



版本	<input type="checkbox"/> 草稿
	<input checked="" type="checkbox"/> 正式发布
	<input type="checkbox"/> 正在修改

SFIT CTP Mini API 应用开发参考手册

(Ver 1.7.3-P2)



上海期货信息技术有限公司

2025 年 7 月

修订历史记录

日期	版本	修订说明	修改者
2021-1-18	0.1	在《CTP Mini _api 说明_v1.3》基础上创建	夏乾坤
2021-2-19	0.2	去掉不支持的接口调用函数及其他修改合并	徐文华 邵悦 夏乾坤
2021-2-25	0.3	1) 增加 <u>接口流程</u> 章节，一方面简要勾画 API 生命周期流程，一方面增加常见的报、撤单流程； 2) 补充 <u>交易接口</u> 、 <u>行情接口</u> 中部分交易所特殊业务规则、字段处理规则、返回常见错误说明等；	夏乾坤
2021-3-5	0.4	增加 <u>V1.5.6版本变更</u> 说明	夏乾坤
2021-3-10	0.5	根据项目组审核修改部分内容，主要包括《 <u>报单</u> 》章节、 <u>V1.5.6版本变更</u> 、 <u>其他修正</u>	徐文华 夏乾坤
2021-3-26	0.6	根据应用运维、测试意见修改： 1) <u>SubscribeMarketData</u> 中示例，采用静态二维数组，避免初学者误解； 2) 勘误： <u>TraderAPI</u> 章节中 <u>OnRtnInstrumentStatus</u> 是通过公有流推送； 3) <u>版本变更</u> 增加终端连接 Session 级别流控说明。	温振辉 廖莹寒 夏乾坤
2021-5-24	0.7	更正各个 <u>ReqOrderInsert</u> 、 <u>ReqExecOrderInsert</u> 、 <u>ReqQuoteInsert</u> 、 <u>ReqOptionSelfCloseInsert</u> 函数中终端输入项目中 OrderRef 字段为必填项目；CTP Mini 不象 CTP 主席对 OrderRef 为空做了加号处理，需要终端自行维护必填；	夏乾坤
2021-6-9	0.8	V1.5.7 版本：配合中金所行情 API 升级到 MdUserAPI V2.0 版本，添加 BandingUpperPrice、BandingLowerPrice 字段； 对应修改： 1) <u>3.2.24 OnRspQryDepthMarketData</u> 2) <u>4.2.8 OnRtnDepthMarketData</u>	夏乾坤
2021-9-10	0.9	本版本不兼容之前版本变更 ，API 接入认证时加入版本控制，请各开发商务必更新 API。具体变化参见以下各章节： 1) <u>版本变更：增加 V1.5.8 变更说明</u> 2) <u>更新 2.5.2.1 TraderAPI 说明</u> 3) <u>交易接口增加章节 3.1.47~3.1.50</u>	夏乾坤

		<p>4) 回应函数增加章节 <u>3.2.49~3.2.53</u></p> <p>5) <u>ReqQryInvestorPositionCombineDetail</u> 接口及回调 <u>RspQryInvestorPositionCombineDetail</u> 不再支持。代替以 <u>ReqQryInvestorPositionForComb</u> 及其回调 <u>RspQryInvestorPositionForComb</u> 实现类似功能。</p> <p>6) <u>ReqQryInvestorPositionDetail</u> 接口及回调 <u>OnRspQryInvestorPositionDetail</u> 根据交易后台设置有条件支持。如查询结果无效,请联系期货公司。</p>	
2021-12-7	1.0	<p>1) 明确字段信息: <u>3.1.47 ReqCombActionInsert</u></p> <p>2) 增加版本变更说明章节 <u>2.3.3V1.5.9</u>, 主要包括: ThostFtdcUserApiDataType.h 头文件增加郑商所组合类型定义、ThostFtdcUserApiStruct.h 头文件中组合单腿汇报结构变化, 增加三个字段;</p>	夏乾坤
2022-1-7	1.0	<p>1) 4.2.8 <u>OnRtnDepthMarketData</u> 章节增加了参数说明: 从 V1.5.8 版本开始增加了大商所组合合约行情; 从 V1.5.9 版本开始增加了郑商所组合合约行情;</p>	夏乾坤
2022-2-21	1.1	<p>1) 新增加章节: <u>2.3.4 V1.6.0</u> 介绍本版本升级信息</p> <p>2) 新增加章节: <u>3.1.51 ReqQryForQuoteParam</u> 查询价差请求;</p> <p>3) 新增加章节: <u>3.2.54 OnRspQryForQuoteParam</u> 查询价差请求响应;</p>	夏乾坤
2022-5-18	1.2	<p>1) 增加章节 <u>2.3.5 V1.6.1</u></p> <p>2) 增加章节 <u>3.1.52 ReqMKBatchOrderAction</u></p> <p>3) 增加章节 <u>3.2.55 OnRspMKBatchOrderAction</u></p> <p>4) 增加章节 <u>3.1.53 SubscribeFlowCtrlWarning</u></p> <p>5) 增加章节 <u>3.1.54 UnSubscribeFlowCtrlWarning</u></p> <p>6) 增加章节 <u>3.2.56 OnRspSubscribeFlowCtrlWarning</u></p> <p>7) 增加章节 <u>3.2.57 OnRspUnSubscribeFlowCtrlWarning</u></p> <p>8) 增加章节 <u>3.2.58 OnRtnFlowCtrlWarning</u></p>	夏乾坤
2022-8-11	1.3	<p>1) 增加章节 <u>2.3.6 V1.6.2</u></p> <p>2) 增加章节 <u>3.1.55ReqQryTraderOffer</u></p> <p>3) 增加章节 <u>3.2.59OnRspQryTraderOffer</u></p>	夏乾坤
2022-11-16	1.4	<p>1) 增加章节 <u>2.3.7V1.6.3</u></p> <p>2) 增加查询郑商所 SPBM 组合产品保证金明细请求及查询回应 章节 <u>3.1.56ReqQryInvestorProdSPBMDetail</u>、</p>	夏乾坤

		<u>3.2.60OnRspQryInvestorProdSPBMDetail</u>	
2023-2-20	1.5	1) 更新 <u>1.2 架构</u> 中架构图，增加广期所； 2) 增加章节 <u>2.3.8V1.6.4</u> 3) 更新 <u>3.2.3OnRspUserLogin</u> 章节，变更结构增加广期所时间字段 <u>CThostFtdcRspUserLoginField</u>	夏乾坤
2023-6-20	1.6	1) 增加 CTP Mini 错误码对应表；	夏乾坤
2023-8-28	1.6.6	1) 为了实现本文档与 CTP Mini API 版本的一致性，从本次版本开始文档版本号与 CTP Mini API 统一； 2) 增加 <u>2.3.9V1.6.6</u> 章节，记录 V1.6.6 版本变化； 3) 更新 <u>2.5.2.1 TraderAP</u> 章节； 4) 增加 SPMM 保证金明细查询接口章节 <u>3.1.57ReqQrySPMMInvestorCommodityGroupMargin</u> 及 SPMM 保证金明细查询响应接口 <u>3.2.62 OnRspQrySPMMInvestorCommodityGroupMargin</u> 5) 更新数据结构 <u>CThostFtdcInputOrderField</u> 、 <u>CThostFtdcOrderField</u> 、 <u>CThostFtdcInputQuoteField</u> 、 <u>CThostFtdcQuoteField</u> ，报单请求域增加 CustomOrderRef 字段，报价请求域增加 CustomQuoteRef 字段。	夏乾坤
2023-12-28	1.6.7	1) 增加 <u>2.3.10 V1.6.7</u> 章节，说明本版本的变化概要； 2) 增加 <u>3.1.58ReqQryRULEInvestorProdMargin</u> 章节查询 Rule 组保明细；对应的响应接口章节 <u>3.2.63 OnRspQryRULEInvestorProdMargin</u> 3) 增加 <u>3.1.59 ReqQryRCAMSInvestorProdMargin</u> 章节查询 RCAMS 组保明细；对应的响应接口章节 <u>3.2.64 OnRspQryRCAMSInvestorProdMargin</u> 4) 增加 <u>3.1.60 ReqQryRCAMSInvestorCombPosition</u> 章节查询 RCAMS 策略组合持仓；对应的响应接口章节 <u>3.2.65 OnRspQryRCAMSInvestorCombPosition</u>	夏乾坤
2024-3-19	1.6.8	1) 增加 <u>2.3.11 V1.6.8</u> 章节，说明本版本的变化概要； 2) 更新 <u>3.1.20 ReqQuotInsert</u> 、 <u>3.2.10 OnRspQuotInsert</u> 参数结构字段； 3) 增加章节 <u>3.1.61 ReqQryControlParam</u> 、 <u>3.2.66 OnRspQryControlParam</u>	夏乾坤

		4) 更新 <u>3.2.3OnRspUserLogin</u> 章节，增加系统类型信息赋值到 <u>CThostFtdcRspUserLoginField</u> 结构体的 <u>SystemName</u> 字段；	
2024-4-24	1.6.9	1) 根据 CTP 产品线软件著作权登记名称的变化，对应本系统软件名称变更：原有的全称“Futures Ideal 个性化投资者交易平台软件”变更为“SFIT CTP 综合交易平台 Mini 交易软件”； 2) 原有的产品名称简称“CTPIIMini”统一变更为“CTP Mini”。	夏乾坤
2024-5-30	1.6.9	1) 增加 <u>2.3.12 V1.6.9</u> 章节，说明本版本的变化概要； 2) 增加 <u>资金变动订阅与取消订阅</u> 章节： <u>3.1.62 ReqSubscribeFundChange</u> 、 <u>3.1.63 ReqUnSubscribeFundChange</u> ；对应的响应接口： <u>3.2.67 OnRspSubscribeFundChange</u> 、 <u>3.2.68 OnRspUnSubscribeFundChange</u> ；订阅成功后主动推送资金变动接口： <u>3.2.69 OnRtnFundChange</u> ； 3) 为了实现报单回报中返回成交均价，修改头文件《 <u>ThostFtdcUserApiStruct.h</u> 》，报单相关的结构体 <u>CThostFtdcOrderField</u> 最后增加了成交均价字段： <u>TThostFtdcPriceType TradeAvgPrice</u> ； 4) 更新 <u>5.错误码对应表</u> 章节。	夏乾坤
2024-9-2	1.7.0	1) 增加 <u>2.3.13 V1.7.0</u> 章节，说明本版本的变化概要； 2) 更新 <u>3.1.15 ReqOrderInsert</u> 章节，报单结构体最后增加报单标志字段 <u>OrderFlag</u> ；如果第一个字符填写‘1’则表示后台对该笔报单不做自成交检查；	夏乾坤
2025-6-12	1.7.3	1) 新增 <u>2.3.14 V1.7.3</u> 章节 2) 更新 <u>5.错误码对应表</u> 章节；	夏乾坤
2025-7-18	1.7.3-P2	1) 新增 <u>2.3.15 V1.7.3-P2</u> 升级说明； 2) 更新 <u>3.1.17 ReqExecOrderInsert</u> 章节 <u>ActionType</u> 、 <u>CloseFlag</u> 字段说明； 3) 更新 <u>3.1.22 ReqOptionSelfCloseInsert</u> 章节，本接口不再支持大商所期权自对冲录入，其他交易所不变； 4) 更新 <u>5.错误码对应表</u> 章节；新增大商所对冲相关报错信息 5) 新增加 <u>3.1.64 ReqOffsetSetting</u> 接口及响应接口 <u>3.2.70 OnRspOffsetSetting</u> ；	夏乾坤

		<ul style="list-style-type: none">6) 新增加 3.1.65 ReqCancelOffsetSetting 接口及响应接口 3.2.71 OnRspCancelOffsetSetting;7) 新增加交易所返回响应接口 3.2.72OnRtnOffsetSetting、3.2.73OnErrRtnCancelOffsetSetting;8) 新增加 3.1.66ReqQryOffsetSetting 查询接口及响应接口 3.2.74OnRspQryOffsetSetting ;	
--	--	--	--



目录

修订历史记录.....	2
目录.....	7
1 系统介绍.....	14
1.1 概述.....	14
1.2 架构.....	14
1.3 特点.....	15
2 API 介绍.....	16
2.1 基本文件	16
2.2 开发平台	17
2.3 版本变更	17
2.3.1 之前版本.....	17
2.3.2 V1.5.8.....	18
2.3.3 V1.5.9.....	30
2.3.4 V1.6.0.....	32
2.3.5 V1.6.1.....	33
2.3.6 V1.6.2.....	34
2.3.7 V1.6.3.....	35
2.3.8 V1.6.4.....	35
2.3.9 V1.6.6.....	36
2.3.10 V1.6.7.....	38
2.3.11 V1.6.8.....	41
2.3.12 V1.6.9.....	44
2.3.13 V1.7.0.....	44
2.3.14 V1.7.3.....	45
2.3.15 V1.7.3-P2.....	45
2.4 网络通信	53
2.4.1 通讯模式.....	53
2.4.2 通讯实现.....	55
2.5 接口分类	56

2.5.1 管理接口.....	56
2.5.2 业务接口.....	57
2.6 接口流程	61
2.6.1 交易	61
2.6.2 行情	65
3 交易接口.....	66
3.1 CTHOSTFDCTRADERAPI.....	66
3.1.1 CreateFtdcTraderApi	66
3.1.2 GetApiVersion.....	67
3.1.3 Release	68
3.1.4 Init.....	69
3.1.5 Join.....	69
3.1.6 GetTradingDay.....	69
3.1.7 RegisterFront.....	70
3.1.8 RegisterSpi.....	70
3.1.9 SubscribePrivateTopic	70
3.1.10 SubscribePublicTopic.....	71
3.1.11 ReqAuthenticate	71
3.1.12 ReqUserLogin.....	72
3.1.13 ReqUserLoginEncrypt.....	73
3.1.14 ReqUserLogout.....	74
3.1.15 ReqOrderInsert.....	75
3.1.16 ReqOrderAction.....	78
3.1.17 ReqExecOrderInsert.....	79
3.1.18 ReqExecOrderAction.....	81
3.1.19 ReqForQuoteInsert.....	82
3.1.20 ReqQuoteInsert.....	83
3.1.21 ReqQuoteAction.....	86
3.1.22 ReqOptionSelfCloseInsert.....	87
3.1.23 ReqOptionSelfCloseAction	88
3.1.24 ReqQryOrder	89
3.1.25 ReqQryTrade	90
3.1.26 ReqQryInvestorPosition	91
3.1.27 ReqQryTradingAccount	92

3.1.28 ReqQryInvestor.....	92
3.1.29 ReqQryTradingCode.....	93
3.1.30 ReqQryInstrumentMarginRate.....	94
3.1.31 ReqQryInstrumentCommissionRate.....	95
3.1.32 ReqQryExchange.....	95
3.1.33 ReqQryInstrument.....	96
3.1.34 ReqQryDepthMarketData.....	97
3.1.35 ReqQryOptionSelfClose.....	97
3.1.36 ReqQryInstrumentStatus.....	98
3.1.37 ReqQryInvestorPositionDetail.....	99
3.1.38 ReqQryInvestorPositionCombineDetail.....	99
3.1.39 ReqQryExchangeMarginRate.....	100
3.1.40 ReqQryExchangeMarginRateAdjust.....	101
3.1.41 ReqQryOptionInstrTradeCost.....	101
3.1.42 ReqQryOptionInstrCommRate.....	102
3.1.43 ReqQryExecOrder.....	103
3.1.44 ReqQryForQuote.....	104
3.1.45 ReqQryQuote.....	105
3.1.46 ReqQryInstrumentOrderCommRate.....	105
3.1.47 ReqCombActionInsert.....	106
3.1.48 ReqQryCombAction.....	107
3.1.49 ReqQryCombInstrument.....	107
3.1.50 ReqQryInvestorPositionForComb.....	108
3.1.51 ReqQryForQuoteParam.....	109
3.1.52 ReqMKBatchOrderAction.....	109
3.1.53 SubscribeFlowCtrlWarning	111
3.1.54 UnSubscribeFlowCtrlWarning.....	112
3.1.55 ReqQryTraderOffer.....	112
3.1.56 ReqQryInvestorProdSPBMDetail.....	113
3.1.57 ReqQrySPMMInvestorCommodityGroupMargin.....	113
3.1.58 ReqQryRULEInvestorProdMargin.....	114
3.1.59 ReqQryRCAMSInvestorProdMargin.....	115
3.1.60 ReqQryRCAMSInvestorCombPosition.....	116
3.1.61 ReqQryControlParam.....	117
3.1.62 ReqSubscribeFundChange.....	118

3.1.63 ReqUnSubscribeFundChange.....	118
3.1.64 ReqOffsetSetting.....	119
3.1.65 ReqCancelOffsetSetting.....	121
3.1.66 ReqQryOffsetSetting	122
3.2 CTHOSTFTDCTRADERSPI.....	122
3.2.1 OnFrontConnected.....	122
3.2.2 OnFrontDisconnected.....	123
3.2.3 OnRspUserLogin.....	123
3.2.4 OnRspAuthenticate.....	124
3.2.5 OnRspOrderInsert.....	124
3.2.6 OnRspOrderAction.....	125
3.2.7 OnRspExecOrderInsert.....	125
3.2.8 OnRspExecOrderAction.....	126
3.2.9 OnRspForQuoteInsert.....	126
3.2.10 OnRspQuoteInsert.....	127
3.2.11 OnRspQuoteAction.....	127
3.2.12 OnRspOptionSelfCloseInsert.....	127
3.2.13 OnRspOptionSelfCloseAction	128
3.2.14 OnRspQryOrder	128
3.2.15 OnRspQryTrade	130
3.2.16 OnRspQryInvestorPosition	131
3.2.17 OnRspQryTradingAccount	133
3.2.18 OnRspQryInvestor	135
3.2.19 OnRspQryTradingCode	135
3.2.20 OnRspQryInstrumentMarginRate	136
3.2.21 OnRspQryInstrumentCommissionRate	137
3.2.22 OnRspQryExchange	137
3.2.23 OnRspQryInstrument	138
3.2.24 OnRspQryDepthMarketData.....	139
3.2.25 OnRspQryInstrumentStatus.....	140
3.2.26 OnRspQryInvestorPositionDetail.....	141
3.2.27 OnRspQryInvestorPositionCombineDetail	142
3.2.28 OnRspQryExchangeMarginRate	143
3.2.29 OnRspQryExchangeMarginRateAdjust	144
3.2.30 OnRspQryOptionInstrTradeCost	145

3.2.31	<i>OnRspQryOptionInstrCommRate</i>	145
3.2.32	<i>OnRspQryExecOrder</i>	146
3.2.33	<i>OnRspQryForQuote</i>	148
3.2.34	<i>OnRspQryQuote</i>	149
3.2.35	<i>OnRspQryOptionSelfClose</i>	150
3.2.36	<i>OnRtnOrder</i>	152
3.2.37	<i>OnRtnTrade</i>	153
3.2.38	<i>OnErrRtnOrderAction</i>	153
3.2.39	<i>OnRtnInstrumentStatus</i>	155
3.2.40	<i>OnRtnExecOrder</i>	155
3.2.41	<i>OnErrRtnExecOrderAction</i>	155
3.2.42	<i>OnErrRtnForQuoteInsert</i>	156
3.2.43	<i>OnRtnQuote</i>	157
3.2.44	<i>OnErrRtnQuoteAction</i>	157
3.2.45	<i>OnRtnForQuoteRsp</i>	158
3.2.46	<i>OnRtnOptionSelfClose</i>	158
3.2.47	<i>OnErrRtnOptionSelfCloseAction</i>	159
3.2.48	<i>OnRspQryInstrumentOrderCommRate</i>	160
3.2.49	<i>OnRspCombActionInsert</i>	160
3.2.50	<i>OnRtnCombAction</i>	161
3.2.51	<i>OnRspQryCombAction</i>	162
3.2.52	<i>OnRspQryCombInstrument</i>	162
3.2.53	<i>OnRspQryInvestorPositionForComb</i>	163
3.2.54	<i>OnRspQryForQuoteParam</i>	164
3.2.55	<i>OnRspMKBatchOrderAction</i>	164
3.2.56	<i>OnRspSubscribeFlowCtrlWarning</i>	165
3.2.57	<i>OnRspUnSubscribeFlowCtrlWarning</i>	165
3.2.58	<i>OnRtnFlowCtrlWarning</i>	166
3.2.59	<i>OnRspQryTraderOffer</i>	166
3.2.60	<i>OnRspQryInvestorProdSPBMDetail</i>	167
3.2.61	<i>OnErrRtnOrderInsert</i>	168
3.2.62	<i>OnRspQrySPMMInvestorCommodityGroupMargin</i>	168
3.2.63	<i>OnRspQryRULEInvestorProdMargin</i>	170
3.2.64	<i>OnRspQryRCAMSInvestorProdMargin</i>	171
3.2.65	<i>OnRspQryRCAMSInvestorCombPosition</i>	172

3.2.66 <i>OnRspQryControlParam</i>	173
3.2.67 <i>OnRspSubscribeFundChange</i>	173
3.2.68 <i>OnRspUnSubscribeFundChange</i>	174
3.2.69 <i>OnRtnFundChange</i>	174
3.2.70 <i>OnRspOffsetSetting</i>	175
3.2.71 <i>OnRspCancelOffsetSetting</i>	175
3.2.72 <i>OnRtnOffsetSetting</i>	175
3.2.73 <i>OnErrRtnCancelOffsetSetting</i>	177
3.2.74 <i>OnRspQryOffsetSetting</i>	178
4 行情接口.....	179
4.1 CTHOSTFTDCMDAPI.....	179
4.1.1 <i>CreateFtdcMdApi</i>	179
4.1.2 <i>GetApiVersion</i>	180
4.1.3 <i>Release</i>	180
4.1.4 <i>Init</i>	180
4.1.5 <i>Join</i>	181
4.1.6 <i>GetTradingDay</i>	181
4.1.7 <i>RegisterFront</i>	182
4.1.8 <i>RegisterSpi</i>	182
4.1.9 <i>SubscribeMarketData</i>	183
4.1.10 <i>UnSubscribeMarketData</i>	184
4.1.11 <i>SubscribeForQuoteRsp</i>	184
4.1.12 <i>UnSubscribeForQuoteRsp</i>	184
4.1.13 <i>ReqUserLogin</i>	185
4.2 CTHOSTFTDCMDSPI	185
4.2.1 <i>OnFrontConnected</i>	185
4.2.2 <i>OnFrontDisconnected</i>	186
4.2.3 <i>OnRspUserLogin</i>	186
4.2.4 <i>OnRspSubMarketData</i>	187
4.2.5 <i>OnRspUnSubMarketData</i>	187
4.2.6 <i>OnRspSubForQuoteRsp</i>	188
4.2.7 <i>OnRspUnSubForQuoteRsp</i>	188
4.2.8 <i>OnRtnDepthMarketData</i>	188
4.2.9 <i>OnRtnForQuoteRsp</i>	190

5 错误码对应表.....	190
---------------	-----



1 系统介绍

为了方便广大 CTP Mini API 开发者更好了解 CTP Mini 产品，本章对产品做一个概括介绍，希望从整体上帮助到每个开发者。

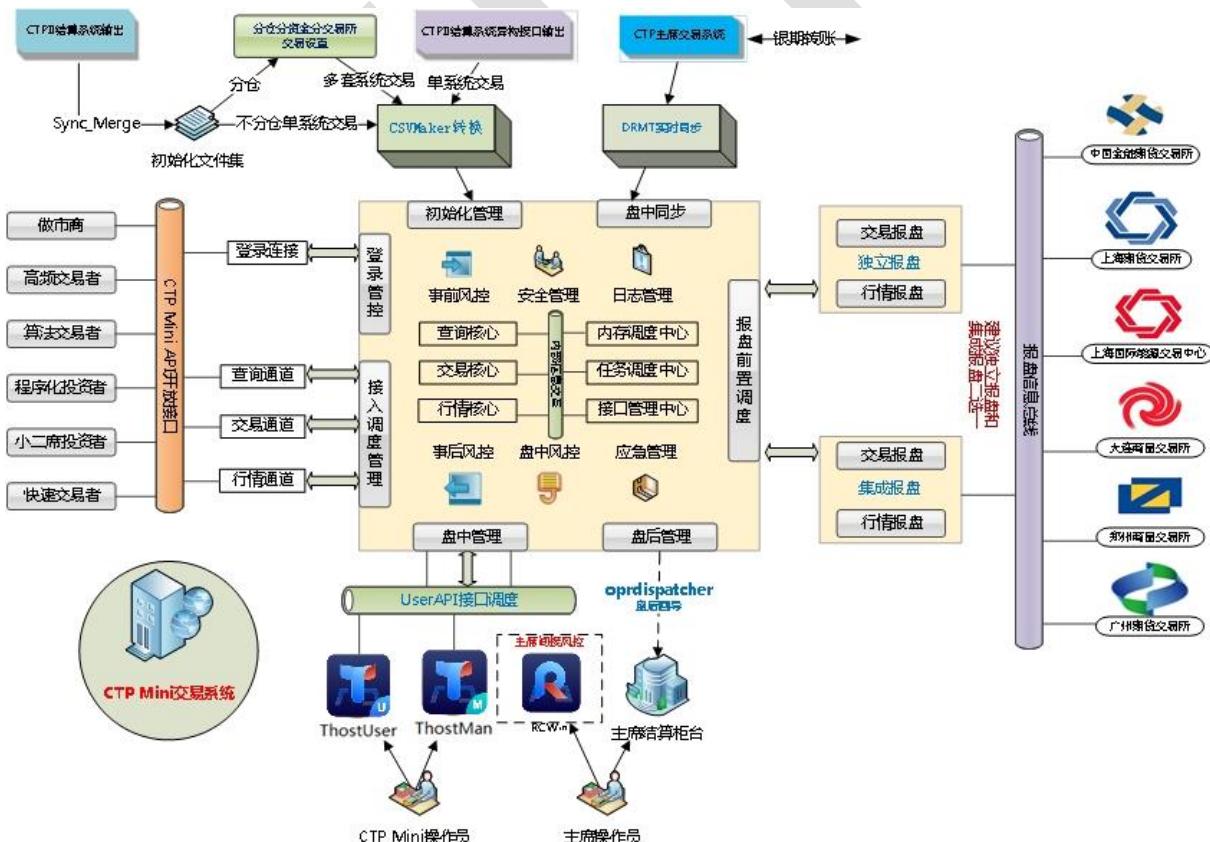
1.1 概述

作为市场细分产品，CTP Mini 为极速客户提供一个专业、极速、轻量级、小投入的期货及其衍生品交易通道，为程序化客户、算法交易者、高频投资者等提供高性价比服务平台，为做市商打造一个独特、独享、开放式接入的完备做市平台。

一台机器、一套系统、多路接入、全部国内交易所业务支持，给您简而美的不二选择！

1.2 架构

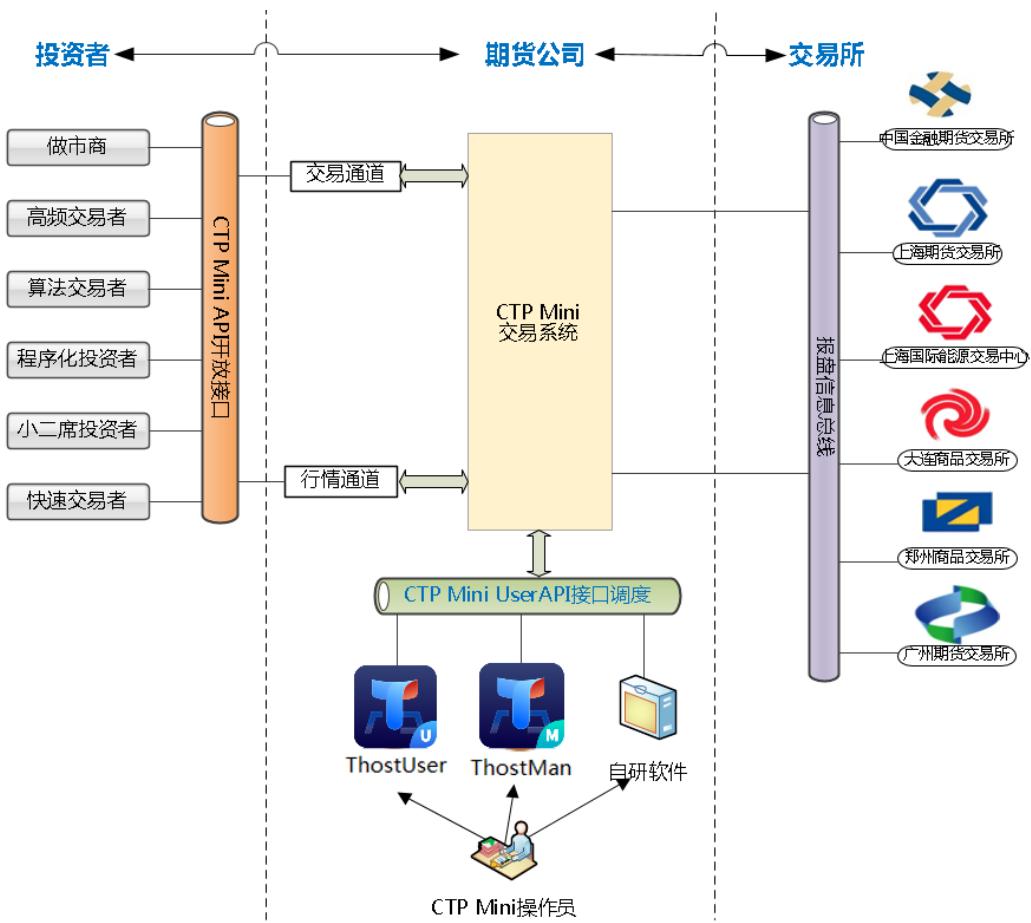
CTP Mini 整体系统架构示意如下图：



从以上架构可以看到，CTP Mini API 开发主要包括两块：

- 投资者交易终端应用 API 开发：TraderAPI（交易）+ MdAPI（行情）；
- 期货公司内部应用 API 开发：UserAPI（交易）+ MdAPI（行情）；

从开发的角度简化流程图如下示意：



1.3 特点

- 小快而灵---满足极速投资者交易需要
 - 高速度、高吞吐量；
 - 五所业务支持，风控合宜；
 - 实时资金同步，快速资金到位；
 - 各种应急处理，紧急风控快速纠偏。
- 持续创新---精益求精，守正出新
 - 分仓、分资金、分交易所交易；
 - 软硬件一体化结合全面提升性能；
 - 服务极速投资者，全面支持做市商；
 - 开放标准化 API 接口，支持 CTP 异构系统接入。
- 未来已来---不辍耕耘，奋斗不息，服务客户
 - 精简架构，全面提升性能；
 - 高性能内存交易，实时快捷风控；

- 自适应均衡通讯，精确重演，完备稽核；
- 特色交易功能支持，想客户之所想，急客户之所需。

2 API 介绍

2.1 基本文件

上海期货信息技术有限公司（简称“上期技术”）于 2018 年 12 月成功上线了 CTP Mini 交易系统。系统使用了开放的主流平台，API 可以运行在 Linux 和 Windows 操作系统上。

CTP Mini 交易系统 API 包括交易 API 接口（TraderAPI）和行情 API 接口（MdAPI）。

终端程序调用 TraderAPI 即可完成与 CTP Mini 交易模块的对接，接收到交易所的报单和成交回报后 TraderAPI 将回调（Callback）相关接口；终端程序调用 MdAPI 即可完成与 CTP Mini 行情模块对接，接收到交易所行情后 MdAPI 将回调（Callback）相关接口。

TraderAPI 把与 CTP Mini 交易系统之间复杂的协议转换、数据同步和网络通信进行了封装。TraderAPI 通过与交易前置机之间的 TCP 连接进行通讯。通过 TraderAPI 建立的连接通道支持断线自动重连。

