.18.OnRspQryPartAccount 方法

会员资金查询应答。当会员系统发出会员资金查询指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryPartAccount(  
    CShfeFtdcRspPartAccountField* pRspPartAccount,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pRspPartAccount:** 指向会员资金应答结构的地址。会员资金应答结构：

```
struct CShfeFtdcRspPartAccountField  
{  
    ///交易日  
    TShfeFtdcDateType   TradingDay;  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType   SettlementGroupID;  
    ///结算编号  
    TShfeFtdcSettlementIDType SettlementID;  
    ///上次结算准备金  
    TShfeFtdcMoneyType PreBalance;  
    ///当前保证金总额  
    TShfeFtdcMoneyType CurrMargin;  
    ///平仓盈亏  
    TShfeFtdcMoneyType CloseProfit;  
    ///期权权利金收支  
    TShfeFtdcMoneyType Premium;  
    ///入金金额  
    TShfeFtdcMoneyType Deposit;  
    ///出金金额  
    TShfeFtdcMoneyType Withdraw;
```

```
///期货结算准备金
TShfeFtdcMoneyType Balance;
///可提资金
TShfeFtdcMoneyType Available;
///资金帐号
TShfeFtdcAccountIDType AccountID;
///冻结的保证金
TShfeFtdcMoneyType FrozenMargin;
///冻结的权利金
TShfeFtdcMoneyType FrozenPremium;
///基本准备金
TShfeFtdcMoneyType BaseReserve;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户资金查询请求的 ID，该 ID 由用户在资金查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.19.OnRspQryOrder 方法

报单查询请求。当会员系统发出报单查询指令后，交易系统返回响应时，该方法会被调用。

**函数原形:**

```
void OnRspQryOrder(
    CShfeFtdcOrderField* pOrder,
    CShfeFtdcRspInfoField* pRspInfo,
```

```
int nRequestID,
bool bIsLast);
```

## 参数:

**pOrder:** 指向报单信息结构的地址。报单信息结构:

```
struct CshfeFtdcOrderField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType   SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///报单编号
    TShfeFtdcOrderSysIDType OrderSysID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType   ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
    ///报单价格条件
    TShfeFtdcOrderPriceTypeType   OrderPriceType;
    ///买卖方向
    TShfeFtdcDirectionType   Direction;
    ///组合开平标志
    TShfeFtdcCombOffsetFlagType   CombOffsetFlag;
    ///组合套保标志
    TShfeFtdcCombHedgeFlagType   CombHedgeFlag;
    ///价格
    TShfeFtdcPriceType   LimitPrice;
    ///数量
    TShfeFtdcVolumeType VolumeTotalOriginal;
    ///有效期类型
    TShfeFtdcTimeConditionType   TimeCondition;
    ///GTD 日期, 未使用
    TShfeFtdcDateType   GTDDate;
    ///成交量类型
    TShfeFtdcVolumeConditionType   VolumeCondition;
    ///最小成交量
    TShfeFtdcVolumeTypeMinVolume;
    ///触发条件
```

```

TShfeFtdcContingentConditionType    ContingentCondition;
///止损价，未使用

TShfeFtdcPriceType    StopPrice;
///强平原因

TShfeFtdcForceCloseReasonType ForceCloseReason;
///本地报单编号

TShfeFtdcOrderLocalIDType    OrderLocalID;
///自动挂起标志

TShfeFtdcBoolType    IsAutoSuspend;
///报单来源

TShfeFtdcOrderSourceType OrderSource;
///报单状态

TShfeFtdcOrderStatusType OrderStatus;
///报单类型

TShfeFtdcOrderTypeType    OrderType;
///今成交数量

TShfeFtdcVolumeType VolumeTraded;
///剩余数量

TShfeFtdcVolumeType VolumeTotal;
///报单日期

TShfeFtdcDateType    InsertDate;
///插入时间

TShfeFtdcTimeType    InsertTime;
///激活时间

TShfeFtdcTimeType    ActiveTime;
///挂起时间

TShfeFtdcTimeType    SuspendTime;
///最后修改时间

TShfeFtdcTimeType    UpdateTime;
///撤销时间

TShfeFtdcTimeType    CancelTime;
///最后修改交易用户代码

TShfeFtdcUserIDType ActiveUserID;
///优先权

TShfeFtdcPriorityTypePriority;
///按时间排队的序号

TShfeFtdcTimeSortIDType    TimeSortID;
///结算会员编号

TShfeFtdcParticipantIDTypeClearingPartID;
///业务单元

TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识

TShfeFtdcBusinessLocalIDType    BusinessLocalID;
///业务发生日期

```

```
TShfeFtdcDateType ActionDay;
//IP 地址
TShfeFtdcIPAddressType IPAddress;
//Mac 地址
TShfeFtdcMacAddressType MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    //错误代码
    TShfeFtdcErrorIDType ErrorID;
    //错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户报单查询请求的 ID，该 ID 由用户在报单查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.20.OnRspQryQuote 方法

报价查询应答。当会员系统发出报价查询指令后，交易系统返回响应时，该方法会被调用。

**函数原形:**

```
void OnRspQryQuote(
    CShfeFtdcQuoteField* pQuote,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

**参数:**

**pQuote:** 指向报价信息结构的地址。报价信息结构:

```
struct CShfeFtdcQuoteField {
```



```

///交易日
TShfeFtdcDateType   TradingDay;
///结算组代码
TShfeFtdcSettlementGroupIDType   SettlementGroupID;
///结算编号
TShfeFtdcSettlementIDType SettlementID;
///报价编号
TShfeFtdcQuoteSysIDType QuoteSysID;
///会员代码
TShfeFtdcParticipantIDType ParticipantID;
///客户代码
TShfeFtdcClientIDType   ClientID;
///交易用户代码
TShfeFtdcUserIDType UserID;
///数量
TShfeFtdcVolumeType Volume;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///本地报价编号
TShfeFtdcOrderLocalIDType   QuoteLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///买方组合开平标志
TShfeFtdcCombOffsetFlagType BidCombOffsetFlag;
///买方组合套保标志
TShfeFtdcCombHedgeFlagType BidCombHedgeFlag;
///买方价格
TShfeFtdcPriceType BidPrice;
///卖方组合开平标志
TShfeFtdcCombOffsetFlagType AskCombOffsetFlag;
///卖方组合套保标志
TShfeFtdcCombHedgeFlagType AskCombHedgeFlag;
///卖方价格
TShfeFtdcPriceType AskPrice;
///插入时间
TShfeFtdcTimeType InsertTime;
///撤销时间
TShfeFtdcTimeType CancelTime;
///成交时间
TShfeFtdcTimeType TradeTime;
///买方报单编号
TShfeFtdcOrderSysIDType BidOrderSysID;
///卖方报单编号
TShfeFtdcOrderSysIDType AskOrderSysID;

```

```
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TShfeFtdcDateType ActionDay;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
///询价编号
TShfeFtdcOrderSysIDType QuoteDemandID;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户报价查询请求的 ID，该 ID 由用户在报价查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.21.OnRspQryTrade 方法

成交单查询应答。当会员系统发出成交单查询指令后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryTrade(
    CShfeFtdcTradeField* pTrade,
    CShfeFtdcRspInfoField* pRspInfo,
```

```
int nRequestID,
bool bIsLast);
```

## 参数:

**pTrade:** 指向成交信息结构的地址。成交信息结构:

```
struct CShfeFtdcTradeField {
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///成交编号
    TShfeFtdcTradeIDType    TradeID;
    ///买卖方向
    TShfeFtdcDirectionType    Direction;
    ///报单编号
    TShfeFtdcOrderSysIDType OrderSysID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType    ClientID;
    ///交易角色
    TShfeFtdcTradingRoleType TradingRole;
    ///资金帐号
    TShfeFtdcAccountIDType    AccountID;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
    ///开平标志
    TShfeFtdcOffsetFlagType    OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType    HedgeFlag;
    ///价格
    TShfeFtdcPriceType    Price;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///成交时间
    TShfeFtdcTimeType    TradeTime;
    ///成交类型
    TShfeFtdcTradeTypeType    TradeType;
    ///成交价来源
    TShfeFtdcPriceSourceType    PriceSource;
    ///交易用户代码
```

```
TShfeFtdcUserIDType UserID;
///本地报单编号
TShfeFtdcOrderLocalIDType OrderLocalID;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TShfeFtdcDateType ActionDay;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户成交单请求的 ID，该 ID 由用户在成交单查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.22.OnRspQryClient 方法

会员客户查询应答。当会员系统发出会员客户查询指令后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryClient(
    CShfeFtdcRspClientField* pRspClient,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
```

**bool bIsLast);**

参数:

**pRspClient:** 指向客户信息结构的地址。客户信息结构:

```
struct CShfeFtdcRspClientField {  
    ///客户代码  
    TShfeFtdcClientIDType    ClientID;  
    ///客户名称  
    TShfeFtdcPartyNameType   ClientName;  
    ///证件类型  
    TShfeFtdcIdCardTypeType   IdentifiedCardType;  
    ///原证件号码  
    TShfeFtdcIdentifiedCardNoV1Type    UseLess;  
    ///交易角色  
    TShfeFtdcTradingRoleType TradingRole;  
    ///客户类型  
    TShfeFtdcClientTypeType   ClientType;  
    ///是否活跃  
    TShfeFtdcBoolType         IsActive;  
    ///会员号  
    TShfeFtdcParticipantIDTypeParticipantID;  
    ///证件号码  
    TShfeFtdcIdentifiedCardNoType IdentifiedCardNo;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType   ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户会员客户查询请求的 ID，该 ID 由用户在会员客户查询时指定。

**blsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.23.OnRspQryPartPosition 方法

会员持仓查询应答。当会员系统发出会员持仓查询指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryPartPosition(
    CShfeFtdcRspPartPositionField* pRspPartPosition,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool blsLast);
```

参数：

**pRspPartPosition:** 指向会员持仓应答结构的地址。会员持仓应答结构：

```
struct CShfeFtdcRspPartPositionField {
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///套保标志
    TShfeFtdcHedgeFlagType    HedgeFlag;
    ///持仓多空方向
    TShfeFtdcPosiDirectionType    PosiDirection;
    ///上日持仓
    TShfeFtdcVolumeTypeYdPosition;
    ///今日持仓
    TShfeFtdcVolumeTypePosition;
    ///多头冻结
    TShfeFtdcVolumeTypeLongFrozen;
    ///空头冻结
    TShfeFtdcVolumeTypeShortFrozen;
    ///昨日多头冻结
    TShfeFtdcVolumeTypeYdLongFrozen;
    ///昨日空头冻结
    TShfeFtdcVolumeTypeYdShortFrozen;
    ///合约代码
```

```
TShfeFtdcInstrumentIDType InstrumentID;  
///会员代码  
TShfeFtdcParticipantIDType ParticipantID;  
///交易角色  
TShfeFtdcTradingRoleType TradingRole;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户会员持仓查询请求的 ID，该 ID 由用户在会员持仓查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.24.OnRspQryClientPosition 方法

客户持仓查询应答。当会员系统发出客户持仓查询指令后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryClientPosition(  
    CShfeFtdcRspClientPositionField* pRspClientPosition,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

**pRspClientPosition:** 指向客户持仓应答结构的地址。客户持仓应答结构:

```
struct CShfeFtdcRspClientPositionField {
```

```

///交易日
TShfeFtdcDateType   TradingDay;
///结算组代码
TShfeFtdcSettlementGroupIDType   SettlementGroupID;
///结算编号
TShfeFtdcSettlementIDType SettlementID;
///套保标志
TShfeFtdcHedgeFlagType   HedgeFlag;
///持仓多空方向
TShfeFtdcPosiDirectionType   PosiDirection;
///上日持仓
TShfeFtdcVolumeTypeYdPosition;
///今日持仓
TShfeFtdcVolumeTypePosition;
///多头冻结
TShfeFtdcVolumeTypeLongFrozen;
///空头冻结
TShfeFtdcVolumeTypeShortFrozen;
///昨日多头冻结
TShfeFtdcVolumeTypeYdLongFrozen;
///昨日空头冻结
TShfeFtdcVolumeTypeYdShortFrozen;
///当日买成交量
TShfeFtdcVolumeTypeBuyTradeVolume;
///当日卖成交量
TShfeFtdcVolumeTypeSellTradeVolume;
///持仓成本
TShfeFtdcMoneyType PositionCost;
///昨日持仓成本
TShfeFtdcMoneyType YdPositionCost;
///占用的保证金
TShfeFtdcMoneyType UseMargin;
///冻结的保证金
TShfeFtdcMoneyType FrozenMargin;
///多头冻结的保证金
TShfeFtdcMoneyType LongFrozenMargin;
///空头冻结的保证金
TShfeFtdcMoneyType ShortFrozenMargin;
///冻结的权利金
TShfeFtdcMoneyType FrozenPremium;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///会员代码
TShfeFtdcParticipantIDType ParticipantID;

```



```
///客户代码
TShfeFtdcClientIDType    ClientID;

};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户客户持仓查询请求的 ID，该 ID 由用户在客户持仓查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.25.OnRspQryInstrument 方法

合约查询应答。当会员系统发出合约查询指令后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryInstrument(
    CShfeFtdcRspInstrumentField* pRspInstrument,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

**pRspInstrument:** 指向合约结构的地址。合约结构:

```
struct CShfeFtdcRspInstrumentField {
    ///结算组代码
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///产品代码
```

```

TShfeFtdcProductIDType ProductID;
///产品组代码
TShfeFtdcProductGroupIDType ProductGroupID;
///基础商品代码
TShfeFtdcInstrumentIDType UnderlyingInstrID;
///产品类型
TShfeFtdcProductClassType ProductClass;
///持仓类型
TShfeFtdcPositionType PositionType;
///执行价
TShfeFtdcPriceType StrikePrice;
///期权类型
TShfeFtdcOptionsType OptionsType;
///合约数量乘数
TShfeFtdcVolumeMultipleType VolumeMultiple;
///合约基础商品乘数
TShfeFtdcUnderlyingMultipleType UnderlyingMultiple;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///合约名称
TShfeFtdcInstrumentNameType InstrumentName;
///交割年份
TShfeFtdcYearType DeliveryYear;
///交割月
TShfeFtdcMonthType DeliveryMonth;
///提前月份
TShfeFtdcAdvanceMonthType AdvanceMonth;
///当前是否交易
TShfeFtdcBoolType IsTrading;
///创建日
TShfeFtdcDateType CreateDate;
///上市日
TShfeFtdcDateType OpenDate;
///到期日
TShfeFtdcDateType ExpireDate;
///开始交割日
TShfeFtdcDateType StartDelivDate;
///最后交割日
TShfeFtdcDateType EndDelivDate;
///挂牌基准价
TShfeFtdcPriceType BasisPrice;
///市价单最大下单量
TShfeFtdcVolumeType MaxMarketOrderVolume;
///市价单最小下单量

```

```
TShfeFtdcVolumeTypeMinMarketOrderVolume;
///限价单最大下单量
TShfeFtdcVolumeTypeMaxLimitOrderVolume;
///限价单最小下单量
TShfeFtdcVolumeTypeMinLimitOrderVolume;
///最小变动价位
TShfeFtdcPriceType PriceTick;
///交割月自然人开仓
TShfeFtdcMonthCountType AllowDelivPersonOpen;
///币种代码
TShfeFtdcCurrencyIDType CurrencyID;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示   | 可能的原因      |
|------|--------|------------|
| 54   | 会话找不到  | 用户未登录      |
| 80   | 用户无此权限 | 只有交易用户能做查询 |

**nRequestID:** 返回用户合约查询请求的 ID，该 ID 由用户在合约查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.26.OnRspQryInstrumentStatus 方法

合约交易状态查询应答。当会员系统发出合约交易状态查询指令后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryInstrumentStatus(
    CShfeFtdcInstrumentStatusField* pInstrumentStatus,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

**pInstrumentStatus:** 指向合约交易状态结构的地址。合约交易状态结构:

```
struct CshfeFtdcInstrumentStatusField {  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;  
    ///合约代码  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///合约交易状态  
    TShfeFtdcInstrumentStatusType  InstrumentStatus;  
    ///交易阶段编号  
    TShfeFtdcTradingSegmentSNTYPE    TradingSegmentSN;  
    ///进入本状态时间  
    TShfeFtdcTimeType    EnterTime;  
    ///进入本状态原因  
    TShfeFtdcInstStatusEnterReasonType  EnterReason;  
    ///进入本状态日期  
    TShfeFtdcDateType EnterDate;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示   | 可能的原因      |
|------|--------|------------|
| 54   | 会话找不到  | 用户未登录      |
| 80   | 用户无此权限 | 只有交易用户能做查询 |

**nRequestID:** 返回用户合约交易状态查询请求的 ID，该 ID 由用户在合约交易状态查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

2.1.27.OnRspQryBulletin 方法

交易所公告查询请求应答。当会员系统发出交易所公告查询指令后，交易系统返回响应时，该方法会被调用。

## 函数原形:

```
void OnRspQryBulletin(
    CShfeFtdcBulletinField* pBulletin,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

## 参数:

**pBulletin:** 指向交易所公告结构的地址。交易所公告结构:

```
struct CShfeFtdcBulletinField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///公告编号
    TShfeFtdcBulletinIDType   BulletinID;
    ///序列号
    TShfeFtdcSequenceNoType   SequenceNo;
    ///公告类型
    TShfeFtdcNewsTypeType   NewsType;
    ///紧急程度
    TShfeFtdcNewsUrgencyType   NewsUrgency;
    ///发送时间
    TShfeFtdcTimeType   SendTime;
    ///消息摘要
    TShfeFtdcAbstractType   Abstract;
    ///消息来源
    TShfeFtdcComeFromType   ComeFrom;
    ///消息正文
    TShfeFtdcContentType   Content;
    ///WEB 地址
    TShfeFtdcURLLinkType   URLLink;
    ///市场代码
    TShfeFtdcMarketIDType   MarketID;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType   ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType   ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示   | 可能的原因      |
|------|--------|------------|
| 54   | 会话找不到  | 用户未登录      |
| 80   | 用户无此权限 | 只有交易用户能做查询 |

**nRequestID:** 返回用户交易所公告查询请求的 ID，该 ID 由用户在交易所公告查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.28.OnRspQryMarketData 方法

普通行情查询应答，当会员系统发出过普通行情查询，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryMarketData(  
    CShfeFtdcMarketDataField* pMarketData,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

**pMarketData:** 返回的市场行情信息的地址。市场行情信息结构:

```
struct CShfeFtdcMarketDataField {  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;  
    ///结算编号  
    TShfeFtdcSettlementIDType SettlementID;  
    ///最新价  
    TShfeFtdcPriceType    LastPrice;  
    ///昨结算  
    TShfeFtdcPriceType    PreSettlementPrice;  
    ///昨收盘  
    TShfeFtdcPriceType    PreClosePrice;  
    ///昨持仓量  
    TShfeFtdcLargeVolumeTypePreOpenInterest;  
    ///今开盘  
    TShfeFtdcPriceType    OpenPrice;
```

```

    ///最高价
    TShfeFtdcPriceType  HighestPrice;
    ///最低价
    TShfeFtdcPriceType  LowestPrice;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///成交金额
    TShfeFtdcMoneyType  Turnover;
    ///持仓量
    TShfeFtdcLargeVolumeTypeOpenInterest;
    ///今收盘
    TShfeFtdcPriceType  ClosePrice;
    ///今结算
    TShfeFtdcPriceType  SettlementPrice;
    ///涨停板价
    TShfeFtdcPriceType  UpperLimitPrice;
    ///跌停板价
    TShfeFtdcPriceType  LowerLimitPrice;
    ///昨虚实度
    TShfeFtdcRatioType  PreDelta;
    ///今虚实度
    TShfeFtdcRatioType  CurrDelta;
    ///最后修改时间
    TShfeFtdcTimeType   UpdateTime;
    ///最后修改毫秒
    TShfeFtdcMillisecType   UpdateMillisec;
    ///合约代码
    TShfeFtdcInstrumentIDTypeInstrumentID;
    ///业务发生日期
    TShfeFtdcDateType  ActionDay;
};
```

**pRspInfo:** 返回用户响应信息的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType      ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType     ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示   | 可能的原因      |
|------|--------|------------|
| 54   | 会话找不到  | 用户未登录      |
| 80   | 用户无此权限 | 只有交易用户能做查询 |

**nRequestID:** 返回用户普通行情查询请求的 ID，该 ID 由用户在普通行情查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.29.OnRspQryHedgeVolume 方法

套保额度应答。当会员系统执行套保额度查询操作后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryHedgeVolume(
    CShfeFtdcHedgeVolumeField* pHedgeVolume,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

**pHedgeVolume:** 指向套保额度量结构的地址。套保额度量结构：

```
struct CShfeFtdcHedgeVolumeField {
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType    ClientID;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
    ///多头套保额度最初申请量，单位为手。
    TShfeFtdcVolumeTypeLongVolumeOriginal;
    ///空头套保额度最初申请量，单位为手。
    TShfeFtdcVolumeTypeShortVolumeOriginal;
    ///多头套保额度，单位为手。
    TShfeFtdcVolumeTypeLongVolume;
    ///空头套保额度，单位为手。
    TShfeFtdcVolumeTypeShortVolume;
```



```
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                                                         |
|------|-----------|---------------------------------------------------------------|
| 54   | 会话找不到     | 用户未登录                                                         |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                                                  |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况;<br>交易所启用跨会员联合套保后, 仅管理员可以做此<br>操作 |

**nRequestID:** 返回用户执行套保额度查询的 ID, 该 ID 由用户在执行套保额度查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.30.OnRtnTrade 方法

成交回报。当发生成交时交易系统会通知会员系统, 该方法会被调用。

函数原形:

```
void OnRtnTrade(CShfeFtdcTradeField* pTrade);
```

参数:

**pTrade:** 指向成交回报结构的地址。注意: 成交回报中的某些字段未使用, 交易系统返回为空值。成交回报结构:

```
struct CShfeFtdcTradeField {  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;  
    ///结算编号  
    TShfeFtdcSettlementIDType SettlementID;  
    ///成交编号
```

```

TShfeFtdcTradeIDType    TradeID;
///买卖方向
TShfeFtdcDirectionType  Direction;
///报单编号
TShfeFtdcOrderSysIDType OrderSysID;
///会员代码
TShfeFtdcParticipantIDType ParticipantID;
///客户代码
TShfeFtdcClientIDType    ClientID;
///交易角色，未使用
TShfeFtdcTradingRoleType TradingRole;
///资金帐号，未使用
TShfeFtdcAccountIDType   AccountID;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///开平标志
TShfeFtdcOffsetFlagType  OffsetFlag;
///套保标志
TShfeFtdcHedgeFlagType   HedgeFlag;
///价格
TShfeFtdcPriceType       Price;
///数量
TShfeFtdcVolumeType      Volume;
///成交时间
TShfeFtdcTimeType        TradeTime;
///成交类型
TShfeFtdcTradeTypeType   TradeType;
///成交价来源，未使用
TShfeFtdcPriceSourceType PriceSource;
///交易用户代码
TShfeFtdcUserIDType      UserID;
///本地报单编号
TShfeFtdcOrderLocalIDType OrderLocalID;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///业务单元，未使用
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TShfeFtdcDateType        ActionDay;
};

```

## 2.1.31.OnRtnOrder 方法

报单回报。当会员系统进行报单录入、报单操作及其它原因（如部分成交）导致报单状态发生变化时，交易系统会主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRtnOrder(CShfeFtdcOrderField* pOrder);
```

参数：

**pOrder**：指向报单回报结构的地址。注意：报单回报中的某些字段未使用，交易系统返回为空值。报单回报结构：

```
struct CShfeFtdcOrderField {  
    ///交易日，未使用  
    TShfeFtdcDateType   TradingDay;  
    ///结算组代码，未使用  
    TShfeFtdcSettlementGroupIDType   SettlementGroupID;  
    ///结算编号，未使用  
    TShfeFtdcSettlementIDType SettlementID;  
    ///报单编号  
    TShfeFtdcOrderSysIDType OrderSysID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType   ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///合约代码  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///报单价格条件  
    TShfeFtdcOrderPriceTypeType   OrderPriceType;  
    ///买卖方向  
    TShfeFtdcDirectionType   Direction;  
    ///组合开平标志  
    TShfeFtdcCombOffsetFlagType   CombOffsetFlag;  
    ///组合套保标志  
    TShfeFtdcCombHedgeFlagType   CombHedgeFlag;  
    ///价格  
    TShfeFtdcPriceType   LimitPrice;  
    ///数量  
    TShfeFtdcVolumeType VolumeTotalOriginal;
```

```

///有效期类型
TShfeFtdcTimeConditionType    TimeCondition;
///GTD 日期
TShfeFtdcDateType    GTDDate;
///成交量类型
TShfeFtdcVolumeConditionType    VolumeCondition;
///最小成交量
TShfeFtdcVolumeTypeMinVolume;
///触发条件
TShfeFtdcContingentConditionType    ContingentCondition;
///止损价
TShfeFtdcPriceType    StopPrice;
///强平原因
TShfeFtdcForceCloseReasonType    ForceCloseReason;
///本地报单编号
TShfeFtdcOrderLocalIDType    OrderLocalID;
///自动挂起标志
TShfeFtdcBoolType    IsAutoSuspend;
///报单来源，未使用
TShfeFtdcOrderSourceType    OrderSource;
///报单状态
TShfeFtdcOrderStatusType    OrderStatus;
///报单类型
TShfeFtdcOrderTypeType    OrderType;
///今成交数量，未使用
TShfeFtdcVolumeTypeVolumeTraded;
///剩余数量
TShfeFtdcVolumeTypeVolumeTotal;
///报单日期
TShfeFtdcDateType    InsertDate;
///插入时间
TShfeFtdcTimeType    InsertTime;
///激活时间，未使用
TShfeFtdcTimeType    ActiveTime;
///挂起时间，未使用
TShfeFtdcTimeType    SuspendTime;
///最后修改时间
TShfeFtdcTimeType    UpdateTime;
///撤销时间，未使用
TShfeFtdcTimeType    CancelTime;
///最后修改交易用户代码
TShfeFtdcUserIDType    ActiveUserID;
///优先权，未使用
TShfeFtdcPriorityTypePriority;

```

```

    ///按时间排队的序号，未使用
    TShfeFtdcTimeSortIDType TimeSortID;
    ///结算会员编号，未使用
    TShfeFtdcParticipantIDType ClearingPartID;
    ///业务单元，未使用
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///业务发生日期
    TShfeFtdcDateType ActionDay;
    ///IP 地址，未使用
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址，未使用
    TShfeFtdcMacAddressType MacAddress;
};

```

### 2.1.32. OnRtnQuote 方法

报价回报。当会员系统进行报价录入、报价操作导致报价状态发生变化时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRtnQuote(CShfeFtdcQuoteField* pQuote);
```

参数：

**pQuote:** 指向报价结构的地址。报价结构：

```

struct CShfeFtdcQuoteField {
    ///交易日
    TShfeFtdcDateType TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///报价编号
    TShfeFtdcQuoteSysIDType QuoteSysID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码

```

```

TShfeFtdcUserIDType UserID;
///数量
TShfeFtdcVolumeType Volume;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///本地报价编号
TShfeFtdcOrderLocalIDType    QuoteLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///买方组合开平标志
TShfeFtdcCombOffsetFlagType BidCombOffsetFlag;
///买方组合套保标志
TShfeFtdcCombHedgeFlagType BidCombHedgeFlag;
///买方价格
TShfeFtdcPriceType BidPrice;
///卖方组合开平标志
TShfeFtdcCombOffsetFlagType AskCombOffsetFlag;
///卖方组合套保标志
TShfeFtdcCombHedgeFlagType AskCombHedgeFlag;
///卖方价格
TShfeFtdcPriceType AskPrice;
///插入时间
TShfeFtdcTimeType InsertTime;
///撤销时间
TShfeFtdcTimeType CancelTime;
///成交时间
TShfeFtdcTimeType TradeTime;
///买方报单编号
TShfeFtdcOrderSysIDType BidOrderSysID;
///卖方报单编号
TShfeFtdcOrderSysIDType AskOrderSysID;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TShfeFtdcDateType ActionDay;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
///询价编号
TShfeFtdcOrderSysIDType QuoteDemandID;
};

```

### 2.1.33.OnRtnExecOrder 方法

执行宣告回报。当会员系统进行执行宣告录入、执行宣告操作导致执行宣告状态发生变化时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRtnExecOrder(CShfeFtdcExecOrderField* pExecOrder);
```

参数：

**pExecOrder:** 指向执行宣告结构的地址。执行宣告结构：

```
struct CShfeFtdcExecOrderField {
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType    ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地执行宣告编号
    TShfeFtdcOrderLocalIDType    ExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///开平标志
    TShfeFtdcOffsetFlagType    OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType    HedgeFlag;
    ///申请的持仓方向，即作为买方（多头）还是卖方（空头）来提出申请
    TShfeFtdcPosiDirectionType    PosiDirection;
    ///期权行权后是否保留期货持仓的标记，未使用
    TShfeFtdcExecOrderPositionFlagType    ReservePositionFlag;
    ///期权行权后生成的期货持仓是否自对冲
    TShfeFtdcExecOrderCloseFlagType CloseFlag;
```

```

///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///执行宣告编号
TShfeFtdcExecOrderSysIDType ExecOrderSysID;
///报单日期
TShfeFtdcDateType InsertDate;
///插入时间
TShfeFtdcTimeType InsertTime;
///撤销时间
TShfeFtdcTimeType CancelTime;
///执行结果
TShfeFtdcExecResultType ExecResult;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TShfeFtdcDateType ActionDay;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};

```

### 2.1.34. OnRtnInstrumentStatus 方法

合约交易状态改变回报。当合约交易状态发生变化时，交易系统会主动通知会员系统，该方法会被调用。

**函数原形：**

```

void OnRtnInstrumentStatus(
    CShfeFtdcInstrumentStatusField* pInstrumentStatus);

```

**参数：**

**pInstrumentStatus：** 指向合约状态结构的地址。合约状态结构：

```

struct CShfeFtdcInstrumentStatusField {
    ///结算组代码
    TShfeFtdcSettlementGroupIDType SettlementGroupID;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
};

```



```

    ///合约交易状态
    TShfeFtdcInstrumentStatusType InstrumentStatus;
    ///交易阶段编号
    TShfeFtdcTradingSegmentSNTYPE TradingSegmentSN;
    ///进入本状态时间
    TShfeFtdcTimeType EnterTime;
    ///进入本状态原因
    TShfeFtdcInstStatusEnterReasonType EnterReason;
    ///进入本状态日期
    TShfeFtdcDateType EnterDate;
};

```

## 2.1.35.OnRtnInsInstrument 方法

增加合约通知。当会员系统登录成功后，交易系统会把系统中增加的合约通过公共流通知给会员系统。

函数原形：

```
void OnRtnInsInstrument(CShfeFtdcInstrumentField* pInstrument);
```

参数：

**pInstrument:** 指向合约结构的地址。合约结构：

```

struct CShfeFtdcInstrumentField {
    ///结算组代码
    TShfeFtdcSettlementGroupIDType SettlementGroupID;
    ///产品代码
    TShfeFtdcProductIDType ProductID;
    ///产品组代码
    TShfeFtdcProductGroupIDType ProductGroupID;
    ///基础商品代码
    TShfeFtdcInstrumentIDType UnderlyingInstrID;
    ///产品类型
    TShfeFtdcProductClassType ProductClass;
    ///持仓类型
    TShfeFtdcPositionType PositionType;
    ///执行价
    TShfeFtdcPriceType StrikePrice;
    ///期权类型
    TShfeFtdcOptionsType OptionsType;
    ///合约数量乘数

```

```

TShfeFtdcVolumeMultipleType  VolumeMultiple;
///合约基础商品乘数
TShfeFtdcUnderlyingMultipleType  UnderlyingMultiple;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///合约名称
TShfeFtdcInstrumentNameType  InstrumentName;
///交割年份
TShfeFtdcYearType  DeliveryYear;
///交割月
TShfeFtdcMonthType  DeliveryMonth;
///提前月份
TShfeFtdcAdvanceMonthType  AdvanceMonth;
///当前是否交易
TShfeFtdcBoolType  IsTrading;
///币种代码
TShfeFtdcCurrencyIDType  CurrencyID;
};

```

### 2.1.36. OnRtnBulletin 方法

公告通知。交易所通过交易系统发布公告时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRtnBulletin(CShfeFtdcBulletinField* pBulletin);
```

参数：

**pBulletin：** 指向公告结构的地址。公告结构：

```

struct CShfeFtdcBulletinField {
    ///交易日
    TShfeFtdcDateType  TradingDay;
    ///公告编号
    TShfeFtdcBulletinIDType  BulletinID;
    ///序列号
    TShfeFtdcSequenceNoType  SequenceNo;
    ///公告类型
    TShfeFtdcNewsTypeType  NewsType;
    ///紧急程度
    TShfeFtdcNewsUrgencyType  NewsUrgency;
};

```

```

    ///发送时间
    TShfeFtdcTimeType   SendTime;
    ///消息摘要
    TShfeFtdcAbstractType   Abstract;
    ///消息来源
    TShfeFtdcComeFromType   ComeFrom;
    ///消息正文
    TShfeFtdcContentType   Content;
    ///WEB 地址
    TShfeFtdcURLLinkType   URLLink;
    ///市场代码
    TShfeFtdcMarketIDType   MarketID;
};

```

### 2.1.37.OnRtnFlowMessageCancel 方法

数据流回退通知。由交易系统发生灾备切换之后，用户重新登录交易系统并订阅某个数据流（私有流或公共流）时，交易系统主动通知会员系统该数据流某些报文被取消了，该方法会被调用。

函数原形：

```

void OnRtnFlowMessageCancel(
    CShfeFtdcFlowMessageCancelField* pFlowMessageCancel);

```

参数：

**pFlowMessageCancel:** 指向数据流回退结构的地址。数据流回退结构：

```

struct CShfeFtdcFlowMessageCancelField{
    ///序列系列号
    TShfeFtdcSequenceSeriesType   SequenceSeries;
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///数据中心代码
    TShfeFtdcDataCenterIDType   DataCenterID;
    ///回退起始序列号
    TShfeFtdcSequenceNoType   StartSequenceNo;
    ///回退结束序列号
    TShfeFtdcSequenceNoType   EndSequenceNo;
};
SequenceSeries: 发生回退的数据流代码（私有流或公共流）

```

回退的报文区间为： (StartSequenceNo, EndSequenceNo]

### 2.1.38.OnErrRtnOrderInsert 方法

报单录入错误回报。当会员系统发出报单录入指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnOrderInsert(  
    CShfeFtdcInputOrderField* pInputOrder,  
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pInputOrder:** 指向报单录入结构的地址，包含了提交报单录入时的输入数据。输入报单结构：

```
struct CShfeFtdcInputOrderField {  
    ///报单编号，该字段由交易系统返回。  
    TShfeFtdcOrderSysIDType OrderSysID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///合约代码  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///报单价格条件  
    TShfeFtdcOrderPriceTypeType OrderPriceType;  
    ///买卖方向  
    TShfeFtdcDirectionType Direction;  
    ///组合开平标志  
    TShfeFtdcCombOffsetFlagType CombOffsetFlag;  
    ///组合套保标志  
    TShfeFtdcCombHedgeFlagType CombHedgeFlag;  
    ///价格  
    TShfeFtdcPriceType LimitPrice;  
    ///数量  
    TShfeFtdcVolumeType VolumeTotalOriginal;  
    ///有效期类型  
    TShfeFtdcTimeConditionType TimeCondition;
```

```
///GTD 日期
TShfeFtdcDateType  GTDDate;
///成交量类型
TShfeFtdcVolumeConditionType  VolumeCondition;
///最小成交量
TShfeFtdcVolumeTypeMinVolume;
///触发条件
TShfeFtdcContingentConditionType  ContingentCondition;
///止损价
TShfeFtdcPriceType  StopPrice;
///强平原因
TShfeFtdcForceCloseReasonType  ForceCloseReason;
///本地报单编号
TShfeFtdcOrderLocalIDType  OrderLocalID;
///自动挂起标志
TShfeFtdcBoolType  IsAutoSuspend;
///业务单元
TShfeFtdcBusinessUnitType  BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType  BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType  IPAddress;
///Mac 地址
TShfeFtdcMacAddressType  MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType  ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType  ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示         | 可能的原因                                   |
|------|--------------|-----------------------------------------|
| 2    | 合约找不到        | 找不到报单中的合约                               |
| 3    | 会员找不到        | 找不到报单中的会员                               |
| 4    | 客户找不到        | 找不到报单中的客户                               |
| 6    | 报单字段错误       | 报单中有字段值不合法（枚举值越界）或发现非强平单中填写了强平原因为非整数倍强平 |
| 12   | 重复的报单        | 报单中的本地报单号重复                             |
| 15   | 客户没有在该会员开户   | 报单中的客户并没有在指定的会员中开户                      |
| 16   | IOC 需在连续交易阶段 | 在非连续交易阶段报入 IOC 单                        |
| 17   | GFA 需在集合竞价阶段 | 在非集合竞价阶段报入 GFA 的报单                      |

|      |                     |                                 |
|------|---------------------|---------------------------------|
| 19   | 数量约束应在 IOC 单上       | 数量约束不是任意数量的报单的时间条件不是 IOC        |
| 20   | GTD 报单过期了           | GTD 报单中的 GTD 日期已经过期了            |
| 21   | 最小数量大于报单数量          | 有最小数量条件的报单的最小数量大于报单数量           |
| 22   | 交易所数据没有同步           | 交易系统尚未初始化完毕, 可以稍后再重试            |
| 23   | 结算组数据没有同步           | 交易系统尚未初始化完毕, 可以稍后再重试            |
| 26   | 当前状态禁止此项操作          | 合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡     |
| 31   | 平仓时客户持仓不足           | 报入平仓报单时, 客户的持仓不足                |
| 32   | 超出客户限仓              | 报入开仓报单时, 超过了客户的一般限仓             |
| 33   | 平仓时会员持仓不足           | 报入平仓报单时, 会员的持仓不足                |
| 34   | 超出会员限仓              | 报入开仓报单时, 超过了会员限仓                |
| 35   | 找不到帐号               | 找不到报单使用的资金帐号                    |
| 36   | 资金不足                | 资金帐号中没有足够的资金                    |
| 37   | 不合法的数量              | 报单数量不是最小报单数量要求的正整数倍, 或者超过最大报单数量 |
| 48   | 价格非最小单位的倍数          | 报单的价格不是合约的最小变动单位的整数倍            |
| 49   | 价格超出涨停板             | 报单的价格高于合约的涨停板                   |
| 50   | 价格跌破跌停板             | 报单的价格低于合约的跌停板                   |
| 51   | 没有交易权限              | 对指定合约、或者客户对指定的合约、或者用户没有交易权限     |
| 52   | 只能平仓                | 会员对指定合约、或者客户对指定的合约、或者用户只有平仓的权限  |
| 53   | 没有此项交易角色            | 会员在指定合约上不具有该客户对应的交易角色           |
| 54   | 会话找不到               | 用户未登录                           |
| 57   | 不能为其他会员操作           | 用户给非其所属的会员进行操作                  |
| 58   | 用户不匹配               | 报单中的用户和登录时的用户不匹配                |
| 72   | 不允许自然人开仓            | 交割月自然人类型的客户发起开仓请求               |
| 78   | GTD 报单没有设定日期        | GTD 报单没有指定 GTD 日期               |
| 79   | 不被支持的报单类型           | 交易所不支持此种报单类型                    |
| 83   | 止损单仅用于连续交易          | 在非连续交易阶段报入止损单                   |
| 84   | 止损单需是 IOC 或 GFD     | 止损单的时间条件既不是 IOC, 也不是 GFD        |
| 95   | 止损报单需说明止损价          | 止损单没有指定止损价                      |
| 96   | 保值额度不足              | 报入套保报单时, 客户的套保额度不足              |
| 98   | 强平单需由管理员使用          | 非管理员用户报了强平单                     |
| 101  | 结算会员不能做交易           | 报单的会员类型为结算会员                    |
| 102  | 无法找到对应结算会员          | 找不到报单会员对应的结算会员                  |
| 103  | 当日套保仓位不能平仓          | 套保仓位不应使用平仓报单进行平仓                |
| 114  | 最优价单不能排队            | 最优价单的时间条件不是 IOC                 |
| 131  | 超出客户合约当日开仓限额        | 客户在某个合约上的开仓量超出开仓限额              |
| 132  | 超出客户产品每秒报单限额        | 客户在某个产品上一秒以内的报单笔数超出限额           |
| 153  | 市价单必须是 GFD 或者 IOC 单 | 市价单时间条件不为 GFD 或 IOC             |
| 154  | 市价单需在连续交易阶段         | 在非连续交易阶段报入市价单                   |
| 155  | 市价单只支持期货期权          | 在非期货期权合约上报入市价单                  |
| 1005 | 无记录                 | 报单所对应的合约记录缺失                    |

## 2.1.39.OnErrRtnOrderAction 方法

报单操作错误回报。当会员系统发出报单操作指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnOrderAction(  
    CShfeFtdcOrderActionField* pOrderAction,  
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pOrderAction**：指向报单操作结构的地址，包含了提交报单操作的输入数据和交易系统返回的报单编号。报单操作结构：

```
struct CShfeFtdcOrderActionField {  
    ///报单编号，该字段由交易系统返回。  
    TShfeFtdcOrderSysIDType OrderSysID;  
    ///本地报单编号  
    TShfeFtdcOrderLocalIDType OrderLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///价格，未使用  
    TShfeFtdcPriceType LimitPrice;  
    ///数量变化，未使用  
    TShfeFtdcVolumeType VolumeChange;  
    ///操作本地编号  
    TShfeFtdcOrderLocalIDType ActionLocalID;  
    ///业务单元，未使用  
    TShfeFtdcBusinessUnitType BusinessUnit;  
    ///本地业务标识  
    TShfeFtdcBusinessLocalIDType BusinessLocalID;  
    ///IP 地址  
    TShfeFtdcIPAddressType IPAddress;  
    ///Mac 地址  
    TShfeFtdcMacAddressType MacAddress;
```

};

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType      ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType     ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|    |              |
|----|--------------|
| 2  | 合约找不到        |
| 3  | 会员找不到        |
| 4  | 客户找不到        |
| 8  | 报单操作字段错误     |
| 15 | 客户没有在该会员开户   |
| 16 | IOC 需在连续交易阶段 |
| 17 | GFA 需在集合竞价阶段 |
| 20 | GTD 报单过期了    |
| 22 | 交易所数据没有同步    |
| 23 | 结算组数据没有同步    |
| 24 | 报单找不到        |
| 26 | 当前状态禁止此项操作   |
| 28 | 报单已经全部成交     |
| 29 | 报单已经撤销       |
| 30 | 没有足够的数量供修改   |
| 31 | 平仓时客户持仓不足    |
| 32 | 超出客户限仓       |
| 33 | 平仓时会员持仓不足    |
| 34 | 超出会员限仓       |
| 35 | 找不到帐号        |
| 36 | 资金不足         |
| 37 | 不合法的数量       |
| 48 | 价格非最小单位的倍数   |
| 49 | 价格超出涨停板      |
| 50 | 价格跌破跌停板      |
| 51 | 没有交易权限       |
| 52 | 只能平仓         |

**可能的原因**

|                                       |
|---------------------------------------|
| 找不到报单操作中的合约                           |
| 报单操作中的会员找不到                           |
| 报单操作中的客户找不到                           |
| 报单操作中有字段值不合法（枚举值越界）                   |
| 客户并没有在指定的会员中开户                        |
| 在非连续交易阶段企图操作 IOC 单                    |
| 在非集合竞价阶段企图操作 GFA 的报单                  |
| GTD 报单中的 GTD 日期已经过期了                  |
| 交易系统尚未初始化完毕，可以稍后再重试                   |
| 交易系统尚未初始化完毕，可以稍后再重试                   |
| 找不到要操作的报单                             |
| 对于激活操作，合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡    |
| 对于其他操作，交易状态不是连续交易或者集合竞价报单             |
| 报单已经全部成交了                             |
| 报单已经撤销了                               |
| 修改报单数量后，剩余报单数量小于 0                    |
| 操作平仓报单时，客户的持仓不足                       |
| 激活报单时，超过了客户的一般限仓                      |
| 操作平仓报单时，会员的持仓不足                       |
| 激活报单时，超过了会员限仓                         |
| 找不到应当使用的资金帐号                          |
| 资金帐号中没有足够的资金                          |
| 修改后报单的数量不是最小报单数量要求的正整数倍，或者不在有效报单数量范围内 |
| 修改后报单的价格不是合约的最小变动单位的整数倍               |
| 修改后报单的价格高于合约的涨停板                      |
| 修改后报单的价格低于合约的跌停板                      |
| 对指定合约、或者客户对指定的合约、或者用户没有交易权限           |
| 会员对指定合约、或者客户对指定的合约、                   |



|     |              |                               |
|-----|--------------|-------------------------------|
|     |              | 或者用户只有平仓的权限                   |
| 54  | 会话找不到        | 用户未登录                         |
| 57  | 不能为其他会员操作    | 用户给非其所属的会员进行操作                |
| 58  | 用户不匹配        | 报单操作中的用户和登录时的用户不匹配            |
| 71  | 不能操作衍生报单     | 用户对衍生报单进行操作                   |
| 72  | 不允许自然人开仓     | 交割月，自然人类型的客户对开仓类型的报单做激活或修改的操作 |
| 76  | 报单已经被挂起      | 在报单挂起时，报单已经被挂起了               |
| 77  | 报单已经被激活      | 在报单激活时，报单已经被激活了               |
| 79  | 不被支持的报单类型    | 交易所不支持此种报单类型，例如：报单修改          |
| 83  | 止损单仅用于连续交易   | 在非连续交易阶段操作止损单                 |
| 95  | 止损报单需说明止损价   | 止损单没有指定止损价                    |
| 96  | 保值额度不足       | 报单激活或修改后，客户的套保额度不足            |
| 98  | 强平单需由管理员使用   | 非管理员用户操作强平单                   |
| 99  | 不能为其他用户操作    | 非授权用户操作同会员的其他用户报入的报单          |
| 131 | 超出客户合约当日开仓限额 | 客户在某个合约上的开仓量超出开仓限额            |
| 133 | 超出客户产品每秒撤单限额 | 客户在某个产品上一秒以内的撤单笔数超出限额         |

### 2.1.40.OnErrRtnQuoteInsert 方法

报价录入错误回报。当会员系统发出报价录入指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnQuoteInsert(  
    CShfeFtdcInputQuoteField* pInputQuote,  
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pInputQuote:** 指向输入报价结构的地址，包含了报价录入操作的输入数据，和交易系统返回的报价编号。输入报价结构：

```
struct CShfeFtdcInputQuoteField {  
    ///报价编号，该字段由交易系统返回。  
    TShfeFtdcQuoteSysIDType QuoteSysID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;
```

```
///数量
TShfeFtdcVolumeType Volume;
///合约代码
TShfeFtdcInstrumentIDType InstrumentID;
///本地报价编号
TShfeFtdcOrderLocalIDType    QuoteLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///买方组合开平标志
TShfeFtdcCombOffsetFlagType BidCombOffsetFlag;
///买方组合套保标志
TShfeFtdcCombHedgeFlagType BidCombHedgeFlag;
///买方价格
TShfeFtdcPriceType BidPrice;
///卖方组合开平标志
TShfeFtdcCombOffsetFlagType AskCombOffsetFlag;
///卖方组合套保标志
TShfeFtdcCombHedgeFlagType AskCombHedgeFlag;
///卖方价格
TShfeFtdcPriceType AskPrice;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
///询价编号
TShfeFtdcOrderSysIDType QuoteDemandID;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示   | 可能的原因             |
|------|--------|-------------------|
| 2    | 合约找不到  | 找不到报价中的合约         |
| 3    | 会员找不到  | 找不到报价中的会员         |
| 4    | 客户找不到  | 找不到报价中的客户         |
| 7    | 报价字段错误 | 报价中有字段值不合法（枚举值越界） |
| 13   | 重复的报价  | 报价中的本地报价号重复       |

|      |              |                                |
|------|--------------|--------------------------------|
| 15   | 客户没有在该会员开户   | 报价中的客户并没有在指定的会员中开户             |
| 22   | 交易所数据没有同步    | 交易系统尚未初始化完毕，可以稍后再重试            |
| 23   | 结算组数据没有同步    | 交易系统尚未初始化完毕，可以稍后再重试            |
| 26   | 当前状态禁止此项操作   | 合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡    |
| 31   | 平仓时客户持仓不足    | 客户的持仓不足                        |
| 32   | 超出客户限仓       | 该报价导致客户的一般持仓超过了限仓              |
| 33   | 平仓时会员持仓不足    | 会员的持仓不足                        |
| 34   | 超出会员限仓       | 该报价导致会员的持仓超过了限仓                |
| 35   | 找不到帐号        | 找不到报价使用的资金帐号                   |
| 36   | 资金不足         | 资金帐号中没有足够的资金                   |
| 37   | 不合法的数量       | 报价数量不是最小报单数量要求的正整数倍，或者超过最大报单数量 |
| 48   | 价格非最小单位的倍数   | 报价的价格不是合约的最小变动单位的整数倍           |
| 49   | 价格超出涨停板      | 报价的价格高于合约的涨停板                  |
| 50   | 价格跌破跌停板      | 报价的价格低于合约的跌停板                  |
| 51   | 没有交易权限       | 对指定合约、或者客户对指定的合约、或者用户没有交易权限    |
| 52   | 只能平仓         | 会员对指定合约、或者客户对指定的合约、或者用户只有平仓的权限 |
| 53   | 没有此项交易角色     | 会员在指定合约上不具有该客户对应的交易角色          |
| 54   | 会话找不到        | 用户未登录                          |
| 57   | 不能为其他会员操作    | 用户给非其所属的会员进行操作                 |
| 58   | 用户不匹配        | 报价中的用户和登录时的用户不匹配               |
| 72   | 不允许自然人开仓     | 交割月自然人类型的客户发起开仓请求              |
| 79   | 不被支持的报单类型    | 交易所不支持此种报单类型                   |
| 96   | 保值额度不足       | 报入报价时，客户的套保额度不足                |
| 98   | 强平单需由管理员使用   | 非管理员用户报了强平单                    |
| 101  | 结算会员不能做交易    | 报价的会员类型为结算会员                   |
| 102  | 无法找到对应结算会员   | 找不到报价会员对应的结算会员                 |
| 103  | 当日套保仓位不能平仓   | 套保仓位不应使用平今仓报价进行平仓              |
| 131  | 超出客户合约当日开仓限额 | 客户在某个合约上的开仓量超出开仓限额             |
| 132  | 超出客户产品每秒报单限额 | 客户在某个产品上一秒以内的报单笔数超出限额          |
| 1005 | 无记录          | 报价所对应的合约记录缺失                   |

## 2.1.41.OnErrRtnQuoteAction 方法

报价操作错误回报。当会员系统发出报价操作指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnQuoteAction(
```

**CShfeFtdcQuoteActionField\* pQuoteAction,**  
**CShfeFtdcRspInfoField\* pRspInfo);**

参数:

**pQuoteAction:** 指向报价操作结构的地址, 包含了报价操作请求的输入数据和交易系统返回的报价编号。报价操作结构:

```
struct CShfeFtdcQuoteActionField {  
    ///报价编号, 该字段由交易系统返回。  
    TShfeFtdcQuoteSysIDType QuoteSysID;  
    ///本地报价编号  
    TShfeFtdcOrderLocalIDType QuoteLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///操作本地编号  
    TShfeFtdcOrderLocalIDType ActionLocalID;  
    ///业务单元  
    TShfeFtdcBusinessUnitType BusinessUnit;  
    ///本地业务标识  
    TShfeFtdcBusinessLocalIDType BusinessLocalID;  
    ///IP 地址  
    TShfeFtdcIPAddressType IPAddress;  
    ///Mac 地址  
    TShfeFtdcMacAddressType MacAddress;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示  | 可能的原因       |
|------|-------|-------------|
| 2    | 合约找不到 | 找不到报价操作中的合约 |

|    |            |                                                                           |
|----|------------|---------------------------------------------------------------------------|
| 3  | 会员找不到      | 报价操作中的会员找不到                                                               |
| 4  | 客户找不到      | 报价操作中的客户找不到                                                               |
| 8  | 报单操作字段错误   | 报价操作中报价衍生的报单中有字段值不合法<br>(例如: 价格不是浮点数或不在有效范围内)                             |
| 9  | 报价操作字段错误   | 报价操作中有字段值不合法(枚举值越界或<br>操作标志为修改、激活或挂起)                                     |
| 15 | 客户没有在该会员开户 | 客户并没有在指定的会员中开户                                                            |
| 22 | 交易所数据没有同步  | 交易系统尚未初始化完毕, 可以稍后再重试                                                      |
| 23 | 结算组数据没有同步  | 交易系统尚未初始化完毕, 可以稍后再重试                                                      |
| 25 | 报价找不到      | 找不到要操作的报价                                                                 |
| 26 | 当前状态禁止此项操作 | 对于激活操作, 合约交易状态不是连续交易、<br>集合竞价报单或者集合竞价平衡<br>对于其他操作, 交易状态不是连续交易或者<br>集合竞价报单 |
| 28 | 报单已经全部成交   | 报价派生的报单已经全部成交了                                                            |
| 29 | 报单已经撤销     | 报价派生的报单已经撤销了                                                              |
| 35 | 找不到帐号      | 找不到应当使用的资金帐号                                                              |
| 36 | 资金不足       | 资金帐号中没有足够的资金                                                              |
| 51 | 没有交易权限     | 对指定合约、或者客户对指定的合约、<br>或者用户没有交易权限                                           |
| 54 | 会话找不到      | 用户未登录                                                                     |
| 57 | 不能为其他会员操作  | 用户给非其所属的会员进行操作                                                            |
| 58 | 用户不匹配      | 报价操作中的用户和登录时的用户不匹配                                                        |
| 70 | 报价已经被取消    | 报价已经被取消                                                                   |
| 99 | 不能为其他用户操作  | 非授权用户操作同会员的其他用户报入的报价                                                      |