MdAPI 与 TraderAPI 类似，与行情前置机之间建立 TCP 连接，实现行情的订阅和接收。

CTP Mini API 是一个基于 C++ 的类库，通过使用或继承类库提供的接口来实现交易和行情转发功能，包括报单录入、报单撤销、报单查询、成交单查询、投资者查询、持仓查询、合约查询、合约交易状态查询、报价录入、询价请求等等。

该类库文件列表如下：

文件名	版本	文件描述
TraderAPI (交易 API)		
ThostFtdcTraderApi.h	V1.5.6	交易接口头文件
ThostFtdcUserApiStruct.h	V1.5.6	定义了一系列业务相关的数据结构的头文件
ThostFtdcUserApiDataType.h	V1.5.6	定义了一系列数据类型的头文件
thosttraderapi.dll	V1.5.6	Windows 动态链接库
thosttraderapi.lib	V1.5.6	Windows 导入库文件
libthosttraderapi.so	V1.5.6	Linux 动态库
MdAPI (行情 API)		
ThostFtdcMdApi.h	V1.5.6	行情接口头文件
ThostFtdcUserApiStruct.h	V1.5.6	定义了一系列业务相关的数据结构的头文件
ThostFtdcUserApiDataType.h	V1.5.6	定义了一系列数据类型的头文件
thostmduserapi.dll	V1.5.6	Windows 动态链接库二进制文件
thostmduserapi.lib	V1.5.6	Windows 导入库文件

libthostmduserapi.so

V1.5.6

Linux 动态库

Windows 版本 api 支持 MSVC++ 12.0 (Visual Studio 2013) 及以上版本；Linux 版本 api 支持基于 gcc4.1.2 及以上版本，依赖于 openssl 库。

以上文件可以在[上海期货信息技术有限公司官网下载](http://www.sfit.com.cn/5_2_DocumentDown_4.htm)：

http://www.sfit.com.cn/5_2_DocumentDown_4.htm

2.2 开发平台

目前发布了以下操作系统平台的版本：

- Windows：包括.h 文件、.dll 文件和.lib 文件。
- RedHat Linux：包括.h 文件和.so 文件。

2.3 版本变更

说明：新版本 API 升级，原则都是向下兼容；特殊不兼容特别说明。

2.3.1 之前版本

V1.5.8 之前版本变更内容摘要如下：

版本号	变更内容
V1.0	正式版本发布
V1.2	增加大商所期权六期接口
V1.3	支持穿透式监管
V1.4.1	后台增加 API 版本控制：1.4.1 之前版本不可以登录 CTP Mini V1.4.1 及以后的交易系统；
V1.5.1	1) 增加申报费率查询及响应函数新接口； <code>ReqQryInstrumentOrderCommRate</code> <code>OnRspQryInstrumentOrderCommRate</code> 2) 增加分仓冻结资金查询功能：CThostFtdcTradingAccountField 结构增加了分仓冻结资金字段 ///分仓冻结资金 TThostFtdcMoneyTypeFrozenPartition;
V1.5.5	在 CTP Mini 的头文件 ThostFtdcUserApiDataType.h 文件中增加枚举值，具体如下： ///保留卖方履约后的期货仓位 <code>#define THOST_FTDC_OSCF_ReserveFuturePosition '4'</code>

	//TAS 合约 #define THOST_FTDC_PC_TAS '7'
V1.5.6	<p>1) 支持中金所顶单业务， CThostFtdcInputQuoteField 增加 ReplaceSysID 字段，表示顶单编号； CThostFtdcQuoteField 增加 ReplaceSysID 字段。 //顶单编号 TThostFtdcOrderSysIDType ReplaceSysID;</p> <p>2) 增加 session 级别流控处理：如果期货公司设置了流控参数 max_package_per_second (每秒最大包个数：小于 0 或空，不流控；等 0，禁止该 session 发送交易类请求；大于 0，依照设置值流控)，则 CTP Mini 后台对终端的报单类请求做流控，包括撤单、报价、撤报价、执行宣告、撤销执行宣告、期权自对冲、撤销期权自对冲、询价；如果超过流控，则通过 RspXXX 通知， ErrorID=-1, ErrorMsg=超过前置流控，对应的请求不予处理；</p> <p>3) 后台优化，API 重新编译，建议终端程序使用最新 API。</p>
V1.5.7	<p>配合中金所行情 API 升级到 MdUserAPI V2.0 版本， ThostFtdcUserApiStruct.h 头文件中 CThostFtdcDepthMarketDataField 域最后添加 BandingUpperPrice、BandingLowerPrice 字段；</p> <p>特别说明：因行情结构变更，不支持新 API 连接老版本后台交易系统，终端开发需要询问期货公司后台交易系统是否已升级到 V1.5.7 及以上版本；</p>

2.3.2 V1.5.8

✓ 本次 API 是重大不兼容变更！

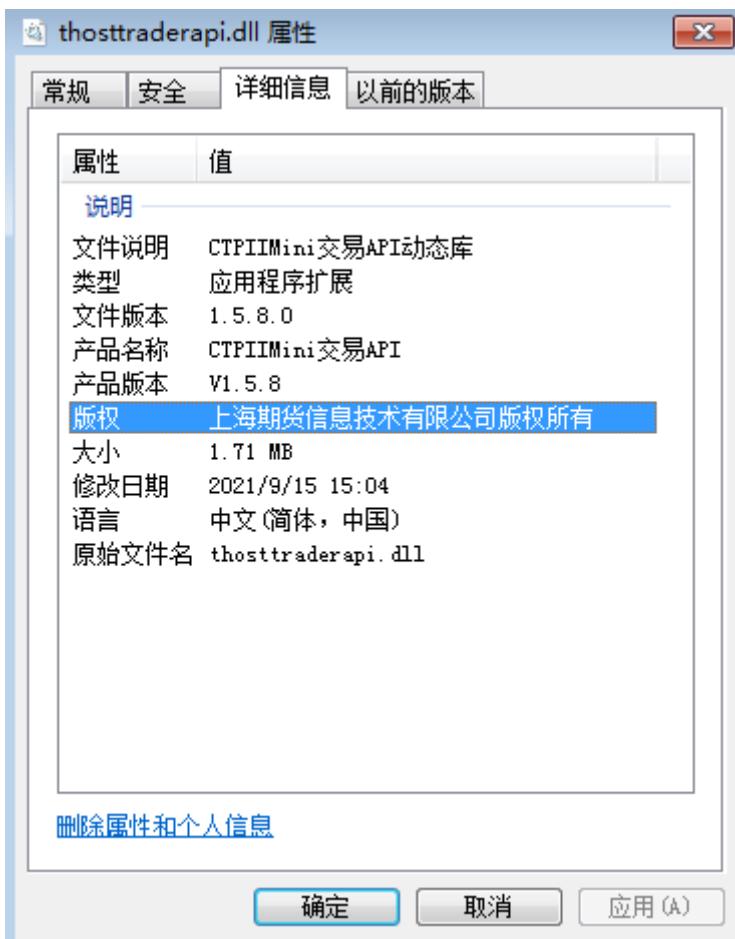
V1.5.8 API 连接 CTP Mini 后台需为 V1.5.8 及以上版本。会员公司升级后台到 V1.5.8 及以上版本时，务必提醒各开发商更新 API 版本；各家开发商采用本版本务必和后台会员公司确认交易系统版本。

不兼容原因：

- 1) 业务变更：支持组合交易，组合保证金算法变更，支持大商所组合保证金优惠，减少资金占用；
- 2) 配套接口变更：数据结构变化、合约字段扩展，行情、交易相关接口与回调接口变化；

✓ Windows 下 DLL 详细版本信息查询

从本版本开始，增加了 Windows 环境的 CTP Mini API DLL 版本右键查看详细版本信息，示意如下图：



✓ 交易 API 变更简要说明如下：

(一)头文件 ThostFtdcUserApiDataType.h 变更六项，具体如下：

1) ///TFtdcExchangeInstIDType 是一个合约在交易所的代码类型，由 31 扩展到 81，支持组合合约代码；

```
typedef char TThostFtdcExchangeInstIDType[81];
```

2) ///TFtdcInstrumentIDType 是一个合约代码类型，由 31 扩展到 81，支持组合合约代码；

```
typedef char TThostFtdcInstrumentIDType[81];
```

3) 增加两个组合枚举值，支持大商所投机套保和套保投机组合

```
///第一腿投机第二腿套保 大商所专用
```

```
#define THOST_FTDC_HF_SpecHedge '6'
```

```
///第一腿套保第二腿投机 大商所专用
```

```
#define THOST_FTDC_HF_HedgeSpec '7'  
///对应数据类型为  
typedef char TThostFtdcHedgeFlagType;
```

4) ///TFtdcCSRCExchangeInstIDType 是一个合约代码类型，由 31 扩展到 81，支持组合合约代码；

```
typedef char TThostFtdcCSRCExchangeInstIDType[81];
```

5) 增加了大商所组合枚举值

```
///TFtdcDceCombinationTypeType 是一个大商所组合类型类型
```

///期货对锁组合

```
#define THOST_FTDC_DCECOMBT_SPL '0'
```

// 期权对锁组合

```
#define THOST_FTDC_DCECOMBT_OPL '1'
```

// 期货跨期组合

```
#define THOST_FTDC_DCECOMBT_SP '2'
```

// 期货跨品种组合

```
#define THOST_FTDC_DCECOMBT_SPC '3'
```

// 买入期权垂直价差组合

```
#define THOST_FTDC_DCECOMBT_BLS '4'
```

// 卖出期权垂直价差组合

```
#define THOST_FTDC_DCECOMBT_BES '5'
```

// 期权日历价差组合

```
#define THOST_FTDC_DCECOMBT_CAS '6'
```

// 期权跨式组合

```
#define THOST_FTDC_DCECOMBT_STD '7'
```

// 期权宽跨式组合

```
#define THOST_FTDC_DCECOMBT_STG '8'
```

// 买入期货期权组合

```
#define THOST_FTDC_DCECOMBT_BFO '9'
```

// 卖出期货期权组合

```
#define THOST_FTDC_DCECOMBT_SFO 'a'
```

```
typedef char TThostFtdcDceCombinationTypeType;
```

6) 组合指令方向类型增加操作员删组合单枚举值

```
#define THOST_FTDC_CMDR_DelComb '2'  
//对应数据类型为：  
typedef char TThostFtdcCombDirectionType;
```

(二)ThostFtdcUserApiStruct.h 数据结构头文件变更 8 个数据结构，具体如下：

1) 新增加//申请组合合约信息

```
struct CThostFtdcComblnstrumentField  
{  
    ///合约代码  
    TThostFtdcInstrumentIDType InstrumentID;  
    ///交易所代码  
    TThostFtdcExchangeIDType ExchangeID;  
    ///组合类型  
    TThostFtdcDceCombinationType Type CombinationType;  
    ///产品代码  
    TThostFtdcInstrumentIDType ProductID;  
    ///保证金优惠比例  
    TThostFtdcRatioType Xparameter;  
};
```

2) 新增加//查询申请组合合约

```
struct CThostFtdcQryComblnstrumentField  
{  
    ///交易所代码  
    TThostFtdcExchangeIDType ExchangeID;  
    ///产品代码  
    TThostFtdcInstrumentIDType ProductID;  
};
```

3) //输入的询价增加一字段 BusinessUnit，可以指定报单席位；

```
struct CThostFtdcInputForQuoteField
```

```
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///合约代码  
    TThostFtdcInstrumentIDType InstrumentID;  
    ///询价引用  
    TThostFtdcOrderRefType ForQuoteRef;  
    ///用户代码  
    TThostFtdcUserIDType UserID;  
    ///交易所代码  
    TThostFtdcExchangeIDType ExchangeID;  
    ///投资单元代码  
    TThostFtdcInvestUnitIDType InvestUnitID;  
    ///IP 地址  
    TThostFtdcIPAddressType IPAddress;  
    ///Mac 地址  
    TThostFtdcMacAddressType MacAddress;  
    ///业务单元--new  
    TThostFtdcBusinessUnitType BusinessUnit;  
};
```

4) ///询价：增加一字段 BusinessUnit，目前作为席位号返回；

```
struct CThostFtdcForQuoteField  
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///合约代码  
    TThostFtdcInstrumentIDType InstrumentID;  
    ///询价引用
```

```
TThostFtdcOrderRefType ForQuoteRef;  
///用户代码  
TThostFtdcUserIDType UserID;  
///本地询价编号  
TThostFtdcOrderLocalIDType ForQuoteLocalID;  
///交易所代码  
TThostFtdcExchangeIDType ExchangeID;  
///会员代码  
TThostFtdcParticipantIDType ParticipantID;  
///客户代码  
TThostFtdcClientIDType ClientID;  
///合约在交易所的代码  
TThostFtdcExchangeInstIDType ExchangeInstID;  
///交易所交易员代码  
TThostFtdcTraderIDType TraderID;  
///安装编号  
TThostFtdcInstallIDType InstallID;  
///报单日期  
TThostFtdcDateType InsertDate;  
///插入时间  
TThostFtdcTimeType InsertTime;  
///询价状态  
TThostFtdcForQuoteStatusType ForQuoteStatus;  
///前置编号  
TThostFtdcFrontIDType FrontID;  
///会话编号  
TThostFtdcSessionIDType SessionID;  
///状态信息  
TThostFtdcErrorMsgType StatusMsg;  
///操作用户代码  
TThostFtdcUserIDType ActiveUserID;  
///经纪公司询价编号  
TThostFtdcSequenceNoType BrokerForQutoSeq;
```

```
///投资单元代码
TThostFtdcInvestUnitIDType InvestUnitID;
///IP 地址
TThostFtdcIPAddressType IPAddress;
///Mac 地址
TThostFtdcMacAddressType MacAddress;
///业务单元--new
TThostFtdcBusinessUnitType BusinessUnit;
};
```

5) ///输入的申请组合：增加一字段 BusinessUnit，可以指定报单席位；增加投资单元代码

InvestUnitID、前值编号 FrontID、会话编号 SessionID 更方便运维标记定位组合；

```
struct CThostFtdcInputCombActionField
```

```
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///组合引用
    TThostFtdcOrderRefType CombActionRef;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///买卖方向
    TThostFtdcDirectionType Direction;
    ///数量
    TThostFtdcVolumeTypeVolume;
    ///组合指令方向
    TThostFtdcCombDirectionTypeCombDirection;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///业务单元--new
};
```

```
TThostFtdcBusinessUnitType BusinessUnit;  
///交易所代码  
TThostFtdcExchangeIDType ExchangeID;  
///IP 地址  
TThostFtdcIPAddressType IPAddress;  
///Mac 地址  
TThostFtdcMacAddressType MacAddress;  
///投资单元代码--new  
TThostFtdcInvestUnitIDType InvestUnitID;  
///前置编号--new  
TThostFtdcFrontIDType FrontID;  
///会话编号--new  
TThostFtdcSessionIDType SessionID;  
};
```

6) ///申请组合：增加组合编号 ComTradeID；

```
struct CThostFtdcCombActionField  
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///合约代码  
    TThostFtdcInstrumentIDType InstrumentID;  
    ///组合引用  
    TThostFtdcOrderRefType CombActionRef;  
    ///用户代码  
    TThostFtdcUserIDType UserID;  
    ///买卖方向  
    TThostFtdcDirectionType Direction;  
    ///数量  
    TThostFtdcVolumeTypeVolume;  
    ///组合指令方向
```

```
TThostFtdcCombDirectionTypeCombDirection;
///投机套保标志
TThostFtdcHedgeFlagType HedgeFlag;
///业务单元--new
TThostFtdcBusinessUnitType BusinessUnit;
///本地申请组合编号
TThostFtdcOrderLocalIDType ActionLocalID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///会员代码
TThostFtdcParticipantIDType ParticipantID;
///客户代码
TThostFtdcClientIDType ClientID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所交易员代码
TThostFtdcTraderIDType TraderID;
///安装编号
TThostFtdcInstallIDType InstallID;
///组合状态
TThostFtdcOrderActionStatusType ActionStatus;
///报单提示序号
TThostFtdcSequenceNoType NotifySequence;
///交易日
TThostFtdcDateType TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;
///序号
TThostFtdcSequenceNoType SequenceNo;
///前置编号
TThostFtdcFrontIDTypeFrontID;
///会话编号
TThostFtdcSessionIDType SessionID;
```

```
///用户端产品信息
TThostFtdcProductInfoType UserProductInfo;
///状态信息
TThostFtdcErrorMsgType StatusMsg;
///IP 地址
TThostFtdcIPAddressType IPAddress;
///Mac 地址
TThostFtdcMacAddressType MacAddress;
///组合编号--new
TThostFtdcTradeIDType ComTradeID;
};
```

7) 新增加///组合单腿汇总

struct CThostFtdcInvestorPositionForCombField 【特别说明，此处为 V1.5.8 版本，在 V1.5.9 版本中增加了配合郑商所组合业务需要的 3 个字段，参见下一章节】

```
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType LegInstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType LegHedgeFlag;
    ///买卖方向
    TThostFtdcDirectionType LegDirection;
    ///数量
    TThostFtdcVolumeType TotalAmt;
    //LegID
    TThostFtdcLegIDType LegID;
    ///组合优先级
};
```

```
TThostFtdcTradeGroupIDType TradeGroupID;  
///组合合约代码  
TThostFtdcInstrumentIDType CombInstrumentID;  
///组合投机套保标志  
TThostFtdcHedgeFlagType CombHedgeFlag;  
///组合类型  
TThostFtdcCombinationTypeType CombinationType;  
};
```

8) 新增加///申请组合单腿汇总查询

```
struct CThostFtdcQryInvestorPositionForCombField  
{  
    ///交易所代码  
    TThostFtdcExchangeIDType ExchangeID;  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///单腿合约代码  
    TThostFtdcInstrumentIDType LegInstrumentID;  
};
```

(三) ThostFtdcTraderApi.h 交易头文件变更，新增 4 个接口及对应回调接口 5 个,有条件使用 2 个接口

1) 新增接口如下：

① ///申请组合录入请求

```
virtual int ReqCombActionInsert(CThostFtdcInputCombActionField  
*pInputCombAction, int nRequestID) = 0;
```

② ///请求查询申请组合合约，根据产品查询

```
virtual int ReqQryCombInstrument(CThostFtdcQryCombInstrumentField*  
pQryCombInstrument, int nRequestID) = 0;
```

③ ///请求查询申请组合

```
virtual int ReqQryCombAction(CThostFtdcQryCombActionField*
```

```
pQryCombAction, int nRequestID) = 0;  
④ ///请求查询单腿汇总  
virtual int
```

```
ReqQryInvestorPositionForComb(CThostFtdcQryInvestorPositionForCombFie  
ld* pQryIPForComb, int nRequestID) = 0;
```

2) 新增回调接口如下：

① ///申请组合通知

```
virtual void OnRtnCombAction(CThostFtdcCombActionField  
*pCombAction);
```

② ///请求查询申请组合合约响应

```
virtual void OnRspQryCombiInstrument(CThostFtdcCombiInstrumentField  
*pCombiInstrument, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast);
```

③ ///请求查询申请组合响应

```
virtual void OnRspQryCombAction(CThostFtdcCombActionField  
*pCombAction, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast);
```

④ ///请求查询组合单腿汇总表响应

```
virtual void  
OnRspQryInvestorPositionForComb(CThostFtdcInvestorPositionForCombFiel  
d *pCombAction, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast);
```

⑤ ///申请组合录入请求响应

```
virtual void  
OnRspCombActionInsert(CThostFtdcInputCombActionField  
*pInputCombAction, CThostFtdcRspInfoField *pRspInfo, int nRequestID,  
bool bIsLast);
```

3) 接口及回调接口有条件使用，如下：

为了优化系统，兼顾性能和业务，对组保持仓接口变化如下：

- ① 在期货公司 CTP Mini 交易后台设置 cal_real_dce_cost 为 “no” 条件下，大商所持仓明细后头不依据交易实时维护更新，持仓明细查询接口及回调接口返回数据为无效数据，包括：ReqQryInvestorPositionDetail、OnRspQryInvestorPositionDetail
- ② 组合持仓明细查询接口及回调接口返回数据为无效数据，非当日组合明细，后续不再提供支持，包括 ReqQryInvestorPositionCombineDetail 、

OnRspQryInvestorPositionCombineDetail;

③ 针对①②，API 新增接口 ReqQryInvestorPositionForComb 实现类似功能。

2.3.3 V1.5.9

本次版本重大变化就是支持郑商所组保业务，包括展期与互换。

本版本变化包括：

1) ReqOrderInsert/ReqOrderAction 后台增加对郑商所套利单的报撤单支持，包括期货组合与期权组合：

- 对应接口后台相应变化处理，包括 OnReqOrderInsert、OnExchRtnOrder、OnExchErrRtnOrderInsert、OnReqOrderAction；
- 本部分为后台变化，接口没有变化，不受影响；

2) ThostFtdcUserApiDataType.h 头文件增加郑商所组合类型定义：

```
// 期货对锁  
#define THOST_FTDC_CZCECOMBT_SPZ '0'  
// 期货跨期  
#define THOST_FTDC_CZCECOMBT_SPD '1'  
// 期货跨品种  
#define THOST_FTDC_CZCECOMBT_IPS '2'  
// 看涨期权垂直价差  
#define THOST_FTDC_CZCECOMBT_BUL '3'  
// 看跌期权垂直价差  
#define THOST_FTDC_CZCECOMBT_BER '4'  
// 看涨期权水平价差  
#define THOST_FTDC_CZCECOMBT_BLT '5'  
// 看跌期权水平价差  
#define THOST_FTDC_CZCECOMBT_BRT '6'  
// 期权跨式组合  
#define THOST_FTDC_CZCECOMBT_STD '7'  
// 期权宽跨式组合  
#define THOST_FTDC_CZCECOMBT_STG '8'  
// 备兑  
#define THOST_FTDC_CZCECOMBT_PRT '9'  
typedef char TThostFtdcCzceCombinationTypeType;
```

3) ThostFtdcUserApiStruct.h 头文件中组合单腿汇总结构变化，配合郑商所组合业务，

最后增加了 3 个字段，如下：

```
struct CThostFtdcInvestorPositionForCombField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType LegInstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType LegHedgeFlag;
    ///买卖方向
    TThostFtdcDirectionType LegDirection;
    ///数量
    TThostFtdcVolumeType TotalAmt;
    //LegID
    TThostFtdcLegIDType LegID;
    ///组合优先级
    TThostFtdcTradeGroupIDType TradeGroupID;
    ///组合合约代码
    TThostFtdcInstrumentIDType CombInstrumentID;
    ///组合投机套保标志
    TThostFtdcHedgeFlagType CombHedgeFlag;
    ///组合类型
    TThostFtdcCombinationTypeType CombinationType;
    ///组合成交编号
    TThostFtdcTradeIDType CombTradeID;
    ///成交日期
    TThostFtdcDateType OpenDate;
    ///成交编号
    TThostFtdcTradeIDType TradeID;
```

```
};
```

2.3.4 V1.6.0

本次版本为了方便终端根据交易所错误代码排查问题，CTP Mini API 返回错误信息中增加交易所错误码，ErrorMsg=[交易所错误码]错误信息。

本次版本开始支持郑商所询价价差限制，对应的 CTP Mini API 增加询价价差查询接口，具体如下：

- ① ThostFtdcUserApiStruct.h 增加询价价差查询相关两结构

```
///查询询价价差
struct CThostFtdcQryForQuoteParamField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

///询价价差
struct CThostFtdcForQuoteParamField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///最新价
    TThostFtdcPriceType LastPrice;
    ///价差
    TThostFtdcPriceType PriceInterval;
};
```

- ② ThostFtdcTraderApi.h 增加询价价差查询及查询响应接口

```
///请求查询询价价差接口
```

```
virtual int  
ReqQryForQuoteParam(CThostFtdcQryForQuoteParamField  
*pQryForQuoteParam, int nRequestID) = 0;  
///请求查询询价价差响应接口  
  
virtual void  
OnRspQryForQuoteParam(CThostFtdcForQuoteParamField  
*pForQuoteParam, CThostFtdcRspInfoField *pRspInfo, int  
nRequestID, bool bIsLast) {};
```

2.3.5 V1.6.1

本次版本有两个接口变动：

- 1) 大商所、郑商所组合交易相关的接口优化，接口 ReqQryInvestorPositionCombineDetail 及回调 RspQryInvestorPositionCombineDetail 不再支持；
- 2) 做市商特色需求新增接口，包括以下接口

- 做市商批量报单操作请求及其响应接口

```
virtual int ReqMKBatchOrderAction(CThostFtdcMKInputOrderActionField  
*pMKInputOrderAction, int nRequestID);  
  
virtual void OnRspMKBatchOrderAction  
( CThostFtdcMKInputOrderActionField *pMKInputOrderAction,  
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

- 订阅席位流控消息及其响应接口

```
virtual int SubscribeFlowCtrlWarning(char *ppTraderID[], int nCount);  
  
virtual void OnRspSubscribeFlowCtrlWarning  
( CThostFtdcSpecificTraderField *pRspSubscribeTraderField,  
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

- 取消订阅席位流控消息及其响应接口：

```
virtual int UnSubscribeFlowCtrlWarning(char *ppTraderID[], int nCount);  
  
virtual void OnRspUnSubscribeFlowCtrlWarning  
( CThostFtdcSpecificTraderField *pRspSubscribeTraderField,  
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

- 席位流控消息推送

```
virtual void OnRtnFlowCtrlWarning( CThostFtdcFlowCtrlWarningField  
*pFlowCtrlWarning);
```

以上接口变化参见 `ThostFtdcTraderApi.h`, 除此之外增加了接口相关的参数及数据结构定义
联系地址：浦东新区乐昌路 399 号 上海期货信息技术有限公司

义，具体参见头文件 ThostFtdcUserApiDataType.h、ThostFtdcUserApiStruct.h。

配合做市商批量接口需要，对现有响应信息结构进行了扩展，增加了字段，具体如下：

```
struct CThostFtdcRspInfoField
{
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
    ///新增加字段：批量交易成功记录数
    TThostFtdcRecordCountType RecordCount;
};
```

2.3.6 V1.6.2

本次版本新增加做市商席位连接状态查询接口：

- 1) ThostFtdcUserApiStruct.h 中修改 struct CThostFtdcQryTraderOfferField；

 ///查询交易员席位信息

```
struct CThostFtdcQryTraderOfferField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易所交易员代码
    TThostFtdcTraderIDType TraderID;
};
```

- 2) ThostFtdcTraderApi.h 中增加 ReqQryTraderOffer 及 OnRspQryTraderOffer 响应接口：

 ///请求查询交易员席位状态响应

```
    virtual void OnRspQryTraderOffer(CThostFtdcTraderOfferField
*pTraderOffer, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
bIsLast) {};
```

 ///请求查询交易员席位状态

```
    virtual int ReqQryTraderOffer(CThostFtdcQryTraderOfferField
*pQryTraderOffer, int nRequestID) = 0;
```

本功能受后台交易系统配置 tserver.ini 中的 mk_trading_mode=yes（开启做市功能）影响，没有开启则无法查询。

另外，本版本修复了穿透式监管中 windows 采集库内存泄漏问题，需要终端开发更新 API，重新编译。

2.3.7 V1.6.3

本版本增加郑商所查询投资者 SPBM 品种明细接口、报价衍生单成交回报中报单系统号增加 a、b 标识的功能：

- 1) 查询投资者 SPBM 品种明细接口变化

- a) ///请求查询投资者 SPBM 品种明细

```
virtual int  
ReqQryInvestorProdSPBMDetail( CThostFtdcQryInvestorProdSPBMDetail  
Field *pQryInvestorProdSPBMDetail, int nRequestID );
```

- b) 请求查询投资者 SPBM 品种明细响应

```
virtual void  
OnRspQryInvestorProdSPBMDetail(CThostFtdcInvestorProdSPBMDetailFi  
eld *pInvestorProdSPBMDetail, CThostFtdcRspInfoField *pRspInfo, int  
nRequestID, bool bIsLast);
```

- c) 相应增加数据结构 CThostFtdcQryInvestorProdSPBMDetailField、
CThostFtdcInvestorProdSPBMDetailField；

- 2) 郑商所报价衍生单成交回报中报单系统号增加 a、b 标识：

- a) tserver 在收到郑商所报价衍生单成交回报后，将卖报价衍生单 OrderSysID 前加“a”，买报价衍生单前加“b”，然后将衍生单回报通过 CTP Mini API 推送给终端；
 - b) 处理原则与 CTP 主席一致；
 - c) 变更受影响接口为： 3.2.37OnRtnTrade。

2.3.8 V1.6.4

本版本增加对广期所业务的支持。

- 头文件 ThostFtdcUserApiStruct.h 中的 struct CThostFtdcRspUserLoginField 增加了广期所官方要求的广期所时间字段：
 - ///广期所时间
 - TThostFtdcTimeType GFEXTIME;
- 受影响接口为： OnRspUserLogin；
- 本变更兼容之前 CTP Mini API 版本。

2.3.9 V1.6.6

本版本支持上期所及能源中心 SPMM 新型组保，支持 SPMM 保证金明细查询，接口如下：

- a) 增加 SPMM 保证金明细查询接口 ReqQrySPMMInvestorCommodityGroupMargin 及响应接口 OnRspQrySPMMInvestorCommodityGroupMargin；

支持报单和报价中客户自定义字段：

- b) 支持报单和报价客户自定义字段：报单请求域增加 CustomOrderRef 字段，报价请求域增加 CustomQuoteRef 字段。相关结构字段变化如下：

```
struct CThostFtdcInputOrderField
```

```
{  
  
    ///经纪公司代码  
  
    TThostFtdcBrokerIDType BrokerID;  
  
    ..... // 中间部分省略  
  
    ///Mac 地址  
  
    TThostFtdcMacAddressType MacAddress;  
  
    ///自定义字段  
  
    TThostFtdcCustomOrderRefType CustomOrderRef;  
  
};
```

```
struct CThostFtdcOrderField
```

```
{  
  
    ///经纪公司代码  
  
    TThostFtdcBrokerIDType BrokerID;  
  
    ..... // 中间部分省略  
  
    ///Mac 地址
```

```
TTHostFtdcMacAddressType MacAddress;

///自定义字段

TTHostFtdcCustomOrderRefType CustomOrderRef;

};

struct CThostFtdcInputQuoteField

{

///经纪公司代码

TTHostFtdcBrokerIDType BrokerID;

..... // 中间部分省略

//顶单编号

TTHostFtdcOrderSysIDType ReplaceSysID;

///自定义字段

TTHostFtdcCustomOrderRefType CustomQuoteRef;

};

struct CThostFtdcQuoteField

{

///经纪公司代码

TTHostFtdcBrokerIDType BrokerID;

..... // 中间部分省略

///Mac 地址

TTHostFtdcMacAddressType MacAddress;
```

```
//顶单编号  
  
TThostFtdcOrderSysIDType ReplaceSysID;  
  
///自定义字段  
  
TThostFtdcCustomOrderRefType CustomQuoteRef;  
  
};
```

2.3.10 V1.6.7

1. 本次更新涉及新型组合保证金，对应的 CTP Mini API 新增加 RULE 相关接口：
a) ///请求查询投资者 RULE 保证金明细

```
① Virtual int  
ReqQryRULEInvestorProdMargin(CThostFtdcQryRULEInvestorProdMarginField* pQryRULEInvestorProdMargin, int nRequestID) = 0;
```

- b) ///请求查询投资者 RULE 保证金响应

```
① virtual void  
OnRspQryRULEInvestorProdMargin(CThostFtdcRULEInvestorProdMarginField* pRULEInvestorProdMargin, CThostFtdcRspInfoField* pRspInfo, int nRequestId, bool bIsLast) {};
```

- c) 对应增加数据结构如下：

```
///查询回报的数据中只包含资金相关的信息：  
struct CThostFtdcRULEInvestorProdMarginField  
{  
    ///交易所代码  
    TThostFtdcExchangeIDType ExchangeID;  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///品种代码  
    TThostFtdcProductIDType ProdFamilyCode;  
    ///合约类型  
    TThostFtdcRULEInstrClassType InstrumentClass;  
    ///商品群号
```

```
TThostFtdcRULECommodityGroupIDType CommodityGroupID;
    /// 品种内对锁保证金
    TThostFtdcMoneyType IntraProdMargin;
    /// 品种间对锁保证金
    TThostFtdcMoneyType InterProdMargin;
    /// 跨品种单腿保证金
    TThostFtdcMoneyType SingleMargin;
    /// 非组合合约保证金
    TThostFtdcMoneyType NonCombMargin;
    /// 附加保证金
    TThostFtdcMoneyType AddOnMargin;
    /// 交易所保证金
    TThostFtdcMoneyType ExchMargin;
    /// 开仓冻结保证金
    TThostFtdcMoneyType OpenFrozenMargin;
    /// 平仓冻结保证金
    TThostFtdcMoneyType CloseFrozenMargin;
    ///品种保证金
    TThostFtdcMoneyType Margin;
    /// 冻结保证金
    TThostFtdcMoneyType FrozenMargin;
};


```

```
/// 查询 RULE 产品保证金
struct CThostFtdcQryRULEInvestorProdMarginField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///品种代码
    TThostFtdcProductIDType ProdFamilyCode;
    ///商品群号

```

```
    TThostFtdcRULECommodityGroupIDType CommodityGroupID;  
};
```

2. 本次更新涉及新型组合保证金，对应的 CTP Mini API 新增加 RCAMS 相关接口：

a) //查询投资者 RCAMS 组合保证金

```
① virtual int  
ReqQryRCAMSInvestorProdMargin(CThostFtdcQryRCAMSInvestorProd  
MarginField* pQryRCAMSInvestorProdMargin, int nRequestID) = 0;
```

b) //请求查询投资者 RCAMS 组合保证金响应

```
① virtual void  
OnRspQryRCAMSInvestorProdMargin(CThostFtdcRCAMSInvestorProd  
MarginField *pRCAMSInvestorProdMargin, CThostFtdcRspInfoField  
*pRspInfo, int nRequestId, bool bIsLast) {};
```

c) //查询 RCAMS 策略组合持仓

```
① virtual int  
ReqQryRCAMSInvestorCombPosition(CThostFtdcQryRCAMSInvestorCo  
mbPositionField* pQryRCAMSInvestorCombPosition, int nRequestID) =  
0;
```

d) //请求查询 RCAMS 策略组合持仓响应

```
① virtual void  
OnRspQryRCAMSInvestorCombPosition(CThostFtdcRCAMSInvestorCo  
mbPositionField *pRCAMSInvestorCombPosition,  
CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast) {};
```

e) 配合中金所策略组合申请及拆分，沿用之前版本申请组合相关接口，增加中金所支持，
涉及：ReqCombActionInsert，OnRspCombActionInsert，OnRtnCombAction；
ReqQryCombAction，OnRspQryCombAction；

f) 对应增加数据结构如下：

```
//查询投资者 RCAMS 组合保证金  
struct CThostFtdcQryRCAMSInvestorProdMarginField  
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///产品代码  
    TThostFtdcInstrumentIDType CombProductID;
```

```
//查询 RCAMS 策略组合持仓
struct CThostFtdcQryRCAMSIInvestorCombPositionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///单腿合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};
```

2.3.11 V1.6.8

1. 配合大商所交易指令支持 GIS 属性，在 CTP Mini TradeAPI 中：
 - a) 报价请求与响应增加 TimeCondition 字段并给以赋值：
 - ReqQuoteInsert、OnRspQuoteInsert 接口增加 TimeCondition 字段，支持大商所 GIS 小节交易有效属性；对应《ThostFtdcUserApiDataType.h.h》文件的 THOST_FTDC_TC_GFS 枚举值；
 - struct CThostFtdcInputQuoteField、struct CThostFtdcQuoteField 增加 TimeCondition 字段；

```
struct CThostFtdcInputQuoteField
```

```
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    .....
    //顶单编号
    TThostFtdcOrderSysIDType ReplaceSysID;
```

///自定义字段

TThostFtdcCustomOrderRefType CustomQuoteRef;

///有效期类型---新增加字段

TThostFtdcTimeConditionType TimeCondition;

};

struct CThostFtdcQuoteField

{

///经纪公司代码

TThostFtdcBrokerIDType BrokerID;

///投资者代码

TThostFtdcInvestorIDType InvestorID;

///合约代码

TThostFtdcInstrumentIDType InstrumentID;

.....

///自定义字段

TThostFtdcCustomOrderRefType CustomQuoteRef;

///有效期类型---新增加字段

TThostFtdcTimeConditionType TimeCondition;

};

- b) 报单已有字段 TimeCondition 属性增加 GIS 选项支持;
2. 对做市商支持，新增加查询部分后台功能与设置功能，需要期货公司授权并开通相关开关才可以使用；配套新增加接口：

a) int ReqQryControlParam(CThostFtdcQryControlParamField *pQryControlParam, int nRequestID);
b) void OnRspQryControlParam(CThostFtdcControlParamField *pControlParam, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

c) 对应增加 CThostFtdcQryControlParamField 、
CThostFtdcControlParamField 参数结构如下：

```
///查询经纪公司用户及开关代码
struct CThostFtdcQryControlParamField
{
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///开关代码
    TThostFtdcControlParamIDType ControlParamID;
};

//查询结果响应
struct CThostFtdcControlParamField
{
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///开关代码
    TThostFtdcControlParamIDType ControlParamID;
    ///开关设置
    TThostFtdcControlParamValueType ControlParamValue;
    ///备注说明
    TThostFtdcMemoType Memo;
}
```

3. 终端登录 API 响应接口 OnRspUserLogin 增加返回交易系统类型信息
- a) 现有接口不变动
- b) 接口返回参数 CThostFtdcRspUserLoginField *pRspUserLogin 对应 SystemName 字段赋值为交易系统类型信息；参数值：1-PRD 生产系统 2-

SIM 仿真系统 3-DEV 开发系统。

2.3.12 V1.6.9

- 1) 增加资金变动订阅及推送接口
 - a) 订阅与取消订阅接口：ReqSubscribeFundChange、ReqUnSubscribeFundChange；
 - b) 订阅及取消订阅对应的响应接口：OnRspSubscribeFundChange、OnRspUnSubscribeFundChange；如后台订阅开关关闭，则返回错误“CTP：资金推送功能未开启”，请联系各家后台经纪公司。
 - c) 订阅成功后接收资金变动推送接口：OnRtnFundChange，接口返回字段复用了现有资金账号查询的接口 **CThostFtdcTradingAccountField**；
- 2) 为了实现报单回报中返回成交均价，修改头文件《ThostFtdcUserApiStruct.h》，报单相关的结构体 **CThostFtdcOrderField** 最后增加了成交均价字段：**TThostFtdcPriceType TradeAvgPrice**，详细结构变更如下：

```
///报单
struct CThostFtdcOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    :::::
    ///自定义字段
    TThostFtdcCustomOrderRefType CustomOrderRef;
    ///成交均价
    TThostFtdcPriceType TradeAvgPrice;
};
```

2.3.13 V1.7.0

本次版本为了配合做市商特色交易处理，通过投资者控制是否做自成交控制报单，CTP Mini API 变化如下：

- a) ThostFtdcUserApiDataType.h 头文件增加以下内容：

```
///////////////////////////////
///TThostFtdcOrderFlagType 是一个报单标志类型，未来可能还有其他扩展
///字符串中的第一个字符代表自成交相关的设置，如设置为'1'则不做自成交检
查；
```

```
///////////////////////////////
```

```
//不做自成交检查
```

```
#define THOST_FTDC_SELFTTRADE_AvoidSelfTradeCheck '1'
```

```
    typedef char TThostFtdcOrderFlagType[21];
```

- b) ThostFtdcUserApiStruct.h 头文件中的报单结构体最后增加 OrderFlag 字段，具体如下：

```
///输入报单
```

```
struct CThostFtdcInputOrderField
```

```
{
```

```
///经纪公司代码
```

```
TThostFtdcBrokerIDType BrokerID;
```

```
///投资者代码
```

```
TThostFtdcInvestorIDType InvestorID;
```

```
.....
```

```
///自定义字段
```

```
TThostFtdcCustomOrderRefType CustomOrderRef;
```

```
///报单标志
```

```
TThostFtdcOrderFlagType OrderFlag;
```

```
};
```

- c) 本次 CTP Mini API 版本变化兼容之前版本。

2.3.14 V1.7.3

本版本更新错误码对应表。

2.3.15 V1.7.3-P2

更新错误码对应表，新增加大商所对冲接口变化的错误码返回信息。

支持大商所行权优化二阶段业务，交易所期权对冲业务变化如下：

- a) 升级后对冲设置的颗粒度更加细化，对冲功能支持变化如下图示意：



- b) 新增期权放弃申请功能，取消“取消到期日自动行权申请”：增加期权放弃申请功能，仅合约到期日有效；
- c) 期权对冲功能：交易 API 端丰富短效申请功能（支持合约，系列，品种级别的设置粒度，合约级别可设置手数），设置后仅当日有效。业务含义和业务处理与现有生产一致。
- d) 行权后期货对冲：取消原行权申请时设置行权后对冲功能。交易 API 端新增短效申请功能（支持合约，品种，交易编码级别的设置粒度），设置后仅当日有效。业务含义和业务处理与现有生产一致。
- e) 履约后期货对冲：交易 API 端丰富短效申请功能（支持合约，品种，交易编码级别的设置粒度），设置后仅当日有效。业务含义和业务处理与现有生产一致。
- f) 期货对冲功能：交易 API 端新增短效申请功能（支持合约，品种，交易编码级别的设置粒度，合约级别可设置手数），设置后仅当日有效。该功能为新增功能，业务含义为对双向期货持仓进行对冲平仓（不依赖于行权或者履约行为）。

配合大商所行权二期业务，CTP Mini API 接口变化较大，请需要行权业务的终端投资者务必对应升级改造，具体 CTP Mini API 接口变化如下：

- a) 新增接口及响应接口：

```

///对冲设置请求
virtual int ReqOffsetSetting(CThostFtdcInputOffsetSettingField
    *pInputOffsetSetting, int nRequestID) = 0;

///撤销对冲设置请求
virtual int ReqCancelOffsetSetting(CThostFtdcInputOffsetSettingField
    *pInputOffsetSetting, int nRequestID) = 0;

///请求查询对冲设置
virtual int ReqQryOffsetSetting(CThostFtdcQryOffsetSettingField
    *pQryOffsetSetting, int nRequestID) = 0;

///对冲设置请求响应

```

```
virtual void OnRspOffsetSetting(CThostFtdcInputOffsetSettingField  
*pInputOffsetSetting, CThostFtdcRspInfoField *pRspInfo, int nRequestID,  
bool bIsLast) {};  
///对冲设置撤销请求响应  
virtual void OnRspCancelOffsetSetting(CThostFtdcInputOffsetSettingField  
*pInputOffsetSetting, CThostFtdcRspInfoField *pRspInfo, int nRequestID,  
bool bIsLast) {};  
///对冲设置通知  
virtual void OnRtnOffsetSetting(CThostFtdcOffsetSettingField  
*pOffsetSetting) {};  
///对冲设置撤销错误回报  
virtual void  
OnErrRtnCancelOffsetSetting(CThostFtdcCancelOffsetSettingField  
*pCancelOffsetSetting, CThostFtdcRspInfoField *pRspInfo) {};  
///请求查询对冲设置响应  
virtual void OnRspQryOffsetSetting(CThostFtdcOffsetSettingField  
*pOffsetSetting, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast) {};
```

b) ThostFtdcUserApiStruct.h 新增：

```
///输入的对冲设置  
struct CThostFtdcInputOffsetSettingField  
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///合约代码  
    TThostFtdcInstrumentIDType InstrumentID;  
    ///标的期货合约代码  
    TThostFtdcInstrumentIDType UnderlyingInstrID;  
    ///产品代码  
    TThostFtdcProductIDType ProductID;  
    ///对冲类型  
    TThostFtdcOffsetTypeType OffsetType;  
    ///申请对冲的合约数量
```

```
TThostFtdcVolumeType    Volume;
///是否对冲
TThostFtdcBoolType   IsOffset;
///请求编号
TThostFtdcRequestIDType RequestID;
///用户代码
TThostFtdcUserIDType    UserID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///IP 地址
TThostFtdcIPAddressType IPAddress;
///Mac 地址
TThostFtdcMacAddressType MacAddress;
};

///对冲设置
struct CThostFtdcOffsetSettingField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///标的期货合约代码
    TThostFtdcInstrumentIDType UnderlyingInstrID;
    ///产品代码
    TThostFtdcProductIDType ProductID;
    ///对冲类型
    TThostFtdcOffsetTypeOffsetType OffsetType;
    ///申请对冲的合约数量
    TThostFtdcVolumeType    Volume;
    ///是否对冲
};
```

```
TThostFtdcBoolType IsOffset;
///请求编号
TThostFtdcRequestIDType RequestID;
///用户代码
TThostFtdcUserIDType UserID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
/// 业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///IP 地址
TThostFtdcIPAddressType IPAddress;
///Mac 地址
TThostFtdcMacAddressType MacAddress;
///交易所合约代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所期权系列号
TThostFtdcExchangeInstIDType ExchangeSerialNo;
///交易所产品代码
TThostFtdcProductIDType ExchangeProductID;
///会员代码
TThostFtdcParticipantIDType ParticipantID;
///客户代码
TThostFtdcClientIDType ClientID;
///交易所交易员代码
TThostFtdcTraderIDType TraderID;
///安装编号
TThostFtdcInstallIDType InstallID;
///对冲提交状态
TThostFtdcOrderSubmitStatusType OrderSubmitStatus;
///交易日
TThostFtdcDateType TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;
///报单日期
TThostFtdcDateType InsertDate;
```

```
    ///插入时间  
    TThostFtdcTimeType InsertTime;  
    ///撤销时间  
    TThostFtdcTimeType CancelTime;  
    ///对冲设置结果  
    TThostFtdcExecResultType ExecResult;  
    ///序号  
    TThostFtdcSequenceNoType SequenceNo;  
    ///前置编号  
    TThostFtdcFrontIDType FrontID;  
    ///会话编号  
    TThostFtdcSessionIDType SessionID;  
    ///状态信息  
    TThostFtdcErrorMsgType StatusMsg;  
    ///操作用户代码  
    TThostFtdcUserIDType ActiveUserID;  
    ///经纪公司报单编号  
    TThostFtdcSequenceNoType BrokerOffsetSettingSeq;  
    ///申请来源  
    TThostFtdcApplySrcType ApplySrc;  
};  
  
///撤销对冲设置  
struct CThostFtdcCancelOffsetSettingField  
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///合约代码  
    TThostFtdcInstrumentIDType InstrumentID;  
    ///标的期货合约代码  
    TThostFtdcInstrumentIDType UnderlyingInstrID;  
    ///产品代码  
    TThostFtdcProductIDType ProductID;
```

```
///对冲类型
TThostFtdcOffsetTypeOffsetType;
///申请对冲的合约数量
TThostFtdcVolumeType    Volume;
///是否对冲
TThostFtdcBoolType     IsOffset;
///请求编号
TThostFtdcRequestIDType RequestID;
///用户代码
TThostFtdcUserIDType UserID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///IP 地址
TThostFtdcIPAddressType IPAddress;
///Mac 地址
TThostFtdcMacAddressType MacAddress;
///交易所合约代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所期权系列号
TThostFtdcExchangeInstIDType ExchangeSerialNo;
///交易所产品代码
TThostFtdcProductIDType ExchangeProductID;
///交易所交易员代码
TThostFtdcTraderIDType   TraderID;
///安装编号
TThostFtdcInstallIDType InstallID;
///会员代码
TThostFtdcParticipantIDType ParticipantID;
///客户代码
TThostFtdcClientIDType   ClientID;
///报单操作状态
TThostFtdcOrderActionStatusType OrderActionStatus;
///状态信息
TThostFtdcErrorMsgType   StatusMsg;
///操作本地编号
```

```
TThostFtdcOrderLocalIDType ActionLocalID;  
///操作日期  
TThostFtdcDateType ActionDate;  
///操作时间  
TThostFtdcTimeType ActionTime;  
};  
  
///查询对冲设置  
struct CThostFtdcQryOffsetSettingField  
{  
    ///经纪公司代码  
    TThostFtdcBrokerIDType BrokerID;  
    ///投资者代码  
    TThostFtdcInvestorIDType InvestorID;  
    ///产品代码  
    TThostFtdcProductIDType ProductID;  
    ///对冲类型  
    TThostFtdcOffsetTypeType OffsetType;  
};
```

c) ThostFtdcUserApiDataType.h 新增：

```
////////////////////////////////////////////////////////////////////////  
///TFtdcOffsetTypeType 是一个对冲类型类型  
////////////////////////////////////////////////////////////////////////  
///期权对冲  
#define THOST_FTDC_OT_OPT_OFFSET '0'  
///期货对冲  
#define THOST_FTDC_OT_FUT_OFFSET '1'  
///行权后期货对冲  
#define THOST_FTDC_OT_EXEC_OFFSET '2'  
///履约后期货对冲  
#define THOST_FTDC_OT_PERFORM_OFFSET '3'  
  
typedef char TThostFtdcOffsetTypeType;
```

```
///////////////////////////////  
///TFtdcApplySrcType 是一个申请来源类型  
///////////////////////////////  
///交易  
#define THOST_FTDC_AS_Trade '0'  
///会服  
#define THOST_FTDC_AS_Member '1'  
  
typedef char TThostFtdcApplySrcType;
```

- d) 在跟随大商所 API 升级到 V1.4.4 版本后，现有接口变化概要如下：
- i. 取消到期日自动行权的报、撤、查业务接口皆不再支持；
 - ii. 现有自对冲接口 ReqOptionSelfCloseInsert 对大商所不可用，其他交易所不变；
 - iii. 新增 ReqOffsetSetting 接口，支持大商所期权对冲、期货对冲、行权后期货对冲、履约后期货对冲；
 - iv. 大商所期权行权调用现有接口 ReqExecOrderInsert，CloseFlag 字段只能填写 ‘1’，否则报错；增加期权放弃申请功能（ActionType= ‘2’）。

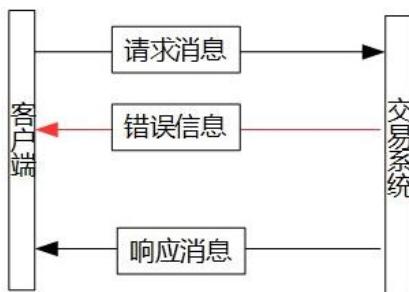
2.4 网络通信

2.4.1 通讯模式

CTP Mini 涉及的通讯模式有三种：

- 1) 对话通讯模式
- 2) 私有通讯模式
- 3) 广播通讯模式

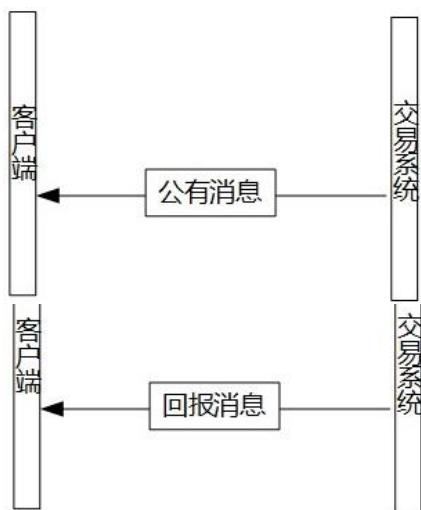
对话通讯模式是指由终端程序主动发起的通讯请求。该请求被 CTP Mini 服务端接收和处理，并给予响应。例如报单、查询等。这种通讯模式与普通的客户 / 服务器模式相同。



私有通讯模式是指服务端主动向某个特定的客户端发出的信息。例如成交回报等。

广播通讯模式是指服务端主动向所有客户端都发出相同的信息。例如合约状态通知。

通讯模式和网络的连接不一定存在简单的一对一的关系。也就是说，一个网络连接中可能



传送多种不同通讯模式的报文，一种通讯模式的报文也可以在多个不同的连接中传送。

CTP Mini API 和后台之间通讯过程都可以简化下图所示：



一般来说，CTP Mini 交易系统中对话模式下被返回的消息成为**响应**。而私有模式和广播模式下被返回的消息被称为**回报**。

2.4.2 通讯实现

对应三种通讯模式，TraderAPI 提供了两个类，分别为：

- CThostFtdcTraderApi
- CThostFtdcTraderSpi

这两个接口对通信协议进行了封装，方便客户端应用程序的开发。客户端通过 CThostFtdcTraderApi 发出操作请求，通过继承 CThostFtdcTraderSpi 并重载回调函数来处理后台服务的响应，进而实现三种通讯模式。

2.4.2.1 对话模式

对话通讯模式下支持对话数据流和查询数据流：

- 对话数据流是一个双向数据流，交易终端发送交易请求，交易系统响应应答，但不维护对话流的状态。系统故障时，对话数据流会重置，通讯途中的数据可能会丢失。
- 查询数据流是一个双向数据流，交易终端发送查询请求，交易系统反馈应答，但不维护查询流的状态。系统故障时，查询数据流会重置，通讯途中的数据可能会丢失。查询数据流只允许同步查询，上一条查询结束后再进行下一次查询，不支持并发查询。

通过对话流进行通讯的编程接口通常如下：

- 请求：

```
int CThostFtdcTraderApi::ReqXXX(CThostFtdcXXXField *pReqXXX, int nRequestID)
```

其中请求接口各参数规则如下：

- ✓ 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具有不同的类；
- ✓ 第二个参数为请求号。请求号由客户端应用程序负责维护，正常情况下每个请求的请求号不要重复，从而可以将响应与请求对应起来。

- 响应：

```
Void CThostFtdcTraderSpi::OnRspXXX(CThostFtdcXXXField
```

```
*pRspXXX, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
```

当收到后台服务应答时，CThostFtdcTraderSpi 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。回调函数的参数规则如下：

- ✓ 第一个参数为响应的具体数据，如果出错或最后一条数据有可能为 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；
- ✓ 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。

2.4.2.2 私有模式

私有通讯模式下支持私有数据流。

私有流是一个单向数据流，由交易系统发向交易终端，用于传送交易员私有的通知和回报信息。私有流是一个可靠的数据流，交易系统维护每个用户的私有流，在一个交易日内，交易终端断线后恢复连接时，可以请求交易系统发送指定序号之后的私有流数据。私有数据流向交易终端提供报单状态报告、成交回报等信息。

私有流中的私有信息包括报单回报、成交回报、报价回报、执行宣告回报等等。

通过私有流接回报的编程接口通常如下：

```
void CThostFtdcTraderApi::OnRtnXXX(CThostFtdcXXXField *pXXX);  
// 或  
void ThostFtdcTraderApi::OnErrRtnXXX(CThostFtdcXXXField pXXX,  
CThostFtdcRspInfoField *pRspInfo);
```

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

2.4.2.3 广播模式

广播通讯模式下支持公共数据流。

公共数据流是一个单向数据流，由交易系统发向交易终端，用于推送信息：公共数据流也是一个可靠的数据流，交易系统维护整个系统的公共数据流，在一个交易日内，交易终端断线恢复连接时，可以请求交易系统发送指定序号之后的公共数据流数据。

公共流中的公共信息包括合约状态通知等。

通过公共流接回报的编程接口通常如下：