## 2.1.42. OnErrRtnExecOrderInsert 方法

执行宣告录入错误回报。当会员系统发出执行宣告录入指令, 出现报错时, 由交易系统主动通知会员系统, 该方法会被调用。

函数原形:

```
void OnErrRtnExecOrderInsert(
    CShfeFtdcInputExecOrderField* pInputExecOrder,
    CShfeFtdcRspInfoField* pRspInfo);
```

参数:

**pInputExecOrder:** 指向输入执行宣告结构的地址。输入执行宣告结构:

```
struct CShfeFtdcInputExecOrderField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
```

```

    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地执行宣告编号
    TShfeFtdcOrderLocalIDType ExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///开平标志
    TShfeFtdcOffsetFlagType OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType HedgeFlag;
    ///申请的持仓方向，即作为买方（多头）还是卖方（空头）来提出申请
    TShfeFtdcPosiDirectionType PosiDirection;
    ///期权行权后是否保留期货持仓的标记，未使用
    TShfeFtdcExecOrderPositionFlagType ReservePositionFlag;
    ///期权行权后生成的期货持仓是否自对冲
    TShfeFtdcExecOrderCloseFlagType CloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
};

```

**pRspInfo**: 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误:

| 错误代码 | 错误提示       | 可能的原因                |
|------|------------|----------------------|
| 2    | 合约找不到      | 找不到执行宣告中的合约          |
| 3    | 会员找不到      | 执行宣告中的会员找不到          |
| 4    | 客户找不到      | 执行宣告中的客户找不到          |
| 15   | 客户没有在该会员开户 | 执行宣告中的客户并没有在指定的会员中开户 |
| 22   | 交易所数据没有同步  | 交易系统尚未初始化完毕，可以稍后再重试  |

|      |                |                                 |
|------|----------------|---------------------------------|
| 23   | 结算组数据没有同步      | 交易系统尚未初始化完毕，可以稍后再重试             |
| 26   | 当前状态禁止此项操作     | 合约交易状态不是连续交易、交易业务处理             |
| 31   | 平仓时客户持仓不足      | 执行宣告录入时，客户的持仓额度不足               |
| 33   | 平仓时会员持仓不足      | 执行宣告录入时，会员的持仓额度不足               |
| 35   | 找不到帐号          | 找不到应当使用的资金帐号                    |
| 36   | 资金不足           | 资金帐号中没有足够的资金                    |
| 37   | 不合法的数量         | 执行宣告数量不合法                       |
| 51   | 没有交易权限         | 对指定合约、或者客户对指定的合约、<br>或者用户没有交易权限 |
| 54   | 会话找不到          | 用户未登录                           |
| 57   | 不能为其他会员操作      | 用户给非其所属的会员进行操作                  |
| 58   | 用户不匹配          | 执行宣告中的用户和登录时的用户不匹配              |
| 79   | 不被支持的报单类型      | 交易所不支持此种报单类型                    |
| 89   | 执行宣告字段错误       | 执行宣告中有字段值不合法（枚举值越界）             |
| 91   | 重复的执行宣告        | 执行宣告中的本地执行宣告号重复                 |
| 94   | 执行宣告只能用于期权     | 执行宣告中的合约是非期权合约                  |
| 101  | 结算会员不能做交易      | 执行宣告的会员类型为结算会员                  |
| 102  | 无法找到对应结算会员     | 找不到执行宣告会员对应的结算会员                |
| 127  | 不在宣告期内         | 不在合约交割期（可行权周期）内                 |
| 129  | 执行或放弃执行宣告不能为开仓 | 执行宣告的开平标志只能限定为平仓                |
| 146  | 只有持有多头仓位才能行权   | 只有期权买方才能行权                      |
| 1005 | 无记录            | 执行宣告所对应的合约记录缺失                  |

## 2.1.43.OnErrRtnExecOrderAction 方法

执行宣告操作错误回报。当会员系统发出执行宣告操作指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnExecOrderAction(
    CShfeFtdcExecOrderActionField* pExecOrderAction,
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pExecOrderAction：**指向执行宣告操作结构的地址。执行宣告操作结构：

```
struct CShfeFtdcExecOrderActionField {
    ///执行宣告编号
    TShfeFtdcExecOrderSysIDType   ExecOrderSysID;
    ///本地执行宣告编号
    TShfeFtdcOrderLocalIDType     ExecOrderLocalID;
    ///报单操作标志
```

```
TShfeFtdcActionFlagType ActionFlag;
///会员代码
TShfeFtdcParticipantIDType ParticipantID;
///客户代码
TShfeFtdcClientIDType ClientID;
///交易用户代码
TShfeFtdcUserIDType UserID;
///操作本地编号
TShfeFtdcOrderLocalIDType ActionLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示       | 可能的原因                       |
|------|------------|-----------------------------|
| 2    | 合约找不到      | 找不到执行宣告操作中的合约               |
| 3    | 会员找不到      | 执行宣告操作中的会员找不到               |
| 4    | 客户找不到      | 执行宣告操作中的客户找不到               |
| 15   | 客户没有在该会员开户 | 执行宣告操作中的客户并没有在指定的会员中开户      |
| 22   | 交易所数据没有同步  | 交易系统尚未初始化完毕，可以稍后再重试         |
| 23   | 结算组数据没有同步  | 交易系统尚未初始化完毕，可以稍后再重试         |
| 26   | 当前状态禁止此项操作 | 合约交易状态不是连续交易、交易业务处理         |
| 35   | 找不到帐号      | 找不到应当使用的资金帐号                |
| 36   | 资金不足       | 资金帐号中没有足够的资金                |
| 51   | 没有交易权限     | 对指定合约、或者客户对指定的合约、或者用户没有交易权限 |
| 54   | 会话找不到      | 用户未登录                       |
| 57   | 不能为其他会员操作  | 用户给非其所属的会员进行操作              |
| 58   | 用户不匹配      | 执行宣告操作中的用户和登录时的用户不匹配        |
| 89   | 执行宣告字段错误   | 执行宣告中有字段值不合法                |
| 90   | 执行宣告操作字段错误 | 执行宣告操作中有字段值不合法（枚举值越界或       |



|      |          |                  |
|------|----------|------------------|
|      |          | 操作标志为修改、激活或挂起)   |
| 92   | 执行宣告已经取消 | 要操作的执行宣告已经取消     |
| 93   | 执行宣告找不到  | 要操作的执行宣告没有找到     |
| 127  | 不在宣告期内   | 不在合约交割期（可行权周期）内  |
| 1005 | 无记录      | 执行宣告操作所对应的合约记录缺失 |

### 2.1.44.OnRspQryExecOrder 方法

执行宣告查询应答。由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRspQryExecOrder(  
    CShfeFtdcExecOrderField* pExecOrder,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pExecOrder：** 指向执行宣告结构的地址，执行宣告结构：

```
struct CShfeFtdcExecOrderField {  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;  
    ///结算编号  
    TShfeFtdcSettlementIDType SettlementID;  
    ///合约编号  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType    ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///本地执行宣告编号  
    TShfeFtdcOrderLocalIDType    ExecOrderLocalID;  
    ///数量  
    TShfeFtdcVolumeType Volume;  
    ///开平标志  
    TShfeFtdcOffsetFlagType    OffsetFlag;  
    ///套保标志
```

```

TShfeFtdcHedgeFlagType HedgeFlag;
///申请的持仓方向，即作为买方（多头）还是卖方（空头）来提出申请
TShfeFtdcPosiDirectionType PosiDirection;
///期权行权后是否保留期货持仓的标记，未使用
TShfeFtdcExecOrderPositionFlagType ReservePositionFlag;
///期权行权后生成的期货持仓是否自对冲
TShfeFtdcExecOrderCloseFlagType CloseFlag;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///执行宣告编号
TShfeFtdcExecOrderSysIDType ExecOrderSysID;
///报单日期
TShfeFtdcDateType InsertDate;
///插入时间
TShfeFtdcTimeType InsertTime;
///撤销时间
TShfeFtdcTimeType CancelTime;
///执行结果
TShfeFtdcExecResultType ExecResult;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TShfeFtdcDateType ActionDay;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户执行宣告查询请求的 ID，该 ID 由用户在执行宣告查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.45.OnRspQryExchangeRate 方法

汇率查询应答。当会员系统发出汇率查询指令后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryExchangeRate(  
    CShfeFtdcRspExchangeRateField* pRspExchangeRate,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pRspExchangeRate:** 指向汇率应答信息结构的地址，汇率应答信息结构：

```
struct CShfeFtdcRspExchangeRateField {  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///币种代码  
    TShfeFtdcCurrencyIDType    CurrencyID;  
    ///外汇交易单位  
    TShfeFtdcRateUnitType    RateUnit;  
    ///外汇中间价  
    TShfeFtdcExRatePriceType    RatePrice;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误：

| 错误代码 | 错误提示  | 可能的原因 |
|------|-------|-------|
| 54   | 会话找不到 | 用户未登录 |

|    |        |            |
|----|--------|------------|
| 80 | 用户无此权限 | 只有交易用户能做查询 |
|----|--------|------------|

**nRequestID:** 返回用户汇率查询请求的 ID，该 ID 由用户在汇率查询时指定。

**bIsLast:** 指示该次返回是否针对 nRequestID 的最后一次返回。

### 2.1.46.OnRspAbandonExecOrderInsert 方法

放弃执行宣告录入应答。当会员系统发出放弃执行宣告录入请求后，交易系统返回响应时，该方法会被调用。

函数原形：

```
void OnRspAbandonExecOrderInsert(  
    CShfeFtdcInputAbandonExecOrderField* pInputAbandonExecOrder,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

**pInputAbandonExecOrder:** 指向输入放弃执行宣告结构的地址，输入放弃执行宣告结构：

```
struct CShfeFtdcInputAbandonExecOrderField {  
    ///合约编号  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///本地放弃执行宣告编号  
    TShfeFtdcOrderLocalIDType AbandonExecOrderLocalID;  
    ///数量  
    TShfeFtdcVolumeType Volume;  
    ///开平标志  
    TShfeFtdcOffsetFlagType OffsetFlag;  
    ///套保标志  
    TShfeFtdcHedgeFlagType HedgeFlag;  
    ///申请放弃的持仓方向，实际只有多头才能提出申请放弃  
    TShfeFtdcPosiDirectionType PosiDirection;
```

```

///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|     |                |
|-----|----------------|
| 2   | 合约找不到          |
| 3   | 会员找不到          |
| 4   | 客户找不到          |
| 15  | 客户没有在该会员开户     |
| 22  | 交易所数据没有同步      |
| 23  | 结算组数据没有同步      |
| 26  | 当前状态禁止此项操作     |
| 31  | 平仓时客户持仓不足      |
| 33  | 平仓时会员持仓不足      |
| 35  | 找不到帐号          |
| 36  | 资金不足           |
| 37  | 不合法的数量         |
| 51  | 没有交易权限         |
| 54  | 会话找不到          |
| 57  | 不能为其他会员操作      |
| 58  | 用户不匹配          |
| 79  | 不被支持的报单类型      |
| 101 | 结算会员不能做交易      |
| 102 | 无法找到对应结算会员     |
| 121 | 放弃执行宣告字段错误     |
| 123 | 重复的放弃执行宣告      |
| 126 | 放弃执行宣告只能用于期权   |
| 128 | 只有持有多头仓位才能放弃行权 |
| 129 | 执行或放弃执行宣告不能为开仓 |
| 149 | 放弃执行宣告申请只能在    |

**可能的原因**

|                             |
|-----------------------------|
| 找不到放弃执行宣告中的合约               |
| 找不到放弃执行宣告中的会员               |
| 找不到放弃执行宣告中的客户               |
| 放弃执行宣告中的客户并没有在指定的会员中开户      |
| 交易系统尚未初始化完毕, 可以稍后再重试        |
| 交易系统尚未初始化完毕, 可以稍后再重试        |
| 合约交易状态不是连续交易、交易业务处理         |
| 放弃执行宣告时, 客户的持仓不足            |
| 放弃执行宣告时, 会员的持仓不足            |
| 找不到应当使用的资金帐号                |
| 资金帐号中没有足够的资金                |
| 放弃执行宣告数量不合法                 |
| 对指定合约、或者客户对指定的合约、或者用户没有交易权限 |
| 用户未登录                       |
| 用户给非其所属的会员进行操作              |
| 放弃执行宣告中的用户和登录时的用户不匹配        |
| 交易所不支持此种报单类型                |
| 放弃执行宣告的会员类型为结算会员            |
| 找不到报单会员对应的结算会员              |
| 放弃执行宣告中有字段值不合法              |
| 放弃执行宣告中的本地放弃执行宣告号重复         |
| 放弃执行宣告中的合约是非期权合约            |
| 只有期权买方才能放弃行权                |
| 放弃执行宣告的开平标志只能限定为平仓          |
| 交易日不是期权到期日                  |

|          |                  |
|----------|------------------|
| 期权到期日提交  |                  |
| 1005 无记录 | 放弃执行宣告所对应的合约记录缺失 |

**nRequestID:** 返回用户放弃执行宣告输入请求的 ID，该 ID 由用户在录入放弃执行宣告时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.47.OnRspAbandonExecOrderAction 方法

放弃执行宣告操作应答。包括放弃执行宣告的撤销、放弃执行宣告的挂起、放弃执行宣告的激活、放弃执行宣告的修改。当会员系统发出放弃执行宣告操作请求后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspAbandonExecOrderAction(  
    CShfeFtdcAbandonExecOrderActionField* pAbandonExecOrderAction,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

**pAbandonExecOrderAction:** 指向放弃执行宣告操作结构的地址，放弃执行宣告操作结构:

```
struct CShfeFtdcAbandonExecOrderActionField {  
    ///放弃执行宣告编号  
    TShfeFtdcExecOrderSysIDType AbandonExecOrderSysID;  
    ///本地放弃执行宣告编号  
    TShfeFtdcOrderLocalIDType AbandonExecOrderLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///操作本地编号  
    TShfeFtdcOrderLocalIDType ActionLocalID;  
    ///业务单元
```

```
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示                   | 可能的原因                                     |
|------|------------------------|-------------------------------------------|
| 2    | 合约找不到                  | 找不到放弃执行宣告操作中的合约                           |
| 3    | 会员找不到                  | 找不到放弃执行宣告操作中的会员                           |
| 4    | 客户找不到                  | 找不到放弃执行宣告操作中的客户                           |
| 15   | 客户没有在该会员开户             | 放弃执行宣告操作中的客户并没有在指定的会员中开户                  |
| 22   | 交易所数据没有同步              | 交易系统尚未初始化完毕, 可以稍后再重试                      |
| 23   | 结算组数据没有同步              | 交易系统尚未初始化完毕, 可以稍后再重试                      |
| 26   | 当前状态禁止此项操作             | 合约交易状态不是连续交易、交易业务处理                       |
| 35   | 找不到帐号                  | 找不到应当使用的资金帐号                              |
| 36   | 资金不足                   | 资金帐号中没有足够的资金                              |
| 51   | 没有交易权限                 | 对指定合约、或者客户对指定的合约、或者用户没有交易权限               |
| 54   | 会话找不到                  | 用户未登录                                     |
| 57   | 不能为其他会员操作              | 用户给非其所属的会员进行操作                            |
| 58   | 用户不匹配                  | 放弃执行宣告操作中的用户和登录时的用户不匹配                    |
| 121  | 放弃执行宣告字段错误             | 放弃执行宣告中有字段值不合法                            |
| 122  | 放弃执行宣告操作字段错误           | 放弃执行宣告操作中有字段值不合法<br>(枚举值越界或操作标志为修改、激活或挂起) |
| 124  | 放弃执行宣告已经取消             | 要操作的放弃执行宣告已经删除                            |
| 125  | 放弃执行宣告找不到              | 要操作的放弃执行宣告找不到                             |
| 149  | 放弃执行宣告申请只能在<br>期权到期日提交 | 交易日不是期权到期日                                |
| 1005 | 无记录                    | 放弃执行宣告操作所对应的合约记录缺失                        |

**nRequestID:** 返回用户放弃执行宣告操作请求的 ID, 该 ID 由用户在放弃执行宣告操作时指定。

**blsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.48.OnRspQryAbandonExecOrder 方法

放弃执行宣告查询应答。当会员系统发出放弃执行宣告查询请求后，交易系统返回响应时，该方法会被调用。

**函数原形:**

```
void OnRspQryAbandonExecOrder(
    CShfeFtdcAbandonExecOrderField* pAbandonExecOrder,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool blsLast);
```

**参数:**

**pAbandonExecOrder:** 指向放弃执行宣告结构的地址，放弃执行宣告结构:

```
struct CShfeFtdcAbandonExecOrderField {
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType    ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地放弃执行宣告编号
    TShfeFtdcOrderLocalIDType    AbandonExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///开平标志
    TShfeFtdcOffsetFlagType    OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType    HedgeFlag;
    ///申请放弃的持仓方向，实际只有多头才能提出申请放弃
```



```
TShfeFtdcPosiDirectionType    PosiDirection;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType  BusinessLocalID;
///放弃执行宣告编号
TShfeFtdcExecOrderSysIDType   AbandonExecOrderSysID;
///报单日期
TShfeFtdcDateType             InsertDate;
///插入时间
TShfeFtdcTimeType              InsertTime;
///撤销时间
TShfeFtdcTimeType              CancelTime;
///放弃执行结果
TShfeFtdcExecResultType        AbandonExecResult;
///结算会员编号
TShfeFtdcParticipantIDType     ClearingPartID;
///业务发生日期
TShfeFtdcDateType              ActionDay;
///IP 地址
TShfeFtdcIPAddressType         IPAddress;
///Mac 地址
TShfeFtdcMacAddressType         MacAddress;

};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType      ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType     ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户放弃执行宣告查询请求的 ID，该 ID 由用户在放弃执行宣告查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.49. OnRtnAbandonExecOrder 方法

放弃执行宣告回报。当会员系统进行放弃执行宣告录入、放弃执行宣告操作导致放弃执行宣告状态发生变化时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRtnAbandonExecOrder(
    CShfeFtdcAbandonExecOrderField* pAbandonExecOrder);
```

参数：

**pAbandonExecOrder：** 指向放弃执行宣告结构的地址，放弃执行宣告结构：

```
struct CShfeFtdcAbandonExecOrderField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType   SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType   ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地放弃执行宣告编号
    TShfeFtdcOrderLocalIDType   AbandonExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///开平标志
    TShfeFtdcOffsetFlagType   OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType   HedgeFlag;
    ///申请放弃的持仓方向，实际只有多头才能提出申请放弃
    TShfeFtdcPosiDirectionType   PosiDirection;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
```

```

TShfeFtdcBusinessLocalIDType BusinessLocalID;
///放弃执行宣告编号
TShfeFtdcExecOrderSysIDType AbandonExecOrderSysID;
///报单日期
TShfeFtdcDateType InsertDate;
///插入时间
TShfeFtdcTimeType InsertTime;
///撤销时间
TShfeFtdcTimeType CancelTime;
///放弃执行结果
TShfeFtdcExecResultType AbandonExecResult;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///业务发生日期
TShfeFtdcDateType ActionDay;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;

};

```

## 2.1.50. OnErrRtnAbandonExecOrderInsert 方法

放弃执行宣告录入错误回报。当会员系统发出放弃执行宣告录入指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```

void OnErrRtnAbandonExecOrderInsert(
    CShfeFtdcInputAbandonExecOrderField* pInputAbandonExecOrder,
    CShfeFtdcRspInfoField* pRspInfo);

```

参数：

**pInputAbandonExecOrder:** 指向放弃执行宣告输入结构的地址，放弃执行宣告输入结构：

```

struct CShfeFtdcInputAbandonExecOrderField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码

```

```
TShfeFtdcClientIDType    ClientID;
///交易用户代码
TShfeFtdcUserIDType      UserID;
///本地放弃执行宣告编号
TShfeFtdcOrderLocalIDType AbandonExecOrderLocalID;
///数量
TShfeFtdcVolumeType      Volume;
///开平标志
TShfeFtdcOffsetFlagType  OffsetFlag;
///套保标志
TShfeFtdcHedgeFlagType   HedgeFlag;
///申请放弃的持仓方向，实际只有多头才能提出申请放弃
TShfeFtdcPosiDirectionType PosiDirection;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType   IPAddress;
///Mac 地址
TShfeFtdcMacAddressType   MacAddress;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType   ErrorMsg;
};
```

可能出现的错误:

**错误代码 错误提示**

2 合约找不到

3 会员找不到

4 客户找不到

15 客户没有在该会员开户

22 交易所数据没有同步

23 结算组数据没有同步

26 当前状态禁止此项操作

31 平仓时客户持仓不足

33 平仓时会员持仓不足

35 找不到帐号

36 资金不足

37 不合法的数量

51 没有交易权限

**可能的原因**

找不到放弃执行宣告中的合约

找不到放弃执行宣告中的会员

找不到放弃执行宣告中的客户

放弃执行宣告中的客户并没有在指定的会员中开户

交易系统尚未初始化完毕，可以稍后再重试

交易系统尚未初始化完毕，可以稍后再重试

合约交易状态不是连续交易、交易业务处理

放弃执行宣告时，客户的持仓不足

放弃执行宣告时，会员的持仓不足

找不到应当使用的资金帐号

资金帐号中没有足够的资金

放弃执行宣告数量不合法

对指定合约、或者客户对指定的合约、

|      |                    |                      |
|------|--------------------|----------------------|
|      |                    | 或者用户没有交易权限           |
| 54   | 会话找不到              | 用户未登录                |
| 57   | 不能为其他会员操作          | 用户给非其所属的会员进行操作       |
| 58   | 用户不匹配              | 放弃执行宣告中的用户和登录时的用户不匹配 |
| 79   | 不被支持的报单类型          | 交易所不支持此种报单类型         |
| 101  | 结算会员不能做交易          | 放弃执行宣告的会员类型为结算会员     |
| 102  | 无法找到对应结算会员         | 找不到放弃执行宣告会员对应的结算会员   |
| 121  | 放弃执行宣告字段错误         | 放弃执行宣告中有字段值不合法       |
| 123  | 重复的放弃执行宣告          | 放弃执行宣告中的本地放弃执行宣告号重复  |
| 126  | 放弃执行宣告只能用于期权       | 放弃执行宣告中的合约是非期权合约     |
| 128  | 只有持有多头仓位才能放弃行权     | 只有期权买方才能放弃行权         |
| 129  | 执行或放弃执行宣告不能为开仓     | 执行宣告的开平标志只能限定为平仓     |
| 149  | 放弃执行宣告申请只能在期权到期日提交 | 交易日不是期权到期日           |
| 1005 | 无记录                | 放弃执行宣告所对应的合约记录缺失     |

2.1.51.OnErrRtnAbandonExecOrderAction 方法

放弃执行宣告操作错误回报。当会员系统发出放弃执行宣告操作指令，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnAbandonExecOrderAction(  
    CShfeFtdcAbandonExecOrderActionField* pAbandonExecOrderAction,  
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pAbandonExecOrderAction：**指向放弃执行宣告操作结构的地址，放弃执行宣告操作结构：

```
struct CShfeFtdcAbandonExecOrderActionField {  
    ///放弃执行宣告编号  
    TShfeFtdcExecOrderSysIDType  AbandonExecOrderSysID;  
    ///本地放弃执行宣告编号  
    TShfeFtdcOrderLocalIDType     AbandonExecOrderLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType  ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDTypeParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType      ClientID;
```

```

///交易用户代码
TShfeFtdcUserIDType UserID;
///操作本地编号
TShfeFtdcOrderLocalIDType    ActionLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType    IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|     |                        |
|-----|------------------------|
| 2   | 合约找不到                  |
| 3   | 会员找不到                  |
| 4   | 客户找不到                  |
| 15  | 客户没有在该会员开户             |
| 22  | 交易所数据没有同步              |
| 23  | 结算组数据没有同步              |
| 26  | 当前状态禁止此项操作             |
| 35  | 找不到帐号                  |
| 36  | 资金不足                   |
| 51  | 没有交易权限                 |
| 54  | 会话找不到                  |
| 57  | 不能为其他会员操作              |
| 58  | 用户不匹配                  |
| 121 | 放弃执行宣告字段错误             |
| 122 | 放弃执行宣告操作字段错误           |
| 124 | 放弃执行宣告已经取消             |
| 125 | 放弃执行宣告找不到              |
| 149 | 放弃执行宣告申请只能在<br>期权到期日提交 |

**可能的原因**

|                                           |
|-------------------------------------------|
| 找不到放弃执行宣告操作中的合约                           |
| 找不到放弃执行宣告操作中的会员                           |
| 找不到放弃执行宣告操作中的客户                           |
| 放弃执行宣告中的客户并没有在指定的会员中开户                    |
| 交易系统尚未初始化完毕, 可以稍后再重试                      |
| 交易系统尚未初始化完毕, 可以稍后再重试                      |
| 合约交易状态不是连续交易、交易业务处理                       |
| 找不到应当使用的资金帐号                              |
| 资金帐号中没有足够的资金                              |
| 对指定合约、或者客户对指定的合约、<br>或者用户没有交易权限           |
| 用户未登录                                     |
| 用户给非其所属的会员进行操作                            |
| 放弃执行宣告操作中的用户和登录时的<br>用户不匹配                |
| 放弃执行宣告中有字段值不合法                            |
| 放弃执行宣告操作中有字段值不合法<br>(枚举值越界或操作标志为修改、激活或挂起) |
| 要操作的放弃执行宣告已经删除                            |
| 要操作的放弃执行宣告找不到                             |
| 交易日不是期权到期日                                |

1005 无记录

放弃执行宣告操作所对应的合约记录缺失

### 2.1.52.OnRspQuoteDemand 方法

询价录入应答，当会员系统发出询价录入请求后，交易系统返回响应时，该方法会被调用。

#### 函数原形：

```
void OnRspQuoteDemand(  
    CShfeFtdcQuoteDemandInfoField* pQuoteDemandInfo,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

#### 参数：

**pQuoteDemandInfo:** 指向询价录入应答的地址，询价信息结构：

```
struct CShfeFtdcQuoteDemandInfoField {  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType    ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///合约代码  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///本地询价编号  
    TShfeFtdcOrderLocalIDType    QuoteDemandLocalID;  
    ///请求时间  
    TShfeFtdcTimeType    DemandTime;  
    ///业务发生日期  
    TShfeFtdcDateType    ActionDay;  
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;
```

};

可能出现的错误:

| 错误代码 | 错误提示                      | 可能的原因                                                                  |
|------|---------------------------|------------------------------------------------------------------------|
| 2    | 合约找不到                     | 找不到询价的合约                                                               |
| 3    | 会员找不到                     | 找不到询价的会员                                                               |
| 4    | 客户找不到                     | 找不到询价的客户                                                               |
| 15   | 客户没有在该会员开户                | 询价的客户并没有在指定的会员中开户                                                      |
| 22   | 交易所数据没有同步                 | 交易系统尚未初始化完毕, 可以稍后再重试                                                   |
| 23   | 结算组数据没有同步                 | 交易系统尚未初始化完毕, 可以稍后再重试                                                   |
| 26   | 当前状态禁止此项操作                | 合约不可交易或合约不处于连续交易状态                                                     |
| 51   | 没有交易权限                    | 对指定合约、或者客户对指定的合约、<br>或者用户没有交易权限                                        |
| 53   | 没有此项交易角色                  | 会员在指定合约上不具有该客户对应的交易角色                                                  |
| 54   | 会话找不到                     | 用户未登录                                                                  |
| 57   | 不能为其他会员操作                 | 用户给非其所属的会员进行操作                                                         |
| 58   | 用户不匹配                     | 询价的用户和登录时的用户不匹配                                                        |
| 88   | 未找到要操作的用户                 | 找不到询价的用户                                                               |
| 101  | 结算会员不能做交易                 | 询价的会员类型为结算会员                                                           |
| 148  | 当前市场价格在合理的价差<br>范围内, 不用询价 | 当前存在买方向报单, 且价格触及涨停板;<br>当前存在卖方向报单, 且价格触及跌停板;<br>当前买卖方向均存在报单, 且价差在合理范围内 |

**nRequestID:** 返回用户询价录入请求的 ID, 该 ID 由用户在询价录入时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.53.OnRtnQuoteDemandNotify 方法

询价请求分发。当交易系统主动通知有对应权限的做市商用户时, 该方法会被调用。

函数原形:

```
void OnRtnQuoteDemandNotify(  
    CShfeFtdcQuoteDemandNotifyField* pQuoteDemandNotify);
```

参数:

**pQuoteDemandNotify:** 指向询价通知地址, 询价通知结构:

```
struct CShfeFtdcQuoteDemandNotifyField {  
    ///合约编号  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///询价日期
```



```

TShfeFtdcDateType DemandDay;
///询价时间
TShfeFtdcTimeType DemandTime;
///询价编号
TShfeFtdcOrderSysIDType QuoteDemandID;

};

```

## 2.1.54. OnRspOptionSelfCloseUpdate 方法

期权自对冲更新应答。当会员系统执行期权自对冲更新后，交易系统返回响应时，该方法会被调用。

函数原形：

```

void OnRspOptionSelfCloseUpdate(
    CShfeFtdcInputOptionSelfCloseField* pInputOptionSelfClose,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

**pInputOptionSelfClose：** 指向期权自对冲更新的地址，期权自对冲更新结构：

```

struct CShfeFtdcInputOptionSelfCloseField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地期权自对冲编号
    TShfeFtdcOrderLocalIDType OptionSelfCloseLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///期权行权的持仓是否自对冲
    TShfeFtdcOptSelfCloseFlagType SelfCloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识

```

```

TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;

};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|      |                              |
|------|------------------------------|
| 2    | 合约找不到                        |
| 3    | 会员找不到                        |
| 4    | 客户找不到                        |
| 15   | 客户没有在该会员开户                   |
| 22   | 交易所数据没有同步                    |
| 23   | 结算组数据没有同步                    |
| 26   | 当前状态禁止此项操作                   |
| 37   | 不合法的数量                       |
| 51   | 没有交易权限                       |
| 53   | 没有此项交易角色                     |
| 54   | 会话找不到                        |
| 57   | 不能为其他会员操作                    |
| 58   | 用户不匹配                        |
| 79   | 不被支持的报单类型                    |
| 101  | 结算会员不能做交易                    |
| 102  | 无法找到对应结算会员                   |
| 127  | 不在宣告期内                       |
| 137  | 期权自对冲字段错误                    |
| 139  | 重复的期权自对冲更新                   |
| 141  | 期权自对冲更新只能用于期权                |
| 144  | 此客户 SelfCloseFlag 不能为保留期权仓位  |
| 145  | 此客户 SelfCloseFlag 不能为自对冲期权仓位 |
| 1005 | 无记录                          |

**可能的原因**

|                                      |
|--------------------------------------|
| 找不到期权自对冲更新的合约                        |
| 期权自对冲更新的会员找不到                        |
| 期权自对冲更新的客户找不到                        |
| 期权自对冲更新的客户并没有在指定的会员中开户               |
| 交易系统尚未初始化完毕, 可以稍后再重试                 |
| 交易系统尚未初始化完毕, 可以稍后再重试                 |
| 合约交易状态既不是连续交易<br>也不是交易业务处理状态         |
| 期权自对冲手数不合法                           |
| 对指定合约、或者客户对指定的合约、<br>或者用户没有交易权限      |
| 会员在指定合约上不具有该客户对应的交易角色                |
| 用户未登录                                |
| 用户给非其所属的会员进行操作                       |
| 期权自对冲更新中的用户和登录用户不匹配                  |
| 交易所不支持此种报单类型                         |
| 申请期权自对冲更新的会员类型为结算会员                  |
| 无法找到期权自对冲更新会员对应的结算会员                 |
| 自对冲期权卖方履约后的期货仓位,<br>只能在交割期(可行权周期)内提交 |
| 期权自对冲更新中有字段值不合法(枚举值越界)               |
| 期权自对冲更新中的本地期权自对冲编号重复                 |
| 期权自对冲更新中的合约是非期权合约                    |
| 只有做市商才能提交保留期权仓位申请                    |
| 做市商只能提交保留期权仓位申请                      |
| 期权自对冲更新所对应的合约记录缺失                    |

**nRequestID:** 返回用户期权自对冲更新请求的 ID，该 ID 由用户在期权自对冲更新时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.55.OnErrRtnOptionSelfCloseUpdate 方法

期权自对冲更新错误回报。当会员系统执行期权自对冲更新，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnOptionSelfCloseUpdate(
    CShfeFtdcInputOptionSelfCloseField* pInputOptionSelfClose,
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pInputOptionSelfClose:** 指向期权自对冲更新结构的地址。期权自对冲更新结构：

```
struct CShfeFtdcInputOptionSelfCloseField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地期权自对冲编号
    TShfeFtdcOrderLocalIDType OptionSelfCloseLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///期权行权的持仓是否自对冲
    TShfeFtdcOptSelfCloseFlagType SelfCloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
```

```

TShfeFtdcMacAddressType MacAddress;

};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType   ErrorMsg;
};

```

可能出现的错误:

**错误代码 错误提示**

|      |                              |
|------|------------------------------|
| 2    | 合约找不到                        |
| 3    | 会员找不到                        |
| 4    | 客户找不到                        |
| 15   | 客户没有在该会员开户                   |
| 22   | 交易所数据没有同步                    |
| 23   | 结算组数据没有同步                    |
| 26   | 当前状态禁止此项操作                   |
| 37   | 不合法的数量                       |
| 51   | 没有交易权限                       |
| 53   | 没有此项交易角色                     |
| 57   | 不能为其他会员操作                    |
| 58   | 用户不匹配                        |
| 79   | 不被支持的报单类型                    |
| 101  | 结算会员不能做交易                    |
| 102  | 无法找到对应结算会员                   |
| 127  | 不在宣告期内                       |
| 137  | 期权自对冲字段错误                    |
| 139  | 重复的期权自对冲更新                   |
| 141  | 期权自对冲更新只能用于期权                |
| 144  | 此客户 SelfCloseFlag 不能为保留期权仓位  |
| 145  | 此客户 SelfCloseFlag 不能为自对冲期权仓位 |
| 1005 | 无记录                          |

**可能的原因**

|                                   |
|-----------------------------------|
| 找不到期权自对冲更新的合约                     |
| 期权自对冲更新的会员找不到                     |
| 期权自对冲更新的客户找不到                     |
| 期权自对冲更新的客户并没有在指定的会员中开户            |
| 交易系统尚未初始化完毕, 可以稍后再重试              |
| 交易系统尚未初始化完毕, 可以稍后再重试              |
| 合约交易状态既不是连续交易也不是交易业务处理状态          |
| 期权自对冲手数不合法                        |
| 对指定合约、或者客户对指定的合约、或者用户没有交易权限       |
| 会员在指定合约上不具有该客户对应的交易角色             |
| 用户给非其所属的会员进行操作                    |
| 期权自对冲更新中的用户和登录用户不匹配               |
| 交易所不支持此种报单类型                      |
| 申请期权自对冲更新的会员类型为结算会员               |
| 无法找到期权自对冲更新会员对应的结算会员              |
| 自对冲期权卖方履约后的期货仓位, 只能在交割期(可行权周期)内提交 |
| 期权自对冲更新中有字段值不合法(枚举值越界)            |
| 期权自对冲更新中的本地期权自对冲编号重复              |
| 期权自对冲更新中的合约是非期权合约                 |
| 只有做市商才能提交保留期权仓位申请                 |
| 做市商只能提交保留期权仓位申请                   |
| 期权自对冲更新所对应的合约记录缺失                 |

## 2.1.56. OnRtnOptionSelfCloseUpdate 方法

期权自对冲更新回报。当会员系统执行期权自对冲更新，导致期权自对冲表发生变化时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnRtnOptionSelfCloseUpdate(  
    CShfeFtdcOptionSelfCloseField* pOptionSelfClose);
```

参数：

**pOptionSelfClose：** 指向期权自对冲结构的地址。期权自对冲结构：

```
struct CShfeFtdcOptionSelfCloseField {  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;  
    ///结算编号  
    TShfeFtdcSettlementIDType SettlementID;  
    ///合约编号  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType    ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///本地期权自对冲编号  
    TShfeFtdcOrderLocalIDType    OptionSelfCloseLocalID;  
    ///数量  
    TShfeFtdcVolumeType Volume;  
    ///期权行权的持仓是否自对冲  
    TShfeFtdcOptSelfCloseFlagType SelfCloseFlag;  
    ///业务单元  
    TShfeFtdcBusinessUnitType BusinessUnit;  
    ///本地业务标识  
    TShfeFtdcBusinessLocalIDType BusinessLocalID;  
    ///IP 地址  
    TShfeFtdcIPAddressType    IPAddress;  
    ///Mac 地址  
    TShfeFtdcMacAddressType MacAddress;  
    ///期权自对冲编号
```

```

TShfeFtdcOptionSelfCloseSysIDType OptionSelfCloseSysID;
///期权自对冲结果
TShfeFtdcExecResultType SelfCloseResult;
///报单日期
TShfeFtdcDateType InsertDate;
///插入时间
TShfeFtdcTimeType InsertTime;
///撤销时间
TShfeFtdcTimeType CancelTime;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///业务发生日期
TShfeFtdcDateType ActionDay;

};

```

## 2.1.57. OnRspOptionSelfCloseAction 方法

期权自对冲操作应答。包括期权自对冲的撤销、期权自对冲的挂起、期权自对冲的激活、期权自对冲的修改。当会员系统执行期权自对冲操作后，交易系统返回响应时，该方法会被调用。

函数原形：

```

void OnRspOptionSelfCloseAction(
    CShfeFtdcOptionSelfCloseActionField* pOptionSelfCloseAction,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

**pOptionSelfCloseAction：**指向期权自对冲操作结构的地址。期权自对冲操作结构：

```

struct CShfeFtdcOptionSelfCloseActionField{
    ///期权自对冲编号
    TShfeFtdcOptionSelfCloseSysIDType OptionSelfCloseSysID;
    ///本地期权自对冲编号
    TShfeFtdcOrderLocalIDType OptionSelfCloseLocalID;
    ///期权自对冲操作标志
    TShfeFtdcActionFlagType ActionFlag;
};

```

```

    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///操作本地编号
    TShfeFtdcOrderLocalIDType ActionLocalID;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
};

```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误:

| 错误代码 | 错误提示        | 可能的原因                                    |
|------|-------------|------------------------------------------|
| 2    | 合约找不到       | 找不到期权自对冲操作中的合约                           |
| 3    | 会员找不到       | 期权自对冲操作中的会员找不到                           |
| 4    | 客户找不到       | 期权自对冲操作中的客户找不到                           |
| 15   | 客户没有在该会员开户  | 期权自对冲操作中的客户并没有在指定的会员中开户                  |
| 22   | 交易所数据没有同步   | 交易系统尚未初始化完毕, 可以稍后再重试                     |
| 23   | 结算组数据没有同步   | 交易系统尚未初始化完毕, 可以稍后再重试                     |
| 26   | 当前状态禁止此项操作  | 合约交易状态既不处于连续交易也不处于交易业务处理状态               |
| 51   | 没有交易权限      | 对指定合约、或者客户对指定的合约、或者交易员没有交易权限             |
| 53   | 没有此项交易角色    | 会员在指定合约上不具有该客户对应的交易角色                    |
| 57   | 不能为其他会员操作   | 用户给非其所属的会员进行操作                           |
| 58   | 用户不匹配       | 期权自对冲操作中的用户和登录用户不匹配                      |
| 138  | 期权自对冲操作字段错误 | 期权自对冲操作中有字段值不合法<br>(枚举值越界或操作标志为修改、激活或挂起) |
| 140  | 期权自对冲更新已经撤销 | 要操作的期权自对冲已经取消                            |

|     |              |                |
|-----|--------------|----------------|
| 142 | 期权自对冲找不到     | 要操作的期权自对冲没有找到  |
| 143 | 期权自对冲操作只能是删除 | 期权自对冲操作的操作类型错误 |

**nRequestID:** 返回用户期权自对冲操作请求的 ID，该 ID 由用户在期权自对冲操作时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.58.OnErrRtnOptionSelfCloseAction 方法

期权自对冲操作错误回报。当会员系统执行期权自对冲操作，出现报错时，由交易系统主动通知会员系统，该方法会被调用。

函数原形：

```
void OnErrRtnOptionSelfCloseAction(  
    CShfeFtdcOptionSelfCloseActionField* pOptionSelfCloseAction,  
    CShfeFtdcRspInfoField* pRspInfo);
```

参数：

**pOptionSelfCloseAction:** 指向期权自对冲操作结构的地址。期权自对冲操作结构：

```
struct CShfeFtdcOptionSelfCloseActionField{  
    ///期权自对冲编号  
    TShfeFtdcOptionSelfCloseSysIDType OptionSelfCloseSysID;  
    ///本地期权自对冲编号  
    TShfeFtdcOrderLocalIDType OptionSelfCloseLocalID;  
    ///期权自对冲操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///操作本地编号  
    TShfeFtdcOrderLocalIDType ActionLocalID;  
    ///业务单元  
    TShfeFtdcBusinessUnitType BusinessUnit;  
    ///本地业务标识  
    TShfeFtdcBusinessLocalIDType BusinessLocalID;  
    ///IP 地址
```



```
TShfeFtdcIPAddressType  IPAddress;

///Mac 地址
TShfeFtdcMacAddressType MacAddress;

};
```

**pRspInfo:** 指向响应信息结构的地址。 响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType      ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType     ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示         | 可能的原因                                    |
|------|--------------|------------------------------------------|
| 2    | 合约找不到        | 找不到期权自对冲操作中的合约                           |
| 3    | 会员找不到        | 期权自对冲操作中的会员找不到                           |
| 4    | 客户找不到        | 期权自对冲操作中的客户找不到                           |
| 15   | 客户没有在该会员开户   | 期权自对冲操作中的客户并没有在指定的会员中开户                  |
| 22   | 交易所数据没有同步    | 交易系统尚未初始化完毕，可以稍后再重试                      |
| 23   | 结算组数据没有同步    | 交易系统尚未初始化完毕，可以稍后再重试                      |
| 26   | 当前状态禁止此项操作   | 合约交易状态既不处于连续交易也不处于交易业务处理状态               |
| 51   | 没有交易权限       | 对指定合约、或者客户对指定的合约、或者交易员没有交易权限             |
| 53   | 没有此项交易角色     | 会员在指定合约上不具有该客户对应的交易角色                    |
| 57   | 不能为其他会员操作    | 用户给非其所属的会员进行操作                           |
| 58   | 用户不匹配        | 期权自对冲操作中的用户和登录用户不匹配                      |
| 138  | 期权自对冲操作字段错误  | 期权自对冲操作中有字段值不合法<br>(枚举值越界或操作标志为修改、激活或挂起) |
| 140  | 期权自对冲更新已经撤销  | 要操作的期权自对冲已经取消                            |
| 142  | 期权自对冲找不到     | 要操作的期权自对冲没有找到                            |
| 143  | 期权自对冲操作只能是删除 | 期权自对冲操作的操作类型错误                           |

## 2.1.59.OnRspQryOptionSelfClose 方法

期权自对冲查询应答。当会员系统发出期权自对冲查询请求后，交易系统返回响应时，该方法会被调用。

**函数原形:**

```
void OnRspQryOptionSelfClose(
    CShfeFtdcOptionSelfCloseField* pOptionSelfClose,
```

```

CShfeFtdcRspInfoField* pRspInfo,
int nRequestID,
bool bIsLast);

```

## 参数:

**pOptionSelfClose:** 指向期权自对冲结构的地址，期权自对冲结构:

```

struct CShfeFtdcOptionSelfCloseField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///结算组代码
    TShfeFtdcSettlementGroupIDType   SettlementGroupID;
    ///结算编号
    TShfeFtdcSettlementIDType SettlementID;
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType   ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地期权自对冲编号
    TShfeFtdcOrderLocalIDType   OptionSelfCloseLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///期权行权的持仓是否自对冲
    TShfeFtdcOptSelfCloseFlagType SelfCloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType   IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
    ///期权自对冲编号
    TShfeFtdcOptionSelfCloseSysIDType OptionSelfCloseSysID;
    ///期权自对冲结果
    TShfeFtdcExecResultType SelfCloseResult;
    ///报单日期
    TShfeFtdcDateType   InsertDate;
    ///插入时间
    TShfeFtdcTimeType   InsertTime;

```

```
///撤销时间
TShfeFtdcTimeType   CancelTime;
///结算会员编号
TShfeFtdcParticipantIDType ClearingPartID;
///业务发生日期
TShfeFtdcDateType    ActionDay;

};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType   ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示      | 可能的原因                       |
|------|-----------|-----------------------------|
| 54   | 会话找不到     | 用户未登录                       |
| 57   | 不能为其他会员操作 | 不能查询其它会员下的情况                |
| 80   | 用户无此权限    | 只有交易用户能做查询、<br>只能查询单个会员下的情况 |

**nRequestID:** 返回用户期权自对冲查询请求的 ID，该 ID 由用户在期权自对冲查询时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.60.OnRspAuthenticate 方法

本方法仅针对自营会员，用于自营会员采集交易终端信息前的认证。

终端认证应答。当自营会员系统执行终端认证操作后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspAuthenticate(
    CShfeFtdcProductAuthField* pProductAuth,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

**pProductAuth:** 指向终端产品认证信息结构的地址，终端产品认证信息结构：

```
struct CShfeFtdcProductAuthField
{
    ///交易终端名称
    TShfeFtdcProductInfoType  AppID;
    ///终端认证授权码
    TShfeFtdcAuthCodeType  AuthCode;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType  ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType  ErrorMsg;
};
```

可能出现的错误：

| 错误代码 | 错误提示 | 可能的原因                |
|------|------|----------------------|
| -1   | 认证失败 | 找不到交易终端对应的授权码或授权码不匹配 |

**nRequestID:** 返回用户终端认证请求的 ID，该 ID 由用户执行终端认证时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.2. CShfeFtdcTraderApi 接口

CShfeFtdcTraderApi 接口提供给用户的功能包括，报单与报价的录入、报单与报价的撤销、报单挂起与激活、报单与报价的查询、成交单查询、会员客户查询、会员持仓查询、客户持仓查询、合约查询、合约交易状态查询和交易所公告查询等功能。

交易系统对每个席位发送的指令速度（每秒发送的指令条数和在途的指令条数）有限制，超过限额，指令发送会失败。具体限额数，请咨询交易所相关部门。

### 2.2.1. CreateFtdcTraderApi 方法

产生一个 CShfeFtdcTraderApi 的一个实例，不能通过 new 来产生。

函数原形：

```
static CShfeFtdcTraderApi* CreateFtdcTraderApi(const char* pszFlowPath = "");
```

参数：

**pszFlowPath**：常量字符指针，用于指定一个文件目录来存贮交易系统发布消息的状态。默认值代表当前目录。

返回值：

返回一个指向 CShfeFtdcTraderApi 实例的指针。

### 2.2.2. GetVersion 方法

获取 API 版本号。

函数原形：

```
const char* GetVersion(int& nMajorVersion, int& nMinorVersion);
```

参数：

**nMajorVersion**：返回主版本号

**nMinorVersion**：返回子版本号

返回值：

返回一个指向版本标识字符串的常量指针。

### 2.2.3. Release 方法

释放当前 API 实例的内部资源，退出 API 工作线程和设置 API 退出信号（只设置退出信号，不释放实例）。

函数原形：

```
int Release();
```

返回值:

- 0, 代表成功。
- -9, 表示未初始化。

## 2.2.4.Init 方法

使会员系统开始与交易系统建立连接, 连接成功后可以进行登录。

函数原形:

```
int Init();
```

返回值:

- 0, 代表成功。
- -5, 表示已登录或重复调用。

## 2.2.5.Join 方法

阻塞 API 工作线程, 当 API 退出信号被触发后, 释放当前 API 实例。

函数原形:

```
int Join();
```

返回值:

- 0, 代表成功。

## 2.2.6.GetTradingDay 方法

获得当前交易日。只有成功登录到交易系统之后才会取到正确的值。

函数原形:

```
const char* GetTradingDay();
```

返回值:

返回一个指向日期信息字符串的常量指针。

### 2.2.7. RegisterSpi 方法

注册一个派生自 CShfeFtdcTraderSpi 接口类的实例，该实例将完成事件处理。

函数原形：

```
void RegisterSpi(CShfeFtdcTraderSpi* pSpi);
```

参数：

**pSpi:** 实现了 CShfeFtdcTraderSpi 接口的实例指针。

### 2.2.8. RegisterFront 方法

设置交易前置的网络通讯地址。交易系统拥有多个交易前置，用户可以同时注册多个交易前置的网络通讯地址。

函数原形：

```
int RegisterFront(const char* pszFrontAddress);
```

参数：

**pszFrontAddress:** 指向交易前置网络通讯地址的指针。服务器地址的格式为：“protocol://ipaddress:port”，如：“tcp://127.0.0.1:17001”。“tcp”代表传输协议，“127.0.0.1”代表服务器地址。“17001”代表服务器端口号。

返回值：

- 0，代表成功。
- -8，表示注册前置网络地址个数超过最大值；
- -10，表示已初始化。

### 2.2.9. RegisterNameServer 方法

设置交易系统 FENS 服务的网络通讯地址。交易系统拥有多个 FENS 服务，

用户可以同时注册多个 FENS 服务的网络通讯地址。

### 函数原形:

```
int RegisterNameServer (const char* pszNsAddress);
```

### 参数:

**pszNsAddress:** 指向交易系统 FENS 服务网络通讯地址的指针。网络地址的格式为: “protocol://ipaddress:port”, 如: ”tcp://127.0.0.1:17001”。 “tcp”代表传输协议, “127.0.0.1”代表服务器地址。 ”17001”代表服务器端口号。

### 返回值:

- 0, 代表成功。
- -8, 表示注册 FENS 服务网络地址个数超过最大值;
- -10, 表示已初始化。

## 2.2.10.SetHeartbeatTimeout 方法

设置网络通信心跳的超时时间。当 TraderAPI 与交易系统的 TCP 连接建立后, 连接会定时发送心跳, 用以检测连接是否正常。该方法用于设置检测心跳超时的时间。交易所建议会员系统将 timeout 值设置为 10 至 30 秒之间。

### 函数原形:

```
int SetHeartbeatTimeout(unsigned int timeout);
```

### 参数:

**timeout:** 心跳超时时间(秒)。若超过 timeout/2 秒未收到交易系统的任何信息时, 将触发回调 OnHeartBeatWarning。若超过 timeout 秒未收到交易系统的任何信息时, 连接将会中断, 将触发回调 OnFrontDisconnected。

参见第一部分 4.9 心跳机制 (Heartbeat)

### 返回值:

- 0, 代表成功。
- -10, 表示已初始化。



### 2.2.11. OpenRequestLog 方法

打开请求日志文件。调用该方法之后，向交易系统发出的请求信息将记录在指定的日志文件中。

函数原形：

```
int OpenRequestLog(const char* pszReqLogFileName);
```

参数：

**pszReqLogFileName:** 请求日志文件名。

返回值：

- 0，代表成功。
- -4，表示打开日志文件失败。

### 2.2.12. OpenResponseLog 方法

打开应答日志文件。调用该方法之后，交易系统返回的信息将记录在指定的日志文件中，包括应答信息和回报信息。

函数原形：

```
int OpenResponseLog(const char* pszRspLogFileName);
```

参数：

**pszRspLogFileName:** 应答日志文件名。

返回值：

- 0，代表成功。
- -4，表示打开日志文件失败。

### 2.2.13.SubscribePrivateTopic 方法

订阅会员私有流。订阅成功后交易系统会根据订阅权限主动发送会员私有流或交易员私有流给会员系统。

函数原形：

```
int SubscribePrivateTopic(TE_RESUME_TYPE nResumeType);
```

参数：

**nResumeType:** 会员私有流重传方式：

- TERT\_RESTART: 从本交易日开始重传；
- TERT\_RESUME: 从上次收到的续传；为保证会员交易数据的完整性，交易所建议使用此方式接收会员私有流，待会员当日交易数据恢复后再处理后续报单业务。
- TERT\_QUICK: 只传送登录后会员私有流的内容；为保证会员交易数据的完整性，交易所不建议使用此方式接收私有流。

返回值：

- 0，代表成功。
- -10，表示已初始化。

### 2.2.14.SubscribePublicTopic 方法

订阅公共流。订阅成功后交易系统会主动发送公共流给会员系统。

函数原形：

```
int SubscribePublicTopic(TE_RESUME_TYPE nResumeType);
```

参数：

**nResumeType:** 公共流重传方式

- TERT\_RESTART: 从本交易日开始重传；
- TERT\_RESUME: 从上次收到的续传；
- TERT\_QUICK: 只传送登录后公共流的内容。

返回值：

- 0，代表成功。
- -10，表示已初始化。

### 2.2.15.SubscribeUserTopic 方法

订阅交易员私有流。订阅成功后交易系统会主动发送交易员私有流给会员系统。

函数原形：

```
int SubscribeUserTopic(TE_RESUME_TYPE nResumeType);
```

参数：

**nResumeType:** 交易员私有流重传方式，

- TERT\_RESTART: 从本交易日开始重传；
- TERT\_RESUME: 从上次收到的续传；为保证会员交易数据的完整性，交易所建议使用此方式接收交易员私有流，待会员和交易员当日交易数据恢复后再处理后续报单业务。
- TERT\_QUICK: 只传送登录后交易员私有流的内容；为保证会员交易数据的完整性，交易所不建议使用此方式接收交易员私有流。