```
void CThostFtdcTraderApi::OnRtnXXX(CThostFtdcXXXField *pXXX);
```

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

2.5 接口分类

2.5.1 管理接口

2.5.1.1 TraderAPI

TraderAPI 中的管理类接口包括以下内容：

接口类型	接口名称	说明
	CThostFtdcTraderApi::CreateFtdcTraderApi	创建 TraderApi 实例

生命周期管 理接口	CThostFtdcTraderApi::GetVersion	获取 API 版本
	CThostFtdcTraderApi::Release	删除接口实例
	CThostFtdcTraderApi::Init	初始化
	CThostFtdcTraderApi::Join	等待接口线程结束运行
参数管理接 口	CThostFtdcTraderApi::RegisterFront	注册前置机网络地址
	CThostFtdcTraderApi::RegisterSpi	注册回调接口
	CThostFtdcTraderApi::GetTradingDay	获取当前交易日，只有登录成功后，才能得到正确的交易日
订阅接口	CThostFtdcTraderApi::SubscribePrivateTopic	订阅私有流
	CThostFtdcTraderApi::SubscribePublicTopic	订阅公共流
通信状态接 口	CThostFtdcTraderSpi::OnFrontConnected	与交易系统建立起通信连接时（还未录前），该方法被调用
	CThostFtdcTraderSpi::OnFrontDisconnected	与交易系统通信连接断开时，该方法调用

2.5.1.2 MdAPI

管理接口是对 API 的生命周期和运行参数进行控制，MDAPI 包括以下列表内容：

接口类型	接口名称	说明
生命周期管 理接口	CThostFtdcMdApi::CreateFtdcMdApi	创建 MdApi 实例
	CThostFtdcMdApi::GetApiVersion	获取 API 版本
	CThostFtdcMdApi::Release	删除接口实例
	CThostFtdcMdApi::Init	初始化
参数管理接 口	CThostFtdcMdApi::Join	等待接口线程结束运行
	CThostFtdcMdApi::RegisterFront	注册前置机网络地址
	CThostFtdcMdApi::RegisterSpi	注册回调接口
通信状态接 口	CThostFtdcMdApi::GetTradingDay	获取当前交易日，只有登录成功后，才能得到正确的交易日
	CThostFtdcMdSpi::OnFrontConnected	与交易系统建立起通信连接时（还未录前），该方法被调用
	CThostFtdcMdSpi::OnFrontDisconnected	与交易系统通信连接断开时，该方法调用

2.5.2 业务接口

2.5.2.1 TraderAPI

和交易业务相关的接口，简要如下列表：

业务类型	业务	请求接口/响应接口	数据流
登录	登录	CThostFtdcTraderApi::ReqUserLogin 用户登录请求	对话流
		CThostFtdcTraderSpi::OnRspUserLogin 用户登录请求响应	
		CThostFtdcTraderApi::ReqUserLoginEncrypt 用户加密登录请求	
客户端认证		CThostFtdcTraderSpi::OnRspUserLoginEncrypt 用户加密登录请求响应	
登出	CThostFtdcTraderApi::ReqUserLogout 登出请求	对话流	
订阅	订阅主题	CThostFtdcTraderApi::ReqAuthenticate 客户端认证请求	对话流
		CThostFtdcTraderSpi::OnRspAuthenticate 客户端认证请求响应	
订阅	订阅主题	CThostFtdcTraderApi::ReqSubscribeTopic	对话流
		CThostFtdcTraderSpi::OnRspSubscribeTopic	



交易	报单录入	CThostFtdcTraderApi::ReqOrderInsert 报单录入请求 CThostFtdcTraderSpi::OnRspOrderInsert 报单录入请求响应	对话流
	报单操作	CThostFtdcTraderApi::ReqOrderAction 报单操作请求 CThostFtdcTraderSpi::OnRspOrderAction 报单操作请求响应	对话流
	执行宣告录入	CThostFtdcTraderApi::ReqExecOrderInsert 执行宣告录入请求 CThostFtdcTraderSpi::OnRspExecOrderInsert 执行宣告录入请求响应	对话流
	执行宣告操作	CThostFtdcTraderApi::ReqExecOrderAction 执行宣告操作请求 CThostFtdcTraderSpi::OnRspExecOrderAction 执行宣告操作请求响应	对话流
	询价录入	CThostFtdcTraderApi::ReqForQuoteInsert 询价录入请求 CThostFtdcTraderSpi::OnRspForQuoteInsert 询价录入请求响应	对话流
	报价录入	CThostFtdcTraderApi::ReqQuoteInsert 报价录入请求 CThostFtdcTraderSpi::OnRspQuoteInsert 报价录入请求响应	对话流
	报价操作	CThostFtdcTraderApi::ReqQuoteAction 报价操作请求 CThostFtdcTraderSpi::OnRspQuoteAction 报价操作请求响应	对话流
	期权自对冲录入	CThostFtdcTraderApi::ReqOptionSelfCloseInsert 期权自对冲录入请求 CThostFtdcTraderSpi::OnRspOptionSelfCloseInsert 期权自对冲录入请求响应	对话流
	期权自对冲操作	CThostFtdcTraderApi::ReqOptionSelfCloseAction 期权自对冲操作请求 CThostFtdcTraderSpi::OnRspOptionSelfCloseAction 期权自对冲操作请求响应	对话流
	申请组合录入请求	CThostFtdcTraderSpi::ReqCombActionInsert 申请组合录入请求 CThostFtdcTraderSpi::OnRspCombActionInsert 申请组合录入请求响应 CThostFtdcTraderSpi::OnRtnCombAction 申请组合通知	V1.5.8 增加
查询	查询报单	CThostFtdcTraderApi::ReqQryOrder 请求查询报单请求 CThostFtdcTraderSpi::OnRspQryOrder 请求查询报单请求响应	对话流
	查询成交	CThostFtdcTraderApi::ReqQryTrade 请求查询成交请求 CThostFtdcTraderSpi::OnRspQryTrade 请求查询成交请求响应	对话流
	查询投资者持仓	CThostFtdcTraderApi::ReqQryInvestorPosition 查询投资者持仓请求 CThostFtdcTraderSpi::OnRspQryInvestorPosition 查询投资者持仓请求响应	对话流
	查询资金账户	CThostFtdcTraderApi::ReqQryTradingAccount 查询资金账户请求 CThostFtdcTraderSpi::OnRspQryTradingAccount 查询资金账户请求响应	对话流
	查询投资者	CThostFtdcTraderApi::ReqQryInvestor 查询投资者请求 CThostFtdcTraderSpi::OnRspQryInvestor 查询投资者请求响应	对话流
	查询交易编码	CThostFtdcTraderApi::ReqQryTradingCode 查询交易编码请求 CThostFtdcTraderSpi::OnRspQryTradingCode 查询交易编码请求响应	对话流
	查询合约保证金率	CThostFtdcTraderApi::ReqQryInstrumentMarginRate 查询合约保证金率请求 CThostFtdcTraderSpi::OnRspQryInstrumentMarginRate 查询合约保证金率请求响应	对话流

	查询合约手续费率	CThostFtdcTraderApi::ReqQryInstrumentCommissionRate 查询合约手续费率请求 CThostFtdcTraderSpi::OnRspQryInstrumentCommissionRate 请求合约手续费率响应	对话流
	查询交易所	CThostFtdcTraderApi::ReqQryExchange 查询交易所请求 CThostFtdcTraderSpi::OnRspReqQryExchange 查询交易所请求响应	对话流
	查询合约	CThostFtdcTraderApi::ReqQryInstrument 查询合约请求 CThostFtdcTraderSpi::OnRspQryInstrument 查询合约请求响应	对话流
	查询行情	CThostFtdcTraderApi::ReqQryDepthMarketData 查询行情请求 CThostFtdcTraderSpi::OnRspQryDepthMarketData 查询行情请求响应	对话流
	查询期权自对冲	CThostFtdcTraderApi::ReqQryOptionSelfClose 查询期权自对冲请求 CThostFtdcTraderSpi::OnRspQryOptionSelfClose 查询期权自对冲请求响应	对话流
	查询合约状态	CThostFtdcTraderApi::ReqQryInstrumentStatus 查询合约状态请求 CThostFtdcTraderSpi::OnRspQryInstrumentStatus 查询合约状态请求响应	对话流
	查询投资者持仓明细	CThostFtdcTraderApi::ReqQryInvestorPositionDetail 查询投资者持仓明细请求 CThostFtdcTraderSpi::OnRspQryInvestorPositionDetail 查询投资者持仓明细请求响应	对话流
	查询投资者组合持仓明细	CThostFtdcTraderApi::ReqQryInvestorPositionCombineDetail 查询投资者组合持仓明细请求 ---本接口在后台系统版本 V1.5.8 及之后取消, 不再支持 CThostFtdcTraderSpi::OnRspQryInvestorPositionCombineDetail 查询投资者组合持仓明细请求响应 ---本接口在后台系统版本 V1.5.8 及之后取消, 不再支持	对话流
	查询交易所保证金率	CThostFtdcTraderApi::ReqQryExchangeMarginRate 请求查询交易所保证金率请求 CThostFtdcTraderSpi::OnRspQryExchangeMarginRate 请求查询交易所保证金率请求响应	对话流
	查询交易所调整保证金率	CThostFtdcTraderApi::ReqQryExchangeMarginRateAdjust 请求查询交易所调整保证金率请求 CThostFtdcTraderSpi::OnRspQryExchangeMarginRateAdjust 请求查询交易所调整保证金率请求响应	对话流
	查询期权交易成本	CThostFtdcTraderApi::ReqQryOptionInstrTradeCost 请求查询期权交易成本请求 CThostFtdcTraderSpi::OnRspQryOptionInstrTradeCost 请求查询期权交易成本请求响应	对话流
	查询期权合约手续费	CThostFtdcTraderApi::ReqQryOptionInstrCommRate 请求查询期权合约手续费请求 CThostFtdcTraderSpi::OnRspQryOptionInstrCommRate 请求查询期权合约手续费请求响应	对话流
	查询执行宣告	CThostFtdcTraderApi::ReqQryExecOrder 请求查询执行宣告请求 CThostFtdcTraderSpi::OnRspQryExecOrder 请求查询执行宣告请求响应	对话流
	查询询价	CThostFtdcTraderApi::ReqQryForQuote 请求查询询价请求 CThostFtdcTraderSpi::OnRspQryForQuote 请求查询询价请求响应	对话流

	查询报价	CThostFtdcTraderApi::ReqQryQuote 请求查询报价请求 CThostFtdcTraderSpi::OnRspQryQuote 请求查询报价请求响应	对话流
	查询申报费	CThostFtdcTraderApi::ReqQryInstrumentOrderCommRate 请求查询申报费请求 CThostFtdcTraderSpi::OnRspQryInstrumentOrderCommRate 请求查询申报费请求响应	对话流
	查询申请组合合约	CThostFtdcTraderApi::ReqQryCombInstrument 请求查询申请组合合约 CThostFtdcTraderSpi::OnRspQryCombInstrument 请求查询申请组合合约请求响应	V1.5.8 增加
	查询申请组合	CThostFtdcTraderApi::ReqQryCombAction 请求查询申请组合 CThostFtdcTraderSpi::OnRspQryCombAction 请求查询申请组合响应	V1.5.8 增加
	查询组合单腿汇总	CThostFtdcTraderApi::ReqQryInvestorPositionForComb 请求查询单腿汇总查询 CThostFtdcTraderSpi::OnRspQryInvestorPositionForComb 请求查询单腿汇总请求响应	V1.5.8 增加
	查询交易员报盘机状态	CThostFtdcTraderApi::ReqQryTraderOffer CThostFtdcTraderSpi::OnRspQryTraderOffer	V1.6.2 版本增加
	查询郑商所 SPBM 组合保金明细	查询郑商所 SPBM 组合产品保证金明细请求及查询响应 CThostFtdcTraderApi::ReqQryInvestorProdSPBMDetail、 CThostFtdcTraderSpi::OnRspQryInvestorProdSPBMDetail	V1.6.3 增加
	查询上期所或能源中心 SPMM 保证金明细	查询上期所或能源中心新型组合产品保证金明细请求及查询响应 CThostFtdcTraderApi::ReqQrySPMMInvestorCommodityGroupMargin3。 CThostFtdcTraderSpi::OnRspQrySPMMInvestorCommodityGr	V1.6.6 增加
	报单通知	OnRtnOrder	私有流
私有回报	成交通知	OnRtnTrade	私有流
	合约交易状态	OnRtnInstrumentStatus	公有流
	执行宣告通知	OnRtnExecOrder	私有流
	报价通知	OnRtnQuote	私有流
	询价通知	OnRtnForQuoteRsp	私有流
	期权自对冲通知	OnRtnOptionSelfClose	私有流
	报单录入错误回报	OnErrRtnOrderInsert	私有流
	报单操作错误回报	OnErrRtnOrderAction	私有流

	执行宣告操作 错误回报	OnErrRtnExecOrderAction	私有流
	询价录入错误 回报	OnErrRtnForQuoteInsert	私有流
	报价操作错误 回报	OnErrRtnQuoteAction	私有流
	期权自对冲操 作错误回报	OnErrRtnOptionSelfCloseAction	私有流

2.5.2.2 MdAPI

和行情业务相关的接口，简要如下列表：

业务类型	业务	请求接口/响应接口	数据流
登录	登录	CThostFtdcMdApi::ReqUserLogin 用户登录请求 CThostFtdcMdSpi::OnRspUserLogin 用户登录请求响应	对话流
	登出	CThostFtdcMdApi::ReqUserLogout 登出请求	对话流
行情	订阅	CThostFtdcMdApi:: SubscribeMarketData 订阅行情 CThostFtdcMdSpi:: OnRspSubMarketData 订阅行情应答 CThostFtdcMdApi:: SubscribeForQuoteRsp 订阅询价 CThostFtdcMdSpi:: OnRspSubForQuoteRsp 订阅询价应答	对话流
	取消订阅	CThostFtdcMdApi:: UnSubscribeMarketData 退订订阅行情 CThostFtdcMdSpi:: OnRspUnSubMarketData 退订订阅行情应答 CThostFtdcMdApi:: UnSubscribeForQuoteRsp 退订订阅询价 CThostFtdcMdSpi:: OnRspUnSubForQuoteRsp 退订订阅询价应答	对话流
	行情通知	CThostFtdcMdSpi:: OnRtnDepthMarketData 深度行情通知 CThostFtdcMdSpi:: OnRtnForQuoteRsp 询价通知	行情流

2.6 接口流程

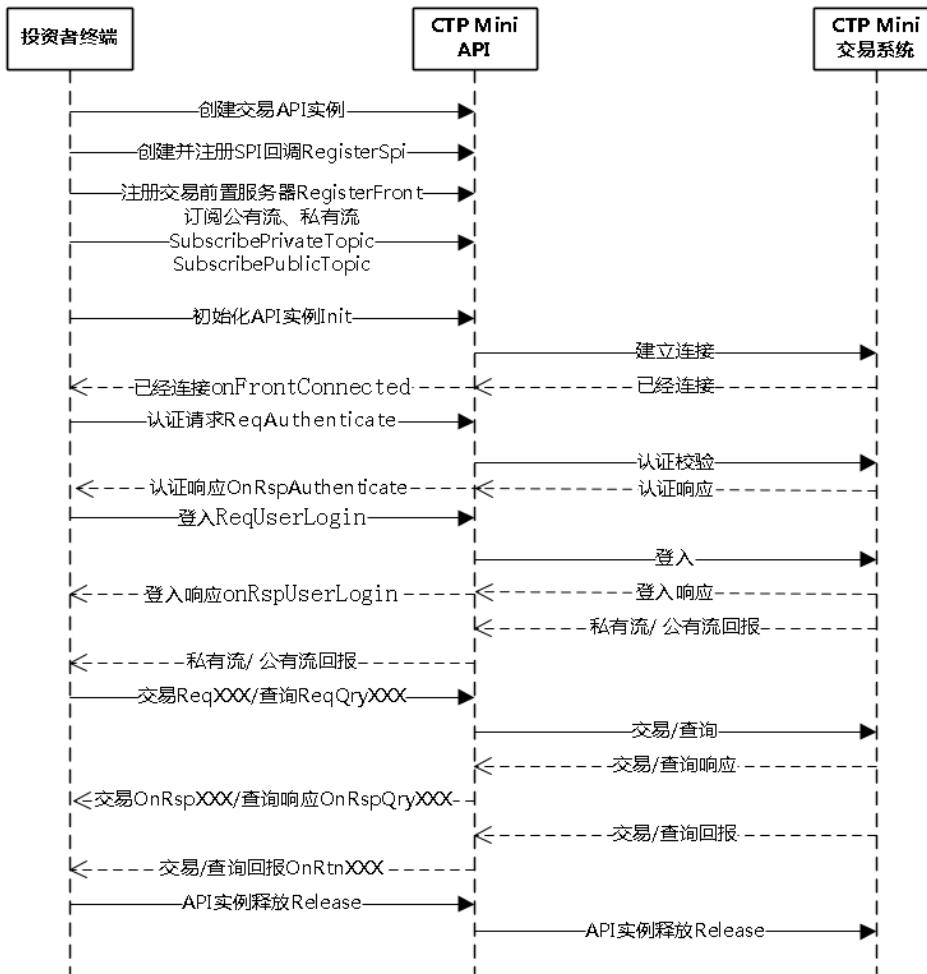
2.6.1 交易

2.6.1.1 全过程

在交易 API 实例初始化后，登陆 CTP Mini 交易系统后就可以进行期货期权交易或查询服务。

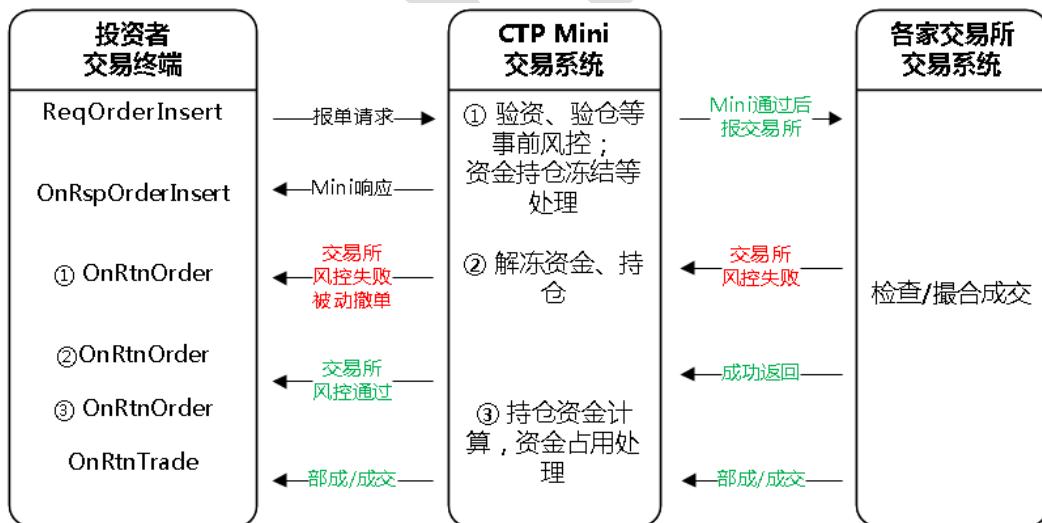
不再进行交易需要退出交易则只需要 release 交易 API 实例即可。

TraderAPI 接口处理流程简要示意如下：



2.6.1.2 报单

报单流程主要包括报单请求、报单回报处理、成交回报处理。从终端投资者到 Mini 交易系统到交易所的交互流程如下图示意：



为了方便理解报单及报单后的回调，通过常见的交易场景给出例子说明如下（以上期所为例）。注意：在以下场景中，虽然交易所回报的序列可能不尽相同，但 CTP Mini API 端收到的回报序列是相同的。

- 场景 1：报单合约 ag1207，报入 1 手，到交易所先未成交，后全部成交 1 手
 - 报单：ReqOrderInsert
 - 回调顺序：
 - ◆ OnRtnOrder (未知单)
 - ◆ 此时 CTP 接收到交易所的未成交报单回报
 - ◆ OnRtnOrder (未成交)
 - ◆ 此时 CTP 接收到交易所的成交回报和全部成交报单回报
 - ◆ OnRtnOrder (全部成交)
 - ◆ OnRtnTrade
- 场景 2：报单合约 ag1207，报入 1 手，到交易所立即全部成交 1 手。（注意，此案例没有收到交易所的未成交回报状态，而是直接收到全部成交状态）
 - 报单：ReqOrderInsert
 - 回调顺序：
 - ◆ OnRtnOrder (未知单)
 - ◆ 此时 CTP 接收到交易所的全部成交报单回报和成交回报
 - ◆ OnRtnOrder (未成交)
 - ◆ OnRtnOrder (全部成交)
 - ◆ OnRtnTrade
- 场景 3：报单合约 ag1207，报入 1 手，到交易所先未成交，然后撤单 1 手
 - 报单：ReqOrderInsert
 - 回调顺序：
 - ◆ OnRtnOrder (未知单)
 - ◆ 此时 CTP 接收到交易所的未成交报单回报
 - ◆ OnRtnOrder (未成交)
 - ◆ ReqOrderAction
 - ◆ 此时 CTP 接收到交易所的撤单报单回报
 - ◆ OnRtnOrder (已撤单)
- 场景 4：报单合约 ag1207，报入 10 手，到交易所先未成交，然后成交 3 手，然后剩余全部成交 7 手
 - 报单：ReqOrderInsert

■ 回调顺序：

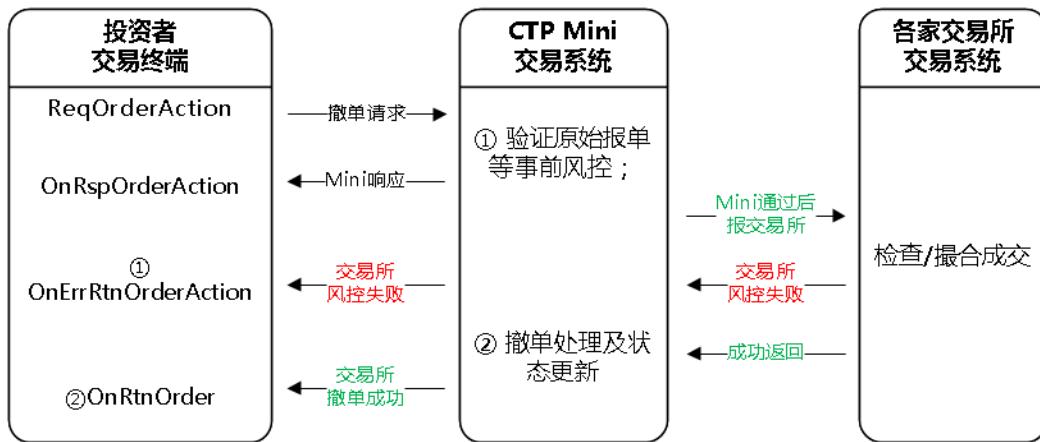
- ◆ OnRtnOrder (未知单)
- ◆ 此时 CTP 接收到交易所的未成交报单回报
- ◆ OnRtnOrder (未成交)
- ◆ 此时 CTP 接收到交易所的成交回报和部分成交报单回报
- ◆ OnRtnOrder (部分成交)
- ◆ OnRtnTrade
- ◆ 此时 CTP 接收到交易所的成交回报和全部成交报单回报
- ◆ OnRtnOrder (全部成交)
- ◆ OnRtnTrade

在某些特殊场景下，由于不同交易所回报序列的不同，导致 CTP Mini API 端收到的回报序列也不同。以上期所和大商所的 FAK 部成部撤单为例：

- 上期所：报单合约 ag1207，报入 10 手 FAK 单，到交易所成交 3 手，其余全部撤销
 - 报单：ReqOrderInsert
 - 回调顺序：
 - ◆ OnRtnOrder (未知单)
 - ◆ 此时 CTP 接收到交易所的已撤单报单回报
 - ◆ OnRtnOrder (已撤单)
 - ◆ 此时 CTP 接收到交易所的成交回报
 - ◆ OnRtnOrder (已撤单)
 - ◆ OnRtnTrade
- 大商所：报单合约 m1207，报入 10 手 FAK 单，到交易所成交 3 手，其余全部撤销
 - 报单：ReqOrderInsert
 - 回调顺序：
 - ◆ OnRtnOrder (未知单)
 - ◆ 此时 CTP 接收到交易所的未成交报单回报
 - ◆ OnRtnOrder (未成交)
 - ◆ 此时 CTP 接收到交易所的成交回报
 - ◆ OnRtnOrder (部分成交)
 - ◆ OnRtnTrade
 - ◆ 此时 CTP 接收到交易所的已撤单报单回报
 - ◆ OnRtnOrder (已撤单)

2.6.1.3 撤单

撤单流程主要包括撤单请求、原始报单校验、撤单返回处理，相关的流程交易如下图：



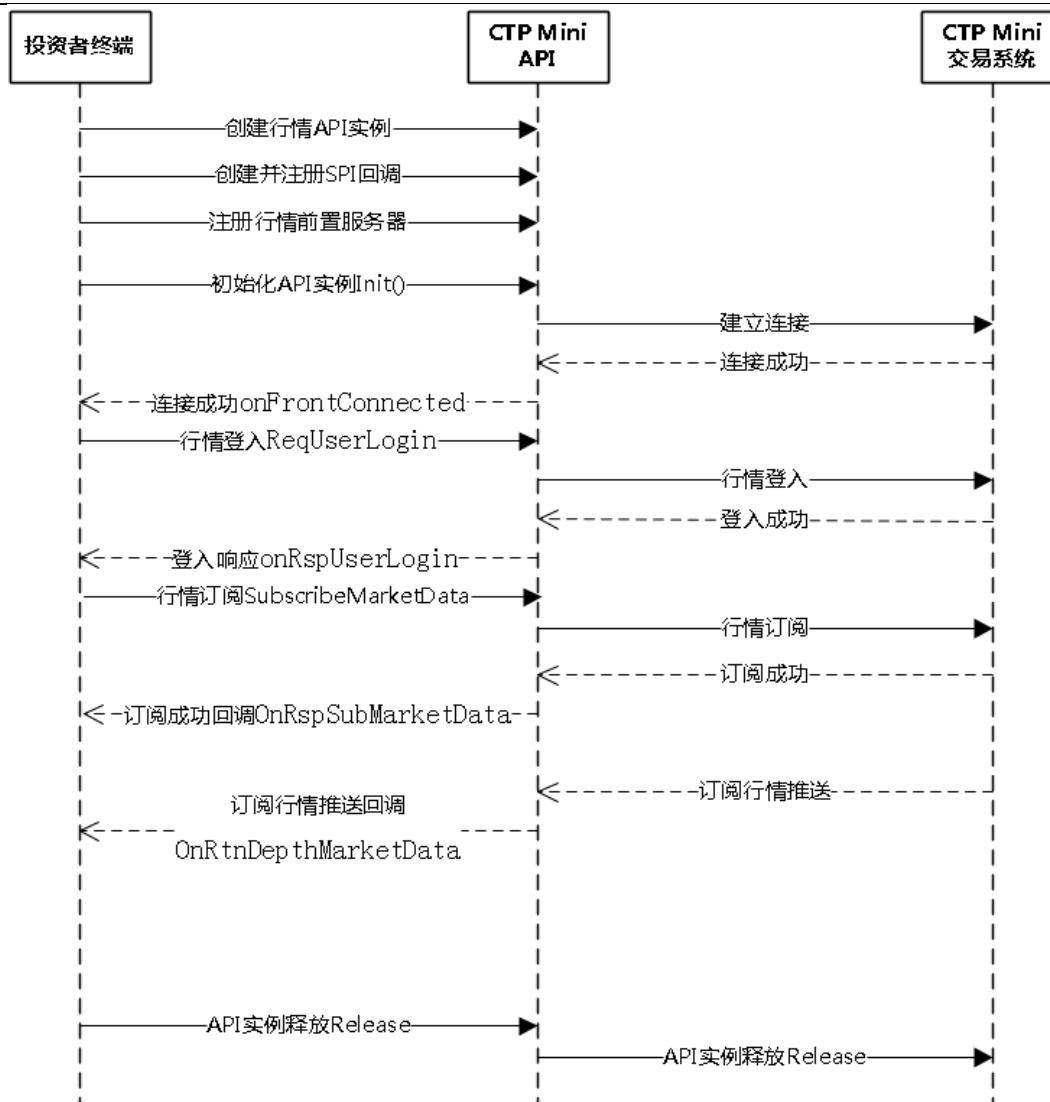
终端开发者通过 ReqOrderAction 指令申请撤单；Mini 交易系统风控检查返回 OnRspOrderAction 响应；通过 Mini 风控后报入交易所，交易所端检查通过实现撤单处理，Mini 交易系统收到交易所回报后返回终端 OnRtnOrder 回报，如果不通过则返回 OnErrRtnOrderAction 回报。

2.6.2 行情

在行情 API 实例初始化后，登陆并订阅行情成功就可以收到后台 CTP Mini 交易系统推送的行情服务。

不再接收行情则退订行情并 release 行情 API 实例即可。

MDAPI 接口的处理流程示意如下：



3 交易接口

特别说明：本节开始的接口参考示例对变量是否为空或非法，如指针为 NULL、数组边界等不做检查与控制，请开发人员严格依照 C/C++ 规范自行添加。

3.1 CThostFtdcTraderApi

3.1.1 CreateFtdcTraderApi

功能：

创建 TraderApi

函数原型：

```
static CThostFtdcTraderApi *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托管系统发布消息的状态。默认值代表当前目录。

返回值：

创建出 CThostFtdcTraderApi 对象

示例：

```
void MyTradeInit ()  
{  
    //初始化 trader api  
    //创建 API 实例  
    CThostFtdcTraderApi *m_pTraderApi = CThostFtdcTraderApi::CreateFtdcTraderApi("flow\\myDir1\\");  
    //创建失败退出  
    assert(m_pTraderApi);  
    //注册 SPI 回调，接收 CTP 响应与回复，可以通过重写父类中的函数实现自己接收响应后的处理逻辑  
    m_pTraderApi->RegisterSpi(m_pSpiApi); //m_pSpiApi 派生自 CThostFtdcTraderSpi 类  
    //注册前置服务器，将 CTP Mini 前置的 IP 地址注册进入 API 实例中  
    m_pTraderApi->RegisterFront((char *)"tcp://172.24.126.17:31503");  
    //订阅公有流与私有流。订阅方式主要有三种，分为断点续传，重传和连接建立开始传三种  
    //订阅私有流  
    m_pTraderApi->SubscribePrivateTopic(THOST_TERT_RESTART);  
    //订阅公有流  
    m_pTraderApi->SubscribePublicTopic(THOST_TERT_RESTART);  
    //准备好后正式启动初始化，启动一个内部线程读写，并去连 CTP 前置  
    m_pTraderApi->Init();  
    //等待接口线程完成各方面调度，完成整个初始化和后台准备工作；对应的在退出 API 实例时需要调用 release 函数释放；  
    m_pTraderApi->Join();  
  
    //创建第二个 api 实例，要区分开 flow 目录，如果需要则继续，  
    CThostFtdcTraderApi *pUserApi2 = CThostFtdcTraderApi::CreateFtdcTraderApi("flow\\ myDir2\\")  
    .....  
}
```

3.1.2 GetApiVersion

功能：

获取 API 的版本信息

函数原型：

```
static const char *GetApiVersion();
```

参数：

无

返回值：

获取到的版本号字符串指针

3.1.3 Release

功能：

删除接口对象本身；不再使用本接口对象时，调用该函数删除接口对象

函数原型：

```
virtual void Release();
```

参数：

无

返回值：

无

示例：

//重新初始化或退出程序前，需要释放初始化创建的句柄；

```
void UnInitAPI ()  
{  
    // 释放 TraderApi  
    if (m_pTraderApi)  
    {  
        m_pTraderApi ->Release();  
        m_pTraderApi = NULL;  
    }  
    // 释放 SpiApi 实例  
    if (m_pSpiApi)  
    {  
        delete m_pSpiApi;  
        m_pSpiApi = NULL;  
    }  
}
```

3.1.4 Init

功能：

初始化运行环境,只有调用后,接口才开始工作

函数原型：

```
virtual void Init(bool bContinuous = false);
```

参数：

bContinuous 为 true 表示线程不休眠；
为 false 表示线程支持休眠；
缺省为 false；

返回值：

无

示例：

参见： [CreateFtdcTraderApi 章节](#)

3.1.5 Join

功能：

等待接口线程结束运行

函数原型：

```
virtual int Join();
```

参数：

无

返回值：

线程退出代码

示例：

参见： [CreateFtdcTraderApi 章节](#)

3.1.6 GetTradingDay

功能：

获取当前交易日；只有登录成功后,才能得到正确的交易日

函数原型：

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

参数：

无

返回值：

成功返回：交易日字符串指针

失败返回：NULL

3.1.7 RegisterFront

功能：

注册前置机网络地址

函数原型：

```
virtual void RegisterFront(char *pszFrontAddress);
```

参数：

pszFrontAddress：前置机网络地址，网络地址的格式为：“protocol://ipaddress:port”

返回值：

无

示例：

参见：[CreateFtdcTraderApi 章节](#)

3.1.8 RegisterSpi

功能：

注册回调接口

函数原型：

```
virtual void RegisterSpi(CThostFtdcTraderSpi *pSpi);
```

参数：

pSpi 派生自回调接口类 CThostFtdcTraderSpi 的实例

返回值：

无

3.1.9 SubscribePrivateTopic

功能：

订阅私有流

该方法要在 Init 方法前调用。若不调用则不会收到私有流数据。

函数原型：

```
virtual void SubscribePrivateTopic(THOST_TE_RESUME_TYPE nResumeType);
```

参数：

nResumeType 私有流重传方式，支持三种方式：

- THOST_TERT_RESTART:从本交易日开始重传
- THOST_TERT_RESUME:从上次收到的续传
 - 序号保存在本地流水文件里，若遇到交易系统清流重启，如果此时不删除原流水文件并继续用 RESUME 模式接入，则可能收不到私有流回报
- THOST_TERT_QUICK:只传送登录后私有流的内容

返回值：

无

示例：

参见： [CreateFtdcTraderApi 章节](#)

3.1.10 SubscribePublicTopic

功能：

订阅公共流

该方法要在 Init 方法前调用。若不调用则不会收到公共流的数据。

函数原型：

```
virtual void SubscribePublicTopic(THOST_TE_RESUME_TYPE nResumeType);
```

参数：

nResumeType 公共流重传方式: 支持三种方式：

- THOST_TERT_RESTART:从本交易日开始重传
- THOST_TERT_RESUME:从上次收到的续传
- THOST_TERT_QUICK:只传送登录后私有流的内容

返回值：

无

示例：

参见： [CreateFtdcTraderApi 章节](#)

3.1.11 ReqAuthenticate

功能：

客户端认证请求

对应响应 OnRspAuthenticate

函数原型：

```
virtual int ReqAuthenticate(CThostFtdcReqAuthenticateField
*pReqAuthenticateField, int nRequestID);
```

参数：

pReqAuthenticateField：客户端认证请求信息结构，各字段如下表解释

字段类型	字段名称	含义	值
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息，如软件开发商、版本号等	无
TThostFtdcAuthCodeType	AuthCode	验证码，需要向期货公司申请得到	必填
TThostFtdcAppIDType	AppID	App 代码；如果没有则要向期货公司申请，申请的 AppID 必须遵循监控中心规范格式	必填

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcReqAuthenticateField structAuth = { 0 };
strcpy(structAuth.BrokerID, "8000");
strcpy(structAuth.UserID, "100001");
strcpy(structAuth.UserProductInfo, "mytest");
strcpy(structAuth.AuthCode, "#34@#@4dfas3AM1k");
strcpy(structAuth.AppID, "mytest");
m_pTraderApi->ReqAuthenticate(&structAuth, m_requestID++);

.....
```

3.1.12 ReqUserLogin

功能：

用户登录请求

函数原型：

```
virtual int ReqUserLogin(CThostFtdcReqUserLoginField *pReqUserLoginField, int
nRequestID);
```

参数：

pReqUserLoginField 登陆用户的基本信息，包含的字段信息如下表

字段类型	字段名称	含义	值
TThostFtdcDateType	TradingDay	交易日	无
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcUserIDType	UserID	用户代码	必填
TThostFtdcPasswordType	Password	密码	必填
TThostFtdcProductInfoType	UserProductInfo	用户产品信息，如软件开发商、版本号等；	无
TThostFtdcProductInfoType	InterfaceProductInfo	接口端产品信息	无
TThostFtdcProtocolInfoType	ProtocolInfo	协议信息	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无
TThostFtdcPasswordType	OneTimePassword	动态密码，如果启用了动态口令，需要填写电子口令牌动态值，mini 不支持	无
TThostFtdcIPAddressType	ClientIPAddress	终端 IP 地址	无
TThostFtdcLoginRemarkType	LoginRemark	登录备注	无

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
int CUserTraderSpi::ReqUserLogin()
{
    //准备登陆用户基本信息
    CThostFtdcReqUserLoginField reqUserLogin;
    memset(&reqUserLogin, 0x00, sizeof(reqUserLogin));
    strcpy(reqUserLogin.BrokerID, "8000");
    strcpy(reqUserLogin.UserID, "8000_admin");
    strcpy(reqUserLogin.Password, "1xqwewe@");

    //调用 Trader API 对应的接口实现用户登陆
    return m_traderApi->ReqUserLogin(&reqUserLogin, ++m_requestID);
}
```

3.1.13 ReqUserLoginEncrypt

功能：

用户加密登录请求

- 当初给出该接口是为了兼容市场上有不加密方式登陆，避免处理错误；

- 本接口与 ReqUserLogin 相同,终端开发参考 ReqUserLogin;都采用加密方式登陆。

函数原型:

```
virtual int ReqUserLoginEncrypt(CThostFtdcReqUserLoginField  
*pReqUserLoginField, int nRequestID);
```

参数:

pReqUserLoginField 登陆用户的基本信息

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```
int CUserTraderSpi::ReqUserLoginEncrypt ()  
{  
    //准备登陆用户基本信息  
    CThostFtdcReqUserLoginField reqUserLogin;  
    memset(&reqUserLogin, 0x00, sizeof(reqUserLogin));  
    strcpy(reqUserLogin.BrokerID, "8000");  
    strcpy(reqUserLogin.UserID, "8000_admin");  
    strcpy(reqUserLogin.Password, "1xqwewe@");  
    //调用 Trader API 对应的接口实现用户登陆  
    return m_traderApi->ReqUserLoginEncrypt (&reqUserLogin, ++m_requestID);  
}
```

3.1.14 ReqUserLogout

功能:

登出请求

函数原型:

```
virtual int ReqUserLogout(CThostFtdcUserLogoutField *pUserLogout, int  
nRequestID);
```

参数:

pUserLogout 登出用户信息，相关结构字段如下表

字段类型	字段名称	含义	值
TTHostFtdcBrokerIDType	BrokerID	经纪公司代码	无
TTHostFtdcUserIDType	UserID	用户代码	无

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
int CUserTraderSpi::ReqUserLogout()
{
    //准备登陆用户基本信息
    CThostFtdcUserLogoutField reqUserLogout;
    memset(&reqUserLogout, 0x00, sizeof(reqUserLogout));
    strcpy(reqUserLogout.BrokerID, "8000");
    strcpy(reqUserLogout.UserID, "8000_admin");

    //调用 Trader API 对应的接口实现用户登陆
    return m_traderApi->ReqUserLogout(&reqUserLogout, ++m_requestID);
}
```

3.1.15 ReqOrderInsert

功能：

报单录入请求

- 录入错误时对应响应 OnRspOrderInsert、OnErrRtnOrderInsert，正确时对应回报 OnRtnOrder、OnRtnTrade；
- 支持限价单、市价单，具体参见交易所支持指令；
- 撤单时使用 ReqOrderAction。