返回值：

- 0，代表成功。
- -10，表示已初始化。

### 2.2.16.ReqUserLogin 方法

用户登录请求。

函数原形：

```
int ReqUserLogin(
    CShfeFtdcReqUserLoginField* pReqUserLoginField,
    int nRequestID);
```

## 参数:

**pReqUserLoginField:** 指向用户登录请求结构的地址。用户登录请求结构:

```

struct CShfeFtdcReqUserLoginField {
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///交易用户代码
    TShfeFtdcUserIDType  UserID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///密码
    TShfeFtdcPasswordType Password;
    ///用户端产品信息
    TShfeFtdcProductInfoType  UserProductInfo;
    ///接口端产品信息
    TShfeFtdcProductInfoType  InterfaceProductInfo;
    ///协议信息
    TShfeFtdcProtocolInfoType ProtocolInfo;
    ///数据中心代码
    TShfeFtdcDataCenterIDType  DataCenterID;
};

```

用户需要填写 **UserProductInfo** 字段,即会员系统的产品信息,如软件开发商、版本号等,例如:"SFIT Trader V100"代表技术公司开发的交易程序和版本号。

**nRequestID:** 用户登录请求的 ID, 该 ID 由用户指定, 管理。

## 返回值:

- 0, 代表成功。
- -2, 表示超出交易在途流控;
- -3, 表示超出交易请求流控;
- -5, 表示已登录或重复调用;
- -6, 表示强制要求填写的字段为空<sup>1</sup> (**UserProductInfo** 未填写);
- -7, 表示启用认证但认证失败;
- -9, 表示未初始化;
- -12, 表示尚未与前置建立连接。

<sup>1</sup> 注意: 空字符串定义: 仅包含空格或无字符的字符串, 以下同。

## 2.2.17. ReqUserLogout 方法

用户登出请求。

函数原形：

```
int ReqUserLogout(  
    CShfeFtdcReqUserLogoutField* pReqUserLogout,  
    int nRequestID);
```

参数：

**pReqUserLogout:** 指向用户登出请求结构的地址。用户登出请求结构：

```
struct CShfeFtdcReqUserLogoutField {  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
};
```

**nRequestID:** 用户登出请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控。

## 2.2.18. ReqUserPasswordUpdate 方法

用户密码修改请求。

函数原形：

```
int ReqUserPasswordUpdate(  
    CShfeFtdcUserPasswordUpdateField* pUserPasswordUpdate,  
    int nRequestID);
```

参数：

**pUserPasswordUpdate:** 指向用户密码修改结构的地址。用户密码修改结构:

```
struct CShfeFtdcUserPasswordUpdateField {
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///旧密码
    TShfeFtdcPasswordType    OldPassword;
    ///新密码
    TShfeFtdcPasswordType    NewPassword;
};
```

**nRequestID:** 用户密码修改请求的 ID，该 ID 由用户指定，管理。

返回值:

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -13，表示会员号不一致；
- -14，表示用户号不一致。

## 2.2.19.ReqSubscribeTopic 方法

订阅主题请求。应当在登录成功后调用。

函数原形:

```
int ReqSubscribeTopic(
    CShfeFtdcDisseminationField* pDissemination,
    int nRequestID);
```

参数:

**pDissemination:** 指向订阅主题结构的地址，包含了要订阅的主题和起始报文的序号。订阅主题结构:

```
struct CShfeFtdcDisseminationField {
```

```

///序列系列号
TShfeFtdcSequenceSeriesType   SequenceSeries;
///序列号
TShfeFtdcSequenceNoType   SequenceNo;
};
SequenceSeries: 要订阅的主题
SequenceNo: <0 表示使用 TERT_QUICK 方式, 其它值表示从该序号开始续传

```

**nRequestID:** 用户订阅主题请求的 ID, 该 ID 由用户指定, 管理。

返回值:

- 0, 代表成功。
- -1, 表示未登录;
- -2, 表示超出交易在途流控;
- -3, 表示超出交易请求流控。

## 2.2.20.ReqQryTopic 方法

查询主题请求。应当在登录成功后调用。

函数原形:

```

int ReqQryTopic(
    CShfeFtdcDisseminationField* pDissemination,
    int nRequestID);

```

参数:

**pDissemination:** 指向查询主题结构的地址, 包含了要查询的主题。订阅主题结构:

```

struct CShfeFtdcDisseminationField {
    ///序列系列号, 填写要查询的主题号
    TShfeFtdcSequenceSeriesType   SequenceSeries;
    ///序列号, 未使用
    TShfeFtdcSequenceNoType   SequenceNo;
};

```

**nRequestID:** 用户查询主题请求的 ID, 该 ID 由用户指定, 管理。

返回值:

- 0, 代表成功。

- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

### 2.2.21.ReqOrderInsert 方法

报单录入请求。

函数原形：

```
int ReqOrderInsert(  
    CShfeFtdcInputOrderField* pInputOrder,  
    int nRequestID);
```

参数：

**pInputOrder：** 指向输入报单结构的地址，输入报单结构：

```
struct CShfeFtdcInputOrderField {  
    ///报单编号，该字段由交易系统返回。  
    TShfeFtdcOrderSysIDType OrderSysID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///合约代码  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///报单价格条件  
    TShfeFtdcOrderPriceTypeType OrderPriceType;  
    ///买卖方向  
    TShfeFtdcDirectionType Direction;  
    ///组合开平标志  
    TShfeFtdcCombOffsetFlagType CombOffsetFlag;  
    ///组合套保标志  
    TShfeFtdcCombHedgeFlagType CombHedgeFlag;  
    ///价格  
    TShfeFtdcPriceType LimitPrice;  
    ///数量  
    TShfeFtdcVolumeType VolumeTotalOriginal;  
    ///有效期类型  
    TShfeFtdcTimeConditionType TimeCondition;
```



```

///GTD 日期, 未使用
TShfeFtdcDateType  GTDDate;
///成交量类型
TShfeFtdcVolumeConditionType  VolumeCondition;
///最小成交量, 成交量类型为最小数量时有效
TShfeFtdcVolumeTypeMinVolume;
///触发条件
TShfeFtdcContingentConditionType  ContingentCondition;
///止损价, 未使用
TShfeFtdcPriceType  StopPrice;
///强平原因
TShfeFtdcForceCloseReasonType  ForceCloseReason;
///本地报单编号*
TShfeFtdcOrderLocalIDType  OrderLocalID;
///自动挂起标志
TShfeFtdcBoolType  IsAutoSuspend;
///业务单元, 未使用
TShfeFtdcBusinessUnitType  BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType  BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType  IPAddress;
///Mac 地址
TShfeFtdcMacAddressType  MacAddress;
};

```

\* OrderLocalID: 本地报单编号, 只能递增(通过字符串比较)。每次登入成功后, 可以从 OnRspUserLogin 的输出参数 CShfeFtdcRspUserLoginField 中获得该席位当日用过的最大本地报单编号 MaxOrderLocalID。

**nRequestID:** 用户报单请求的 ID, 该 ID 由用户指定, 管理。在一次会话中, 该 ID 允许重复。

**返回值:**

- 0, 代表成功;
- -1, 表示未登录;
- -2, 表示超出交易在途流控;
- -3, 表示超出交易请求流控;
- -6, 表示强制要求填写的字段为空 (OrderLocalID 为空);
- -11, 表示重复的 id (OrderLocalID 不是递增<sup>2</sup>)。

<sup>2</sup> 递增的含义: 涉及本地编号的请求共用 LocalID 编号序列, 对于非空字符串的本地编号要求递增, 指共用

业务说明：

当前交易系统支持的报单类型如下表所示：

| 价 格 条 件<br>OrderPriceType | 时 间 条 件<br>TimeCondition | 成 交 量 条 件<br>VolumeCondition | 触 发 条 件<br>ContingentCondition |
|---------------------------|--------------------------|------------------------------|--------------------------------|
| 限价                        | 当日有效                     | 任意数量 AV                      | 立即                             |
| 市价                        |                          |                              |                                |
| 限价                        | 立即完成, 否则撤<br>销           | 任意数量 AV                      | 立即                             |
|                           |                          | 最小数量 MV                      |                                |
| 市价                        |                          | 全部数量 CV                      |                                |

2.2.22.ReqOrderAction 方法

报单操作请求，包括报单的撤销、报单的挂起、报单的激活、报单的修改。

函数原形：

```
int ReqOrderAction(  
    CShfeFtdcOrderActionField* pOrderAction,  
    int nRequestID);
```

参数：

**pOrderAction：** 指向报单操作结构的地址。报单操作结构：

```
struct CShfeFtdcOrderActionField {  
    ///报单编号*  
    TShfeFtdcOrderSysIDType OrderSysID;  
    ///本地报单编号*  
    TShfeFtdcOrderLocalIDType OrderLocalID;  
    ///报单操作标志  
    TShfeFtdcActionFlagType ActionFlag;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///价格，未使用  
    TShfeFtdcPriceType LimitPrice;  
    ///操作本地编号*
```

的 LocalID 编号序列递增，以下同。

```

TShfeFtdcOrderLocalIDType    ActionLocalID;
///数量变化, 未使用
TShfeFtdcVolumeType VolumeChange;
///业务单元, 未使用
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType    IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;
};
* OrderSysID, OrderLocalID 是指要进行操作的目标报单, 可以任填一个。
* ActionLocalID: 操作本地编号, 如果非空字符串则要求递增。

```

**nRequestID:** 用户报单操作请求的 ID, 该 ID 由用户指定, 管理。

### 返回值:

- 0, 代表成功。
- -1, 表示未登录;
- -2, 表示超出交易在途流控;
- -3, 表示超出交易请求流控;
- -6, 表示强制要求填写的字段为空 (OrderSysID 和 OrderLocalID 均为空);
- -11, 表示重复的 id (ActionLocalID 不是递增)。

### 业务说明:

当前暂不支持报单修改功能。

## 2.2.23.ReqQuoteInsert 方法

报价录入请求。

### 函数原形:

```

int ReqQuoteInsert(
    CShfeFtdcInputQuoteField* pInputQuote,
    int nRequestID);

```

**参数:****pInputQuote:** 指向输入报价结构的地址。输入报价结构:

```

struct CShfeFtdcInputQuoteField {
    ///报价编号，该字段由交易系统返回。
    TShfeFtdcQuoteSysIDType QuoteSysID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///合约代码
    TShfeFtdcInstrumentIDType InstrumentID;
    ///本地报价编号
    TShfeFtdcOrderLocalIDType QuoteLocalID;
    ///业务单元，未使用
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///买方组合开平标志
    TShfeFtdcCombOffsetFlagType BidCombOffsetFlag;
    ///买方组合套保标志
    TShfeFtdcCombHedgeFlagType BidCombHedgeFlag;
    ///买方价格
    TShfeFtdcPriceType BidPrice;
    ///卖方组合开平标志
    TShfeFtdcCombOffsetFlagType AskCombOffsetFlag;
    ///卖方组合套保标志
    TShfeFtdcCombHedgeFlagType AskCombHedgeFlag;
    ///卖方价格
    TShfeFtdcPriceType AskPrice;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
    ///询价编号
    TShfeFtdcOrderSysIDType QuoteDemandID;
};

```

**nRequestID:** 用户报价请求的 ID，该 ID 由用户指定，管理。

## 返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（QuoteLocalID 为空）；
- -11，表示重复的 id（QuoteLocalID 不是递增）。

## 2.2.24.ReqQuoteAction 方法

报价操作请求，包括报价的撤销、报价的挂起、报价的激活、报价的修改。

### 函数原形：

```
int ReqQuoteAction(
    CShfeFtdcQuoteActionField* pQuoteAction,
    int nRequestID);
```

### 参数：

**pQuoteAction：**指向报价操作结构的地址。报价操作结构：

```
struct CShfeFtdcQuoteActionField {
    ///报价编号
    TShfeFtdcQuoteSysIDType QuoteSysID;
    ///本地报价编号
    TShfeFtdcOrderLocalIDType QuoteLocalID;
    ///报单操作标志
    TShfeFtdcActionFlagType ActionFlag;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///操作本地编号
    TShfeFtdcOrderLocalIDType ActionLocalID;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
```

```

TShfeFtdcBusinessLocalIDType BusinessLocalID;
//IP 地址
TShfeFtdcIPAddressType IPAddress;
//Mac 地址
TShfeFtdcMacAddressType MacAddress;

};

```

**nRequestID:** 用户报价操作请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（QuoteSysID 和 QuoteLocalID 均为空）；
- -11，表示重复的 id（ActionLocalID 不是递增）。

业务说明：

当前仅支持报价的撤销。

## 2.2.25. ReqExecOrderInsert 方法

执行宣告录入请求，只有期权买方才能录入执行宣告请求。

函数原形：

```

int ReqExecOrderInsert(
    CShfeFtdcInputExecOrderField* pInputExecOrder,
    int nRequestID);

```

参数：

**pInputExecOrder:** 指向输入执行宣告结构的地址。输入执行宣告结构：

```

struct CShfeFtdcInputExecOrderField {
    //合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    //会员代码
    TShfeFtdcParticipantIDType ParticipantID;

```

```

    ///客户代码
    TShfeFtdcClientIDType    ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType    UserID;
    ///本地执行宣告编号
    TShfeFtdcOrderLocalIDType    ExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType    Volume;
    ///开平标志
    TShfeFtdcOffsetFlagType    OffsetFlag;
    ///套保标志
    TShfeFtdcHedgeFlagType    HedgeFlag;
    ///申请的持仓方向，即作为买方（多头）还是卖方（空头）来提出申请
    TShfeFtdcPosiDirectionType    PosiDirection;
    ///期权行权后是否保留期货持仓的标记，未使用
    TShfeFtdcExecOrderPositionFlagType    ReservePositionFlag;
    ///期权行权后生成的期货持仓是否自对冲
    TShfeFtdcExecOrderCloseFlagType    CloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType    BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType    BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType    IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType    MacAddress;
};

```

**nRequestID:** 用户执行宣告录入请求的 ID，该 ID 由用户指定，管理。

## 返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（ExecOrderLocalID 为空）；
- -11，表示重复的 id（ExecOrderLocalID 不是递增）。

## 2.2.26. ReqExecOrderAction 方法

执行宣告操作请求。

函数原形：

```
int ReqExecOrderAction(
    CShfeFtdcExecOrderActionField* pExecOrderAction,
    int nRequestID);
```

参数：

**pExecOrderAction:** 指向执行宣告操作结构的地址。执行宣告操作结构：

```
struct CShfeFtdcExecOrderActionField {
    ///执行宣告编号
    TShfeFtdcExecOrderSysIDType  ExecOrderSysID;
    ///本地执行宣告编号
    TShfeFtdcOrderLocalIDType  ExecOrderLocalID;
    ///报单操作标志
    TShfeFtdcActionFlagType  ActionFlag;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType  ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///操作本地编号
    TShfeFtdcOrderLocalIDType  ActionLocalID;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType  IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
};
```

**nRequestID:** 用户执行宣告操作请求的 ID，该 ID 由用户指定，管理。

返回值：

■ 0，代表成功。



- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（ExecOrderSysID 和 ExecOrderLocalID 均为空）；
- -11，表示重复的 id（ActionLocalID 不是递增）。

### 业务说明：

当前仅支持执行宣告的撤销。

## 2.2.27.ReqQryPartAccount 方法

会员资金查询请求。

### 函数原形：

```
int ReqQryPartAccount(
    CShfeFtdcQryPartAccountField* pQryPartAccount,
    int nRequestID);
```

### 参数：

**pQryPartAccount:** 指向会员资金查询结构的地址。会员资金查询结构：

```
struct CShfeFtdcQryPartAccountField {
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDTypePartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDTypePartIDEnd;
    ///资金帐号，可选
    TShfeFtdcAccountIDType AccountID;
};
```

**nRequestID:** 用户会员资金查询请求的 ID，该 ID 由用户指定，管理。

### 返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；

- -3，表示超出查询请求流控。

## 2.2.28.ReqQryOrder 方法

报单查询请求。

函数原形：

```
int ReqQryOrder(
    CShfeFtdcQryOrderField* pQryOrder,
    int nRequestID);
```

参数：

**pQryOrder：** 指向报单查询结构的地址。各个查询条件之间是与的关系。

如果某个可选的查询条件为空，则忽略该查询条件。报单查询结构：

```
struct CShfeFtdcQryOrderField {
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///报单编号，可选
    TShfeFtdcOrderSysIDType OrderSysID;
    ///合约代码，可选
    TShfeFtdcInstrumentIDType InstrumentID;
    ///客户代码，可选
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码，可选
    TShfeFtdcUserIDType UserID;
    ///开始时间，可选
    TShfeFtdcTimeType TimeStart;
    ///结束时间，可选
    TShfeFtdcTimeType TimeEnd;
};
```

**nRequestID：** 用户报单查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；

- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.29. ReqQryQuote 方法

报价查询请求。

函数原形：

```
int ReqQryQuote(
    CShfeFtdcQryQuoteField* pQryQuote,
    int nRequestID);
```

参数：

**pQryQuote：** 指向报价查询结构的地址。报价查询结构：

```
struct CShfeFtdcQryQuoteField {
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///报价编号，可选
    TShfeFtdcQuoteSysIDType QuoteSysID;
    ///客户代码，可选
    TShfeFtdcClientIDType ClientID;
    ///合约代码，可选
    TShfeFtdcInstrumentIDType InstrumentID;
    ///交易用户代码，可选
    TShfeFtdcUserIDType UserID;
};
```

**nRequestID：** 用户报价查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.30. ReqQryTrade 方法

成交单查询请求。

函数原形：

```
int ReqQryTrade(
    CShfeFtdcQryTradeField* pQryTrade,
    int nRequestID);
```

参数：

**pQryTrade:** 指向成交查询结构的地址。成交查询结构：

```
struct CShfeFtdcQryTradeField {
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///起始合约代码，可选
    TShfeFtdcInstrumentIDType InstIDStart;
    ///结束合约代码，可选
    TShfeFtdcInstrumentIDType InstIDEnd;
    ///成交编号，可选
    TShfeFtdcTradeIDType TradeID;
    ///客户代码，可选
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码，可选
    TShfeFtdcUserIDType UserID;
    ///开始时间，可选
    TShfeFtdcTimeType TimeStart;
    ///结束时间，可选
    TShfeFtdcTimeType TimeEnd;
};
```

**nRequestID:** 用户成交单查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

### 2.2.31. ReqQryClient 方法

会员客户查询请求。

函数原形：

```
int ReqQryClient(
    CShfeFtdcQryClientField* pQryClient,
    int nRequestID);
```

参数：

**pQryClient:** 指向客户查询结构的地址。客户查询结构：

```
struct CShfeFtdcQryClientField {
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///起始客户代码，可选
    TShfeFtdcClientIDType      ClientIDStart;
    ///结束客户代码，可选
    TShfeFtdcClientIDType      ClientIDEnd;
};
```

**nRequestID:** 用户客户查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

### 2.2.32. ReqQryPartPosition 方法

会员持仓查询请求。

函数原形：

```
int ReqQryPartPosition(
    CShfeFtdcQryPartPositionField* pQryPartPosition,
```

```
int nRequestID);
```

### 参数:

**pQryPartPosition:** 指向会员持仓查询结构的地址。会员持仓查询结构:

```
struct CShfeFtdcQryPartPositionField {
    ///起始会员代码, 只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码, 只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///起始合约代码, 可选
    TShfeFtdcInstrumentIDType InstIDStart;
    ///结束合约代码, 可选
    TShfeFtdcInstrumentIDType InstIDEnd;
};
```

**nRequestID:** 用户会员持仓查询请求的 ID, 该 ID 由用户指定, 管理。

### 返回值:

- 0, 代表成功。
- -1, 表示未登录;
- -2, 表示超出查询在途流控;
- -3, 表示超出查询请求流控。

## 2.2.33. ReqQryClientPosition 方法

客户持仓查询请求。

### 函数原形:

```
int ReqQryClientPosition(
    CShfeFtdcQryClientPositionField* pQryClientPosition,
    int nRequestID);
```

### 参数:

**pQryClientPosition:** 指向客户持仓查询结构的地址。客户持仓查询结构:

```
struct CShfeFtdcQryClientPositionField {
    ///起始会员代码, 只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码, 只能是本会员
```

```

TShfeFtdcParticipantIDType PartIDEnd;
///起始客户代码，可选
TShfeFtdcClientIDType      ClientIDStart;
///结束客户代码，可选
TShfeFtdcClientIDType      ClientIDEnd;
///起始合约代码，可选
TShfeFtdcInstrumentIDType InstIDStart;
///结束合约代码，可选
TShfeFtdcInstrumentIDType InstIDEnd;
///客户类型，可选
TShfeFtdcClientTypeType    ClientType;
};

```

**nRequestID:** 用户客户持仓查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.34. ReqQryInstrument 方法

合约查询请求。

函数原形：

```

int ReqQryInstrument(
    CShfeFtdcQryInstrumentField* pQryInstrument,
    int nRequestID);

```

参数：

**pQryInstrument:** 指向合约查询结构的地址。合约查询结构：

```

struct CShfeFtdcQryInstrumentField {
    ///结算组代码，可选
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;
    ///产品组代码，可选
    TShfeFtdcProductGroupIDType    ProductGroupID;
    ///产品代码，可选
    TShfeFtdcProductIDType    ProductID;
};

```

```

    ///合约代码，可选
    TShfeFtdcInstrumentIDType InstrumentID;
};

```

**nRequestID:** 用户合约查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.35. ReqQryInstrumentStatus 方法

合约交易状态查询请求。

函数原形：

```

int ReqQryInstrumentStatus(
    CShfeFtdcQryInstrumentStatusField* pQryInstrumentStatus,
    int nRequestID);

```

参数：

**pQryInstrumentStatus:** 指向合约状态查询结构的地址。合约状态查询结构：

```

struct CShfeFtdcQryInstrumentStatusField {
    ///起始合约代码，可选
    TShfeFtdcInstrumentIDType InstIDStart;
    ///结束合约代码，可选
    TShfeFtdcInstrumentIDType InstIDEnd;
};

```

**nRequestID:** 用户合约状态查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；



- -3，表示超出查询请求流控。

### 2.2.36.ReqQryMarketData 方法

普通行情查询请求。

函数原形：

```
int ReqQryMarketData(
    CShfeFtdcQryMarketDataField* pQryMarketData,
    int nRequestID);
```

参数：

**pQryMarketData：** 指向行情查询结构的地址。行情查询结构：

```
struct CShfeFtdcQryMarketDataField {
    ///产品代码，可选
    TShfeFtdcProductIDType ProductID;
    ///合约代码，可选
    TShfeFtdcInstrumentIDType InstrumentID;
};
```

**nRequestID：** 用户普通行情查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

### 2.2.37.ReqQryBulletin 方法

交易所公告查询请求。

函数原形：

```
int ReqQryBulletin(
    CShfeFtdcQryBulletinField* pQryBulletin,
    int nRequestID);
```

**参数:**

**pQryBulletin:** 指向公告查询结构的地址。公告查询结构:

```
struct CShfeFtdcQryBulletinField {
    ///交易日, 可选
    TShfeFtdcDateType    TradingDay;
    ///市场代码, 可选
    TShfeFtdcMarketIDType    MarketID;
    ///公告编号, 可选
    TShfeFtdcBulletinIDType    BulletinID;
    ///公告类型, 可选
    TShfeFtdcNewsTypeType    NewsType;
    ///紧急程度, 可选
    TShfeFtdcNewsUrgencyType    NewsUrgency;
};
```

**nRequestID:** 用户公告查询请求的 ID, 该 ID 由用户指定, 管理。

**返回值:**

- 0, 代表成功。
- -1, 表示未登录;
- -2, 表示超出查询在途流控;
- -3, 表示超出查询请求流控。

**2.2.38.ReqQryHedgeVolume 方法**

套保额度查询请求。

**函数原形:**

```
int ReqQryHedgeVolume(
    CShfeFtdcQryHedgeVolumeField* pQryHedgeVolume,
    int nRequestID);
```

**参数:**

**pQryHedgeVolume:** 指向套保额度查询结构的地址。套保额度查询结构:

```
struct CshfeFtdcQryHedgeVolumeField {
{
    ///起始会员代码, 只能是本会员
```

```

TShfeFtdcParticipantIDType PartIDStart;
///结束会员代码，只能是本会员
TShfeFtdcParticipantIDType PartIDEnd;
///起始客户代码，可选
TShfeFtdcClientIDType      ClientIDStart;
///结束客户代码，可选
TShfeFtdcClientIDType      ClientIDEnd;
///起始合约代码，可选
TShfeFtdcInstrumentIDType InstIDStart;
///结束合约代码，可选
TShfeFtdcInstrumentIDType InstIDEnd;
};

```

**nRequestID:** 用户套保额度查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

业务说明：

本功能当前暂不支持。

## 2.2.39. ReqQryExecOrder 方法

执行宣告查询请求。

函数原形：

```

int ReqQryExecOrder(
    CShfeFtdcQryExecOrderField* pQryExecOrder,
    int nRequestID);

```

参数：

**pQryExecOrder:** 指向执行宣告查询结构的地址。执行宣告查询结构：

```

struct CShfeFtdcQryExecOrderField
{
    ///起始会员代码，只能是本会员

```

```

TShfeFtdcParticipantIDType PartIDStart;
///结束会员代码，只能是本会员
TShfeFtdcParticipantIDType PartIDEnd;
///执行宣告编号，可选
TShfeFtdcExecOrderSysIDType ExecOrderSysID;
///合约代码，可选
TShfeFtdcInstrumentIDType InstrumentID;
///客户代码，可选
TShfeFtdcClientIDType ClientID;
///交易用户代码，可选
TShfeFtdcUserIDType UserID;
///开始时间，可选
TShfeFtdcTimeType TimeStart;
///结束时间，可选
TShfeFtdcTimeType TimeEnd;
};

```

**nRequestID:** 用户执行宣告查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.40. ReqQryExchangeRate 方法

汇率查询请求。

函数原形：

```

int ReqQryExchangeRate(
    CShfeFtdcQryExchangeRateField* pQryExchangeRate,
    int nRequestID);

```

参数：

**pQryExchangeRate:** 指向汇率查询结构的地址。汇率查询结构：

```

struct CShfeFtdcQryExchangeRateField
{
    ///币种代码

```

```
TShfeFtdcCurrencyIDType CurrencyID;
};
```

**nRequestID:** 用户汇率查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.41. ReqAbandonExecOrderInsert 方法

放弃执行宣告录入请求，只有期权买方才有权放弃执行。

函数原形：

```
int ReqAbandonExecOrderInsert(
    CShfeFtdcInputAbandonExecOrderField* pInputAbandonExecOrder,
    int nRequestID);
```

参数：

**pInputAbandonExecOrder:** 指向输入放弃执行宣告结构的地址。输入放弃执行宣告结构：

```
struct CShfeFtdcInputAbandonExecOrderField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地放弃执行宣告编号
    TShfeFtdcOrderLocalIDType AbandonExecOrderLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///开平标志
    TShfeFtdcOffsetFlagType OffsetFlag;
```

```

    ///套保标志
    TShfeFtdcHedgeFlagType HedgeFlag;
    ///申请放弃的持仓方向，实际只有多头才能提出申请放弃
    TShfeFtdcPosiDirectionType PosiDirection;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
};