函数原型：

```
virtual int ReqOrderInsert(CThostFtdcInputOrderField *pInputOrder, int nRequestID);
```

参数：

plnputOrder: 报单信息, 相关字段如下表

字段类型	字段名称	含义	值
TTHostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TTHostFtdcInvestorIDType	InvestorID	投资者代码	必填
TTHostFtdcInstrumentIDType	InstrumentID	合约代码	必填
TTHostFtdcOrderRefType	OrderRef	报单引用, 本地会话全局唯一编号, 由终端维护	必填
TTHostFtdcUserIDType	UserID	用户代码	无
TTHostFtdcOrderPriceTypeType	OrderPriceType	报单价格类型, 枚举类型;	必填
TTHostFtdcDirectionType	Direction	买、卖方向	必填
TTHostFtdcCombOffsetFlagType	CombOffsetFlag	开、平标志; 特别注意: 上期和能源有平今指令, 平仓和平昨指令相同; 其他交易所只有平仓;	必填
TTHostFtdcCombHedgeFlagType	CombHedgeFlag	投机、套保等标志	必填
TTHostFtdcPriceType	LimitPrice	价格, 只有 OrderPriceType 是限价单的时候需要填写, 填写的时候注意价格要是最小变动价位(查询合约可得)的整数倍, 否则会被拒单; 对 TAS 交易, LimitPrice 必须为 0;	必填
TTHostFtdcVolumeType	VolumeTotalOriginal	报单数量, 必须大于 0	必填
TTHostFtdcTimeConditionType	TimeCondition	有效期类型, 包括 IOC、GFS、GFD、GTD、GTC、GFA 等 V1.6.8 开始启用 GFS 支持大商所 GIS 属性	必填
TTHostFtdcDateType	GTDDate	GTD 日期	无
TTHostFtdcVolumeConditionType	VolumeCondition	成交量类型, 包括任意数量、最小数量、全部数量等	必填
TTHostFtdcVolumeType	MinVolume	最小成交量, 当成交量类型是最小数量时, 需要通过该字段指定数量	无
TTHostFtdcContingentConditionType	ContingentCondition	触发条件	必填
TTHostFtdcPriceType	StopPrice	止损价	无
TTHostFtdcForceCloseReasonType	ForceCloseReason	强平原因, 填写非强平标志 THOST_FTDC_FCC_NotForceClose	无
TTHostFtdcBoolType	IsAutoSuspend	自动挂起标志	无
TTHostFtdcBusinessUnitType	BusinessUnit	业务单元; 字段复用, 可以通过该字段指定席位(TraderID) 报单	无
TTHostFtdcRequestIDType	RequestID	请求编号	无
TTHostFtdcBoolType	UserForceClose	用户强平标志	无
TTHostFtdcBoolType	IsSwapOrder	互换单标志	无
TTHostFtdcExchangeIDType	ExchangeID	交易所代码, 全部大写 CFFEX、CZCE、DCE、INE、SHFE	建议填写
TTHostFtdcInvestUnitIDType	InvestUnitID	投资单元代码, 如填写需和 InvestorID 一样	无
TTHostFtdcAccountIDType	AccountID	投资者帐号	无
TTHostFtdcCurrencyIDType	CurrencyID	币种代码, 默认为 CNY	无
TTHostFtdcClientIDType	ClientID	客户代码	无
TTHostFtdcIPAddressType	IPAddress	IP 地址	无
TTHostFtdcMacAddressType	MacAddress	Mac 地址	无
TTHostFtdcCustomOrderRefType	CustomOrderRef	客户报单自定义字段	无
TTHostFtdcOrderFlagType	OrderFlag	报单标志, 未来可能扩展各种投资者自己控制后台开关; 可以不设置, 缺省为 0; 如果设置为 THOST_FTDC_SELFTRADE_AvoidSelfTradeCheck 或'1', 则代表后台对该笔报单不做自成交检查	无

- 几种限价单有效期、成交量、最小成交量的关系

字段	普通	FAK	FOK
TimeCondition	THOST_FTDC_TC_GFD	THOST_FTDC_TC_IOC	THOST_FTDC_TC_IOC
VolumeCondition	THOST_FTDC_VC_AV	THOST_FTDC_VC_AV/THOST_FTDC_VC_MV	THOST_FTDC_VC_CV
MinVolume	不需要填	如果 VolumeCondition 为 THOST_FTDC_VC_AV，则不需要填。如果为 THOST_FTDC_VC_MV，则设为要求的最小成交的手数	不需要填

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```
//以下以限价单报单为例说明
int CUserTraderSpi::reqOrderInsert()
{
    //准备根据投资者报单要求准备报单基本信息
    CThostFtdcInputOrderField pInputOrder;
    memset(&pInputOrder, 0x00, sizeof(pInputOrder));
    strcpy(pInputOrder.BrokerID, "8000"); //经纪公司代码
    strcpy(pInputOrder.InvestorID, "000073"); //投资者代码
    strcpy(pInputOrder.UserID, "000073"); //用户代码
    strcpy(pInputOrder.InstrumentID, "al1201"); //合约代码
    strcpy(pInputOrder.ExchangeID, "SHFE"); //交易所代码
    sprintf(pInputOrder.OrderRef, "%d", m_orderRef++); //报单引用编号
    pInputOrder.OrderPriceType = THOST_FTDC_OPT_LimitPrice; //限价单
    pInputOrder.Direction = THOST_FTDC_D_Buy; //买入
    pInputOrder.CombHedgeFlag[0] = THOST_FTDC_HF_Speculation; //投机单
    pInputOrder.LimitPrice = 17350; //限价价格
    pInputOrder.VolumeTotalOriginal = 2; //报单数量
    pInputOrder.CombOffsetFlag[0] = THOST_FTDC_OF_Open; //开仓
    pInputOrder.TimeCondition = THOST_FTDC_TC_GFD; //当日有效
    pInputOrder.VolumeCondition = THOST_FTDC_VC_AV; //任意数量成交
    pInputOrder.MinVolume = 1; //最小成交量手数
    pInputOrder.ContingentCondition = THOST_FTDC_CC_Immediately; //触发条件：立即单
    pInputOrder.ForceCloseReason = THOST_FTDC_FCC_NotForceClose; //强平原因：非强平
    pInputOrder.IsAutoSuspend = 0; //不自动挂起
    strcpy(pInputOrder.InvestUnitID, "000073"); //投资单元
}
```

```

pInputOrder.UserForceClose = 0;//不是强平单，普通单
pInputOrder.IsSwapOrder = 0;//不是互换单，普通单
//发起报单
m_traderApi->ReqOrderInsert(&pInputOrder, m_requestID++);
}

```

3.1.16 ReqOrderAction

功能：

报单操作请求

函数原型：

```
virtual int ReqOrderAction(CThostFtdcInputOrderActionField *pInputOrderAction,
int nRequestID);
```

参数：

pInputOrderAction：报单操作信息，字段信息如下表

字段类型	字段名称	含义	值
TTHostFtdcBrokerIDType	BrokerID	经纪公司代码	无
TTHostFtdcInvestorIDType	InvestorID	投资者代码	无
TTHostFtdcOrderRefType	OrderRef	报单引用，对应要撤销的那笔报单的报单引用	必填*1
TTHostFtdcOrderActionRefType	OrderActionRef	报单操作引用	无
TTHostFtdcRequestIDType	RequestID	请求编号	无
TTHostFtdcFrontIDType	FrontID	前置编号，对应要撤销的那笔报单的前置编号	必填*1
TTHostFtdcSessionIDType	SessionID	会话编号，对应要撤销的那笔报单的会话编号	必填*1
TTHostFtdcExchangeIDType	ExchangeID	交易所代码，对应要撤销的那笔报单的交易所 ID	必填*2
TTHostFtdcOrderSysIDType	OrderSysID	报单编号，与 OrderRef 二选一填写；对应要撤销的那笔报单的报单编号	必填*2
TTHostFtdcActionFlagType	ActionFlag	操作标志，只支持删除，不支持修改、激活、挂起	必填
TTHostFtdcPriceType	LimitPrice	价格	无
TTHostFtdcVolumeType	VolumeChange	数量变化	无
TTHostFtdcUserIDType	UserID	用户代码	无
TTHostFtdcInstrumentIDType	InstrumentID	合约代码	必填*2
TTHostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TTHostFtdcIPAddressType	IPAddress	IP 地址，填写本机 IP 地址，不自动获取	无
TTHostFtdcMacAddressType	MacAddress	Mac 地址，填写本机 MAC 地址，不自动获取	无

- 撤单需要提供原始报单唯一标识序号，支持两种方式：

- ExchangeID（交易所编号）+OrderSysID（报单系统编号），此时 InstrumentID 必填；
- FrontID（前置编号）+ SessionID（会话链接编号）+ OrderRef（报单编号）

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
//第一种方法，使用 OrderSysID 撤单
CThostFtdcInputOrderActionField reqOrderAction = { 0 };
memset(&reqOrderAction, 0x00, sizeof(reqOrderAction));
strcpy(reqOrderAction.BrokerID, "8000");
strcpy(reqOrderAction.InvestorID, "100001");
strcpy(reqOrderAction.UserID, "100001");
strcpy(reqOrderAction.OrderSysID, "131"); //对应要撤报单的 OrderSysID，前补空格
strcpy(reqOrderAction.ExchangeID, "SHFE");
strcpy(reqOrderAction.InstrumentID, "rb2101");
reqOrderAction.ActionFlag = THOST_FTDC_AF_Delete;
m_traderApi -> ReqOrderAction(&reqOrderAction, m_requestID++);

//第二种方法，使用 FrontID+SessionID+OrderRef 撤单
CThostFtdcInputOrderActionField reqOrderAction = { 0 };
memset(&reqOrderAction, 0x00, sizeof(reqOrderAction));
strcpy(reqOrderAction.BrokerID, "8000");
strcpy(reqOrderAction.InvestorID, "100001");
strcpy(reqOrderAction.UserID, "100001");
reqOrderAction.FrontID = 1; //对应要撤报单的 FrontID
reqOrderAction.SessionID = -788541; //对应要撤报单的 sessionid
strcpy(reqOrderAction.OrderRef, "32343"); //对应要撤报单的 OrderRef
strcpy(reqOrderAction.ExchangeID, "SHFE");
strcpy(reqOrderAction.InstrumentID, "rb2101");
reqOrderAction.ActionFlag = THOST_FTDC_AF_Delete;
m_traderApi -> ReqOrderAction(&reqOrderAction, m_requestID++);
```

3.1.17 ReqExecOrderInsert

功能：

执行宣告录入请求

函数原型：

```
virtual int ReqExecOrderInsert(CThostFtdcInputExecOrderField *pInputExecOrder,
int nRequestID);
```

参数：

pInputExecOrder：执行宣告录入信息，相关字段如下表

字段类型	字段名称	含义	值
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	必填

TThostFtdcOrderRefType	ExecOrderRef	执行宣告引用, 用户维护	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcVolumeType	Volume	数量	必填
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcBusinessUnitType	BusinessUnit	业务单元; 字段复用, 可以通过该字段指定席位 (TraderID) 报单	无
TThostFtdcOffsetFlagType	OffsetFlag	开平标志	必填
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	必填
TThostFtdcActionTypeType	ActionType	执行类型: '1' 代表执行; '2' 代表放弃	必填
TThostFtdcPosiDirectionType	PosiDirection	保留头寸申请的持仓方向	多头
TThostFtdcExecOrderPositionFlagType	ReservePositionFlag	期权行权后是否保留期货头寸的标记,	无
TThostFtdcExecOrderCloseFlagType	CloseFlag	期权行权后生成的头寸是否自动平仓 对大商所, 只能填'1'	必填
TThostFtdcExchangeIDType	ExchangeID	交易所代码	无
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcAccountIDType	AccountID	投资者帐号	无
TThostFtdcCurrencyIDType	CurrencyID	币种代码	无
TThostFtdcClientIDType	ClientID	客户代码	无
TThostFtdcIPAddressType	IPAddress	IP 地址, 填写本机 IP 地址, 不自动获取	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```

CThostFtdcInputExecOrderField reqExecOrderInsert = { 0 };

memset(&reqExecOrderInsert, 0x00, sizeof(reqExecOrderInsert));

strcpy(reqExecOrderInsert.BrokerID, "8000");

strcpy(reqExecOrderInsert.InvestorID, "100001");

strcpy(reqExecOrderInsert.InstrumentID, "rb2101");

strcpy(reqExecOrderInsert.ExchangeID, "SHFE");

strcpy(reqExecOrderInsert.ExecOrderRef, "00001");

strcpy(reqExecOrderInsert.UserID, "100001");

reqExecOrderInsert.Volume = 1;

reqExecOrderInsert.RequestID = 1;

reqExecOrderInsert.OffsetFlag = THOST_FTDC_OF_Close;//开平标志

reqExecOrderInsert.HedgeFlag = THOST_FTDC_HF_Speculation;//投机套保标志

reqExecOrderInsert.ActionType = THOST_FTDC_ACTP_Exec;//执行类型类型

reqExecOrderInsert.PosiDirection = THOST_FTDC_PD_Long;//持仓多空方向类型

```

```

reqExecOrderInsert.ReservePositionFlag = THOST_FTDC_EOPF_Reserve;//期权行权后是否保留期货头寸的标记
reqExecOrderInsert.CloseFlag = THOST_FTDC_EOCF_NotToClose;//期权行权后生成的头寸是否自动平仓类型
m_pTraderApi->ReqExecOrderInsert(&reqExecOrderInsert, m_requestID++);

```

3.1.18 ReqExecOrderAction

功能：

执行宣告操作请求

函数原型：

```

virtual int ReqExecOrderAction(CThostFtdcInputExecOrderActionField
*pInputExecOrderAction, int nRequestID);

```

参数：

pInputExecOrderAction：执行宣告操作信息，相关字段如下表

字段类型	字段名称	含义	值
TTHostFtdcBrokerIDType	BrokerID	经纪公司代码	无
TTHostFtdcInvestorIDType	InvestorID	投资者代码	无
TTHostFtdcOrderActionRefType	ExecOrderActionRef	报单操作引用	无
TTHostFtdcOrderRefType	ExecOrderRef	报单引用	必填*1
TTHostFtdcRequestIDType	RequestID	请求编号	无
TTHostFtdcFrontIDType	FrontID	前置编号	必填*1
TTHostFtdcSessionIDType	SessionID	会话编号	必填*1
TTHostFtdcExchangeIDType	ExchangeID	交易所代码	必填*2
TTHostFtdcExecOrderSysIDType	ExecOrderSysID	执行宣告操作编号	必填*2
TTHostFtdcActionFlagType	ActionFlag	操作标志，只支持删除，不支持修改	必填
TTHostFtdcUserIDType	UserID	用户代码	无
TTHostFtdcInstrumentIDType	InstrumentID	合约代码	必填*2
TTHostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TTHostFtdcIPAddressType	IPAddress	IP 地址，填写本机 IP 地址，不自动获取	无
TTHostFtdcMacAddressType	MacAddress	Mac 地址	无

查找行权记录的必填项：

ExecOrderSysID+ ExchangeID+ InstrumentID

或者

ExecOrderRef+ SessionID+ FrontID

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```

CThostFtdcInputExecOrderField reqExecOrderAction = { 0 };

memset(&reqExecOrderAction, 0x00, sizeof(reqExecOrderAction));

strcpy(reqExecOrderAction.BrokerID, "8000");

strcpy(reqExecOrderAction.InvestorID, "100001");

reqExecOrderAction.ExecOrderActionRef = 1;

strcpy(reqExecOrderAction.ExecOrderRef, "00000003");

reqExecOrderAction.FrontID = 1;

reqExecOrderAction.SessionID = 7844256;

strcpy(reqExecOrderAction.ExchangeID, "SHFE");

strcpy(reqExecOrderAction.ExecOrderSysID, "285");

reqExecOrderAction.ActionFlag = THOST_FTDC_AF_Delete;//删除

strcpy(reqExecOrderAction.UserID, "100001");

strcpy(reqExecOrderAction.InstrumentID, "rb2102");

m_pTraderApi->ReqExecOrderAction(&reqExecOrderAction, m_requestID++);

```

3.1.19 ReqForQuoteInsert

功能：

做市商询价录入请求

函数原型：

```
virtual int ReqForQuoteInsert(CThostFtdcInputForQuoteField *pInputForQuote,
int nRequestID);
```

参数：

pInputForQuote：询价录入请求信息，相关字段如下表

字段类型	字段名称	含义	值
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	必填
TThostFtdcOrderRefType	ForQuoteRef	询价引用，终端维护	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcExchangeIDType	ExchangeID	交易所代码	无
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcIPAddressType	IPAddress	IP 地址，填写本机 IP 地址，不自动获取	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无
TThostFtdcBusinessUnitType	BusinessUnit	业务单元，V1.5.8 版本增加	无

- 如果询价价差不在交易所询价价差范围内，则会被拒绝；
- 禁止过于频繁询价；

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcInputForQuoteField reqInputForQuote = { 0 };

memset(&reqInputForQuote, 0x00, sizeof(reqInputForQuote));

strcpy(reqInputForQuote.BrokerID, "8000");

strcpy(reqInputForQuote.InvestorID, "100001");

strcpy(reqInputForQuote.InstrumentID, "rb2102");

strcpy(reqInputForQuote.UserID, "100001");

strcpy(reqInputForQuote.ExchangeID, "SHFE");

m_pTraderApi->ReqForQuoteInsert(&reqInputForQuote, m_requestID++);
```

3.1.20 ReqQuoteInsert

功能：

做市商报价录入请求

- 如果出错，则返回响应 OnRspQuoteInsert；
- 正确则推送 OnRtnQuote、OnRtnOrder 和 OnRtnTrade；
- 单边报价和双边报价，都是用一个接口 ReqQuoteInsert；
- 单边报价时，另一边报单数量填 0；
- 双边报价时，Ask/BidOrderRef 要么全部填写，要么都不填。
- 各家交易所报价对比如下表（本处为参考，不一定为交易所最新规则）：

中金所		大商所	郑商所	上期所/能源中心
是否支持平仓	支持	支持	支持	支持
是否必须关联询价	询价通知没有询价编号，不用填	询价通知有编号，但是非必填，如做市商要完成义务率则要填	询价通知有编号，但是非必填，如做市商要完成义务率则要填	询价通知有询价编号
是否支持单边报价	不支持，提示“CTP:报单字段有误”	六期及之前：支持买卖单边，数量填 0 七期只支持卖单边报价	不支持，提示“报单字段有误”	不支持，提示“报单字段有误”
是否支持市价单	不支持，交易所提示：“已撤单报单被拒绝 CFFEX:价格跌破跌停板”	不支持，交易所提示：已撤单报单被拒绝 DCE:期权价格必须大于等于 TICK!	不支持，交易所提示：“已撤单报单被拒绝 CZCE:出错：限价单价格不能为 0 或负数”	不支持

双边报价 买卖数量 可否不一致	可以不一致	可以不一致	不可以, 提示“报单字段有误”	不可以, 提示报单字段有误
双边报价 卖价是否必须大于买价	一定是低价买, 高价卖。否则报错, 触发自成交	必须, 否则交易所提示: 已撤单报单被拒绝 DCE: 双边报价指令卖价必须大于买价	必须, 否则交易所提示: 已撤单报单被拒绝 CZCE: 出错: 卖价必需大于买价	一定是低价买, 高价卖。否则会形成自成交
是否支持套保属性	支持	不支持, 提示“该期权合约只支持投机类型报单”	不支持, 提示“该期权合约只支持投机类型报单”	支持
是否会撤销前一次报价	不会撤销; 在顶单上线后对指定顶单编号的报价会自动撤销, 同时不支持撤衍生单;	不会撤销 接入大商所七期之后 支持自动撤销	会撤销	会撤销
是否支持撤单腿	可以 (顶单上线后不支持)	可以	不可以	不可以
双边报价开平可否不一致	可以	可以	可以	可以

函数原型:

```
virtual int ReqQuoteInsert(CThostFtdcInputQuoteField *pInputQuote, int nRequestID);
```

参数:

pInputQuote: 报价录入请求信息, 相关字段如下表

字段类型	字段名称	含义	值
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	必填
TThostFtdcOrderRefType	QuoteRef	报价引用, 终端维护	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcPriceType	AskPrice	卖价格	必填
TThostFtdcPriceType	BidPrice	买价格	必填
TThostFtdcVolumeType	AskVolume	卖数量	必填
TThostFtdcVolumeType	BidVolume	买数量	必填
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcBusinessUnitType	BusinessUnit	业务单元; 字段复用, 可以通过该字段指定席位(TraderID) 报单	无
TThostFtdcOffsetFlagType	AskOffsetFlag	卖开平标志	必填
TThostFtdcOffsetFlagType	BidOffsetFlag	买开平标志	必填
TThostFtdcHedgeFlagType	AskHedgeFlag	卖投机套保标志	必填
TThostFtdcHedgeFlagType	BidHedgeFlag	买投机套保标志	必填
TThostFtdcOrderRefType	AskOrderRef	衍生卖报单引用, 终端维护	选填
TThostFtdcOrderRefType	BidOrderRef	衍生买报单引用, 终端维护	选填
TThostFtdcOrderSysIDType	ForQuoteSysID	应价编号, 唯一定位一笔询价	选填
TThostFtdcExchangeIDType	ExchangeID	交易所代码	无
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcClientIDType	ClientID	客户代码	无

TThostFtdcIPAddressType	IPAddress	IP 地址, 写本机 IP 地址, 不自动获取	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无
TThostFtdcOrderSysIDType	ReplaceSysID	顶单编号	选填
TThostFtdcCustomOrderRefType	CustomQuoteRef	客户自定义报价编号	选填
TThostFtdcTimeConditionType	TimeCondition	有效期类型, 支持大商所 GIS 本小节有效 ///本节有效---原有头文件中枚举值启用 #define THOST_FTDC_TC_GFS '2'	选填, V1.6.8 增加字段

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```
int CUserTraderApi::reqQuoteInsert()
{
    CThostFtdcInputQuoteField reqInputQuote;

    memset(&reqInputQuote, 0x00, sizeof(reqInputQuote));

    strcpy(reqInputQuote.BrokerID, "8000");
    strcpy(reqInputQuote.InvestorID, "000073");
    strcpy(reqInputQuote.InstrumentID, "fu1202C5100");
    strcpy(reqInputQuote.UserID, "000073");
    strcpy(reqInputQuote.ExchangelD, "SHFE");
    sprintf(reqInputQuote.QuoteRef, "%012d", m_orderRef++);
    sprintf(reqInputQuote.AskOrderRef, "%012d", m_orderRef++);
    sprintf(reqInputQuote.BidOrderRef, "%012d", m_orderRef++);

    reqInputQuote.AskPrice = 370;
    reqInputQuote.BidPrice = 5;
    reqInputQuote.AskVolume = 1;
    reqInputQuote.BidVolume = 1;

    reqInputQuote.AskOffsetFlag = THOST_FTDC_OF_Open;//卖开平标志
    reqInputQuote.BidOffsetFlag = THOST_FTDC_OF_Open; //买开平标志
    reqInputQuote.AskHedgeFlag = THOST_FTDC_HF_Speculation; //卖投机套保标志
    reqInputQuote.BidHedgeFlag = THOST_FTDC_HF_Speculation; //买投机套保标志

    m_traderApi->ReqQuoteInsert(&reqInputQuote, m_requestID++);
}
```

3.1.21 ReqQuoteAction

功能：

报价操作请求,撤销报价

函数原型：

```
virtual int ReqQuoteAction(CThostFtdcInputQuoteActionField
* pInputQuoteAction, int nRequestId);
```

参数：

pInputQuoteAction：报价操作请求信息，相关字段如下表所示：

字段类型	字段名称	含义	值
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	无
TThostFtdcInvestorIDType	InvestorID	投资者代码	无
TThostFtdcOrderActionRefType	QuoteActionRef	报价操作引用	无
TThostFtdcOrderRefType	QuoteRef	报价引用	必填*1
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcFrontIDType	FrontID	对应要撤销的报价的前置编号	必填*1
TThostFtdcSessionIDType	SessionID	对应要撤销的报价的会话编号	必填*1
TThostFtdcExchangeIDType	ExchangeID	交易所代码	必填*2
TThostFtdcOrderSysIDType	QuoteSysID	对应要撤销的报价的编号	必填*2
TThostFtdcActionFlagType	ActionFlag	操作标志，支持删除，目前不支持改单。	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcInstrumentIDType	InstrumentID	合约代码	必填*2
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcClientIDType	ClientID	客户代码	无
TThostFtdcIPAddressType	IPAddress	IP 地址，填写本机 IP 地址，不自动获取	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

必填*1、必填*2：两组选一组必填，能对应要撤的报价单。

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
int CUserTraderApi::reqQuoteAction()
{
    CThostFtdcInputQuoteActionField reqQuoteAction;
    memset(&reqQuoteAction, 0x00, sizeof(reqQuoteAction));
    strcpy(reqQuoteAction.BrokerID, "8000");
    strcpy(reqQuoteAction.InvestorID, "000073");
```

```

strcpy(reqQuoteAction.InstrumentID, "fu1202C5000");

strcpy(reqQuoteAction.UserID, "000073");

strcpy(reqQuoteAction.ExchangeID, "SHFE");

sprintf(reqQuoteAction.QuoteRef, "%012d", m_orderRef++);

sprintf(reqQuoteAction.QuoteSysID, "%12d", 12);

reqQuoteAction.ActionFlag = THOST_FTDC_AF_Delete;

return m_traderApi->ReqQuoteAction(&reqQuoteAction, m_requestID++);

}

```

3.1.22 ReqOptionSelfCloseInsert

功能：

期权自对冲录入请求；

V1.7.3-P2 版本开始，不支持大商所期权自对冲录入业务。

函数原型：

```
virtual int ReqOptionSelfCloseInsert(CThostFtdcInputOptionSelfCloseField *pInputOptionSelfClose, int nRequestID);
```

参数：

pInputOptionSelfClose：期权自对冲录入请求信息，各字段如下表

字段类型	字段名称	含义	值
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	必填
TThostFtdcOrderRefType	OptionSelfCloseRef	期权自对冲引用，终端维护	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcVolumeType	Volume	数量，上期所数量必填，大商所可以不填	无
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	无
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	无
TThostFtdcOptSelfCloseFlagType	OptSelfCloseFlag	期权行权的头寸是否自对冲； 1 自对冲期权仓位 2 保留期权仓位 3 自对冲卖方履约后的期货仓位 4 保留卖方履约后的期货仓位	必填
TThostFtdcExchangeIDType	ExchangeID	交易所代码	必填
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcAccountIDType	AccountID	投资者帐号	无
TThostFtdcCurrencyIDType	CurrencyID	币种代码	无
TThostFtdcClientIDType	ClientID	客户代码	无
TThostFtdcIPAddressType	IPAddress	IP 地址，填写本机 IP 地址，不自动获取	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

- InstrumentID 期权的合约代码命名规则是：标的期货合约+看涨/看跌+执行价

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcInputOptionSelfCloseField reqOptionSelfClose = { 0 };

memset(&reqOptionSelfClose, 0x00, sizeof(reqOptionSelfClose));

strcpy(reqOptionSelfClose.BrokerID, "8000");

strcpy(reqOptionSelfClose.InvestorID, "100001");

strcpy(reqOptionSelfClose.InstrumentID, "cu2102C43000");

strcpy(reqOptionSelfClose.UserID, "100001");

reqOptionSelfClose.Volume = 1;

reqOptionSelfClose.HedgeFlag = THOST_FTDC_HF_Speculation;

reqOptionSelfClose.OptSelfCloseFlag = THOST_FTDC_OSCF_CloseSelfOptionPosition;

strcpy(reqOptionSelfClose.ExchangeID, "SHFE");

strcpy(reqOptionSelfClose.AccountID, "100001");

strcpy(reqOptionSelfClose.CurrencyID, "CNY");

m_pTraderApi->ReqOptionSelfCloseInsert(&reqOptionSelfClose, m_requestID++);
```

3.1.23 ReqOptionSelfCloseAction

功能：

期权自对冲操作请求

函数原型：

```
virtual int ReqOptionSelfCloseAction(CThostFtdcInputOptionSelfCloseActionField
*pInputOptionSelfCloseAction, int nRequestID);
```

参数：

pInputOptionSelfCloseAction：期权自对冲操作请求，相关字段如下表

字段类型	字段名称	含义	值
TTHostFtdcBrokerIDType	BrokerID	经纪公司代码	无
TTHostFtdcInvestorIDType	InvestorID	投资者代码	无
TTHostFtdcOrderActionRefType	OptionSelfCloseActionRef	期权自对冲操作引用	无
TTHostFtdcOrderRefType	OptionSelfCloseRef	对应要撤销的期权自对冲的引用	必填*1
TTHostFtdcRequestIDType	RequestID	请求编号	无

TThostFtdcFrontIDType	FrontID	对应要撤销的期权自对冲的前置编号	必填*1
TThostFtdcSessionIDType	SessionID	对应要撤销的期权自对冲的会话编号	必填*1
TThostFtdcExchangeIDType	ExchangeID	对应要撤销的期权自对冲的交易所编号	必填*2
TThostFtdcOrderSysIDType	OptionSelfCloseSysID	期权自对冲报单编号	必填*2
TThostFtdcActionFlagType	ActionFlag	操作标志：支持删除，不支持修改	必填
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcInstrumentIDType	InstrumentID	合约代码：对应要撤销的期权自对冲的合约代码	必填*2
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcIPAddressType	IPAddress	IP 地址：填写本机 IP 地址，不自动获取	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CHostFtdcInputOptionSelfCloseActionField reqOptionSelfCloseAction = { 0 };
memset(&reqOptionSelfCloseAction, 0x00, sizeof(reqOptionSelfCloseAction));
strcpy(reqOptionSelfCloseAction.BrokerID, "8000");
strcpy(reqOptionSelfCloseAction.InvestorID, "100001");
strcpy(reqOptionSelfCloseAction.OptionSelfCloseRef, "000258");//期权自对冲引用
reqOptionSelfCloseAction.FrontID = 1;
reqOptionSelfCloseAction.SessionID = 6442531;
strcpy(reqOptionSelfCloseAction.ExchangeID, "SHFE");
reqOptionSelfCloseAction.ActionFlag = THOST_FTDC_AF_Delete;
strcpy(reqOptionSelfCloseAction.UserID, "100001");
strcpy(reqOptionSelfCloseAction.InstrumentID, "cu2102C43000");
m_pTraderApi->ReqOptionSelfCloseAction(&reqOptionSelfCloseAction, m_requestID++);
```

3.1.24 ReqQryOrder

功能：

请求查询报单

函数原型：

```
virtual int ReqQryOrder(CThostFtdcQryOrderField *pQryOrder, int nRequestID);
```

参数：

pQryOrder: 请求查询报单信息，各字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcOrderSysIDType	OrderSysID	报单编号	是
TThostFtdcTimeType	InsertTimeStart	开始时间	是
TThostFtdcTimeType	InsertTimeEnd	结束时间	是

nRequestID: 终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryOrderField reqQryOrder = { 0 };

memset(&reqQryOrder, 0x00, sizeof(reqQryOrder));

strcpy(reqQryOrder.BrokerID, "8000");

strcpy(reqQryOrder.InvestorID, "100001");

strcpy(reqQryOrder.ExchangeID, "SHFE");

m_pTraderApi->ReqQryOrder(&reqQryOrder, m_requestID++);
```

3.1.25 ReqQryTrade

功能：

请求查询成交

函数原型：

```
virtual int ReqQryTrade(CThostFtdcQryTradeField *pQryTrade, int nRequestID);
```

参数：

pQryTrade: 请求查询成交信息，各字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcTradeIDType	TradeID	成交编号	是
TThostFtdcTimeType	TradeTimeStart	开始时间	是
TThostFtdcTimeType	TradeTimeEnd	结束时间	是

nRequestId: 终端请求编号

返回值:

0: 成功

-1: 失败

示例: 略

3.1.26 ReqQryInvestorPosition

功能:

请求查询投资者持仓

函数原型:

```
virtual     int      ReqQryInvestorPosition(CThostFtdcQryInvestorPositionField  
*pQryInvestorPosition, int nRequestId);
```

参数:

pQryInvestorPosition: 请求查询投资者持仓信息, 相关字段如下表

字段类型	字段名称	含义	过滤条件
TTHostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TTHostFtdcInvestorIDType	InvestorID	投资者代码	是
TTHostFtdcInstrumentIDType	InstrumentID	合约代码	是

- 如果各个字段都不填写, 则查询所有持仓

nRequestId: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```
CThostFtdcQryInvestorPositionField reqQryInvestorPosition= { 0 };  
  
memset(&reqQryInvestorPosition, 0x00, sizeof(reqQryInvestorPosition));  
  
strcpy(reqQryInvestorPosition.BrokerID, "8000");  
  
strcpy (reqQryInvestorPosition.InvestorID, "100001");  
  
strcpy (reqQryInvestorPosition.InstrumentID, "rb2102");//不填写合约则返回所有持仓  
  
m_pTraderApi->ReqQryInvestorPosition(&reqQryInvestorPosition, m_requestID++);
```

3.1.27 ReqQryTradingAccount

功能：

请求查询资金账户

函数原型：

```
virtual int ReqQryTradingAccount(CThostFtdcQryTradingAccountField *pQryTradingAccount, int nRequestID);
```

参数：

pQryTradingAccount：请求查询资金账户信息，相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcCurrencyIDType	CurrencyID	币种代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryTradingAccountField reqQryTradingAccount = { 0 };
strcpy(reqQryTradingAccount.BrokerID, "8000");
strcpy(reqQryTradingAccount.InvestorID, "100001");
strcpy(reqQryTradingAccount.CurrencyID, "CNY");
m_pTraderApi->ReqQryTradingAccount(&reqQryTradingAccount, m_requestID++);
```

3.1.28 ReqQryInvestor

功能：

请求查询投资者

函数原型：

```
virtual int ReqQryInvestor(CThostFtdcQryInvestorField *pQryInvestor, int nRequestID);
```

参数：

pQryInvestor：请求查询投资者信息,相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```
CThostFtdcQryInvestorField reqQryInvestor = { 0 };

memset(&reqQryInvestor, 0x00, sizeof(reqQryInvestor));

strcpy(reqQryInvestor.BrokerID, "8000");

strcpy(reqQryInvestor.InvestorID, "100001");

m_pTraderApi->ReqQryInvestor(&reqQryInvestor, m_requestID++);
```

3.1.29 ReqQryTradingCode

功能:

请求查询交易编码

函数原型:

```
virtual int ReqQryTradingCode(CThostFtdcQryTradingCodeField
*pQryTradingCode, int nRequestID);
```

参数:

pQryTradingCode: 请求查询交易编码信息, 相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcClientIDType	ClientID	客户代码	是
TThostFtdcClientIDTypeType	ClientIDType	交易编码类型	是

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

略

3.1.30 ReqQryInstrumentMarginRate

功能：

请求查询合约保证金率

- 如果 InstrumentID 填空，则返回所有期货合约对应的保证金率，否则返回相应 InstrumentID 的保证金率。

函数原型：

```
virtual int ReqQryInstrumentMarginRate  
(CThostFtdcQryInstrumentMarginRateField *pQryInstrumentMarginRate, int  
nRequestID);
```

参数：

pQryInstrumentMarginRate：请求查询合约保证金率信息，相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryInstrumentMarginRateField reqQryInstrumentMarginRate = { 0 };  
  
memset(&reqQryInstrumentMarginRate, 0x00, sizeof(reqQryInstrumentMarginRate));  
  
strcpy(reqQryInstrumentMarginRate.BrokerID, "8000");  
  
strcpy(reqQryInstrumentMarginRate.InvestorID, "100001");  
  
strcpy(reqQryInstrumentMarginRate.InstrumentID, "rb2102");  
  
reqQryInstrumentMarginRate.HedgeFlag = THOST_FTDC_HF_Speculation;  
  
m_pTraderApi->ReqQryInstrumentMarginRate(&reqQryInstrumentMarginRate, m_requestID++);
```

3.1.31 ReqQryInstrumentCommissionRate

功能：

请求查询合约手续费率

- 如果 InstrumentID 填空，则返回所有期货合约对应的手续费率；

函数原型：

```
virtual int ReqQryInstrumentCommissionRate
(CThostFtdcQryInstrumentCommissionRateField *pQryInstrumentCommissionRate,
int nRequestID);
```

参数：

pQryInstrumentCommissionRate：请求查询合约手续费率信息,相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryInstrumentCommissionRateField reqQryInstrumentCommissionRate = { 0 };

memset(&reqQryInstrumentCommissionRate, 0x00, sizeof(reqQryInstrumentCommissionRate));
strcpy(reqQryInstrumentCommissionRate.BrokerID, "8000");
strcpy(reqQryInstrumentCommissionRate.InvestorID, "100001");
strcpy(reqQryInstrumentCommissionRate.InstrumentID, "rb2102");
m_pTraderApi->ReqQryInstrumentCommissionRate(&reqQryInstrumentCommissionRate, m_requestID++);
```

3.1.32 ReqQryExchange

功能：

请求查询交易所

函数原型：

```
virtual int ReqQryExchange(CThostFtdcQryExchangeField *pQryExchange, int
```

nRequestID);

参数：

pQryExchange：请求查询交易所信息，相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：略

3.1.33 ReqQryInstrument

功能：

请求查询合约；

- 填空可以查询到所有合约；

函数原型：

```
virtual int ReqQryInstrument(CThostFtdcQryInstrumentField *pQryInstrument, int  
nRequestID);
```

参数：

pQryInstrument：请求查询合约信息，相关字段信息如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	是
TThostFtdcInstrumentIDType	ProductID	产品代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryInstrumentField reqQryInstrument = { 0 };
```

```
memset(&reqQryInstrument, 0x00, sizeof(reqQryInstrument));
strcpy(reqQryInstrument.InstrumentID, "rb2102");
strcpy(reqQryInstrument.ExchangeID, "SHFE");
m_pTraderApi->ReqQryInstrument(&reqQryInstrument, m_requestID++);
```

3.1.34 ReqQryDepthMarketData

功能：

请求查询行情

只能查询当前快照，不能查询历史行情。

函数原型：

```
virtual int ReqQryDepthMarketData(CThostFtdcQryDepthMarketDataField
*pQryDepthMarketData, int nRequestID);
```

参数：

pQryDepthMarketData：请求查询行情信息，相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：略

3.1.35 ReqQryOptionSelfClose

功能：

请求查询期权自对冲

函数原型：

```
virtual int ReqQryOptionSelfClose(CThostFtdcQryOptionSelfCloseField
*pQryOptionSelfClose, int nRequestID);
```

参数：

pQryOptionSelfClose：请求查询期权自对冲，相关字段说明如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcOrderSysIDType	OptionSelfCloseSysID	期权自对冲编号	是
TThostFtdcTimeType	InsertTimeStart	开始时间	是
TThostFtdcTimeType	InsertTimeEnd	结束时间	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryOptionSelfCloseField reqQryOptionSelfClose = { 0 };

strcpy(reqQryOptionSelfClose.BrokerID, "8000");
strcpy(reqQryOptionSelfClose.InvestorID, "100001");
strcpy(reqQryOptionSelfClose.InstrumentID, "rb2102");
strcpy(reqQryOptionSelfClose.ExchangeID, "SHFE");

m_pTraderApi->ReqQryOptionSelfClose(&reqQryOptionSelfClose, m_requestID++);
```

3.1.36 ReqQryInstrumentStatus

功能：

请求查询合约状态

函数原型：

```
virtual int ReqQryInstrumentStatus(CThostFtdcQryInstrumentStatusField *pQryInstrumentStatus, int nRequestID);
```

参数：

pQryInstrumentStatus：请求查询合约状态信息,相关字段说明如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：略

3.1.37 ReqQryInvestorPositionDetail

功能：

请求查询投资者持仓明细

- 本接口从后台 V1.5.8 版本开始根据交易后台有条件下提供支持：
 - V1.5.8 支持大商所持仓明细查询；
 - V1.7.0 支持广期所持仓明细查询；

函数原型：

```
virtual int ReqQryInvestorPositionDetail  
(CThostFtdcQryInvestorPositionDetailField *pQryInvestorPositionDetail, int  
nRequestID);
```

参数：

pQryInvestorPositionDetail：请求查询投资者持仓明细信息，相关字段信息如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

略

3.1.38 ReqQryInvestorPositionCombineDetail

功能：

请求查询投资者组合持仓明细

- 本接口从 V1.5.8 版本开始不再提供支持；

函数原型：

```
virtual int ReqQryInvestorPositionCombineDetail  
(CThostFtdcQryInvestorPositionCombineDetailField  
*pQryInvestorPositionCombineDetail, int nRequestID);
```

参数：

pQryInvestorPositionCombineDetail：请求查询投资者持仓明细信息，相关字段如下

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	ComblnstrumentID	组合持仓合约编码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

略

3.1.39 ReqQryExchangeMarginRate

功能：

请求查询交易所保证金率

函数原型：

```
virtual int ReqQryExchangeMarginRate(CThostFtdcQryExchangeMarginRateField  
*pQryExchangeMarginRate, int nRequestID);
```

参数：

pQryExchangeMarginRate：请求查询交易所保证金率信息，相关字段如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	是

nRequestID：终端请求编号

返回值：

0: 成功

-1: 失败

示例：略

3.1.40 ReqQryExchangeMarginRateAdjust

功能：

请求查询交易所调整保证金率

函数原型：

```
virtual int ReqQryExchangeMarginRateAdjust  
(CThostFtdcQryExchangeMarginRateAdjustField *pQryExchangeMarginRateAdjust,  
int nRequestID);
```

参数：

pQryExchangeMarginRateAdjust：请求查询交易所调整保证金率信息，相关字段如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	是

nRequestID：终端请求编号

返回值：

0: 成功

-1: 失败

示例：略

3.1.41 ReqQryOptionInstrTradeCost

功能：

请求查询期权交易成本

函数原型：

```
virtual int ReqQryOptionInstrTradeCost(CThostFtdcQryOptionInstrTradeCostField  
*pQryOptionInstrTradeCost, int nRequestID);
```

参数：

pQryOptionInstrTradeCost：请求查询期权交易成本，相关字段如下表

字段类型	字段名称	含义	过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	是
TThostFtdcPriceType	InputPrice	期权合约报价	否
TThostFtdcPriceType	UnderlyingPrice	标的价格	否

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryOptionInstrTradeCostField reqQryOptionInstrTradeCost = { 0 };

memset(&reqQryOptionInstrTradeCost, 0x00, sizeof(reqQryOptionInstrTradeCost));

strcpy(reqQryOptionInstrTradeCost.BrokerID, "8000");

strcpy(reqQryOptionInstrTradeCost.InvestorID, "100001");

strcpy(reqQryOptionInstrTradeCost.InstrumentID, "rb2102");

reqQryOptionInstrTradeCost.HedgeFlag = THOST_FTDC_HF_Speculation;

reqQryOptionInstrTradeCost.InputPrice = 300;

m_pTraderApi->ReqQryOptionInstrTradeCost(&reqQryOptionInstrTradeCost, m_requestID++);
```

3.1.42 ReqQryOptionInstrCommRate

功能：

请求查询期权合约手续费

函数原型：

```
virtual int ReqQryOptionInstrCommRate
(CThostFtdcQryOptionInstrCommRateField *pQryOptionInstrCommRate, int
nRequestID);
```

参数：

pQryOptionInstrCommRate：请求查询期权合约手续费信息，相关字段如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是

TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

示例:

```
CThostFtdcQryOptionInstrCommRateField reqQryOptionInstrCommRate = { 0 };

memset(&reqQryOptionInstrCommRate, 0x00, sizeof(reqQryOptionInstrCommRate));

strcpy(reqQryOptionInstrCommRate.BrokerID, "8000");
strcpy(reqQryOptionInstrCommRate.InvestorID, "100001");
strcpy(reqQryOptionInstrCommRate.InstrumentID, "rb2102");

m_pTraderApi->ReqQryOptionInstrCommRate(&reqQryOptionInstrCommRate, m_requestID++);
```

3.1.43 ReqQryExecOrder

功能:

请求查询执行宣告

函数原型:

```
virtual int ReqQryExecOrder(CThostFtdcQryExecOrderField *pQryExecOrder, int
nRequestID);
```

参数:

pQryExecOrder: 请求查询执行宣告信息, 相关字段如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcExecOrderSysIDType	ExecOrderSysID	执行宣告编号	是
TThostFtdcTimeType	InsertTimeStart	开始时间	是
TThostFtdcTimeType	InsertTimeEnd	结束时间	是

nRequestID: 终端请求编号

返回值:

0: 成功

-1：失败

示例：

```
CThostFtdcQryExecOrderField reqQryExecOrder = { 0 };

memset(&reqQryExecOrder, 0x00, sizeof(reqQryExecOrder));
strcpy(reqQryExecOrder.BrokerID, "8000");
strcpy(reqQryExecOrder.InvestorID, "100001");
strcpy(reqQryExecOrder.InstrumentID, "rb2102");
strcpy(reqQryExecOrder.ExchangeID, "SHFE");
m_pTraderApi->ReqQryExecOrder(&reqQryExecOrder, m_requestID++);
```

3.1.44 ReqQryForQuote

功能：

请求查询询价

函数原型：

```
virtual int ReqQryForQuote(CThostFtdcQryForQuoteField *pQryForQuote, int nRequestID);
```

参数：

pQryForQuote：请求查询询价，各字段信息如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcTimeType	InsertTimeStart	开始时间	是
TThostFtdcTimeType	InsertTimeEnd	结束时间	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

略

3.1.45 ReqQryQuote

功能：

请求查询报价

函数原型：

```
virtual int ReqQryQuote(CThostFtdcQryQuoteField *pQryQuote, int nRequestID);
```

参数：

pQryQuote：请求查询报价信息，相关字段信息如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcOrderSysIDType	QuoteSysID	报价编号	是
TThostFtdcTimeType	InsertTimeStart	开始时间	是
TThostFtdcTimeType	InsertTimeEnd	结束时间	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

略

3.1.46 ReqQryInstrumentOrderCommRate

功能：

请求查询经纪公司报单手续费

函数原型：

```
virtual int ReqQryInstrumentOrderCommRate
(CThostFtdcQryInstrumentOrderCommRateField *pQryInstrumentOrderCommRate,
int nRequestID);
```

参数：

pQryInstrumentOrderCommRate：请求查询报单手续费信息，相关字段如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是

TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

```
CThostFtdcQryInstrumentOrderCommRateField reqQryInstrumentOrderCommRate = { 0 };
memset(&reqQryInstrumentOrderCommRate, 0x00, sizeof(reqQryInstrumentOrderCommRate));
strcpy(reqQryInstrumentOrderCommRate.BrokerID, "8000");
strcpy(reqQryInstrumentOrderCommRate.InvestorID, "100001");
strcpy(reqQryInstrumentOrderCommRate.InstrumentID, "rb2102");
m_pTraderApi->ReqQryInstrumentOrderCommRate(&reqQryInstrumentOrderCommRate, 1);
```

3.1.47 ReqCombActionInsert

功能：

申请组合/解锁录入请求

函数原型：

```
virtual int ReqCombActionInsert(CThostFtdcInputCombActionField
*pInputCombAction, int nRequestID) = 0;
```

参数：

pInputCombAction：申请组合录入请求字段，包括以下内容：

字段类型	字段名称	含义	是否必须
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcOrderRefType	CombActionRef	组合引用	是
TThostFtdcUserIDType	UserID	用户代码	是
TThostFtdcDirectionType	Direction	组合买卖方向	是
TThostFtdcVolumeType	Volume	数量	是
TThostFtdcCombDirectionType	CombDirection	组合指令方向：组合、解锁	是
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	是
TThostFtdcBusinessUnitType	BusinessUnit	业务单元--new	否
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是

TThostFtdcIPAddressType	IPAddress	IP 地址	否
TThostFtdcMacAddressType	MacAddress	Mac 地址	否
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码--new	否
TThostFtdcFrontIDType	FrontID	前置编号--new	否
TThostFtdcSessionIDType	SessionID	会话编号--new	否

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

3.1.48 ReqQryCombAction

功能:

请求查询申请组合;

函数原型:

```
virtual int ReqQryCombAction(CThostFtdcQryCombActionField*
pQryCombAction, int nRequestID) = 0;
```

参数:

pQryCombAction, 申请组合查询:

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是

nRequestID: 终端请求编号

返回值:

0: 成功

-1: 失败

3.1.49 ReqQryComInstrument

功能:

请求查询申请组合合约;

函数原型：

```
virtual int ReqQryCombInstrument(CThostFtdcQryCombInstrumentField*  
pQryCombInstrument, int nRequestID) = 0;
```

参数：

pQryCombInstrument 查询申请组合合约：

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcInstrumentIDType	ProductID	产品代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

3.1.50 ReqQryInvestorPositionForComb

功能：

请求查询组合仓中单腿汇总信息；

函数原型：

```
virtual int ReqQryInvestorPositionForComb  
(CThostFtdcQryInvestorPositionForCombField* pQryIPForComb, int nRequestID) = 0;
```

参数：

pQryIPForComb 申请组合单腿汇总查询：

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	LegInstrumentID	单腿合约代码	是

nRequestID：终端请求编号

返回值：

0：成功

-1: 失败

3.1.51 ReqQryForQuoteParam

功能：

请求查询询价价差参数；如果查询条件全为空，则查询全部询价价差参数信息。

函数原型：

```
virtual int ReqQryForQuoteParam(CThostFtdcQryForQuoteParamField  
*pQryForQuoteParam, int nRequestID) = 0;
```

参数：

pQryForQuoteParam 询价价差查询条件：

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInstrumentIDType	InstrumentID	合约代码	是
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是

nRequestID：终端请求编号

返回值：

0: 成功

-1: 失败

3.1.52 ReqMKBatchOrderAction

功能：

做市商批量报单操作请求。

V1.6.1 版本开始支持，支持两种方式筛选方式批量撤单/一键撤单，二选一：

- 根据产品、合约、报单状态组合条件实现条件操作，ActionMode=THOST_FTDC_MKAM_Product。
 - 终端输入条件：产品 ProductID + 合约 InstrumentID + 定单状态 OrderStatus，其中 ProductID 必填，InstrumentID 和 OrderStatus 选填。
- 按照客户 orderref 起、止编号进行条件操作，ActionMode=THOST_FTDC_MKAM_OrderRef。
 - 终端输入条件：上限条件 UpperLimit+下限条件 LowerLimit+前置号 FrontID+会话号 SessionID+定单状态 OrderStatus，其中 UpperLimit, LowerLimit+,

- FrontID, SessionID 必填, 定单状态 OrderStatus 选填;
- 该模式下, 上限条件 UpperLimit 和下限条件 LowerLimit 必须填写纯数字, 例如: " 1" , " 01" , " 001" ... " 000000000001" , 类似格式有效。对应要用该模式撤销的报单, 报单引用的编号, 必须是" 000000000001" 12 位, 前补 0 格式, 其他格式无法与上下限条件匹配;
 - 每次接口调用后返回成功上报的可撤单数量。
 - 性能提醒: 如果单合约报单量较大, 批量撤单模式 ActionMode=THOST_FTDC_MKAM_Product 时, 有一定性能损耗; 此时推荐使用 ActionMode=THOST_FTDC_MKAM_OrderRef。

CTP Mini 后台交易系统参数控制:

- 为控制大量撤单影响系统性能及其他不可知风险, 后台增加最大撤单数量控制; 如最大撤单数量初始值设置为 100。

函数原型:

```
virtual     int     ReqMKBatchOrderAction(CThostFtdcMKInputOrderActionField
*pMKInputOrderAction, int nRequestID) = 0;
```

参数:

pMKInputOrderAction 做市商报单操作条件:

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	无
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcOrderActionRefType	OrderActionRef	报单操作引用	无
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcFrontIDType	FrontID	前置编号	ActionMode=THOST_FTDC_MKAM_OrderRef 时必填
TThostFtdcSessionIDType	SessionID	会话编号	ActionMode=THOST_FTDC_MKAM_OrderRef 时必填
TThostFtdcExchangeIDType	ExchangeID	交易所代码	必填
TThostFtdcActionFlagType	ActionFlag	操作标志只支持删除, 不支持修改	必填
TThostFtdcMKActionModeType	ActionMode	操作模式	必填

TThostFtdcLowerLimitType	LowerLimit	下限	ActionMode= THOST_FTDC_MKAM_OrderRef 时必填，且该模式下必须填写纯数字
TThostFtdcUpperLimitType	UpperLimit	上限	ActionMode= THOST_FTDC_MKAM_OrderRef 时必填，且该模式下必须填写纯数字
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcInstrumentIDType	ProductID	产品代码	ActionMode= THOST_FTDC_MKAM_Product 时必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	ActionMode= THOST_FTDC_MKAM_Product 时可选
TThostFtdcOrderStatusType	OrderStatus	报单状态	可选
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	无
TThostFtdcIPAddressType	IPAddress	IP 地址	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

nRequestID：终端请求编号

返回值：

- 0：成功
- 1：失败

3.1.53 SubscribeFlowCtrlWarning

功能：

订阅交易所流控警告。

该方法必须在登录成功之后调用。

订阅请求，可以在盘中随时调用，即可以实现动态管理。

函数原型：

```
virtual int SubscribeFlowCtrlWarning(char *ppTraderID[], int nCount) = 0;
```

参数：

ppTraderID：席位 ID 数组，填写需要订阅的席位号；

nCount：要订阅的的席位个数；

返回值：

- 0：成功
- 1：交易或查询会话没有创建

-2：用户没有登录

3.1.54 UnSubscribeFlowCtrlWarning

功能：

取消订阅交易所流控警告。

取消订阅请求，可以在盘中随时调用，即可以实现动态管理。

函数原型：

```
virtual int UnSubscribeFlowCtrlWarning (char *ppTraderID[], int nCount) = 0;
```

参数：

ppTraderID：需要取消订阅的席位号；

nCount：要取消订阅的席位个数；

返回值：

0：成功

-1：交易或查询会话没有创建

-2：用户没有登录

3.1.55 ReqQryTraderOffer

功能：

请求查询交易员席位状态。

特别说明：该查询功能需要后台做市商功能开关开启，同时席位号需要得到交易系统管理人员授权。

函数原型：

```
virtual int ReqQryTraderOffer(CThostFtdcQryTraderOfferField *pQryTraderOffer,  
int nRequestID) = 0;
```

参数：

pQryTraderOffer：需要查询交易所及对应的席位号；

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcTraderIDType	TraderID	交易所交易员代码（席位号）	是

nRequestID：终端请求编号

返回值：

- 0：成功
- 1：交易或查询会话没有创建
- 2：重复登录
- 3：用户没有登录

3.1.56 ReqQryInvestorProdSPBMDetail

功能：

请求查询投资者 SPBM 品种保证金明细。

函数原型：

```
virtual int ReqQryInvestorProdSPBMDetail  
( CThostFtdcQryInvestorProdSPBMDetailField *pQryInvestorProdSPBMDetail, int  
nRequestID) = 0;
```

参数：

pQryInvestorProdSPBMDetail: SPBM 品种保证金明细查询条件

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	ProdFamilyCode	品种代码	是

nRequestID: 终端请求编号

返回值：

- 0：成功
- 1：交易或查询会话没有创建
- 2：重复登录
- 3：用户没有登录

3.1.57 ReqQrySPMMInvestorCommodityGroupMargin

功能：

请求查询投资者上期所或能源中心 SPMM 商品群保证金明细。

函数原型：

```
virtual int ReqQrySPMMInvestorCommodityGroupMargin
(CThostFtdcQrySPMMInvestorCommodityGroupMarginField
*pQrySPMMInvestorCommodityGroupMargin, int nRequestID) = 0;
```

参数：

pQrySPMMInvestorCommodityGroupMargin：SPMM 品种保证金明细查询条件

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcSPMMProductIDType	CommodityGroupID	商品群代码	是

nRequestID：终端请求编号

返回值：

- 0：成功
- 1：交易或查询会话没有创建
- 2：重复登录
- 3：用户没有登录

3.1.58 ReqQryRULEInvestorProdMargin

功能：

请求查询大商所投资者 RULE 保证金明细。

函数原型：

```
Virtual int
ReqQryRULEInvestorProdMargin( CThostFtdcQryRULEInvestorProdMarginField*
pQryRULEInvestorProdMargin, int nRequestID) = 0;
```

参数：

pQryRULEInvestorProdMargin: RULE 保证金明细查询条件

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcExchangeIDType	ExchangeID	交易所代码	是
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcProductIDType	ProdFamilyCode	品种代码	是
TThostFtdcRULECommodityGroupIDType	CommodityGroupID	商品群代码	是

nRequestID: 终端请求编号

返回值：

- 0: 成功
- 1: 查询会话没有创建
- 2: 重复登录
- 3: 用户没有登录

3.1.59 ReqQryRCAMSInvestorProdMargin

功能：

请求查询中金所投资者 RCAMS 保证金明细。

函数原型：

```
virtual int ReqQryRCAMSInvestorProdMargin(CThostFtdcQryRCAMSInvestorProdMarginField*
```

```
pQryRCAMSInvestorProdMargin, int nRequestID) = 0;
```

参数：

pQryRCAMSInvestorProdMargin: RCAMS 保证金明细查询条件

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是

TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	CombProductID	产品代码	是

nRequestID：终端请求编号

返回值：

- 0：成功
- 1：查询会话没有创建
- 2：重复登录
- 3：用户没有登录

3.1.60 ReqQryRCAMSInvestorCombPosition

功能：

请求查询 RCAMS 策略组合持仓。

函数原型：

virtual

int

```
ReqQryRCAMSInvestorCombPosition(CThostFtdcQryRCAMSInvestorCombPositionField*
d* pQryRCAMSInvestorCombPosition, int nRequestID) = 0;
```

参数：

pQryRCAMSInvestorCombPosition：查询条件

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcInstrumentIDType	InstrumentID	单腿合约代码	是

nRequestID：终端请求编号

返回值：

- 0：成功
- 1：查询会话没有创建
- 2：重复登录
- 3：用户没有登录

3.1.61 ReqQryControlParam

功能：

请求查询部分后台功能与设置。

特殊说明：本接口需要期货公司授权并设置相关参数后才可以使用。

第一期开放的开关或功能参数如下表（后续根据市场需求进一步开放）：

ControlParamID	ControlParamValue	Memo	其他说明
S	1: PRD 生产系统 2: SIM 仿真系统 3: DEV 开发系统	交易系统类型	查询后台 CTP Mini 交易系统类型
C	1-开启； 0-未开启	自成交检测	是否开启自成交检查
P	1-开启； 0-未开启	会话流控	是否开启 Session 级别流控
T	1-开启； 0-未开启	做市商批量交易开关	是否开启做市商批量功能
L	最大失败登录次数值	最大登录尝试次数	登录失败超过最大次数后锁定不允许登录
其他值	Invalid ControlParamID	无效参数	

函数原型：

```
int ReqQryControlParam(CThostFtdcQryControlParamField
*pQryControlParam, int nRequestId);
```

参数：

CThostFtdcQryControlParamField: 查询条件

字段类型	字段名称	含义	是否可作为过滤条件
TTHostFtdcInvestorIDType	InvestorID	投资者代码	是
TTHostFtdcControlParamIDType	ControlParamID	开关代码	是

nRequestId：终端请求编号

返回值：

0：成功

- 1: 查询会话没有创建
- 2: 重复登录
- 3: 用户没有登录

3.1.62 ReqSubscribeFundChange

功能：

订阅资金变动。

该方法必须在登录成功之后调用。

函数原型：

```
virtual int ReqSubscribeFundChange(int nRequestID) = 0;
```

参数：

nRequestID: 终端请求编号

返回值：

0: 成功

-1: 交易或查询会话没有创建

-2: 用户没有登录

3.1.63 ReqUnSubscribeFundChange

功能：

取消订阅资金变动。

该方法必须在登录成功之后调用。

函数原型：

```
virtual int ReqUnSubscribeFundChange(int nRequestID) = 0;
```

参数：

nRequestID: 终端请求编号

返回值：

0：成功

-1：交易或查询会话没有创建

-2：用户没有登录

3.1.64 ReqOffsetSetting

功能：

大商所对冲设置请求；

注意：本接口仅适用大商所，并且需要后台交易系统升级到支持大商所官方 API V1.4.4 及以上版本。

函数原型：

```
virtual int ReqOffsetSetting(CThostFtdcInputOffsetSettingField
*pInputOffsetSetting, int nRequestID);
```

参数：

pInputOffsetSetting：输入的对冲设置，相关字段信息如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	见后说明
TThostFtdcInstrumentIDType	UnderlyingInstrID	标的期货合约代码	见后说明
TThostFtdcProductIDType	ProductID	产品代码	见后说明
TThostFtdcOffsetTypeType	OffsetType	对冲类型	必填
TThostFtdcVolumeType	Volume	申请对冲的合约数量	合约级对冲必填
TThostFtdcBoolType	IsOffset	是否对冲	必填
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcExchangeIDType	ExchangeID	交易所代码	必填 "DCE"
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	可以指定席位
TThostFtdcIPAddressType	IPAddress	IP 地址	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

关于 InstrumentID、UnderlyingInstrID、ProductID 填写规范：

(1) 期权对冲，选填 InstrumentID(期权合约)、UnderlyingInstrID(标的期货合约)、ProductID(期权产品)三个字段，至少一个不为空，**后续大商所支持三个都为空另行通知**；

(2) 期货对冲，选填 InstrumentID(期货合约)、ProductID(期货品种)字段，UnderlyingInstrID 字段不填；

(3) 行权后对冲、履约后对冲，选填 UnderlyingInstrID(标的期货合约)、ProductID(期权品种)字段，InstrumentID 不填；

Volume：设置合约级别的期权对冲和期货对冲且 IsOffset=1 时，要求填入的 Volume 在 [1,99999999] 区间内，其他情况不检查手数字段，99999999 表示全部手数。

RequestID：对于同一“投资者+对冲粒度+对冲类型”的对冲，后台只会保留一笔；若用户需要区分每笔不同的报单，建议使用 RequestID 字段区分，此字段只做回显，不做其他检查。

例：

(1) 期权对冲

品种级别对冲：ProductID 需要填具体，UnderlyingInstrID、InstrumentID 为空

期权系列级别对冲：ProductID 为空或具体，UnderlyingInstrID 需要填具体，InstrumentID 为空；

合约级别对冲：ProductID、UnderlyingInstrID 为空或者具体，InstrumentID 需要填具体，Volume 在 [1,99999999] 区间内，99999999 表示全部手数；

(2) 行权后对冲、履约后对冲

客户级别对冲：ProductID、UnderlyingInstrID 为空

品种级别对冲：ProductID 需要填具体，UnderlyingInstrID 为空

期权系列级别对冲：ProductID 为空或者具体，UnderlyingInstrID 为具体

(3) 期货对冲

客户级别对冲：ProductID、InstrumentID 为空

品种级别对冲：ProductID 需要填具体，InstrumentID 为空

合约级别对冲：ProductID 为空或者具体，InstrumentID 需要填具体，Volume 在 [1,99999999] 区间内，99999999 表示全部手数

nRequestId：终端请求编号

返回值：

0：成功

-1：失败

3.1.65 ReqCancelOffsetSetting

功能：

撤销对冲设置请求；

注意：本接口仅适用大商所，并且需要后台交易系统升级到支持大商所官方 API V1.4.4 及以上版本。

函数原型：

```
virtual      int      ReqCancelOffsetSetting(CThostFtdcInputOffsetSettingField
*pInputOffsetSetting, int nRequestID) = 0;
```