```

**nRequestID:** 用户放弃执行宣告录入请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（AbandonExecOrderLocalID 为空）；
- -11，表示重复的 id（AbandonExecOrderLocalID 不是递增）。

## 2.2.42.ReqAbandonExecOrderAction 方法

放弃执行宣告操作请求。

函数原形：

```

int ReqAbandonExecOrderAction(
    CShfeFtdcAbandonExecOrderActionField* pAbandonExecOrderAction,
    int nRequestID);

```

参数：

**pAbandonExecOrderAction:** 指向放弃执行宣告操作结构的地址。放弃执行宣告操作结构：

```

struct CShfeFtdcAbandonExecOrderActionField {

```

```

    ///放弃执行宣告编号
    TShfeFtdcExecOrderSysIDType AbandonExecOrderSysID;
    ///本地放弃执行宣告编号
    TShfeFtdcOrderLocalIDType AbandonExecOrderLocalID;
    ///报单操作标志
    TShfeFtdcActionFlagType ActionFlag;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///操作本地编号
    TShfeFtdcOrderLocalIDType ActionLocalID;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
};

```

**nRequestID:** 用户放弃执行宣告操作请求的 ID，该 ID 由用户指定，管理。

## 返回值:

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（AbandonExecOrderLocalID 和 AbandonExecOrderSysID 均为空）；
- -11，表示重复的 id（ActionLocalID 不是递增）。

## 业务说明:

当前仅支持放弃执行宣告的撤销。

## 2.2.43. ReqQryAbandonExecOrder 方法

放弃执行宣告查询请求。

函数原形：

```
int ReqQryAbandonExecOrder(
    CShfeFtdcQryAbandonExecOrderField* pQryAbandonExecOrder,
    int nRequestID);
```

参数：

**pQryAbandonExecOrder：** 指向放弃执行宣告查询结构的地址。放弃执行宣告查询结构：

```
struct CShfeFtdcQryAbandonExecOrderField
{
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///放弃执行宣告编号，可选
    TShfeFtdcExecOrderSysIDType AbandonExecOrderSysID;
    ///合约代码，可选
    TShfeFtdcInstrumentIDType InstrumentID;
    ///客户代码，可选
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码，可选
    TShfeFtdcUserIDType UserID;
    ///开始时间，可选
    TShfeFtdcTimeType TimeStart;
    ///结束时间，可选
    TShfeFtdcTimeType TimeEnd;
};
```

**nRequestID：** 用户放弃执行宣告查询请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；



- -3，表示超出查询请求流控。

## 2.2.44. ReqQuoteDemand 方法

询价录入请求。

函数原形：

```
int ReqQuoteDemand(  
    CShfeFtdcInputQuoteDemandField* pInputQuoteDemand,  
    int nRequestID);
```

参数：

**pInputQuoteDemand：**指向询价录入请求结构的地址。询价录入请求结构：

```
struct CShfeFtdcInputQuoteDemandField  
{  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///客户代码  
    TShfeFtdcClientIDType ClientID;  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///合约代码  
    TShfeFtdcInstrumentIDType InstrumentID;  
    ///本地询价编号  
    TShfeFtdcOrderLocalIDType QuoteDemandLocalID;  
};
```

**nRequestID：**用户询价录入请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（QuoteDemandLocalID 为空）。

## 2.2.45. ReqOptionSelfCloseUpdate 方法

期权自对冲更新请求，普通客户可以申请自对冲期权仓位；期权卖方可以申请自对冲卖方履约后的期货仓位；期权做市商可以申请保留期权仓位。期权自对冲更新请求针对相同的会员、客户、合约和自对冲类型，只保留最新申请的一条记录。

### 函数原形：

```
int ReqOptionSelfCloseUpdate(
    CShfeFtdcInputOptionSelfCloseField* pInputOptionSelfClose,
    int nRequestID);
```

### 参数：

**pInputOptionSelfClose：** 指向期权自对冲更新结构的地址。期权自对冲更新结构：

```
struct CShfeFtdcInputOptionSelfCloseField {
    ///合约编号
    TShfeFtdcInstrumentIDType InstrumentID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///本地期权自对冲编号
    TShfeFtdcOrderLocalIDType OptionSelfCloseLocalID;
    ///数量
    TShfeFtdcVolumeType Volume;
    ///期权行权的持仓是否自对冲
    TShfeFtdcOptSelfCloseFlagType SelfCloseFlag;
    ///业务单元
    TShfeFtdcBusinessUnitType BusinessUnit;
    ///本地业务标识
    TShfeFtdcBusinessLocalIDType BusinessLocalID;
    ///IP 地址
    TShfeFtdcIPAddressType IPAddress;
    ///Mac 地址
    TShfeFtdcMacAddressType MacAddress;
};
```

**nRequestID:** 用户期权自对冲更新请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（OptionSelfCloseLocalID 为空）；
- -11，表示重复的 id（OptionSelfCloseLocalID 不是递增）。

## 2.2.46. ReqOptionSelfCloseAction 方法

期权自对冲操作请求。

函数原形：

```
int ReqOptionSelfCloseAction(
    CShfeFtdcOptionSelfCloseActionField* pOptionSelfCloseAction,
    int nRequestID);
```

参数：

**pOptionSelfCloseAction:** 指向期权自对冲操作结构的地址。期权自对冲操作结构：

```
struct CShfeFtdcOptionSelfCloseActionField {
    ///期权自对冲编号
    TShfeFtdcOptionSelfCloseSysIDType OptionSelfCloseSysID;
    ///本地期权自对冲编号
    TShfeFtdcOrderLocalIDType OptionSelfCloseLocalID;
    ///期权自对冲操作标志
    TShfeFtdcActionFlagType ActionFlag;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///客户代码
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///操作本地编号
```

```

TShfeFtdcOrderLocalIDType   ActionLocalID;
///业务单元
TShfeFtdcBusinessUnitType BusinessUnit;
///本地业务标识
TShfeFtdcBusinessLocalIDType BusinessLocalID;
///IP 地址
TShfeFtdcIPAddressType   IPAddress;
///Mac 地址
TShfeFtdcMacAddressType MacAddress;

};

```

**nRequestID:** 用户期权自对冲操作请求的 ID，该 ID 由用户指定，管理。

返回值:

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出交易在途流控；
- -3，表示超出交易请求流控；
- -6，表示强制要求填写的字段为空（OptionSelfCloseLocalID 和 OptionSelfCloseSysID 均为空）；
- -11，表示重复的 id（ActionLocalID 不是递增）。

业务说明:

当前仅支持期权自对冲的撤销。

## 2.2.47. ReqQryOptionSelfClose 方法

期权自对冲查询请求。

函数原形:

```

int ReqQryOptionSelfClose(
    CShfeFtdcQryOptionSelfCloseField* pQryOptionSelfClose,
    int nRequestID);

```

参数:

**pQryOptionSelfClose:** 指向期权自对冲查询结构的地址。期权自对冲查询结构:

```

struct CShfeFtdcQryOptionSelfCloseField
{
    ///起始会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDStart;
    ///结束会员代码，只能是本会员
    TShfeFtdcParticipantIDType PartIDEnd;
    ///期权自对冲编号，可选
    TShfeFtdcOptionSelfCloseSysIDType OptionSelfCloseSysID;
    ///合约代码，可选
    TShfeFtdcInstrumentIDType InstrumentID;
    ///客户代码，可选
    TShfeFtdcClientIDType ClientID;
    ///交易用户代码，可选
    TShfeFtdcUserIDType UserID;
    ///开始时间，可选
    TShfeFtdcTimeType TimeStart;
    ///结束时间，可选
    TShfeFtdcTimeType TimeEnd;
};

```

**nRequestID:** 用户期权自对冲查询请求的 ID，该 ID 由用户指定，管理。

返回值:

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出查询在途流控；
- -3，表示超出查询请求流控。

## 2.2.48. ReqAuthenticate 方法

本方法仅针对自营会员，用于自营会员采集交易终端信息前的认证。

终端认证请求。

函数原形:

```

int ReqAuthenticate(
    CShfeFtdcProductAuthField* pProductAuth,
    int nRequestID);

```

参数:

**pProductAuth:** 指向终端产品认证信息结构的地址。终端产品认证信息请求结构:

```
struct CShfeFtdcProductAuthField
{
    ///交易终端名称
    TShfeFtdcProductInfoType AppID;
    ///终端认证授权码
    TShfeFtdcAuthCodeType AuthCode;
};
```

**nRequestID:** 用户终端产品认证信息请求的 ID, 该 ID 由用户指定, 管理。

返回值:

- 0, 代表成功。
- -2, 表示超出交易在途流控;
- -3, 表示超出交易请求流控;
- -5, 表示已登录或重复调用 (不允许登录后调用或重复调用);
- -9, 表示未初始化;
- -12, 表示尚未与前置建立连接。

### 3. TraderAPI 接口开发示例

```
// tradeapitest.cpp :
// 一个简单的例子，介绍CShfeFtdcTraderApi和CShfeFtdcTraderSpi接口的使用。
// 本例将演示一个报单录入操作的过程

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "FtdcTraderApi.h"

class CSimpleHandler : public CShfeFtdcTraderSpi
{
public:
    // 构造函数，需要一个有效的指向CShfeFtdcTraderApi实例的指针
    CSimpleHandler(CShfeFtdcTraderApi *pTraderApi) : m_pTraderApi(pTraderApi) {}

    ~CSimpleHandler() {}

    // 当会员系统与交易系统建立起通信连接，会员系统需要进行登录
    virtual void OnFrontConnected()
    {
        CShfeFtdcReqUserLoginField reqUserLogin;
        memset(&reqUserLogin, 0, sizeof(reqUserLogin));
        strcpy(reqUserLogin.ParticipantID, "0888");
        strcpy(reqUserLogin.UserID, "0888c1c");
        strcpy(reqUserLogin.Password, "1");
        strcpy(reqUserLogin.UserProductInfo, "Test TraderAPI v2.00");
        // 发出登录请求
        int ret = m_pTraderApi->ReqUserLogin(&reqUserLogin, 0);
        if (ret != 0)
        {
            printf("ReqUserLogin Fail ret = %d\n", ret);
            exit(-1);
        }
    }

    // 当会员系统与交易系统通信连接断开时，该方法被调用
    virtual void OnFrontDisconnected(int nReason)
    {
        // 当发生这个情况后，API会自动重新连接，会员系统可不作处理
    }
}
```

```

        printf("OnFrontDisconnected Reason = %#x.\n", nReason);
    }

    // 当会员系统发出登录请求之后, 该方法会被调用, 通知会员系统登录是否成功
    virtual void OnRspUserLogin(CShfeFtdcRspUserLoginField *pRspUserLogin, CShfeFtdcRspInfoField
    *pRspInfo, int nRequestID, bool bIsLast)
    {
        printf("OnRspUserLogin:\n");
        printf("ErrorCode=[%d], ErrorMsg=[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        printf("RequestID=[%d], Chain=[%d]\n", nRequestID, bIsLast);

        if (pRspInfo->ErrorID != 0) {
            // 登录失败, 会员系统需进行错误处理
            printf("Failed to login, errorcode=%d errormsg=%s requestid=%d chain=%d",
            pRspInfo->ErrorID, pRspInfo->ErrorMsg, nRequestID, bIsLast);
            exit(-1);
        }

        // 登录成功, 发出报单录入请求
        CShfeFtdcInputOrderField ord;
        memset(&ord, 0, sizeof(ord));

        // 会员代码
        strcpy(ord.ParticipantID, "0888");
        // 客户代码
        strcpy(ord.ClientID, "08880001");
        // 交易用户代码
        strcpy(ord.UserID, "0888c1c");
        // 合约代码
        strcpy(ord.InstrumentID, "cu2511");
        // 报单价格条件
        ord.OrderPriceType = SHFE_FTDC_OPT_LimitPrice;
        // 买卖方向
        ord.Direction = SHFE_FTDC_D_Buy;
        // 组合开平标志
        strcpy(ord.CombOffsetFlag, "0");
        // 组合套保标志
        strcpy(ord.CombHedgeFlag, "1");
        // 价格
        ord.LimitPrice = 74000;
        // 数量
        ord.VolumeTotalOriginal = 10;
        // 有效期类型
        ord.TimeCondition = SHFE_FTDC_TC_GFD;
    }

```



```

// GTD日期
strcpy(ord.GTDDate, "");
// 成交量类型
ord.VolumeCondition = SHFE_FTDC_VC_AV;
// 最小成交量
ord.MinVolume = 0;
// 触发条件
ord.ContingentCondition = SHFE_FTDC_CC_Immediately;
// 止损价
ord.StopPrice = 0;
// 强平原因
ord.ForceCloseReason = SHFE_FTDC_FCC_NotForceClose;
// 本地报单编号
strcpy(ord.OrderLocalID, "0000000001");
// 自动挂起标志
ord.IsAutoSuspend = 0;

int ret = m_pTraderApi ->ReqOrderInsert(&ord, 1);
if (ret != 0)
{
    printf("ReqOrderInsert Fail ret = %d\n", ret);
    exit(-1);
}

}

// 报单录入应答
virtual void OnRspOrderInsert(CShfeFtdcInputOrderField *pInputOrder, CShfeFtdcRspInfoField *pRspInfo,
int nRequestID, bool bIsLast)
{
    // 输出报单录入结果
    printf("ErrorCode=[%d], ErrorMsg=[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);

    // 报单录入完成
    exit(0);
};

///报单回报
virtual void OnRtnOrder(CShfeFtdcOrderField *pOrder)
{
    printf("OnRtnOrder:\n");
    printf("OrderSysID=[%s]\n", pOrder->OrderSysID);
}

```

```
// 针对用户请求的出错通知
virtual void OnRspError(CShfeFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {
    printf("OnRspError:\n");
    printf("ErrorCode=[%d], ErrorMsg=[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
    printf("RequestID=[%d], Chain=[%d]\n", nRequestID, bIsLast);
    // 会员系统需进行错误处理
    exit(-1);
}

private:
    // 指向CShfeFtdcTraderApi实例的指针
    CShfeFtdcTraderApi *m_pTraderApi;
};

int main()
{
    // 产生一个CShfeFtdcTraderApi实例
    CShfeFtdcTraderApi *pTraderApi = CShfeFtdcTraderApi::CreateFtdcTraderApi();
    // 产生一个事件处理的实例
    CSimpleHandler sh(pTraderApi);
    // 注册一事件处理的实例
    pTraderApi->RegisterSpi(&sh);

    // 订阅私有流
    pTraderApi->SubscribePrivateTopic(TERT_RESUME);

    // 订阅公共流
    pTraderApi->SubscribePublicTopic(TERT_RESUME);

    //设置心跳超时时间
    pTraderApi->SetHeartbeatTimeout(19);

    // 设置交易系统前置NameServer的地址
    pTraderApi->RegisterNameServer("tcp://172.16.0.31:17001");

    // 使会员系统开始与交易系统建立连接
    pTraderApi->Init();

    // 等待API实例退出
    pTraderApi->Join();

    return 0;
}
```

## 第三部分 MduserAPI 参考手册

本部分主要提供给行情接收系统开发商阅读，内容包括：

第一章为 MduserAPI 接口分类。

第二章为 MduserAPI 接口说明。

第三章为 MduserAPI 接口开发示例。

# 1. MduserAPI 接口分类

## 1.1. 管理接口

MduserAPI 的管理接口用于对 API 的生命周期和运行参数进行控制。

| 接口类型     | 接口名称                                         | 说明                   |
|----------|----------------------------------------------|----------------------|
| 生命周期管理接口 | CShfeFtdcMduserApi::CreateFtdcMduserApi      | 创建 MduserApi 实例      |
|          | CShfeFtdcMduserApi::GetVersion               | 获取 API 版本            |
|          | CShfeFtdcMduserApi::Release                  | 删除接口实例               |
|          | CShfeFtdcMduserApi::Init                     | 初始化                  |
|          | CShfeFtdcMduserApi::Join                     | 等待接口线程结束运行           |
|          | CShfeFtdcMduserApi::GetTradingDay            | 获得当前交易日              |
| 参数管理接口   | CShfeFtdcMduserApi::RegisterSpi              | 注册回调接口               |
|          | CShfeFtdcMduserApi::RegisterFront            | 注册前置网络地址             |
|          | CShfeFtdcMduserApi::RegisterNameServer       | 注册 NameServer 网络地址   |
|          | CShfeFtdcMduserApi::SetHeartbeatTimeout      | 设置心跳超时时间             |
| 订阅接口     | CShfeFtdcMduserApi::SubscribeMarketDataTopic | 订阅行情                 |
| 日志接口     | CShfeFtdcMduserApi::OpenRequestLog           | 打开请求日志文件             |
|          | CShfeFtdcMduserApi::OpenResponseLog          | 打开应答日志文件             |
| 通信状态接口   | CShfeFtdcMduserSpi::OnFrontConnected         | 与交易系统建立起通信连接时，该方法被调用 |
|          | CShfeFtdcMduserSpi::OnFrontDisconnected      | 与交易系统通信连接断开时，该方法被调用  |
|          | CShfeFtdcMduserSpi::OnHeartBeatWarning       | 当长时间未收到报文时，该方法被调用    |
|          | CShfeFtdcMduserSpi::OnPackageStart           | 报文回调开始通知             |
|          | CShfeFtdcMduserSpi::OnPackageEnd             | 报文回调结束通知             |
| 灾备接口     | CShfeFtdcMduserSpi::OnRtnFlowMessageCancel   | 数据流回退通知              |

## 1.2. 业务接口

| 业务类型 | 业务     | 请求接口 / 响应接口                                                                              | 数据流 |
|------|--------|------------------------------------------------------------------------------------------|-----|
| 登录   | 登录     | CShfeFtdcMduserApi::ReqUserLogin<br>CShfeFtdcMduserSpi::OnRspUserLogin                   | 对话流 |
|      | 登出     | CShfeFtdcMduserApi::ReqUserLogout<br>CShfeFtdcMduserSpi::OnRspUserLogout                 | 对话流 |
|      | 修改用户口令 | CShfeFtdcMduserApi::ReqUserPasswordUpdate<br>CShfeFtdcMduserSpi::OnRspUserPasswordUpdate | 对话流 |
| 订阅   | 订阅主题   | CShfeFtdcMduserApi::ReqSubscribeTopic                                                    | 对话流 |

| 业务类型 | 业务   | 请求接口 / 响应接口                                                          | 数据流 |
|------|------|----------------------------------------------------------------------|-----|
|      |      | CShfeFtdcMduserSpi::OnRspSubscribeTopic                              |     |
|      | 查询主题 | CShfeFtdcMduserApi::ReqQryTopic<br>CShfeFtdcMduserSpi::OnRspQryTopic | 查询流 |
| 行情   | 行情通知 | CShfeFtdcMduserSpi::OnRtnDepthMarketData                             | 行情流 |
| 错误应答 | 错误应答 | CShfeFtdcMduserSpi::OnRspError                                       | 对话流 |

## 2. MduserAPI 接口说明

### 2.1. CShfeFtdcMduserSpi 接口

CShfeFtdcMduserSpi 实现了事件通知接口。用户必须派生 CShfeFtdcMduserSpi 接口，编写事件处理方法来处理需要的事件。

#### 2.1.1. OnFrontConnected 方法

当行情接收系统与交易系统建立起 TCP 虚链路（连接）后，该方法被调用。该连接是由 API 自动建立的。

函数原形：

```
void OnFrontConnected();
```

注意：OnFrontConnected 被调用仅说明 TCP 连接成功，行情接收系统必须自行登录，才能进行后续的业务操作。

#### 2.1.2. OnFrontDisconnected 方法

当行情接收系统与交易系统通信连接断开时，该方法被调用。当发生这个情况后，API 会自动重新连接。自动重连地址，可能是原来注册的地址，也可能是系统支持的其它可用的通信地址，它由程序自动选择。

函数原形：

```
void OnFrontDisconnected(int nReason);
```

参数：

**nReason:** 连接断开原因：

- 0x1001，表示网络读失败；
- 0x1002，表示网络写失败；

- 0x2001, 表示接收心跳超时;
- 0x2002, 表示报文加密失败;
- 0x2003, 表示报文解密失败;
- 0x2004, 表示收到未订阅主题的报文;
- 0x2005, 表示收到的报文序号不连续;
- 0x2006, 表示报文长度不合法;
- 0x2007, 表示报文转换错误;
- 0x2008, 表示登录错误前置服务。

### 2.1.3. OnHeartBeatWarning 方法

心跳超时警告。当长时间未收到报文时, 该方法被调用。缺省超时告警时间设置为 5 秒。如果调用过 `SetHeartbeatTimeout(unsigned int timeout)` 自行设定心跳超时, 则警告时间为 `timeout/2`。

函数原形:

```
void OnHeartBeatWarning(int nTimeLapse);
```

参数:

**nTimeLapse:** 距离上次接收报文的时间

### 2.1.4. OnPackageStart 方法

报文回调开始通知。当 API 收到一个报文后, 如果该报文属于行情流, 则调用本方法, 然后是各数据域的回调, 最后是报文回调结束通知。

函数原形:

```
void OnPackageStart(int nTopicID, int nSequenceNo);
```

参数:

**nTopicID:** 主题代码 (如行情主题 1001)。

**nSequenceNo:** 报文序号。

### 2.1.5. OnPackageEnd 方法

报文回调结束通知。当 API 收到一个报文后，如果该报文属于行情流，则调用报文回调开始通知，然后是各数据域的回调，最后调用本方法。

函数原形：

```
void OnPackageEnd(int nTopicID, int nSequenceNo);
```

参数：

**nTopicID**：主题代码（如行情主题 1001）。

**nSequenceNo**：报文序号。

### 2.1.6. OnRspUserLogin 方法

当行情接收系统发出登录请求之后，交易系统返回响应时，该方法会被调用，通知行情接收系统登录是否成功。

函数原形：

```
void OnRspUserLogin(
    CShfeFtdcRspUserLoginField* pRspUserLogin,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

**pRspUserLogin**：返回用户登录信息的地址。

用户登录信息结构：

```
struct CShfeFtdcRspUserLoginField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///登录成功时间
    TShfeFtdcTimeType   LoginTime;
    ///最大本地报单号，未使用
    TShfeFtdcOrderLocalIDType   MaxOrderLocalID;
    ///交易用户代码
    TShfeFtdcUserIDType   UserID;
```



```

///会员代码
TShfeFtdcParticipantIDType ParticipantID;
///交易系统名称
TShfeFtdcTradingSystemNameType TradingSystemName;
///数据中心代码
TShfeFtdcDataCenterIDType DataCenterID;
///会员私有流当前长度，未使用
TShfeFtdcSequenceNoType PrivateFlowSize;
///交易员私有流当前长度，未使用
TShfeFtdcSequenceNoType UserFlowSize;
///业务发生日期
TShfeFtdcDateType ActionDay;};

```

**pRspInfo:** 返回用户响应信息的地址。错误代码为 0 时，表示操作成功，以下同。响应信息结构：

```

struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType ErrorMsg;
};

```

可能出现的错误：

**错误代码 错误提示**

|     |             |
|-----|-------------|
| 3   | 会员找不到       |
| 45  | 结算组初始化状态不对  |
| 59  | 用户重复登录      |
| 60  | 用户名或口令错误    |
| 62  | 用户不活跃       |
| 64  | 用户不属于此会员    |
| 65  | 错误的登录 IP 地址 |
| 75  | 前置不活跃       |
| 106 | 重复的会话       |
| 135 | 用户认证失败      |
| 136 | 用户没有权限直连前置  |

**可能的原因**

|                        |
|------------------------|
| 登录的会员代码错误              |
| 交易系统尚未初始化完毕，可以稍后再重试    |
| 不同 IP 地址的相同用户重复登录      |
| 登录的用户代码或口令错误           |
| 交易系统不允许该用户登录           |
| 登录的会员代码错误              |
| 发起登录的计算机没有交易所允许的 IP 地址 |
| 交易系统前置不活跃              |
| 相同会话多次重复登录             |
| 用户密钥校验失败               |
| 用户没有权限直连前置             |

**nRequestID:** 返回用户登录请求的 ID，该 ID 由用户在登录时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.7. OnRspUserLogout 方法

当行情接收系统发出登出请求之后，交易系统返回响应时，该方法会被调

用，通知行情接收系统登出是否成功。

函数原形:

```
void OnRspUserLogout(  
    CShfeFtdcRspUserLogoutField* pRspUserLogout,  
    CShfeFtdcRspInfoField* pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

**pRspUserLogout:** 返回用户登出信息的地址。用户登出信息结构:

```
struct CShfeFtdcRspUserLogoutField {  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
};
```

**pRspInfo:** 返回用户响应信息的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示       | 可能的原因               |
|------|------------|---------------------|
| 45   | 结算组初始化状态不对 | 交易系统尚未初始化完毕，可以稍后再重试 |
| 66   | 用户尚未登录     | 尚未登录                |
| 67   | 并没有以此用户登录  | 登出时的用户和登录时不同        |
| 68   | 并没有以此会员登录  | 登出时的会员和登录时不同        |

**nRequestID:** 返回用户登出请求的 ID，该 ID 由用户在登出时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

2.1.8. OnRspSubscribeTopic 方法

订阅主题应答。当行情接收系统发出订阅主题指令后，交易系统返回响应时，该方法会被调用。

函数原形:

```
void OnRspSubscribeTopic(
    CShfeFtdcDisseminationField* pDissemination,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

**pDissemination:** 指向订阅主题结构的地址, 包含了要订阅的主题和起始报文的序号。订阅主题结构:

```
struct CShfeFtdcDisseminationField {
    ///序列系列号
    TShfeFtdcSequenceSeriesType    SequenceSeries;
    ///序列号
    TShfeFtdcSequenceNoType    SequenceNo;
};
```

**pRspInfo:** 指向响应信息结构的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};
```

可能出现的错误:

| 错误代码 | 错误提示        | 可能的原因                  |
|------|-------------|------------------------|
| 1    | 无效的会话或主题不存在 | 订阅的主题不存在或用户没有相应主题的订阅权限 |

**nRequestID:** 返回用户订阅主题请求的 ID, 该 ID 由用户在订阅主题时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

2.1.9. OnRspQryTopic 方法

查询主题应答。当行情接收系统发出查询主题指令后, 交易系统返回响应时, 该方法会被调用。

函数原形:

```
void OnRspQryTopic(
    CShfeFtdcDisseminationField* pDissemination,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

**参数：**

**pDissemination：**指向查询主题结构的地址，包含了要查询的主题和该主题报文个数。查询主题结构：

```
struct CShfeFtdcDisseminationField {
    ///序列系列号
    TShfeFtdcSequenceSeriesType    SequenceSeries;
    ///序列号
    TShfeFtdcSequenceNoType    SequenceNo;
};
```

**pRspInfo：**指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};
```