参数：

pInputOffsetSetting：输入的对冲设置，相关字段信息如下表

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	必填
TThostFtdcInvestorIDType	InvestorID	投资者代码	必填
TThostFtdcInstrumentIDType	InstrumentID	合约代码	和报入一致
TThostFtdcInstrumentIDType	UnderlyingInstrID	标的期货合约代码	和报入一致
TThostFtdcProductIDType	ProductID	产品代码	和报入一致
TThostFtdcOffsetTypeType	OffsetType	对冲类型	必填
TThostFtdcVolumeType	Volume	申请对冲的合约数量	无
TThostFtdcBoolType	IsOffset	是否对冲	无
TThostFtdcRequestIDType	RequestID	请求编号	无
TThostFtdcUserIDType	UserID	用户代码	无
TThostFtdcExchangeIDType	ExchangeID	交易所代码	必填"DCE"
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	无
TThostFtdcIPAddressType	IPAddress	IP 地址	无
TThostFtdcMacAddressType	MacAddress	Mac 地址	无

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

3.1.66 ReqQryOffsetSetting

功能：

投资者对冲设置查询，投资者对冲设置查询响应 OnRspQryOffsetSetting

注意：该接口仅适用大商所。

函数原型：

```
virtual int ReqQryOffsetSetting(CThostFtdcQryOffsetSettingField  
*pQryOffsetSetting, int nRequestID) = 0;
```

参数：

pQryOffsetSetting：查询对冲设置，相关字段如下：

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	是
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcProductIDType	ProductID	产品代码	是
TThostFtdcOffsetTypeType	OffsetType	对冲类型	是

nRequestID：终端请求编号

返回值：

0：成功

-1：查询会话没有创建

-2：重复登录

-3：用户没有登录

3.2 CThostFtdcTraderSpi

3.2.1 OnFrontConnected

功能：

当客户端与交易后台建立起通信连接时（还未登录前），该方法被调用。

函数原型：

```
virtual void OnFrontConnected();
```

参数：

无；

3.2.2 OnFrontDisconnected

功能：

当客户端与交易后台通信连接断开时，该方法被调用。

当发生这个情况后，API 会自动重新连接，客户端可不做处理。

函数原型：

virtual void OnFrontDisconnected(int nReason);

参数：

nReason：错误原因代码，含义如下

- -3 关闭连接
- -4 网络读失败
- -5 网络写失败
- -6 读订阅流水请求出错
- -7 序列号错误
- -8 读心跳出错
- -9 错误的网络包大小

3.2.3 OnRspUserLogin

功能：

登录请求响应

函数原型：

**virtual void OnRspUserLogin(CThostFtdcRspUserLoginField *pRspUserLogin,
CThostFtdcRspInfoField *pRsplInfo, int nRequestId, bool bIsLast);**

参数：

pRspUserLogin：登录请求响应信息，对应的结构各字段如下表：

字段类型	字段名称	含义	备注
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcTimeType	LoginTime	登录成功时间	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcSystemNameType	SystemName	交易系统名称	V1.6.8 版本 给以赋值
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcOrderRefType	MaxOrderRef	最大报单引用	

TThostFtdcTimeType	SHFETime	上期所时间	
TThostFtdcTimeType	DCETime	大商所时间	
TThostFtdcTimeType	CZCETime	郑商所时间	
TThostFtdcTimeType	FFEXTime	中金所时间	
TThostFtdcTimeType	INETime	能源中心时间	
TThostFtdcTimeType	GFEXTime	广期所时间	V1.6.4 版本 开始增加

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true；

3.2.4 OnRspAuthenticate

功能：

客户端认证响应

函数原型：

```
virtual void OnRspAuthenticate(CThostFtdcRspAuthenticateField
*pRspAuthenticateField, CThostFtdcRsplInfoField *pRspInfo, int nRequestID, bool
bIsLast);
```

参数：

pRspAuthenticateField: 客户端认证响应信息，对应的结构各字段如下表：

字段类型	字段名称	含义	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息	
TThostFtdcClientAppIDType	AppID	App 代码	

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.5 OnRspOrderInsert

功能：

报单录入请求响应

函数原型：

```
virtual void OnRspOrderInsert(CThostFtdcInputOrderField *pInputOrder,
CThostFtdcRsplInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pInputOrder: 报单录入请求响应信息，对应的结构各字段参见《ReqOrderInsert》章节。

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.6 OnRspOrderAction

功能：

报单操作请求响应

函数原型：

```
virtual void OnRspOrderAction(CThostFtdcInputOrderActionField  
*pInputOrderAction, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast);
```

参数：

pInputOrderAction: 报单操作请求响应信息，对应的结构各字段参见《ReqOrderAction》章节。

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.7 OnRspExecOrderInsert

功能：

执行宣告录入请求响应

函数原型：

```
virtual void OnRspExecOrderInsert(CThostFtdcInputExecOrderField  
*pInputExecOrder, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast);
```

参数：

pInputExecOrder : 执行宣告录入请求响应信息，对应的结构各字段参见《ReqExecOrderInsert》章节。

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.8 OnRspExecOrderAction

功能:

执行宣告操作请求响应

函数原型:

```
virtual void OnRspExecOrderAction(CThostFtdcInputExecOrderActionField  
*pInputExecOrderAction, CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool  
bIsLast);
```

参数:

pInputExecOrderAction: 执行宣告操作请求响应信息, 对应的结构各字段参见《ReqExecOrderAction》章节。

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息 ErrorMsg;

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.9 OnRspForQuoteInsert

功能:

询价录入请求响应

- 只在录入错误的情况下返回, 如字段校验失败, 可以通过 ErrorID 和 ErrorMsg 查看错误信息;

函数原型:

```
virtual void OnRspForQuoteInsert(CThostFtdcInputForQuoteField  
*pInputForQuote, CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool  
bIsLast);
```

参数:

pInputForQuote: 询价录入请求响应信息, 对应的结构各字段参见《ReqForQuoteInsert》章节;

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息 ErrorMsg;

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.10 OnRspQuoteInsert

功能：

报价录入请求响应

- 只在录入错误的情况下返回，如字段校验失败等；可以通过 ErrorID 和 ErrorMsg 查看错误信息；

函数原型：

```
virtual void OnRspQuoteInsert(CThostFtdcInputQuoteField *pInputQuote,  
CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast);
```

参数：

pInputQuote: 报价录入请求响应信息，对应的结构各字段参见《ReqQuoteInsert》章节。

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.11 OnRspQuoteAction

功能：

报价操作请求响应

函数原型：

```
virtual void OnRspQuoteAction(CThostFtdcInputQuoteActionField  
*pInputQuoteAction, CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool  
bIsLast);
```

参数：

pInputQuoteAction: 报价操作请求响应信息，对应的结构各字段参见《ReqQuoteAction》章节。

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.12 OnRspOptionSelfCloseInsert

功能：

期权自对冲录入请求响应

函数原型：

virtual void

OnRspOptionSelfCloseInsert(CThostFtdcInputOptionSelfCloseField

*pInputOptionSelfClose, CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool
bIsLast);

参数：

pInputOptionSelfClose: 期权自对冲录入请求响应信息，对应的结构各字段参见
《ReqOptionSelfCloseInsert》章节。

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.13 OnRspOptionSelfCloseAction

功能：

期权自对冲操作请求响应

函数原型：

virtual void OnRspOptionSelfCloseAction
(CThostFtdcInputOptionSelfCloseActionField *pInputOptionSelfCloseAction,
CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast);

参数：

pInputOptionSelfCloseAction: 期权自对冲操作请求响应信息，对应的结构各字段参见
《ReqOptionSelfCloseAction》章节。

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.14 OnRspQryOrder

功能：

请求查询报单响应

函数原型：

virtual void OnRspQryOrder(CThostFtdcOrderField *pOrder,
CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast);

参数：

pOrder: 请求查询报单响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderRefType	OrderRef	报单引用	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcOrderPriceTypeType	OrderPriceType	报单价格条件	
TThostFtdcDirectionType	Direction	买卖方向	
TThostFtdcCombOffsetFlagType	CombOffsetFlag	组合开平标志	
TThostFtdcCombHedgeFlagType	CombHedgeFlag	组合投机套保标志	
TThostFtdcPriceType	LimitPrice	价格	
TThostFtdcVolumeType	VolumeTotalOriginal	数量	
TThostFtdcTimeConditionType	TimeCondition	有效期类型	
TThostFtdcDateType	GTDDate	GTD 日期	
TThostFtdcVolumeConditionType	VolumeCondition	成交量类型	
TThostFtdcVolumeType	MinVolume	最小成交量	
TThostFtdcContingentConditionType	ContingentCondition	触发条件	
TThostFtdcPriceType	StopPrice	止损价	
TThostFtdcForceCloseReasonType	ForceCloseReason	强平原因	
TThostFtdcBoolType	IsAutoSuspend	自动挂起标志	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcOrderLocalIDType	OrderLocalID	本地报单编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderSubmitStatusType	OrderSubmitStatus	报单提交状态	
TThostFtdcSequenceNoType	NotifySequence	报单提示序号	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcOrderSysIDType	OrderSysID	报单编号	
TThostFtdcOrderSourceType	OrderSource	报单来源	
TThostFtdcOrderStatusType	OrderStatus	报单状态	
TThostFtdcOrderTypeType	OrderType	报单类型	
TThostFtdcVolumeType	VolumeTraded	今成交量	
TThostFtdcVolumeType	VolumeTotal	剩余数量	
TThostFtdcDateType	InsertDate	报单日期	
TThostFtdcTimeType	InsertTime	委托时间	
TThostFtdcTimeType	ActiveTime	激活时间	
TThostFtdcTimeType	SuspendTime	挂起时间	

TThostFtdcTimeType	UpdateTime	最后修改时间	
TThostFtdcTimeType	CancelTime	撤销时间	
TThostFtdcTraderIDType	ActiveTraderID	最后修改交易所交易员代码	
TThostFtdcParticipantIDType	ClearingPartID	结算会员编号	
TThostFtdcSequenceNoType	SequenceNo	序号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcBoolType	UserForceClose	用户强评标志	
TThostFtdcUserIDType	ActiveUserID	操作用户代码	
TThostFtdcSequenceNoType	BrokerOrderSeq	经纪公司报单编号	
TThostFtdcOrderSysIDType	RelativeOrderSysID	相关报单	
TThostFtdcVolumeType	ZCETotalTradedVolume	郑商所成交数量	
TThostFtdcBoolType	IsSwapOrder	互换单标志	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcAccountIDType	AccountID	资金账号	
TThostFtdcCurrencyIDType	CurrencyID	币种代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcCustomOrderRefType	CustomOrderRef	客户自定义字段	
TThostFtdcPriceType	TradeAvgPrice	成交均价	V1.6.9 增加

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.15 OnRspQryTrade

功能：

请求查询成交响应

函数原型：

```
virtual void OnRspQryTrade(CThostFtdcTradeField *pTrade,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pTrade： 请求查询成交响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	

TThostFtdcInstrumentIDType	InstrumentID	合约代码
TThostFtdcOrderRefType	OrderRef	报单引用
TThostFtdcUserIDType	UserID	用户代码
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcTradeIDType	TradeID	成交编号
TThostFtdcDirectionType	Direction	买卖方向
TThostFtdcOrderSysIDType	OrderSysID	报单编号
TThostFtdcParticipantIDType	ParticipantID	会员代码
TThostFtdcClientIDType	ClientID	客户代码
TThostFtdcTradingRoleType	TradingRole	交易角色
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码
TThostFtdcOffsetFlagType	OffsetFlag	开平标志
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志
TThostFtdcPriceType	Price	成交价格；对于 TAS 交易成交价格没有意义；
TThostFtdcVolumeType	Volume	数量
TThostFtdcDateType	TradeDate	成交时期
TThostFtdcTimeType	TradeTime	成交时间
TThostFtdcTradeTypeType	TradeType	成交类型；对于 TAS 交易，为普通成交 THOST_FTDC_TRDT_Common
TThostFtdcPriceSourceType	PriceSource	成交价来源
TThostFtdcTraderIDType	TraderID	交易所交易员代码
TThostFtdcOrderLocalIDType	OrderLocalID	本地报单编号
TThostFtdcParticipantIDType	ClearingPartID	结算会员编号
TThostFtdcBusinessUnitType	BusinessUnit	业务单元
TThostFtdcSequenceNoType	SequenceNo	序号
TThostFtdcDateType	TradingDay	交易日
TThostFtdcSettlementIDType	SettlementID	结算编号
TThostFtdcSequenceNoType	BrokerOrderSeq	经纪公司报单编号
TThostFtdcTradeSourceType	TradeSource	成交来源

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.16 OnRspQryInvestorPosition

功能：

请求查询投资者持仓响应

函数原型：

```
virtual void OnRspQryInvestorPosition(CThostFtdcInvestorPositionField
*pInvestorPosition, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
```

bIsLast) ;

参数：

pInvestorPosition: 请求查询投资者持仓响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcPosiDirectionType	PosiDirection	持仓多空方向	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcPositionDateType	PositionDate	持仓日期标志，‘1’表示当前交易日持仓；‘2’表示是历史仓（昨仓）。只有上期/能源交易所的合约才可能有 PositionDate 为‘2’的持仓记录	
TThostFtdcVolumeType	YdPosition	上日持仓，静态值，为当前交易日开始时客户该合约的历史仓位，并不会随着今日平仓而减小	
TThostFtdcVolumeType	Position	今日持仓，总持仓，随着客户开、平仓而变动	
TThostFtdcVolumeType	LongFrozen	多头冻结	
TThostFtdcVolumeType	ShortFrozen	空头冻结，平多头仓位报单未成交的数量	
TThostFtdcMoneyType	LongFrozenAmount	开仓冻结金额	
TThostFtdcMoneyType	ShortFrozenAmount	开仓冻结金额	
TThostFtdcVolumeType	OpenVolume	开仓量	
TThostFtdcVolumeType	CloseVolume	平仓量	
TThostFtdcMoneyType	OpenAmount	开仓金额	
TThostFtdcMoneyType	CloseAmount	平仓金额	
TThostFtdcMoneyType	PositionCost	持仓成本	
TThostFtdcMoneyType	PreMargin	上次占用的保证金	
TThostFtdcMoneyType	UseMargin	占用的保证金	
TThostFtdcMoneyType	FrozenMargin	冻结的保证金	
TThostFtdcMoneyType	FrozenCash	冻结的资金	
TThostFtdcMoneyType	FrozenCommission	冻结的手续费	
TThostFtdcMoneyType	CashIn	资金差额	
TThostFtdcMoneyType	Commission	手续费	
TThostFtdcMoneyType	CloseProfit	平仓盈亏	
TThostFtdcMoneyType	PositionProfit	持仓盈亏	
TThostFtdcPriceType	PreSettlementPrice	上次结算价	
TThostFtdcPriceType	SettlementPrice	本次结算价	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcMoneyType	OpenCost	开仓成本	

TThostFtdcMoneyType	ExchangeMargin	交易所保证金	
TThostFtdcVolumeType	CombPosition	组合成交形成的持仓	
TThostFtdcVolumeType	CombLongFrozen	组合多头冻结	
TThostFtdcVolumeType	CombShortFrozen	组合空头冻结	
TThostFtdcMoneyType	CloseProfitByDate	逐日盯市平仓盈亏	
TThostFtdcMoneyType	CloseProfitByTrade	逐笔对冲平仓盈亏	
TThostFtdcVolumeType	TodayPosition	今日持仓, 今日新持仓数量, 随着客户开仓平今而变动	
TThostFtdcRatioType	MarginRateByMoney	保证金率	
TThostFtdcRatioType	MarginRateByVolume	保证金率(按手数)	
TThostFtdcVolumeType	StrikeFrozen	执行冻结	
TThostFtdcMoneyType	StrikeFrozenAmount	执行冻结金额	
TThostFtdcVolumeType	AbandonFrozen	放弃执行冻结	

- 开头这 6 个字段(InstrumentID、BrokerID、InvestorID、PosiDirection、HedgeFlag、PositionDate) 称之为持仓记录的键值, 通过这 6 个字段就可以定位唯一一条持仓记录;
- 对于非上期/能源的交易所, 合约的昨仓和今仓在一条记录里面, 而上期/能源是分成了两条记录体现;

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息 ErrorMsg;

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.17 OnRspQryTradingAccount

功能:

请求查询资金账户响应

函数原型:

```
virtual void OnRspQryTradingAccount(CThostFtdcTradingAccountField *pTradingAccount, CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast);
```

参数:

pTradingAccount: 请求查询资金账户响应信息, 对应的结构各字段如下表:

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcAccountIDType	AccountID	投资者帐号	
TThostFtdcMoneyType	PreMortgage	上次质押金额	
TThostFtdcMoneyType	PreCredit	上次信用额度	
TThostFtdcMoneyType	PreDeposit	上次存款额	
TThostFtdcMoneyType	PreBalance	上次结算准备金	

TThostFtdcMoneyType	PreMargin	上次占用的保证金
TThostFtdcMoneyType	InterestBase	利息基数
TThostFtdcMoneyType	Interest	利息收入
TThostFtdcMoneyType	Deposit	入金金额
TThostFtdcMoneyType	Withdraw	出金金额
TThostFtdcMoneyType	FrozenMargin	冻结的保证金
TThostFtdcMoneyType	FrozenCash	冻结的资金
TThostFtdcMoneyType	FrozenCommission	冻结的手续费
TThostFtdcMoneyType	CurrMargin	当前保证金总额
TThostFtdcMoneyType	CashIn	资金差额
TThostFtdcMoneyType	Commission	手续费
TThostFtdcMoneyType	CloseProfit	平仓盈亏
TThostFtdcMoneyType	PositionProfit	持仓盈亏
TThostFtdcMoneyType	Balance	期货结算准备金
TThostFtdcMoneyType	Available	可用资金
TThostFtdcMoneyType	WithdrawQuota	可取资金
TThostFtdcMoneyType	Reserve	基本准备金
TThostFtdcDateType	TradingDay	交易日
TThostFtdcSettlementIDType	SettlementID	结算编号
TThostFtdcMoneyType	Credit	信用额度
TThostFtdcMoneyType	Mortgage	质押金额
TThostFtdcMoneyType	ExchangeMargin	交易所保证金
TThostFtdcMoneyType	DeliveryMargin	投资者交割保证金
TThostFtdcMoneyType	ExchangeDeliveryMargin	交易所交割保证金
TThostFtdcMoneyType	ReserveBalance	保底期货结算准备金
TThostFtdcCurrencyIDType	CurrencyID	币种代码
TThostFtdcMoneyType	PreFundMortgageln	上次货币质入金额
TThostFtdcMoneyType	PreFundMortgageOut	上次货币质出金额
TThostFtdcMoneyType	FundMortgageln	货币质入金额
TThostFtdcMoneyType	FundMortgageOut	货币质出金额
TThostFtdcMoneyType	FundMortgageAvailable	货币质押余额
TThostFtdcMoneyType	MortgageableFund	可质押货币金额
TThostFtdcMoneyType	SpecProductMargin	特殊产品占用保证金
TThostFtdcMoneyType	SpecProductFrozenMargin	特殊产品冻结保证金
TThostFtdcMoneyType	SpecProductCommission	特殊产品手续费
TThostFtdcMoneyType	SpecProductFrozenCommission	特殊产品冻结手续费
TThostFtdcMoneyType	SpecProductPositionProfit	特殊产品持仓盈亏
TThostFtdcMoneyType	SpecProductCloseProfit	特殊产品平仓盈亏
TThostFtdcMoneyType	SpecProductPositionProfitByAlg	根据持仓盈亏算法计算的特殊产品持仓盈亏
TThostFtdcMoneyType	SpecProductExchangeMargin	特殊产品交易所保证金
TThostFtdcMoneyType	FrozenPartition	分仓冻结资金

pRsplInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.18 OnRspQryInvestor

功能：

请求查询投资者响应

函数原型：

```
virtual void OnRspQryInvestor(CThostFtdcInvestorField *pInvestor,
CThostFtdcRspInfoField *pRsplInfo, int nRequestId, bool bIsLast);
```

参数：

pInvestor: 请求查询投资者响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorGroupID	投资者分组代码	
TThostFtdcPartyNameType	InvestorName	投资者名称	
TThostFtdcIdentifiedCardTypeType	IdentifiedCardType	证件类型	
TThostFtdcIdentifiedCardNoType	IdentifiedCardNo	证件号码	
TThostFtdcBoolType	IsActive	是否活跃	
TThostFtdcTelephoneType	Telephone	联系电话	
TThostFtdcAddressType	Address	通讯地址	
TThostFtdcDateType	OpenDate	开户日期	
TThostFtdcMobileType	Mobile	手机	
TThostFtdcInvestorIDType	CommModelID	手续费率模板代码	
TThostFtdcInvestorIDType	MarginModelID	保证金率模板代码	

pRsplInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.19 OnRspQryTradingCode

功能：

请求查询交易编码响应

函数原型：

```
virtual void OnRspQryTradingCode(CThostFtdcTradingCodeField
*pTradingCode, CThostFtdcRspInfoField *pRsplInfo, int nRequestId, bool bIsLast);
```

参数：

pTradingCode: 请求查询交易编码响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcBoolType	IsActive	是否活跃	
TThostFtdcClientIDTypeType	ClientIDType	交易编码类型	

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.20 OnRspQryInstrumentMarginRate

功能：

请求查询合约保证金率响应

函数原型：

virtual void

OnRspQryInstrumentMarginRate(CThostFtdcInstrumentMarginRateField

***pInstrumentMarginRate, CThostFtdcRsplInfoField *pRsplInfo, int nRequestID,**
bool bIsLast);

参数：

pInstrumentMarginRate: 请求查询合约保证金率响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInvestorRangeType	InvestorRange	投资者范围	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcRatioType	LongMarginRatioByMoney	多头保证金率	
TThostFtdcMoneyType	LongMarginRatioByVolume	多头保证金费	
TThostFtdcRatioType	ShortMarginRatioByMoney	空头保证金率	
TThostFtdcMoneyType	ShortMarginRatioByVolume	空头保证金费	
TThostFtdcBoolType	IsRelative	是否相对交易所收取	

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.21 OnRspQryInstrumentCommissionRate

功能：

请求查询合约手续费率响应

函数原型：

```
virtual void
OnRspQryInstrumentCommissionRate(CThostFtdcInstrumentCommissionRateField
    *pInstrumentCommissionRate, CThostFtdcRspInfoField *pRspInfo, int
nRequestID, bool bIsLast);
```

参数：

pInstrumentCommissionRate: 请求查询合约手续费率响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInvestorRangeType	InvestorRange	投资者范围	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcRatioType	OpenRatioByMoney	开仓手续费率	
TThostFtdcRatioType	OpenRatioByVolume	开仓手续费	
TThostFtdcRatioType	CloseRatioByMoney	平仓手续费率	
TThostFtdcRatioType	CloseRatioByVolume	平仓手续费	
TThostFtdcRatioType	CloseTodayRatioByMoney	平今手续费率	
TThostFtdcRatioType	CloseTodayRatioByVolume	平今手续费	

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.22 OnRspQryExchange

功能：

请求查询交易所响应

函数原型：

```
virtual void OnRspQryExchange(CThostFtdcExchangeField *pExchange,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pExchange: 请求查询交易所响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcExchangeIDType	ExchangeID	交易所代码	

TThostFtdcExchangeNameType	ExchangeName	交易所名称	
TThostFtdcExchangePropertyType	ExchangeProperty	交易所属性	

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息ErrorMsg;

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.23 OnRspQryInstrument

功能:

请求查询合约响应

函数原型:

```
virtual void OnRspQryInstrument(CThostFtdcInstrumentField *pInstrument,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数:

pInstrument: 请求查询合约响应信息, 对应的结构各字段如下表:

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcInstrumentNameType	InstrumentName	合约名称	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcInstrumentIDType	ProductID	产品代码	
TThostFtdcProductClassType	ProductClass	产品类型	
TThostFtdcYearType	DeliveryYear	交割年份	
TThostFtdcMonthType	DeliveryMonth	交割月	
TThostFtdcVolumeType	MaxMarketOrderVolume	市价单最大下单量	
TThostFtdcVolumeType	MinMarketOrderVolume	市价单最小下单量	
TThostFtdcVolumeType	MaxLimitOrderVolume	限价单最大下单量	
TThostFtdcVolumeType	MinLimitOrderVolume	限价单最小下单量	
TThostFtdcVolumeMultipleType	VolumeMultiple	合约数量乘数	
TThostFtdcPriceType	PriceTick	最小变动价位	
TThostFtdcDateType	CreateDate	创建日	
TThostFtdcDateType	OpenDate	上市日	
TThostFtdcDateType	ExpireDate	到期日	
TThostFtdcDateType	StartDelivDate	开始交割日	
TThostFtdcDateType	EndDelivDate	结束交割日	
TThostFtdcInstLifePhaseType	InstLifePhase	合约生命周期状态	
TThostFtdcBoolType	IsTrading	当前是否交易	
TThostFtdcPositionTypeType	PositionType	持仓类型	
TThostFtdcPositionDateTypeType	PositionDateType	持仓日期类型	
TThostFtdcRatioType	LongMarginRatio	多头保证金率	

TThostFtdcRatioType	ShortMarginRatio	空头保证金率	
TThostFtdcMaxMarginSideAlgorithmType	MaxMarginSideAlgorithm	是否使用大额单边保证金算法	
TThostFtdcInstrumentIDType	UnderlyingInstrID	基础商品代码	
TThostFtdcPriceType	StrikePrice	执行价	
TThostFtdcOptionsTypeType	OptionsType	期权类型	
TThostFtdcUnderlyingMultipleType	UnderlyingMultiple	合约基础商品乘数	
TThostFtdcCombinationTypeType	CombinationType	组合类型	

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.24 OnRspQryDepthMarketData

功能：

请求查询行情响应

函数原型：

```
virtual void OnRspQryDepthMarketData(CThostFtdcDepthMarketDataField
*pDepthMarketData, CThostFtdcRsplInfoField *pRsplInfo, int nRequestID, bool
bIsLast);
```

参数：

pDepthMarketData：请求查询行情响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcPriceType	LastPrice	最新价	
TThostFtdcPriceType	PreSettlementPrice	上次结算价	
TThostFtdcPriceType	PreClosePrice	昨收盘	
TThostFtdcLargeVolumeType	PreOpenInterest	昨持仓量	
TThostFtdcPriceType	OpenPrice	今开盘	
TThostFtdcPriceType	HighestPrice	最高价	
TThostFtdcPriceType	LowestPrice	最低价	
TThostFtdcVolumeType	Volume	数量	
TThostFtdcMoneyType	Turnover	成交金额	
TThostFtdcLargeVolumeType	OpenInterest	持仓量	
TThostFtdcPriceType	ClosePrice	今收盘	
TThostFtdcPriceType	SettlementPrice	本次结算价	
TThostFtdcPriceType	UpperLimitPrice	涨停板价	

TThostFtdcPriceType	LowerLimitPrice	跌停板价	
TThostFtdcRatioType	PreDelta	昨虚实度	
TThostFtdcRatioType	CurrDelta	今虚实度	
TThostFtdcTimeType	UpdateTime	最后修改时间	
TThostFtdcMillisecondType	UpdateMillisecond	最后修改毫秒	
TThostFtdcPriceType	BidPrice1	申买价一	
TThostFtdcVolumeType	BidVolume1	申买量一	
TThostFtdcPriceType	AskPrice1	申卖价一	
TThostFtdcVolumeType	AskVolume1	申卖量一	
TThostFtdcPriceType	BidPrice2	申买价二	
TThostFtdcVolumeType	BidVolume2	申买量二	
TThostFtdcPriceType	AskPrice2	申卖价二	
TThostFtdcVolumeType	AskVolume2	申卖量二	
TThostFtdcPriceType	BidPrice3	申买价三	
TThostFtdcVolumeType	BidVolume3	申买量三	
TThostFtdcPriceType	AskPrice3	申卖价三	
TThostFtdcVolumeType	AskVolume3	申卖量三	
TThostFtdcPriceType	BidPrice4	申买价四	
TThostFtdcVolumeType	BidVolume4	申买量四	
TThostFtdcPriceType	AskPrice4	申卖价四	
TThostFtdcVolumeType	AskVolume4	申卖量四	
TThostFtdcPriceType	BidPrice5	申买价五	
TThostFtdcVolumeType	BidVolume5	申买量五	
TThostFtdcPriceType	AskPrice5	申卖价五	
TThostFtdcVolumeType	AskVolume5	申卖量五	
TThostFtdcPriceType	AveragePrice	当日均价	
TThostFtdcDateType	ActionDay	业务日期	
TThostFtdcPriceType	BandingUpperPrice	上带价	V1.5.7 新增
TThostFtdcPriceType	BandingLowerPrice	下带价	V1.5.7 新增

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.25 OnRspQryInstrumentStatus

功能：

请求查询合约状态响应

函数原型：

```
virtual void OnRspQryInstrumentStatus(CThostFtdcInstrumentStatusField
*pInstrumentStatus, CThostFtdcRsplInfoField *pRsplInfo, int nRequestId, bool
bIsLast);
```

参数：

pInstrumentStatus: 请求查询合约状态响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcSettlementGroupIDType	SettlementGroupID	结算组代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInstrumentStatusType	InstrumentStatus	合约交易状态	
TThostFtdcTradingSegmentSNType	TradingSegmentSN	交易阶段编号	
TThostFtdcTimeType	EnterTime	进入时间	
TThostFtdcInstStatusEnterReasonType	EnterReason	进入本状态原因	

pRsplInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.26 OnRspQryInvestorPositionDetail

功能：

请求查询投资者持仓明细响应

- 本接口从后台 V1.5.8 版本开始根据交易后台有条件提供支持：
 - V1.5.8 支持大商所持仓明细查询；
 - V1.7.0 支持广期所持仓明细查询；

函数原型：

```
virtual void OnRspQryInvestorPositionDetail
(CThostFtdcInvestorPositionDetailField
 *pInvestorPositionDetail,
 CThostFtdcRsplInfoField *pRsplInfo, int nRequestId, bool bIsLast);
```

参数：

pInvestorPositionDetail: 请求查询投资者持仓明细响应信息，对应的结构各字段如下

表：

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	

TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcDirectionType	Direction	买卖	
TThostFtdcDateType	OpenDate	开仓日期	
TThostFtdcTradeIDType	TradeID	成交编号	
TThostFtdcVolumeType	Volume	数量	
TThostFtdcPriceType	OpenPrice	开仓价	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcTradeTypeType	TradeType	成交类型	
TThostFtdcInstrumentIDType	ComblnstrumentID	组合合约代码	
TThostFtdcExchangeIDType	ExchangelD	交易所代码	
TThostFtdcMoneyType	CloseProfitByDate	逐日盯市平仓盈亏	
TThostFtdcMoneyType	CloseProfitByTrade	逐笔对冲平仓盈亏	
TThostFtdcMoneyType	PositionProfitByDate	逐日盯市持仓盈亏	
TThostFtdcMoneyType	PositionProfitByTrade	逐笔对冲持仓盈亏	
TThostFtdcMoneyType	Margin	投资者保证金	
TThostFtdcMoneyType	ExchMargin	交易所保证金	
TThostFtdcRatioType	MarginRateByMoney	保证金率	
TThostFtdcRatioType	MarginRateByVolume	保证金率(按手数)	
TThostFtdcPriceType	LastSettlementPrice	昨结算价	
TThostFtdcPriceType	SettlementPrice	结算价	
TThostFtdcVolumeType	CloseVolume	平仓量	
TThostFtdcMoneyType	CloseAmount	平仓金额	

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.27 OnRspQryInvestorPositionCombineDetail