**nRequestID：**返回用户查询主题请求的 ID，该 ID 由用户在订阅主题时指定。

**bIsLast：**指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.1.10.OnRspError 方法

针对用户请求的出错通知。

**函数原形：**

```
void OnRspError(
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

**参数：**

**pRspInfo:** 返回用户响应信息的地址。响应信息结构:

```
struct CShfeFtdcRspInfoField {  
    ///错误代码  
    TShfeFtdcErrorIDType    ErrorID;  
    ///错误信息  
    TShfeFtdcErrorMsgType    ErrorMsg;  
};
```

可能出现的错误:

| 错误代码 | 错误提示                       | 可能的原因          |
|------|----------------------------|----------------|
| 1    | Not Login                  | 尚未登录           |
|      | Too High FTD Version       | FTD 版本太高       |
|      | Unrecognized ftd tid       | FTD 报文头错误      |
| 151  | 版本校验未通过                    | 行情 API 版本校验未通过 |
| 997  | api authentication failure | 非法的 API 接入     |
|      | api crypt info failure     | 未完成查询加密信息请求    |

**nRequestID:** 返回用户操作请求的 ID，该 ID 由用户在操作请求时指定。

**bIsLast:** 指示该次返回是否为针对 nRequestID 的最后一次返回。

### 2.1.11.OnRtnDepthMarketData 方法

行情通知。当行情发生变化时，交易系统会通知行情接收系统，该方法会被调用。

函数原形:

```
void OnRtnDepthMarketData(  
    CShfeFtdcDepthMarketDataField* pDepthMarketData);
```

参数:

**pDepthMarketData:** 指向行情结构的地址。注意：一档行情中该结构的部分字段（买二-买五，卖二-卖五）是无意义的。行情结构:

```
struct CShfeFtdcDepthMarketDataField  
{  
    ///交易日  
    TShfeFtdcDateType    TradingDay;  
    ///结算组代码  
    TShfeFtdcSettlementGroupIDType    SettlementGroupID;  
    ///结算编号  
    TShfeFtdcSettlementIDType SettlementID;  
    ///最新价
```

```

TShfeFtdcPriceType LastPrice;
///昨结算
TShfeFtdcPriceType PreSettlementPrice;
///昨收盘
TShfeFtdcPriceType PreClosePrice;
///昨持仓量
TShfeFtdcLargeVolumeTypePreOpenInterest;
///今开盘
TShfeFtdcPriceType OpenPrice;
///最高价
TShfeFtdcPriceType HighestPrice;
///最低价
TShfeFtdcPriceType LowestPrice;
///数量
TShfeFtdcVolumeTypeVolume;
///成交金额
TShfeFtdcMoneyType Turnover;
///持仓量
TShfeFtdcLargeVolumeTypeOpenInterest;
///今收盘
TShfeFtdcPriceType ClosePrice;
///今结算
TShfeFtdcPriceType SettlementPrice;
///涨停板价
TShfeFtdcPriceType UpperLimitPrice;
///跌停板价
TShfeFtdcPriceType LowerLimitPrice;
///昨虚实度
TShfeFtdcRatioType PreDelta;
///今虚实度
TShfeFtdcRatioType CurrDelta;
///最后修改时间
TShfeFtdcTimeType UpdateTime;
///最后修改毫秒
TShfeFtdcMillisecType UpdateMillisec;
///合约代码
TShfeFtdcInstrumentIDTypeInstrumentID;
///申买价一
TShfeFtdcPriceType BidPrice1;
///申买量一
TShfeFtdcVolumeTypeBidVolume1;
///申卖价一
TShfeFtdcPriceType AskPrice1;
///申卖量一

```

```

TShfeFtdcVolumeTypeAskVolume1;
///申买价二
TShfeFtdcPriceType BidPrice2;
///申买量二
TShfeFtdcVolumeTypeBidVolume2;
///申卖价二
TShfeFtdcPriceType AskPrice2;
///申卖量二
TShfeFtdcVolumeTypeAskVolume2;
///申买价三
TShfeFtdcPriceType BidPrice3;
///申买量三
TShfeFtdcVolumeTypeBidVolume3;
///申卖价三
TShfeFtdcPriceType AskPrice3;
///申卖量三
TShfeFtdcVolumeTypeAskVolume3;
///申买价四
TShfeFtdcPriceType BidPrice4;
///申买量四
TShfeFtdcVolumeTypeBidVolume4;
///申卖价四
TShfeFtdcPriceType AskPrice4;
///申卖量四
TShfeFtdcVolumeTypeAskVolume4;
///申买价五
TShfeFtdcPriceType BidPrice5;
///申买量五
TShfeFtdcVolumeTypeBidVolume5;
///申卖价五
TShfeFtdcPriceType AskPrice5;
///申卖量五
TShfeFtdcVolumeTypeAskVolume5;
///业务发生日期
TShfeFtdcDateType ActionDay;
};

```

## 2.1.12.OnRtnFlowMessageCancel 方法

数据流回退通知。由交易系统发生灾备切换之后，用户重新登录交易系统并订阅某个数据流时，交易系统主动通知行情接收系统该数据流某些报文被取

消了，该方法会被调用。

### 函数原形：

```
void OnRtnFlowMessageCancel(
    CShfeFtdcFlowMessageCancelField* pFlowMessageCancel);
```

### 参数：

**pFlowMessageCancel:** 指向数据流回退结构的地址。数据流回退结构：

```
struct CShfeFtdcFlowMessageCancelField
{
    ///序列系列号
    TShfeFtdcSequenceSeriesType    SequenceSeries;
    ///交易日
    TShfeFtdcDateType    TradingDay;
    ///数据中心代码
    TShfeFtdcDataCenterIDType    DataCenterID;
    ///回退起始序列号
    TShfeFtdcSequenceNoType    StartSequenceNo;
    ///回退结束序列号
    TShfeFtdcSequenceNoType    EndSequenceNo;
};
SequenceSeries: 发生回退的数据流代码（行情流）
回退的报文区间为：（StartSequenceNo, EndSequenceNo]
```

## 2.1.13.OnRspUserPasswordUpdate 方法

用户密码修改应答。当行情接收系统发出用户密码修改指令后，交易系统返回响应时，该方法会被调用。

### 函数原形：

```
void OnRspUserPasswordUpdate(
    CShfeFtdcUserPasswordUpdateField* pUserPasswordUpdate,
    CShfeFtdcRspInfoField* pRspInfo,
    int nRequestID,
    bool bIsLast);
```

### 参数：

**pUserPasswordUpdate:** 指向用户密码修改结构的地址，包含了用户密码



修改请求的输入数据。用户密码修改结构：

```
struct CShfeFtdcUserPasswordUpdateField {
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///旧密码
    TShfeFtdcPasswordType    OldPassword;
    ///新密码
    TShfeFtdcPasswordType    NewPassword;
};
```

**pRspInfo**：指向响应信息结构的地址。响应信息结构：

```
struct CShfeFtdcRspInfoField {
    ///错误代码
    TShfeFtdcErrorIDType    ErrorID;
    ///错误信息
    TShfeFtdcErrorMsgType    ErrorMsg;
};
```

**nRequestID**：返回用户密码修改请求的 ID，该 ID 由用户在密码修改时指定。

**bIsLast**：指示该次返回是否为针对 nRequestID 的最后一次返回。

## 2.2. CShfeFtdcMduserApi 接口

CShfeFtdcMduserApi 接口提供给用户的功能包括，登入/登出、行情订阅等功能。

### 2.2.1. CreateFtdcMduserApi 方法

产生一个 CShfeFtdcMduserApi 的一个实例，不能通过 new 来产生。

函数原形：

```
static CShfeFtdcMduserApi* CreateFtdcMduserApi(const char* pszFlowPath = "");
```

参数：

**pszFlowPath:** 常量字符指针, 用于指定一个文件目录来存贮交易系统发布消息的状态。默认值代表当前目录。

### 返回值:

返回一个指向 CShfeFtdcMduserApi 实例的指针。

## 2.2.2. GetVersion 方法

获取 API 版本号。

### 函数原形:

```
const char* GetVersion(int& nMajorVersion, int& nMinorVersion);
```

### 参数:

**nMajorVersion:** 返回主版本号

**nMajorVersion:** 返回子版本号

### 返回值:

返回一个指向版本标识字符串的常量指针。

## 2.2.3. Release 方法

释放当前 API 实例的内部资源, 退出 API 工作线程和设置 API 退出信号(只设置退出信号, 不释放实例)。

### 函数原形:

```
int Release();
```

### 返回值:

- 0, 代表成功。
- -9, 表示未初始化。

### 2.2.4.Init 方法

使行情接收系统开始与交易系统建立连接，连接成功后可以进行登录。

函数原形：

```
int Init();
```

返回值：

- 0，代表成功。
- -5，表示已登录或重复调用。

### 2.2.5.Join 方法

阻塞 API 工作线程，当 API 退出信号被触发后，释放当前 API 实例。

函数原形：

```
int Join();
```

返回值：

- 0，代表成功。

### 2.2.6.GetTradingDay 方法

获得当前交易日。只有成功登录到交易系统之后才会取到正确的值。

函数原形：

```
const char* GetTradingDay();
```

返回值：

返回一个指向日期信息字符串的常量指针。

### 2.2.7.RegisterSpi 方法

注册一个派生自 CShfeFtdcMduserSpi 接口类的实例，该实例将完成事件处

理。

### 函数原形:

```
void RegisterSpi(CShfeFtdcMduserSpi* pSpi);
```

### 参数:

**pSpi:** 实现了 CShfeFtdcMduserSpi 接口的实例指针。

## 2.2.8. RegisterFront 方法

设置行情前置的网络通讯地址。交易系统拥有多个行情前置，用户可以同时注册多个行情前置的网络通讯地址。

### 函数原形:

```
int RegisterFront(const char* pszFrontAddress);
```

### 参数:

**pszFrontAddress:** 指向行情前置网络通讯地址的指针。服务器地址的格式为：“protocol://ipaddress:port”，如：“tcp://127.0.0.1:17001”。“tcp”代表传输协议，“127.0.0.1”代表服务器地址。“17001”代表服务器端口号。

### 返回值:

- 0，代表成功。
- -8，表示注册前置网络地址个数超过最大值；
- -10，表示已初始化。

## 2.2.9. RegisterNameServer 方法

设置交易系统 FENS 服务的网络通讯地址。交易系统拥有多个 FENS 服务，用户可以同时注册多个 FENS 服务的网络通讯地址。

### 函数原形:

```
int RegisterNameServer (const char* pszNsAddress);
```

**参数：**

**pszNsAddress:** 指向交易系统 FENS 服务网络通讯地址的指针。网络地址的格式为：“protocol://ipaddress:port”，如：“tcp://127.0.0.1:17001”。“tcp”代表传输协议，“127.0.0.1”代表服务器地址。“17001”代表服务器端口号。

**返回值：**

- 0，代表成功。
- -8，表示注册 FENS 服务网络地址个数超过最大值；
- -10，表示已初始化。

**2.2.10.SetHeartbeatTimeout 方法**

设置网络通信心跳的超时时间。当 MduserAPI 与交易系统的 TCP 连接建立后，连接会定时发送心跳，用以检测连接是否正常。该方法用于设置检测心跳超时的时间。交易所建议行情接收系统将 timeout 值设置为 10 至 30 秒之间。

**函数原形：**

```
int SetHeartbeatTimeout(unsigned int timeout);
```

**参数：**

**timeout:** 心跳超时时间（秒）。若超过 timeout/2 秒未收到交易系统的任何信息时，将触发回调 OnHeartBeatWarning。若超过 timeout 秒未收到交易系统的任何信息时，连接将会中断，将触发回调 OnFrontDisconnected。

**返回值：**

- 0，代表成功。
- -10，表示已初始化。

**2.2.11.OpenRequestLog 方法**

打开请求日志文件。调用该方法之后，向交易系统发出的请求信息将记录在指定的日志文件中。

函数原形:

```
int OpenRequestLog(const char* pszReqLogFileName);
```

参数:

**pszReqLogFileName:** 请求日志文件名。

返回值:

- 0, 代表成功。
- -4, 表示打开日志文件失败;

## 2.2.12.OpenResponseLog 方法

打开应答日志文件。调用该方法之后, 交易系统返回的信息将记录在指定的日志文件中, 包括应答信息和回报信息。

函数原形:

```
int OpenResponseLog(const char* pszRspLogFileName);
```

参数:

**pszRspLogFileName:** 应答日志文件名。

返回值:

- 0, 代表成功。
- -4, 表示打开日志文件失败;

## 2.2.13.SubscribeMarketDataTopic 方法

订阅行情。订阅后交易系统会主动发出行情通知给行情接收系统。

函数原形:

```
int SubscribeMarketDataTopic(int nTopicID, TE_RESUME_TYPE nResumeType);
```

参数:

**nTopicID:** 要订阅的行情主题, 由交易所公布。

**nResumeType:** 行情重传方式:

- TERT\_RESTART: 从本交易日开始重传;
- TERT\_RESUME: 从上次收到的续传;
- TERT\_QUICK: 先传送当前行情快照,再传送之后的行情。交易所建议会员使用此方式快速恢复行情。

返回值:

- 0, 代表成功。
- -8, 表示订阅市场行情主题个数超过最大值;
- -10, 表示已初始化。

## 2.2.14.ReqUserLogin 方法

用户登录请求。

函数原形:

```
int ReqUserLogin(
    CShfeFtdcReqUserLoginField* pReqUserLoginField,
    int nRequestID);
```

参数:

**pReqUserLoginField:** 指向用户登录请求结构的地址。用户登录请求结构:

```
struct CShfeFtdcReqUserLoginField {
    ///交易日
    TShfeFtdcDateType   TradingDay;
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
    ///密码
    TShfeFtdcPasswordType Password;
    ///用户端产品信息
    TShfeFtdcProductInfoType UserProductInfo;
    ///接口端产品信息
    TShfeFtdcProductInfoType InterfaceProductInfo;
    ///协议信息
    TShfeFtdcProtocolInfoType ProtocolInfo;
```

```
///数据中心代码
```

```
TShfeFtdcDataCenterIDType    DataCenterID;
```

```
};
```

用户需要填写 **UserProductInfo** 字段，即行情接收系统的产品信息，如软件开发商、版本号等，例如：“SFIT Mduser V100”代表技术公司开发的行情接收程序和版本号。

**nRequestID**：用户登录请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -2，表示超出行情在途流控；
- -3，表示超出行情请求流控；
- -5，表示已登录；
- -6，表示强制要求填写的字段为空（**UserProductInfo** 未填写）；
- -9，表示未初始化；
- -12，尚未与前置建立连接。

## 2.2.15. ReqUserLogout 方法

用户登出请求。

函数原形：

```
int ReqUserLogout(
    CShfeFtdcReqUserLogoutField* pReqUserLogout,
    int nRequestID);
```

参数：

**pReqUserLogout**：指向用户登出请求结构的地址。用户登出请求结构：

```
struct CShfeFtdcReqUserLogoutField {
    ///交易用户代码
    TShfeFtdcUserIDType UserID;
    ///会员代码
    TShfeFtdcParticipantIDType ParticipantID;
};
```

**nRequestID**：用户登出请求的 ID，该 ID 由用户指定，管理。

返回值：



- 0，代表成功。
- -1，表示未登录；
- -2，表示超出行情在途流控；
- -3，表示超出行情请求流控。

## 2.2.16. ReqSubscribeTopic 方法

订阅主题请求。

函数原形：

```
int ReqSubscribeTopic(
    CShfeFtdcDisseminationField* pDissemination,
    int nRequestID);
```

参数：

**pDissemination**：指向订阅主题结构的地址，包含了要订阅的主题和起始报文的序号。订阅主题结构：

```
struct CShfeFtdcDisseminationField {
    ///序列系列号
    TShfeFtdcSequenceSeriesType  SequenceSeries;
    ///序列号
    TShfeFtdcSequenceNoType  SequenceNo;
};
SequenceSeries: 要订阅的主题
SequenceNo: <0 表示使用快照方式，其它值表示从该序号开始续传
```

**nRequestID**：用户订阅主题请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出行情在途流控；
- -3，表示超出行情请求流控；
- -8，表示订阅行情主题个数超出限制。

## 2.2.17. ReqQryTopic 方法

查询主题请求。

函数原形：

```
int ReqQryTopic(  
    CShfeFtdcDisseminationField* pDissemination,  
    int nRequestID);
```

参数：

**pDissemination**：指向查询主题结构的地址，包含了要查询的主题。订阅主题结构：

```
struct CShfeFtdcDisseminationField {  
    ///序列系列号，填写要查询的主题号  
    TShfeFtdcSequenceSeriesType    SequenceSeries;  
    ///序列号，未使用  
    TShfeFtdcSequenceNoType    SequenceNo;  
};
```

**nRequestID**：用户查询主题请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出行情在途流控；
- -3，表示超出行情请求流控。

## 2.2.18. ReqUserPasswordUpdate 方法

用户密码修改请求。

函数原形：

```
int ReqUserPasswordUpdate(  
    CShfeFtdcUserPasswordUpdateField* pUserPasswordUpdate,  
    int nRequestID);
```

参数：

**pUserPasswordUpdate:** 指向用户密码修改结构的地址，用户密码修改结构：

```
struct CShfeFtdcUserPasswordUpdateField {  
    ///交易用户代码  
    TShfeFtdcUserIDType UserID;  
    ///会员代码  
    TShfeFtdcParticipantIDType ParticipantID;  
    ///旧密码  
    TShfeFtdcPasswordType    OldPassword;  
    ///新密码  
    TShfeFtdcPasswordType    NewPassword;  
};
```

**nRequestID:** 用户密码修改请求的 ID，该 ID 由用户指定，管理。

返回值：

- 0，代表成功。
- -1，表示未登录；
- -2，表示超出行情在途流控；
- -3，表示超出行情请求流控；
- -13，表示会员号不一致；
- -14，表示用户号不一致。

业务说明：

当前版本暂不支持该功能。

### 3. MduserAPI 接口开发示例

```
// mdusertest.cpp :
// 一个简单的例子，介绍CShfeFtdcMduserApi和CShfeFtdcMduserSpi接口的使用。
// 当行情字段值为DBL_MAX（1.7976931348623157e+308）时，该字段实际为空
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <float.h>
#include "FtdcMduserApi.h"

class CSimpleHandler : public CShfeFtdcMduserSpi
{
public:
    // 构造函数，需要一个有效的指向CShfeFtdcMduserApi实例的指针
    CSimpleHandler(CShfeFtdcMduserApi *pMduserApi) : m_pMduserApi(pMduserApi) {}

    ~CSimpleHandler() {}

    // 当行情接收系统与交易系统建立起通信连接后，需要进行登录
    void OnFrontConnected()
    {
        CShfeFtdcReqUserLoginField reqUserLogin;
        memset(&reqUserLogin, 0, sizeof(reqUserLogin));
        strcpy(reqUserLogin.ParticipantID, "0888");
        strcpy(reqUserLogin.UserID, "0888c1c");
        strcpy(reqUserLogin.Password, "1");
        strcpy(reqUserLogin.UserProductInfo, "TestMduserAPI V2.00");
        // 发出登录请求
        int ret = m_pMduserApi->ReqUserLogin(&reqUserLogin, 0);
        if (ret != 0)
        {
            printf("ReqUserLogin Fail ret = %d\n", ret);
        }
    }

    // 当行情接收系统与交易系统通信连接断开时，该方法被调用
    void OnFrontDisconnected(int nReason) {
        // 当发生这个情况后，API会自动重新连接，可不做处理
        printf("OnFrontDisconnected Reason = %#x.\n", nReason);
    }
}
```

```

// 当交易系统返回登录应答之后，该方法会被调用，通知登录是否成功
void OnRspUserLogin(CShfeFtdcRspUserLoginField *pRspUserLogin, CShfeFtdcRspInfoField *pRspInfo,
int nRequestID, bool bIsLast) {
    printf("OnRspUserLogin: ErrorCode=[%d], ErrorMsg=[%s]\n",
pRspInfo->ErrorID, pRspInfo->ErrorMsg);
    printf("RequestID=[%d], Chain=[%d]\n", nRequestID, bIsLast);
    if (pRspInfo->ErrorID != 0) {
        // 登录失败，需进行错误处理
        printf("Failed to login, errorcode=%d errormsg=%s requestid=%d chain=%d",
pRspInfo->ErrorID, pRspInfo->ErrorMsg, nRequestID, bIsLast);
    }
}

// 深度行情通知，交易系统会主动通知
void OnRtnDepthMarketData(CShfeFtdcDepthMarketDataField *pMarketData) {
    // 行情接收系统按需处理返回的数据
    if(pMarketData->OpenPrice!=DBL_MAX)
    {
        printf("OpenPrice=%.2f\n",pMarketData->OpenPrice);
    }
}

// 针对用户请求的出错通知
void OnRspError(CShfeFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {
    printf("OnRspError:\n");
    printf("ErrorCode=[%d], ErrorMsg=[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
    printf("RequestID=[%d], Chain=[%d]\n", nRequestID, bIsLast);
    // 行情接收系统需进行错误处理
}

private:
    // 指向CShfeFtdcMduserApi实例的指针
    CShfeFtdcMduserApi *m_pMduserApi;
};

int main()
{
    // 产生一个CShfeFtdcMduserApi实例
    CShfeFtdcMduserApi *pMduserApi = CShfeFtdcMduserApi::CreateFtdcMduserApi();
    // 产生一个事件处理的实例
    CSimpleHandler sh(pMduserApi);
    // 注册一事件处理的实例
    pMduserApi->RegisterSpi(&sh);
    // 注册需要的深度行情主题

```

```
pMduserApi->SubscribeMarketDataTopic (1001, TERT_RESUME);  
//设置心跳超时时间  
pMduserApi->SetHeartbeatTimeout(19);  
// 设置交易系统行情前置NameServer的地址  
pMduserApi->RegisterNameServer("tcp://192.168.1.1:17011");  
  
// 开始与交易系统行情前置建立连接  
pMduserApi->Init();  
// 等待API实例退出  
pMduserApi->Join();  
  
return 0;  
}
```

## 第四部分 附录

# 1. 错误编码列表

| 错误编号 | 错误提示                 | 错误原因                                    |
|------|----------------------|-----------------------------------------|
| -1   | 认证失败                 | 找不到交易终端对应的授权码或授权码不匹配                    |
| 1    | 无效的会话或主题不存在          | 订阅的主题不存在、订阅的主题数超过了上限或用户没有相应主题的订阅权限      |
|      | Not Login            | 用户尚未登录                                  |
|      | Too High FTD Version | FTD 版本太高                                |
|      | Unrecognized ftd tid | FTD 报文头错误                               |
| 2    | 合约找不到                | 在各项操作中发现合约找不到                           |
| 3    | 会员找不到                | 在各项操作中发现会员找不到                           |
| 4    | 客户找不到                | 在各项操作中发现客户找不到                           |
| 6    | 报单字段错误               | 插入报单时，发现报单中有字段值不合法（枚举值越界）               |
|      |                      | 插入报单时，发现非强平单中设置了强平原因                    |
| 7    | 报价字段错误               | 插入报价时，发现报价中有字段值不合法（枚举值越界）               |
| 8    | 报单操作字段错误             | 报单操作时，发现报单操作中有字段值不合法（枚举值越界）             |
|      |                      | 报价操作衍生的报单中有字段值不合法（例如：价格不是浮点数或不在有效范围内）   |
| 9    | 报价操作字段错误             | 报价操作时，发现报价操作中有字段值不合法（枚举值越界或操作标志为修改或挂起）  |
| 12   | 重复的报单                | 插入报单时，本地报单号重复                           |
| 13   | 重复的报价                | 插入报价时，本地报价号重复                           |
| 15   | 客户没有在该会员开户           | 在各项操作时，发现指定的客户并没有在指定的会员中开户              |
| 16   | IOC 需在连续交易阶段         | 在非连续交易阶段插入 IOC 单                        |
| 17   | GFA 需在集合竞价阶段         | 在非集合竞价阶段插入 GFA 的报单                      |
| 19   | 数量约束应在 IOC 单上        | 在插入数量约束不是任意数量的报单时，发现时间条件不是 IOC          |
| 20   | GTD 报单过期了            | 在插入 GTD 报单时，发现 GTD 日期已经过期了              |
| 21   | 最小数量大于报单数量           | 在插入有最小数量条件的报单时，发现最小数量大于报单数量             |
| 22   | 交易所数据没有同步            | 进行各类业务操作时，发现交易所的数据还没有同步                 |
| 23   | 结算组数据没有同步            | 进行各类业务操作时，发现结算组的数据还没有同步                 |
| 24   | 报单找不到                | 进行报单操作时，发现要操作的报单找不到                     |
| 25   | 报价找不到                | 进行报价操作时，发现要操作的报价找不到                     |
|      |                      | 在插入报单/报价时，发现合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡 |



|    |            |                                                                                     |
|----|------------|-------------------------------------------------------------------------------------|
| 26 | 当前状态禁止此项操作 | 在报单/报价操作时，<br>对于激活操作，发现合约交易状态不是连续交易、集合竞价报单或者集合竞价平衡<br>对于其他操作，发现合约交易状态不是连续交易或者集合竞价报单 |
|    |            | 在做执行宣告/放弃执行宣告的插入/操作、期权自对冲的更新/操作时，发现合约交易状态不是连续交易、交易业务处理                              |
|    |            | 在做询价录入时，发现合约不可交易或合约不处于连接交易状态                                                        |
| 28 | 报单已经全部成交   | 在报单/报价操作时，发现报单已经全部成交了                                                               |
| 29 | 报单已经撤销     | 在报单/报价操作时，发现报单已经撤销了                                                                 |
| 30 | 没有足够的数量供修改 | 修改报单数量后，剩余报单数量小于 0                                                                  |
| 31 | 平仓时客户持仓不足  | 在各类可能造成平仓的操作时，发现客户的持仓不足                                                             |
| 32 | 超出客户限仓     | 在各类可能开仓操作时，发现超过了客户的一般限仓                                                             |
| 33 | 平仓时会员持仓不足  | 在各类可能造成平仓的操作时，发现会员的持仓不足                                                             |
| 34 | 超出会员限仓     | 在各类可能开仓操作时，发现超过了会员限仓                                                                |
| 35 | 找不到帐号      | 在各类操作时，发现找不到该操作应当使用的帐号                                                              |
| 36 | 资金不足       | 在各类操作时，发现帐号中没有足够的资金                                                                 |
| 37 | 不合法的数量     | 在插入报单、报单操作、报价录入、执行宣告录入、放弃执行宣告录入、期权自对冲更新时，数量不是最小报单数量要求的正整数倍，或者超过最大报单数量               |
| 45 | 结算组初始化状态不对 | 交易系统尚未初始化完毕                                                                         |
| 48 | 价格非最小单位的倍数 | 在各类操作时，价格不是合约的最小变动单位的整数倍                                                            |
| 49 | 价格超出涨停板    | 在各类操作时，价格高于合约的涨停板                                                                   |
| 50 | 价格跌破跌停板    | 在各类操作时，价格低于合约的跌停板                                                                   |
| 51 | 没有交易权限     | 在进行各类操作时，发现会员对指定合约、或者客户对指定的合约、或者用户没有交易权限                                            |
| 52 | 只能平仓       | 在进行各类可能造成开仓的操作时，发现会员对指定合约、或者客户对指定的合约、或者用户只有平仓的权限                                    |
| 53 | 没有此项交易角色   | 在插入报单、插入报价、期权自对冲更新/操作时，该会员在指定合约上不具有该客户对应的交易角色                                       |
| 54 | 会话找不到      | 在各类操作中，发现会话找不到                                                                      |
| 57 | 不能为其他会员操作  | 在各项操作中，发现用户给非其所属的会员进行操作                                                             |
| 58 | 用户不匹配      | 在各项操作时，发现操作的用户和会话的用户不匹配                                                             |
| 59 | 用户重复登录     | 同一用户在不同 IP 地址重复登录                                                                   |
| 60 | 用户名或口令错误   | 在用户登录或者修改口令时，发现用户名找不到或者口令错误                                                         |
| 62 | 用户不活跃      | 在用户登录或者修改口令时，发现该用户没有登录、交易或改密码的权限                                                    |

|     |                 |                                       |
|-----|-----------------|---------------------------------------|
| 64  | 用户不属于此会员        | 在用户登录时，发现用户不属于指定的会员                   |
| 65  | 错误的登录 IP 地址     | 在用户登录时，发现用户的 IP 地址不合法                 |
| 66  | 用户尚未登录          | 在用户登出或者修改口令时，发现该用户尚未登录                |
| 67  | 并没有以此用户登录       | 在用户登出时，发现登出时的用户与登录时不同                 |
| 68  | 并没有以此会员登录       | 在用户登出或者修改口令时，发现用户并不是以此会员登录的           |
| 70  | 报价已经被取消         | 在报价操作时，发现报价已经被取消了                     |
| 71  | 不能操作衍生报单        | 在报单操作时，发现用户操作的是衍生报单                   |
| 72  | 不允许自然人开仓        | 交割月，自然人类型的客户发起开仓请求或对开仓类型的报单做激活或修改的操作  |
| 75  | 前置不活跃           | 交易系统前置不活跃                             |
| 76  | 报单已经被挂起         | 在报单挂起时，发现报单已经被挂起了                     |
| 77  | 报单已经被激活         | 在报单激活时，发现报单已经被激活了                     |
| 78  | GTD 报单没有设定日期    | 在插入 GTD 报单时，没有指定 GTD 日期               |
| 79  | 不被支持的报单类型       | 在插入各类报单时，发现目前本交易所还不支持此种报单类型           |
| 80  | 用户无此权限          | 在进行各类操作时，发现用户没有相应操作的权限                |
| 83  | 止损单仅用于连续交易      | 在非连续交易阶段企图插入/操作止损单                    |
| 84  | 止损单需是 IOC 或 GFD | 在插入止损单时，发现时间条件既不是 IOC，也不是 GFD         |
| 88  | 未找到要操作的用户       | 做询价操作时，找不到询价的用户                       |
| 89  | 执行宣告字段错误        | 插入/操作执行宣告时，发现执行宣告中有字段值不合法（枚举值越界）      |
| 90  | 执行宣告操作字段错误      | 执行宣告操作时，发现执行宣告操作中有字段值不合法（枚举值越界）       |
| 91  | 重复的执行宣告         | 插入执行宣告时，本地执行宣告编号重复                    |
| 92  | 执行宣告已经取消        | 在执行宣告操作时，发现执行宣告已经取消了                  |
| 93  | 执行宣告找不到         | 进行执行宣告操作时，发现要操作的执行宣告找不到               |
| 94  | 执行宣告只能用于期权      | 插入执行宣告时，发现合约是非期权合约                    |
| 95  | 止损报单需说明止损价      | 在插入/操作止损单时，发现没有指定止损价                  |
| 96  | 保值额度不足          | 在插入/操作套保报单、插入报价时，发现客户的套保额度不足          |
| 98  | 强平单需由管理员使用      | 非管理员用户报了强平单                           |
| 99  | 不能为其他用户操作       | 在报单/报价操作时，发现非授权用户企图操作同会员的其他用户插入的报单/报价 |
| 100 | 错误的用户类型         | 在用户登录时，发现用户类型是行情用户                    |
| 101 | 结算会员不能做交易       | 使用结算会员做交易相关的操作                        |
| 102 | 无法找到对应结算会       | 在进行各类操作时，发现找不到会员对应的结算会员               |

|     |                |                                 |
|-----|----------------|---------------------------------|
|     | 员              |                                 |
| 103 | 当日套保仓位不能平仓     | 企图对套保仓位进行插入平今仓报单                |
| 106 | 重复的会话          | 在同一个会话上发起两次登录                   |
| 114 | 最优价单不能排队       | 在插入最优价单时, 发现时间条件不是 IOC          |
| 121 | 放弃执行宣告字段错误     | 在插入/操作放弃执行宣告时, 发现放弃执行宣告中有字段不合法  |
| 122 | 放弃执行宣告操作字段错误   | 放弃执行宣告操作时, 发现放弃执行宣告操作中有字段不合法    |
| 123 | 重复的放弃执行宣告      | 插入放弃执行宣告时, 本地放弃执行宣告编号重复         |
| 124 | 放弃执行宣告已经取消     | 在放弃执行宣告操作时, 发现执行宣告已经取消了         |
| 125 | 放弃执行宣告找不到      | 进行放弃执行宣告操作时, 发现要操作的放弃执行宣告找不到    |
| 126 | 放弃执行宣告只能用于期权   | 插入放弃执行宣告时, 发现合约是非期权合约           |
| 127 | 不在宣告期内         | 插入执行宣告或者放弃执行宣告时, 发现不在限定的期限内     |
| 128 | 只有持有多头仓位才能放弃行权 | 期权卖方不允许放弃行权                     |
| 129 | 执行或放弃执行宣告不能为开仓 | 插入执行宣告或者放弃执行宣告时, 发现开平仓标记是开仓     |
| 131 | 超出客户合约当日开仓限额   | 客户在某个合约上当日开仓手数超出限额              |
| 132 | 超出客户产品每秒报单限额   | 客户在某个产品上一秒以内的报单笔数超出限额           |
| 133 | 超出客户产品每秒撤单限额   | 客户在某个产品上一秒以内的撤单笔数超出限额           |
| 134 | API 校验未通过      | 使用了非正式渠道发布的 API 库               |
| 135 | 用户认证失败         | 开发者软件未经过交易所审核认证                 |
| 136 | 用户没有权限直连前置     | 只能通过 Fens 服务器获得前置地址的用户使用了直接连接模式 |
| 137 | 期权自对冲字段错误      | 期权自对冲更新中有字段值不合法 (枚举值越界)         |
| 138 | 期权自对冲操作字段错误    | 期权自对冲操作中有字段值不合法 (枚举值越界)         |
| 139 | 重复的期权自对冲更新     | 期权自对冲更新中的本地期权自对冲编号重复            |
| 140 | 期权自对冲更新已经撤销    | 要操作的期权自对冲更新已经取消                 |
| 141 | 期权自对冲更新只能用于期权  | 期权自对冲更新中的合约是非期权合约               |
| 142 | 期权自对冲找不到       | 要操作的期权自对冲没有找到                   |
| 143 | 期权自对冲操作只能是删除   | 期权自对冲操作的操作类型错误                  |

|      |                                        |                                                |
|------|----------------------------------------|------------------------------------------------|
| 144  | 此客户 SelfCloseFlag 不能为保留期权仓位            | 期权自对冲更新中的 SelfCloseFlag 和客户类型不匹配               |
| 145  | 此客户 SelfCloseFlag 不能为自对冲期权仓位           | 期权自对冲更新中的 SelfCloseFlag 和客户类型不匹配               |
| 146  | 只有持有多头仓位才能行权                           | 只有期权买方才能提交执行宣告录入请求                             |
| 147  | 用户新密码不符合要求（长度不低于 8 位，必须包含数字、大写字母和小写字母） | 用户修改新密码，必须要符合要求，即长度不低于 8 位，必须包含数字、大写字母和小写字母。   |
| 148  | 当前市场价格在合理的价差范围内，不用询价。                  | 当前市场价格如果在合理的价差范围内，客户的询价是不会发送给做市商的，即客户的询价是无意义的。 |
| 149  | 放弃执行宣告申请只能在期权到期日提交                     | 只有在期权的到期日，才能提交对应期权的放弃执行宣告事件。                   |
| 150  | 自营会员登陆前未认证或认证失败                        | 自营会员只有在终端信息经过认证后才能进行登录。                        |
| 151  | 版本校验未通过                                | API 版本校验未通过                                    |
| 153  | 市价单必须为 GFD 或者 IOC 单                    | 插入市价单时，发现有效期类型既不是 IOC 也不是 GFD                  |
| 154  | 市价单需在连续交易阶段                            | 插入市价单时，发现合约状态此时不在连续交易阶段                        |
| 155  | 市价单只支持期货期权                             | 插入市价单时，发现产品类型不是期货或期权                           |
| 997  | api authentication failure             | 非法的 API 接入                                     |
|      | api crypt info failure                 | API 加密信息查询失败                                   |
| 998  | query frequency is too high            | 发送的查询频率太高                                      |
| 999  | the last query result is on way        | 有未发送完的查询响应数据                                   |
| 1005 | 无记录                                    | 在各类操作时，发现合约所对应的记录缺失                            |

## 2. 枚举值列表

| 序号 | 枚举描述   | 枚举前缀 | 枚举名称             | 编码描述     | 编码名称                  | 编码数值 |
|----|--------|------|------------------|----------|-----------------------|------|
| 1  | 交易角色   | ER   | TradingRole      | 代理       | Broker                | 1    |
|    |        |      |                  | 自营       | Host                  | 2    |
| 2  | 产品类型   | PC   | ProductClass     | 期货       | Futures               | 1    |
|    |        |      |                  | 期权       | Options               | 2    |
|    |        |      |                  | 组合       | Combination           | 3    |
|    |        |      |                  | 即期       | Spot                  | 4    |
|    |        |      |                  | 期转现      | EFP                   | 5    |
|    |        |      |                  | 结算价交易    | TAS                   | 6    |
|    |        |      |                  | 套利       | Spread                | 7    |
| 3  | 期权类型   | OT   | OptionsType      | 非期权      | NotOptions            | 0    |
|    |        |      |                  | 看涨       | CallOptions           | 1    |
|    |        |      |                  | 看跌       | PutOptions            | 2    |
| 4  | 合约交易状态 | IS   | InstrumentStatus | 开盘前      | BeforeTrading         | 0    |
|    |        |      |                  | 非交易      | NoTrading             | 1    |
|    |        |      |                  | 连续交易     | Continuous            | 2    |
|    |        |      |                  | 集合竞价报单   | AuctionOrdering       | 3    |
|    |        |      |                  | 集合竞价价格平衡 | AuctionBalance        | 4    |
|    |        |      |                  | 集合竞价撮合   | AuctionMatch          | 5    |
|    |        |      |                  | 收盘       | Closed                | 6    |
| 5  | 买卖方向   | D    | Direction        | 交易业务处理   | TransactionProcessing | 7    |
|    |        |      |                  | 买        | Buy                   | 0    |
| 6  | 持仓类型   | PT   | PositionType     | 卖        | Sell                  | 1    |
|    |        |      |                  | 净持仓      | Net                   | 1    |
| 7  | 持仓多空方向 | PD   | PosiDirection    | 综合持仓     | Gross                 | 2    |
|    |        |      |                  | 净        | Net                   | 1    |
|    |        |      |                  | 多头       | Long                  | 2    |
| 8  | 套保标志   | HF   | HedgeFlag        | 空头       | Short                 | 3    |
|    |        |      |                  | 一般       | Speculation           | 1    |
|    |        |      |                  | 套保       | Hedge                 | 3    |
| 9  | 客户类型   | CT   | ClientType       | 无        | None                  | N    |
|    |        |      |                  | 自然人      | Person                | 0    |
|    |        |      |                  | 法人       | Company               | 1    |

| 序号 | 枚举描述       | 枚举前缀 | 枚举名称                  | 编码描述      | 编码名称                    | 编码数值 |
|----|------------|------|-----------------------|-----------|-------------------------|------|
|    |            |      |                       | 投资基金      | Fund                    | 2    |
| 10 | 品种进入交易状态原因 | IER  | InstStatusEnterReason | 自动切换      | Automatic               | 1    |
|    |            |      |                       | 手动切换      | Manual                  | 2    |
|    |            |      |                       | 熔断        | Fuse                    | 3    |
|    |            |      |                       | 熔断手动      | FuseManual              | 4    |
| 11 | 报单价格条件     | OPT  | OrderPriceType        | 任意价       | AnyPrice                | 1    |
|    |            |      |                       | 限价        | LimitPrice              | 2    |
|    |            |      |                       | 最优价       | BestPrice               | 3    |
| 12 | 开平标志       | OF   | OffsetFlag            | 开仓        | Open                    | 0    |
|    |            |      |                       | 平仓        | Close                   | 1    |
|    |            |      |                       | 强平        | ForceClose              | 2    |
|    |            |      |                       | 平今        | CloseToday              | 3    |
|    |            |      |                       | 平昨        | CloseYesterday          | 4    |
|    |            |      |                       | 无         | None                    | N    |
| 13 | 强平原因       | FCC  | ForceCloseReason      | 非强平       | NotForceClose           | 0    |
|    |            |      |                       | 资金不足      | LackDeposit             | 1    |
|    |            |      |                       | 客户超仓      | ClientOverPositionLimit | 2    |
|    |            |      |                       | 会员超仓      | MemberOverPositionLimit | 3    |
|    |            |      |                       | 持仓非整数倍    | NotMultiple             | 4    |
|    |            |      |                       | 违规        | Violation               | 5    |
|    |            |      |                       | 其它        | Other                   | 6    |
|    |            |      |                       | 自然人临近交割   | PersonDeliv             | 7    |
|    |            |      |                       | 客户套保超仓    | HedgeOverPositionLimit  | 8    |
| 14 | 报单状态       | OST  | OrderStatus           | 全部成交      | AllTraded               | 0    |
|    |            |      |                       | 部分成交还在队列中 | PartTradedQueueing      | 1    |
|    |            |      |                       | 部分成交不在队列中 | PartTradedNotQueueing   | 2    |
|    |            |      |                       | 未成交还在队列中  | NoTradeQueueing         | 3    |
|    |            |      |                       | 未成交不在队列中  | NoTradeNotQueueing      | 4    |
|    |            |      |                       | 撤单        | Canceled                | 5    |
| 15 | 报单类型       | ORDT | OrderType             | 正常        | Normal                  | 0    |
|    |            |      |                       | 报价衍生      | DeriveFromQuote         | 1    |
|    |            |      |                       | 组合衍生      | DeriveFromCombin        | 2    |

| 序号 | 枚举描述  | 枚举前缀 | 枚举名称                | 编码描述       | 编码名称               | 编码数值 |
|----|-------|------|---------------------|------------|--------------------|------|
|    |       |      |                     |            | ation              |      |
| 16 | 有效期类型 | TC   | TimeCondition       | 立即完成, 否则撤销 | IOC                | 1    |
|    |       |      |                     | 本节有效       | GFS                | 2    |
|    |       |      |                     | 当日有效       | GFD                | 3    |
|    |       |      |                     | 指定日期前有效    | GTD                | 4    |
|    |       |      |                     | 撤销前有效      | GTC                | 5    |
|    |       |      |                     | 集合竞价有效     | GFA                | 6    |
| 17 | 成交量类型 | VC   | VolumeCondition     | 任何数量       | AV                 | 1    |
|    |       |      |                     | 最小数量       | MV                 | 2    |
|    |       |      |                     | 全部数量       | CV                 | 3    |
| 18 | 触发条件  | CC   | ContingentCondition | 立即         | Immediately        | 1    |
|    |       |      |                     | 止损         | Touch              | 2    |
| 19 | 操作标志  | AF   | ActionFlag          | 删除         | Delete             | 0    |
|    |       |      |                     | 挂起         | Suspend            | 1    |
|    |       |      |                     | 激活         | Active             | 2    |
|    |       |      |                     | 修改         | Modify             | 3    |
| 20 | 报单来源  | OSRC | OrderSource         | 来自参与者      | Participant        | 0    |
|    |       |      |                     | 来自管理员      | Administrator      | 1    |
| 21 | 成交类型  | TRDT | TradeType           | 普通成交       | Common             | 0    |
|    |       |      |                     | 期权执行       | OptionsExecution   | 1    |
|    |       |      |                     | OTC 成交     | OTC                | 2    |
|    |       |      |                     | 期转现衍生成交    | EFPDerived         | 3    |
|    |       |      |                     | 组合衍生成交     | CombinationDerived | 4    |
|    |       |      |                     | 大笔交易成交     | BlockTrade         | 5    |
|    |       |      |                     | 套利衍生成交     | SpreadDerived      | 6    |
| 22 | 成交价来源 | PSRC | PriceSource         | 前成交价       | LastPrice          | 0    |
|    |       |      |                     | 买委托价       | Buy                | 1    |
|    |       |      |                     | 卖委托价       | Sell               | 2    |
|    |       |      |                     | 推导价        | Implied            | 3    |
| 23 | 执行结果  | OER  | ExecResult          | 没有执行       | NoExec             | n    |
|    |       |      |                     | 已经取消       | Canceled           | c    |
|    |       |      |                     | 执行成功       | OK                 | 0    |
|    |       |      |                     | 期权持仓不够     | NoPosition         | 1    |
|    |       |      |                     | 资金不够       | NoDeposit          | 2    |
|    |       |      |                     | 会员不存在      | NoParticipant      | 3    |
|    |       |      |                     | 客户不存在      | NoClient           | 4    |

| 序号 | 枚举描述          | 枚举前缀 | 枚举名称                  | 编码描述          | 编码名称                        | 编码数值 |
|----|---------------|------|-----------------------|---------------|-----------------------------|------|
|    |               |      |                       | 合约不存在         | NoInstrument                | 6    |
|    |               |      |                       | 没有执行权限        | NoRight                     | 7    |
|    |               |      |                       | 不合理的数量        | InvalidVolume               | 8    |
|    |               |      |                       | 没有足够的历史成交     | NoEnoughHistoryTrade        | 9    |
| 24 | 期权行权后是否保留持仓标记 | EOPF | ExecOrderPositionFlag | 保留            | Reserve                     | 0    |
|    |               |      |                       | 不保留           | UnReserve                   | 1    |
| 25 | 期权行权后是否自动平仓   | EOCF | ExecOrderCloseFlag    | 自动平仓          | AutoClose                   | 0    |
|    |               |      |                       | 免于自动平仓        | NotToClose                  | 1    |
| 26 | 期权行权是否自对冲类型   | OSCF | OptSelfCloseFlag      | 自对冲期权仓位       | CloseSelfOptionPosition     | 0    |
|    |               |      |                       | 保留期权仓位        | ReserveOptionPosition       | 1    |
|    |               |      |                       | 自对冲卖方履约后的期货仓位 | SellCloseSelfFuturePosition | 2    |



### 3. 数据类型列表

| 数据类型名                          | 基本数据类型   | 数据类型说明   |
|--------------------------------|----------|----------|
| TShfeFtdcErrorIDType           | int      | 错误代码     |
| TShfeFtdcPriorityType          | int      | 优先权      |
| TShfeFtdcSettlementIDType      | int      | 结算编号     |
| TShfeFtdcMonthCountType        | int      | 月份数量     |
| TShfeFtdcTradingSegmentSNTYPE  | int      | 交易阶段编号   |
| TShfeFtdcVolumeType            | int      | 数量       |
| TShfeFtdcTimeSortIDType        | int      | 按时间排队的序号 |
| TShfeFtdcSequenceNoType        | int      | 序列号      |
| TShfeFtdcBulletinIDType        | int      | 公告编号     |
| TShfeFtdcMillisecType          | int      | 时间（毫秒）   |
| TShfeFtdcVolumeMultipleType    | int      | 合约数量乘数   |
| TShfeFtdcParticipantIDType     | char[11] | 会员代码     |
| TShfeFtdcUserIDType            | char[16] | 交易用户代码   |
| TShfeFtdcPasswordType          | char[41] | 密码       |
| TShfeFtdcClientIDType          | char[11] | 客户代码     |
| TShfeFtdcInstrumentIDType      | char[31] | 合约代码     |
| TShfeFtdcProductIDType         | char[9]  | 产品代码     |
| TShfeFtdcDateType              | char[9]  | 日期       |
| TShfeFtdcTimeType              | char[9]  | 时间       |
| TShfeFtdcInstrumentNameType    | char[21] | 合约名称     |
| TShfeFtdcProductGroupIDType    | char[9]  | 产品组代码    |
| TShfeFtdcMarketIDType          | char[9]  | 市场代码     |
| TShfeFtdcSettlementGroupIDType | char[9]  | 结算组代码    |
| TShfeFtdcOrderSysIDType        | char[13] | 报单编号     |
| TShfeFtdcExecOrderSysIDType    | char[13] | 执行宣告系统编号 |
| TShfeFtdcQuoteSysIDType        | char[13] | 报价编号     |
| TShfeFtdcTradeIDType           | char[13] | 成交编号     |
| TShfeFtdcOrderLocalIDType      | char[13] | 本地报单编号   |
| TShfeFtdcComeFromType          | char[21] | 消息来源     |
| TShfeFtdcAccountIDType         | char[13] | 资金帐号     |
| TShfeFtdcNewsTypeType          | char[3]  | 公告类型     |
| TShfeFtdcAdvanceMonthType      | char[4]  | 提前月份     |
| TShfeFtdcIPAddressType         | char[16] | IP 地址    |
| TShfeFtdcProductInfoType       | char[41] | 产品信息     |
| TShfeFtdcProtocolInfoType      | char[41] | 协议信息     |

| 数据类型名                              | 基本数据类型    | 数据类型说明     |
|------------------------------------|-----------|------------|
| TShfeFtdcBusinessUnitType          | char[21]  | 业务单元       |
| TShfeFtdcTradingSystemNameType     | char[61]  | 交易系统名称     |
| TShfeFtdcTradingRoleType           | char      | 交易角色       |
| TShfeFtdcProductClassType          | char      | 产品类型       |
| TShfeFtdcOptionsTypeType           | char      | 期权类型       |
| TShfeFtdcInstrumentStatusType      | char      | 合约交易状态     |
| TShfeFtdcDirectionType             | char      | 买卖方向       |
| TShfeFtdcPositionTypeType          | char      | 持仓类型       |
| TShfeFtdcPosiDirectionType         | char      | 持仓多空方向     |
| TShfeFtdcHedgeFlagType             | char      | 套保标志       |
| TShfeFtdcClientTypeType            | char      | 客户类型       |
| TShfeFtdcInstStatusEnterReasonType | char      | 品种进入交易状态原因 |
| TShfeFtdcOrderPriceTypeType        | char      | 报单价格条件     |
| TShfeFtdcOffsetFlagType            | char      | 开平标志       |
| TShfeFtdcForceCloseReasonType      | char      | 强平原因       |
| TShfeFtdcOrderStatusType           | char      | 报单状态       |
| TShfeFtdcOrderTypeType             | char      | 报单类型       |
| TShfeFtdcTimeConditionType         | char      | 有效期类型      |
| TShfeFtdcVolumeConditionType       | char      | 成交量类型      |
| TShfeFtdcContingentConditionType   | char      | 触发条件       |
| TShfeFtdcActionFlagType            | char      | 操作标志       |
| TShfeFtdcOrderSourceType           | char      | 报单来源       |
| TShfeFtdcTradeTypeType             | char      | 成交类型       |
| TShfeFtdcPriceSourceType           | char      | 成交价来源      |
| TShfeFtdcExecResultType            | char      | 执行结果       |
| TShfeFtdcYearType                  | int       | 年份         |
| TShfeFtdcMonthType                 | int       | 月份         |
| TShfeFtdcBoolType                  | int       | 布尔型        |
| TShfeFtdcPriceType                 | double    | 价格         |
| TShfeFtdcUnderlyingMultipleType    | double    | 合约基础商品乘数   |
| TShfeFtdcCombOffsetFlagType        | char[5]   | 组合开平标志     |
| TShfeFtdcCombHedgeFlagType         | char[5]   | 组合套保标志     |
| TShfeFtdcRatioType                 | double    | 比率         |
| TShfeFtdcMoneyType                 | double    | 资金         |
| TShfeFtdcLargeVolumeType           | double    | 大额数量       |
| TShfeFtdcNewsUrgencyType           | char      | 紧急程度       |
| TShfeFtdcSequenceSeriesType        | short     | 序列系列号      |
| TShfeFtdcErrorMsgType              | char[81]  | 错误信息       |
| TShfeFtdcAbstractType              | char[81]  | 消息摘要       |
| TShfeFtdcContentType               | char[501] | 消息正文       |

| 数据类型名                              | 基本数据类型    | 数据类型说明            |
|------------------------------------|-----------|-------------------|
| TShfeFtdcURLLinkType               | char[201] | WEB 地址            |
| TShfeFtdcIdentifiedCardNoType      | char[51]  | 证件号码              |
| TShfeFtdcIdentifiedCardNoV1Type    | char[21]  | 原证件号码             |
| TShfeFtdcPartyNameType             | char[81]  | 参与人名称             |
| TShfeFtdcIdCardTypeType            | char[16]  | 证件类型              |
| TShfeFtdcDataCenterIDType          | int       | 数据中心代码            |
| TShfeFtdcBusinessLocalIDType       | int       | 本地业务标志            |
| TShfeFtdcCurrencyIDType            | char[4]   | 币种代码              |
| TShfeFtdcRateUnitType              | int       | 外汇交易单位            |
| TShfeFtdcExRatePriceType           | double    | 外汇中间价             |
| TShfeFtdcExecOrderPositionFlagType | char      | 期权行权后是否保留期货持仓的标记  |
| TShfeFtdcExecOrderCloseFlagType    | char      | 期权行权后生成的期货持仓是否自对冲 |
| TShfeFtdcMacAddressType            | char[21]  | Mac 地址信息          |
| TShfeFtdcOptionSelfCloseSysIDType  | char[13]  | 期权自对冲系统编号         |
| TShfeFtdcOptSelfCloseFlagType      | char      | 期权行权的持仓是否自对冲      |
| TShfeFtdcAuthCodeType              | char[17]  | 终端认证授权码类型         |

## 4. API 返回值列表

| 返回值 | 返回值含义        |
|-----|--------------|
| 0   | 成功           |
| -1  | 未登录          |
| -2  | 超出在途流控       |
| -3  | 超出请求流控       |
| -4  | 文件不存在或文件读写失败 |
| -5  | 已登录或重复调用     |
| -6  | 强制要求填写的字段为空  |
| -7  | 启用认证但认证失败    |
| -8  | 超过最大个数       |
| -9  | 未初始化         |
| -10 | 已初始化         |
| -11 | 重复的 id       |
| -12 | 尚未与前置建立连接    |
| -13 | 会员号不一致       |
| -14 | 用户号不一致       |