功能：

请求查询投资者持仓明细响应

- 本接口从 V1.5.8 版本开始不再提供支持；

函数原型：

```
virtual void OnRspQryInvestorPositionCombineDetail
(CThostFtdcInvestorPositionCombineDetailField
*pInvestorPositionCombineDetail, CThostFtdcRsplnfoField *pRspInfo, int
nRequestID, bool bIsLast);
```

参数：

pInvestorPositionCombineDetail: 请求查询投资者持仓明细响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcDateType	OpenDate	开仓日期	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcTradeIDType	ComTradeID	组合编号	
TThostFtdcTradeIDType	TradeID	撮合编号	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcDirectionType	Direction	买卖	
TThostFtdcVolumeType	TotalAmt	持仓量	
TThostFtdcMoneyType	Margin	投资者保证金	
TThostFtdcMoneyType	ExchMargin	交易所保证金	
TThostFtdcRatioType	MarginRateByMoney	保证金率	
TThostFtdcRatioType	MarginRateByVolume	保证金率(按手数)	
TThostFtdcLegIDType	LegID	单腿编号	
TThostFtdcLegMultipleType	LegMultiple	单腿乘数	
TThostFtdcInstrumentIDType	ComblnstrumentID	组合持仓合约编码	
TThostFtdcTradeGroupIDType	TradeGroupID	成交组号	

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.28 OnRspQryExchangeMarginRate

功能：

请求查询交易所保证金率响应

函数原型：

virtual void

OnRspQryExchangeMarginRate(CThostFtdcExchangeMarginRateField *pExchangeMarginRate, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

参数：

pExchangeMarginRate: 请求查询交易所保证金率响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
------	------	------	----

TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcRatioType	LongMarginRatioByMoney	多头保证金率	
TThostFtdcMoneyType	LongMarginRatioByVolume	多头保证金费	
TThostFtdcRatioType	ShortMarginRatioByMoney	空头保证金率	
TThostFtdcMoneyType	ShortMarginRatioByVolume	空头保证金费	

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.29 OnRspQryExchangeMarginRateAdjust

功能：

请求查询交易所调整保证金率响应

函数原型：

```
virtual void OnRspQryExchangeMarginRateAdjust
(CThostFtdcExchangeMarginRateAdjustField *pExchangeMarginRateAdjust,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pExchangeMarginRateAdjust: 请求查询交易所调整保证金率响应信息，对应的结构

各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcRatioType	LongMarginRatioByMoney	跟随交易所投资者多头保证金率	
TThostFtdcMoneyType	LongMarginRatioByVolume	跟随交易所投资者多头保证金费	
TThostFtdcRatioType	ShortMarginRatioByMoney	跟随交易所投资者空头保证金率	
TThostFtdcMoneyType	ShortMarginRatioByVolume	跟随交易所投资者空头保证金费	
TThostFtdcRatioType	ExchLongMarginRatioByMoney	交易所多头保证金率	
TThostFtdcMoneyType	ExchLongMarginRatioByVolume	交易所多头保证金费	
TThostFtdcRatioType	ExchShortMarginRatioByMoney	交易所空头保证金率	
TThostFtdcMoneyType	ExchShortMarginRatioByVolume	交易所空头保证金费	
TThostFtdcRatioType	NoLongMarginRatioByMoney	不跟随交易所投资者多头保证金率	

TThostFtdcMoneyType	NoLongMarginRatioByVolume	不跟随交易所投资者多头保证金费	
TThostFtdcRatioType	NoShortMarginRatioByMoney	不跟随交易所投资者空头保证金率	
TThostFtdcMoneyType	NoShortMarginRatioByVolume	不跟随交易所投资者空头保证金费	

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息ErrorMsg;

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.30 OnRspQryOptionInstrTradeCost

功能:

请求查询期权交易成本响应

函数原型:

```
virtual void OnRspQryOptionInstrTradeCost
(CThostFtdcOptionInstrTradeCostField *pOptionInstrTradeCost,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数:

pOptionInstrTradeCost: 请求查询期权交易成本响应信息, 对应的结构各字段如下表:

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcMoneyType	FixedMargin	期权合约保证金不变部分	
TThostFtdcMoneyType	MiniMargin	期权合约最小保证金	
TThostFtdcMoneyType	Royalty	期权合约权利金	
TThostFtdcMoneyType	ExchFixedMargin	交易所期权合约保证金不变部分	
TThostFtdcMoneyType	ExchMiniMargin	交易所期权合约最小保证金	

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息ErrorMsg;

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.31 OnRspQryOptionInstrCommRate

功能:

请求查询期权合约手续费响应

函数原型：

```
virtual void OnRspQryOptionInstrCommRate
(CThostFtdcOptionInstrCommRateField *pOptionInstrCommRate,
CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast);
```

参数：

pOptionInstrCommRate: 请求查询期权合约手续费响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInvestorRangeType	InvestorRange	投资者范围	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcRatioType	OpenRatioByMoney	开仓手续费率	
TThostFtdcRatioType	OpenRatioByVolume	开仓手续费	
TThostFtdcRatioType	CloseRatioByMoney	平仓手续费率	
TThostFtdcRatioType	CloseRatioByVolume	平仓手续费	
TThostFtdcRatioType	CloseTodayRatioByMoney	平今手续费率	
TThostFtdcRatioType	CloseTodayRatioByVolume	平今手续费	
TThostFtdcRatioType	StrikeRatioByMoney	执行手续费率	
TThostFtdcRatioType	StrikeRatioByVolume	执行手续费	

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 Errormsg；

nRequestId: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.32 OnRspQryExecOrder

功能：

请求查询执行宣告响应

函数原型：

```
virtual void OnRspQryExecOrder(CThostFtdcExecOrderField *pExecOrder,
CThostFtdcRspInfoField *pRspInfo, int nRequestId, bool bIsLast);
```

参数：

pExecOrder: 请求查询执行宣告响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderRefType	ExecOrderRef	执行宣告引用	

TThostFtdcUserIDType	UserID	用户代码
TThostFtdcVolumeType	Volume	数量
TThostFtdcRequestIDType	RequestID	请求编号
TThostFtdcBusinessUnitType	BusinessUnit	业务单元
TThostFtdcOffsetFlagType	OffsetFlag	开平标志
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志
TThostFtdcActionTypeType	ActionType	执行类型
TThostFtdcPosiDirectionType	PosiDirection	保留头寸申请的持仓方向
TThostFtdcExecOrderPositionFlagType	ReservePositionFlag	期权行权后是否保留期货头寸的标记
TThostFtdcExecOrderCloseFlagType	CloseFlag	期权行权后生成的头寸是否自动平仓
TThostFtdcOrderLocalIDType	ExecOrderLocalID	本地执行宣告编号
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcParticipantIDType	ParticipantID	会员代码
TThostFtdcClientIDType	ClientID	客户代码
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码
TThostFtdcTraderIDType	TraderID	交易所交易员代码
TThostFtdcInstallIDType	InstallID	安装编号
TThostFtdcOrderSubmitStatusType	OrderSubmitStatus	执行宣告提交状态
TThostFtdcSequenceNoType	NotifySequence	报单提示序号
TThostFtdcDateType	TradingDay	交易日
TThostFtdcSettlementIDType	SettlementID	结算编号
TThostFtdcExecOrderSysIDType	ExecOrderSysID	执行宣告编号
TThostFtdcDateType	InsertDate	报单日期
TThostFtdcTimeType	InsertTime	插入时间
TThostFtdcTimeType	CancelTime	撤销时间
TThostFtdcExecResultType	ExecResult	执行结果
TThostFtdcParticipantIDType	ClearingPartID	结算会员编号
TThostFtdcSequenceNoType	SequenceNo	序号
TThostFtdcFrontIDType	FrontID	前置编号
TThostFtdcSessionIDType	SessionID	会话编号
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息
TThostFtdcErrorMsgType	StatusMsg	状态信息
TThostFtdcUserIDType	ActiveUserID	操作用户代码
TThostFtdcSequenceNoType	BrokerExecOrderSeq	经纪公司报单编号
TThostFtdcBranchIDType	BranchID	营业部编号
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码
TThostFtdcAccountIDType	AccountID	资金账号
TThostFtdcCurrencyIDType	CurrencyID	币种代码
TThostFtdcIPAddressType	IPAddress	IP 地址
TThostFtdcMacAddressType	MacAddress	Mac 地址

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.33 OnRspQryForQuote

功能：

请求查询询价响应

函数原型：

```
virtual void OnRspQryForQuote(CThostFtdcForQuoteField *pForQuote,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pForQuote： 请求查询询价响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderRefType	ForQuoteRef	询价引用	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcOrderLocalIDType	ForQuoteLocalID	本地询价编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcExchangeInstIDType	ExchangelInstID	合约在交易所的代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcDateType	InsertDate	报单日期	
TThostFtdcTimeType	InsertTime	插入时间	
TThostFtdcForQuoteStatusType	ForQuoteStatus	询价状态	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcUserIDType	ActiveUserID	操作用户代码	
TThostFtdcSequenceNoType	BrokerForQuoSeq	经纪公司询价编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元，常用作席位号	V1.5.8 增加

pRspInfo： 响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID： 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.34 OnRspQryQuote

功能:

请求查询报价响应

函数原型:

```
virtual void OnRspQryQuote(CThostFtdcQuoteField *pQuote,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数:

pQuote: 请求查询报价响应信息, 对应的结构各字段如下表:

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderRefType	QuoteRef	报价引用	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcPriceType	AskPrice	卖价格	
TThostFtdcPriceType	BidPrice	买价格	
TThostFtdcVolumeType	AskVolume	卖数量	
TThostFtdcVolumeType	BidVolume	买数量	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcOffsetFlagType	AskOffsetFlag	卖开平标志	
TThostFtdcOffsetFlagType	BidOffsetFlag	买开平标志	
TThostFtdcHedgeFlagType	AskHedgeFlag	卖投机套保标志	
TThostFtdcHedgeFlagType	BidHedgeFlag	买投机套保标志	
TThostFtdcOrderLocalIDType	QuoteLocalID	本地报价编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcSequenceNoType	NotifySequence	报价提示序号	
TThostFtdcOrderSubmitStatusType	OrderSubmitStatus	报价提交状态	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcOrderSysIDType	QuoteSysID	报价编号	
TThostFtdcDateType	InsertDate	报单日期	
TThostFtdcTimeType	InsertTime	插入时间	

TThostFtdcTimeType	CancelTime	撤销时间	
TThostFtdcOrderStatusType	QuoteStatus	报价状态	
TThostFtdcParticipantIDType	ClearingPartID	结算会员编号	
TThostFtdcSequenceNoType	SequenceNo	序号	
TThostFtdcOrderSysIDType	AskOrderSysID	卖方报单编号	
TThostFtdcOrderSysIDType	BidOrderSysID	买方报单编号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcUserIDType	ActiveUserID	操作用户代码	
TThostFtdcSequenceNoType	BrokerQuoteSeq	经纪公司报价编号	
TThostFtdcOrderRefType	AskOrderRef	衍生卖报单引用	
TThostFtdcOrderRefType	BidOrderRef	衍生买报单引用	
TThostFtdcOrderSysIDType	ForQuoteSysID	应价编号	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcAccountIDType	AccountID	资金账号	
TThostFtdcCurrencyIDType	CurrencyID	币种代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcOrderSysIDType	ReplaceSysID	顶单编号	
TThostFtdcCustomOrderRefType	CustomQuoteRef	客户自定义报价编号	
TThostFtdcTimeConditionType	TimeCondition	有效期类型	V1.6.8 增加字段

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.35 OnRspQryOptionSelfClose

功能：

请求查询期权自对冲响应

函数原型：

```
virtual void OnRspQryOptionSelfClose(CThostFtdcOptionSelfCloseField
*pOptionSelfClose, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
bIsLast);
```

参数：

pOptionSelfClose: 请求查询期权自对冲响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderRefType	OptionSelfCloseRef	期权自对冲引用	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcVolumeType	Volume	数量	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcOptSelfCloseFlagType	OptSelfCloseFlag	期权行权的头寸是否自对冲	
TThostFtdcOrderLocalIDType	OptionSelfCloseLocalID	本地期权自对冲编号	
TThostFtdcExchangeIDType	ExchangelD	交易所代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcExchangeInstIDType	ExchangelInstID	合约在交易所的代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderSubmitStatusType	OrderSubmitStatus	期权自对冲提交状态	
TThostFtdcSequenceNoType	NotifySequence	报单提示序号	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcOrderSysIDType	OptionSelfCloseSysID	期权自对冲编号	
TThostFtdcDateType	InsertDate	报单日期	
TThostFtdcTimeType	InsertTime	插入时间	
TThostFtdcTimeType	CancelTime	撤销时间	
TThostFtdcExecResultType	ExecResult	自对冲结果	
TThostFtdcParticipantIDType	ClearingPartID	结算会员编号	
TThostFtdcSequenceNoType	SequenceNo	序号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcUserIDType	ActiveUserID	操作用户代码	
TThostFtdcSequenceNoType	BrokerOptionSelfCloseSeq	经纪公司报单编号	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcAccountIDType	AccountID	资金账号	
TThostFtdcCurrencyIDType	CurrencyID	币种代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.36 OnRtnOrder

功能:

报单通知

- 对于做市商, 报价录入成功后, 返回相应的 OnRtnOrder;

函数原型:

virtual void OnRtnOrder(CThostFtdcOrderField *pOrder);

参数:

pOrder: 报单通知信息, 对应的结构各字段参见《OnRspQryOrder》章节。

- 在处理报单回报信息时, 需要确定原始报单, 有三个路径可以标识唯一的报单
 - FrontID (Mini 后台前置编号) + SessionID (会话编号) + OrderRef (终端请求时自己填写的对应报单引用编号);
 - 报单时维护本地报单;
 - 建议本地维护 OrderRef 字段, 确保同一会话中的 OrderRef 唯一;
 - ExchangeID + TraderID + OrderLocalID (每日单调递增)
 - CTP Mini 内部维护的一组序号, 终端无法改变;
 - ExchangeID + OrderSysID (报单报入到交易所时交易所给编的唯一编号)
 - 这一组字段也可以用于撤单时指定唯一一笔订单;
- 以上三个路径根据交易的不同阶段稍有区别, 终端开发商可以依照自己开发程序需要去甄别。
- 当前报单状态, Mini API 提供了三个字段分别表示不同含义, 提供终端判断
 - OrderStatus 是枚举类型, 表示报单状态

OrderStatus 报单状态类型	状态说明
#define THOST_FTDC_OST_AllTraded '0'	全部成交, 交易所返回的最后状态;
#define THOST_FTDC_OST_PartTradedQueueing '1'	部分成交还在队列中, 交易所返回的中间状态;
#define THOST_FTDC_OST_PartTradedNotQueueing '2'	部分成交不在队列中
#define THOST_FTDC_OST_NoTradeQueueing '3'	未成交还在队列中, 表示报单已经报送交易所, 处于挂单为未成交状态;
#define THOST_FTDC_OST_NoTradeNotQueueing '4'	未成交不在队列中
#define THOST_FTDC_OST_Canceled '5'	撤单; 在主动撤单或被动撤单时返回的状态;
#define THOST_FTDC_OST_Unknown 'a'	未知, 表示 Mini 收到报单并通过风控检查, 交易所报单状态未知
#define THOST_FTDC_OST_NotTouched 'b'	尚未触发
#define THOST_FTDC_OST_Touched 'c'	已触发

- 如果报单被交易所拒绝, 此处就返回被动撤单;

- OrderStatus 为 THOST_FTDC_OST_Canceled (撤单), 可以结合 OrderSysID 是否为空来判断是撤单还是被动撤单; StatusMsg 字段有具体的拒单原因, 是由交易所直接返回的描述信息;
- ◆ 常见的交易所拒单原因如下:
 - 非交易时间段报单
 - 价格超出涨跌停板
 - 限价单价格不对, 不是最小变动价的整数倍或价格非最小单位的倍数;
 - 进入交割期合约报单数量交易所有要求, 不合法数量
- StatusMsg: 当被动撤单时会包含交易所的拒单信息
- 当前挂单及成交量
 - VolumeTotalOriginal 数量, 表示原始报单数量
 - VolumeTraded 今成交量, 该笔原始报单目前的累计成交量
 - VolumeTotal 剩余数量, 该笔报单目前剩余未成交数量
 - 本处的数量因为是中间状态, 因微小时间差回报异步导致并非 100% 可靠, 建议以成交回报 OnRtnTrade 中对应返回数量为准判断报单是否成交, 以及成交量和价格; 同时 Mini 交易系统后台也是以成交回报为准更新持仓状态的, 建议终端开发与后台同标准处理。

3.2.37 OnRtnTrade

功能:

成交通知

函数原型:

`virtual void OnRtnTrade(CThostFtdcTradeField *pTrade);`

参数:

pTrade: 成交通知信息, 相关字段信息参考《[OnRspQryTrade](#)》章节

- 唯一确定一笔成交的标志是组合键值: ExchangeID 交易所代码 + TradeID 交易所对每一笔成交的编号 (交易所唯一) + Direction 买卖方向,
 - 除郑商所外的交易所对成交时互为对手方的两笔成交编号是一样的, 所以如果对全部交易所适用还需要加上 Direction 买卖方向字段;
- 如果有成交的话对应的报单状态是变化的, 所以成交回报之前会先有报单回报;

3.2.38 OnErrRtnOrderAction

功能:

报单操作错误回报

- 常见的撤单失败原因如下：

- 已成交或已撤单，不能再撤
- 无权限
- 原始报单找不到
- 报单全部成交

函数原型：

```
virtual void OnErrRtnOrderAction(CThostFtdcOrderActionField
*pOrderAction, CThostFtdcRspInfoField *pRspInfo);
```

参数：

pOrderAction 报单操作错误回报信息；相关字段信息如下表所示：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcOrderActionRefType	OrderActionRef	报单操作引用	
TThostFtdcOrderRefType	OrderRef	报单引用	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcOrderSysIDType	OrderSysID	报单编号	
TThostFtdcActionFlagType	ActionFlag	操作标志	
TThostFtdcPriceType	LimitPrice	价格	
TThostFtdcVolumeType	VolumeChange	数量变化	
TThostFtdcDateType	ActionDate	操作日期	
TThostFtdcTimeType	ActionTime	操作时间	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderLocalIDType	OrderLocalID	本地报单编号	
TThostFtdcOrderLocalIDType	ActionLocalID	操作本地编号	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcOrderActionStatusType	OrderActionStatus	报单操作状态	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	

TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.39 OnRtnInstrumentStatus

功能：

合约交易状态通知

- 用户在登录后，无需订阅便能收到全量的合约状态变化，该变化在盘中实时推送；
- 期货合约状态推送到产品级别，期权也是推送到产品级别；

函数原型：

```
virtual void OnRtnInstrumentStatus(CThostFtdcInstrumentStatusField *pInstrumentStatus);
```

参数：

pInstrumentStatus：合约交易状态通知信息，相关字段信息参考《[OnRspQryInstrumentStatus](#)》章节。

3.2.40 OnRtnExecOrder

功能：

执行宣告通知

函数原型：

```
virtual void OnRtnExecOrder(CThostFtdcExecOrderField *pExecOrder);
```

参数：

pExecOrder：执行宣告通知信息，对应的结构各字段参见《[OnRspQryExecOrder](#)》章节

3.2.41 OnErrRtnExecOrderAction

功能：

执行宣告操作错误回报

函数原型：

```
virtual void OnErrRtnExecOrderAction(CThostFtdcExecOrderActionField
```

*pExecOrderAction, CThostFtdcRspInfoField *pRspInfo);

参数：

pExecOrderAction: 执行宣告操作错误回报信息，相关字段参考下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcOrderActionRefType	ExecOrderActionRef	执行宣告操作引用	
TThostFtdcOrderRefType	ExecOrderRef	执行宣告引用	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcExecOrderSysIDType	ExecOrderSysID	执行宣告操作编号	
TThostFtdcActionFlagType	ActionFlag	操作标志	
TThostFtdcDateType	ActionDate	操作日期	
TThostFtdcTimeType	ActionTime	操作时间	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderLocalIDType	ExecOrderLocalID	本地执行宣告编号	
TThostFtdcOrderLocalIDType	ActionLocalID	操作本地编号	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcOrderActionStatusType	OrderActionStatus	报单操作状态	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcActionTypeType	ActionType	执行类型	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

3.2.42 OnErrRtnForQuoteInsert

功能：

询价录入错误回报

- 被交易所拒绝的询价通过此接口返回；可以通过 Errord 和 ErrorMsg 查看错误信息

函数原型：

virtual void OnErrRtnForQuoteInsert(CThostFtdcInputForQuoteField

*pInputForQuote, CThostFtdcRspInfoField *pRspInfo);

参数：

pInputForQuote：询价录入错误回报信息，对应的结构各字段参见《ReqForQuoteInsert》章节。

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 Errormsg；

3.2.43 OnRtnQuote

功能：

报价通知

- 当报价录入成功后，Mini 交易系统转发交易所该报价回报；

函数原型：

virtual void OnRtnQuote(CThostFtdcQuoteField *pQuote);

参数：

pQuote：报价通知信息，相关结构字段信息参见《OnRspQryQuote》章节

3.2.44 OnErrRtnQuoteAction

功能：

报价操作错误回报

函数原型：

virtual void OnErrRtnQuoteAction(CThostFtdcQuoteActionField

***pQuoteAction, CThostFtdcRspInfoField *pRspInfo);**

参数：

pQuoteAction：报价操作错误回报信息，相关字段信息参考下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcOrderActionRefType	QuoteActionRef	报价操作引用	
TThostFtdcOrderRefType	QuoteRef	报价引用	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcExchangeIDType	ExchangelD	交易所代码	
TThostFtdcOrderSysIDType	QuoteSysID	报价操作编号	
TThostFtdcActionFlagType	ActionFlag	操作标志	
TThostFtdcDateType	ActionDate	操作日期	

TThostFtdcTimeType	ActionTime	操作时间	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderLocalIDType	QuoteLocalID	本地报价编号	
TThostFtdcOrderLocalIDType	ActionLocalID	操作本地编号	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcOrderActionStatusType	OrderActionStatus	报单操作状态	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

3.2.45 OnRtnForQuoteRsp

功能：

询价通知

函数原型：

```
virtual void OnRtnForQuoteRsp(CThostFtdcForQuoteRspField
*pForQuoteRsp);
```

参数：

pForQuoteRsp： 询价通知信息，相关字段信息参考下表：

字段类型	字段名称	字段说明	备注
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderSysIDType	ForQuoteSysID	询价编号	
TThostFtdcTimeType	ForQuoteTime	询价时间	
TThostFtdcDateType	ActionDay	业务日期	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	

3.2.46 OnRtnOptionSelfClose

功能：

期权自对冲通知

函数原型：

```
virtual void OnRtnOptionSelfClose(CThostFtdcOptionSelfCloseField
*pOptionSelfClose);
```

参数：

pOptionSelfClose：期权自对冲通知信息，相关结构字段信息参考《[OnRspQryOptionSelfClose](#)》章节。

3.2.47 OnErrRtnOptionSelfCloseAction

功能：

期权自对冲操作错误回报

函数原型：

```
virtual void OnErrRtnOptionSelfCloseAction
(CThostFtdcOptionSelfCloseActionField
* pOptionSelfCloseAction,
CThostFtdcRspInfoField * pRspInfo);
```

参数：

pOptionSelfCloseAction: 期权自对冲操作错误回报信息，

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcOrderActionRefType	OptionSelfCloseActionRef	期权自对冲操作引用	
TThostFtdcOrderRefType	OptionSelfCloseRef	期权自对冲引用	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcOrderSysIDType	OptionSelfCloseSysID	期权自对冲操作编号	
TThostFtdcActionFlagType	ActionFlag	操作标志	
TThostFtdcDateType	ActionDate	操作日期	
TThostFtdcTimeType	ActionTime	操作时间	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderLocalIDType	OptionSelfCloseLocalID	本地期权自对冲编号	
TThostFtdcOrderLocalIDType	ActionLocalID	操作本地编号	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcOrderActionStatusType	OrderActionStatus	报单操作状态	
TThostFtdcUserIDType	UserID	用户代码	

TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcBranchIDType	BranchID	营业部编号	
TThostFtdcInvestUnitIDType	InvestUnitID	投资单元代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

3.2.48 OnRspQryInstrumentOrderCommRate

功能：

请求查询申报费响应

函数原型：

```
virtual void OnRspQryInstrumentOrderCommRate
(CThostFtdcInstrumentOrderCommRateField *pInstrumentOrderCommRate,
CThostFtdcRsplInfoField *pRsplInfo, int nRequestID, bool bIsLast);
```

参数：

pInstrumentOrderCommRate: 请求查询申报费响应信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInvestorRangeType	InvestorRange	投资者范围	
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcRatioType	OrderCommByVolume	报单手续费	
TThostFtdcRatioType	OrderActionCommByVolume	撤单手续费	

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.49 OnRspCombActionInsert

功能：

申请组合录入请求响应

函数原型：

```
virtual void OnRspCombActionInsert(CThostFtdcInputCombActionField
*pInputCombAction, CThostFtdcRsplInfoField *pRsplInfo, int nRequestID, bool bIsLast)
```

{};

参数：pInputCombAction, 参见章节《[3.1.47 ReqCombActionInsert](#)》

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.50 OnRtnCombAction

功能：

申请组合通知

函数原型：

virtual void OnRtnCombAction(CThostFtdcCombActionField *pCombAction) {};

参数：

pCombAction 申请组合单腿汇总查询：

字段类型	字段名称	含义	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcOrderRefType	CombActionRef	组合引用	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcDirectionType	Direction	买卖方向	
TThostFtdcVolumeType	Volume	数量	
TThostFtdcCombDirectionType	CombDirection	组合指令方向	
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元--new	
TThostFtdcOrderLocalIDType	ActionLocalID	本地申请组合编号	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderActionStatusType	ActionStatus	组合状态	
TThostFtdcSequenceNoType	NotifySequence	报单提示序号	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	

TThostFtdcSequenceNoType	SequenceNo	序号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcProductInfoType	UserProductInfo	用户端产品信息	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcTradeIDType	ComTradeID	组合编号--new	

3.2.51 OnRspQryCombAction

功能：

请求查询申请组合响应

函数原型：

```
virtual void OnRspQryCombAction(CThostFtdcCombActionField *pCombAction,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

参数：

pCombAction 申请组合单腿汇总查询，参见《3.2.50 OnRtnCombAction》章节；

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.52 OnRspQryCombiInstrument

功能：

请求查询申请组合合约响应

函数原型：

```
virtual void OnRspQryCombiInstrument(CThostFtdcCombiInstrumentField
*pCombiInstrument, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
{};
```

参数：

pCombiInstrument 申请组合合约信息：

字段类型	字段名称	含义	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	

TThostFtdcDceCombinationType	CombinationType	组合类型	
TThostFtdcInstrumentID	ProductID	产品代码	
TThostFtdcRatio	Xparameter	保证金优惠比例	

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.53 OnRspQryInvestorPositionForComb

功能：

请求查询组合单腿汇总表响应

函数原型：

```
virtual void OnRspQryInvestorPositionForComb
(CThostFtdcInvestorPositionForCombField *pCombAction, CThostFtdcRspInfoField
*pRspInfo, int nRequestID, bool bIsLast) {};
```

参数：

pCombAction 组合单腿汇总：

字段类型	字段名称	含义	备注
TThostFtdcExchangeID	ExchangeID	交易所代码	
TThostFtdcBrokerID	BrokerID	经纪公司代码	
TThostFtdcInvestorID	InvestorID	投资者代码	
TThostFtdcInstrumentID	LegInstrumentID	合约代码	
TThostFtdcHedgeFlag	LegHedgeFlag	投机套保标志	
TThostFtdcDirection	LegDirection	买卖方向	
TThostFtdcVolume	TotalAmt	数量	
TThostFtdcLegID	LegID	LegID	
TThostFtdcTradeGroupID	TradeGroupID	组合优先级	大商所有效，相同组合优先级的腿为一对组合
TThostFtdcInstrumentID	CombInstrumentID	组合合约代码	
TThostFtdcHedgeFlag	CombHedgeFlag	组合投机套保标志	
TThostFtdcCombinationType	CombinationType	组合类型	
TThostFtdcTradeID	CombTradeID	组合成交编号	郑商所有效，编号相同的为一对组合
TThostFtdcDate	OpenDate	成交日期	郑商所有效
TThostFtdcTradeID	TradeID	成交编号	郑商所有效

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.54 OnRspQryForQuoteParam

功能:

请求查询询价价差响应。

函数原型:

```
virtual void OnRspQryForQuoteParam(CThostFtdcForQuoteParamField  
*pForQuoteParam, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast);
```

参数:

pForQuoteParam 查询询价价差信息:

字段类型	字段名称	含义	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcPriceType	LastPrice	最新价	
TThostFtdcPriceType	PriceInterval	价差	

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息 ErrorMsg;

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.55 OnRspMKBatchOrderAction

功能:

做市商批量报单操作请求响应。

函数原型:

```
virtual void OnRspMKBatchOrderAction(CThostFtdcMKInputOrderActionField  
*pMKInputOrderAction, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool  
bIsLast) {};
```

参数:

pMKInputOrderAction, 参见章节《[3.1.52 ReqMKBatchOrderAction](#)》

pRspInfo: 响应信息结构体, 包含响应代码 ErrorID、响应信息 ErrorMsg;

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.56 OnRspSubscribeFlowCtrlWarning

功能：

订阅流控警告应答。

函数原型：

```
virtual void  
OnRspSubscribeFlowCtrlWarning(CThostFtdcSpecificTraderField  
*pRspSubscribeTraderField, CThostFtdcRspInfoField *pRspInfo, int  
nRequestID, bool bIsLast) {};
```

参数：

pRspSubscribeTraderField 指定的席位：

字段类型	字段名称	含义	备注
TThostFtdcTraderIDType	TraderID	席位代码	

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.57 OnRspUnSubscribeFlowCtrlWarning

功能：

取消订阅流控警告应答。

函数原型：

```
virtual void  
OnRspUnSubscribeFlowCtrlWarning(CThostFtdcSpecificTraderField  
*pRspSubscribeTraderField, CThostFtdcRspInfoField *pRspInfo, int  
nRequestID, bool bIsLast) {};
```

参数：

pRspSubscribeTraderField 指定的席位：

字段类型	字段名称	含义	备注
TThostFtdcTraderIDType	TraderID	席位代码	

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.58 OnRtnFlowCtrlWarning

功能:

交易所席位流控警告

函数原型:

```
virtual void OnRtnFlowCtrlWarning(CThostFtdcFlowCtrlWarningField
*pFlowCtrlWarning) {};
```

参数:

pFlowCtrlWarning 流控警告通知:

字段类型	字段名称	含义	备注
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcLongTimeType	TriggerTime	触发时间	
unsigned int	Tgid	tid 组号 (交易所编号)	
unsigned int	CurPkgCnt	瞬时流量值	

3.2.59 OnRspQryTraderOffer

功能:

请求查询交易员席位状态响应

函数原型:

```
virtual void OnRspQryTraderOffer(CThostFtdcTraderOfferField *pTraderOffer,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

参数:

pTraderOffer 结构体如下:

字段类型	字段名称	含义
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcTraderIDType	TraderID	交易所交易员代码
TThostFtdcParticipantIDType	ParticipantID	会员代码
TThostFtdcPasswordType	Password	密码
TThostFtdcInstallIDType	InstallID	安装编号
TThostFtdcOrderLocalIDType	OrderLocalID	本地报单编号
TThostFtdcTraderConnectStatusType	TraderConnectStatus	交易所交易员连接状态 ///没有任何连接

		#define THOST_FTDC_TCS_NotConnected '1' ///已经连接 #define THOST_FTDC_TCS_Connected '2'
TThostFtdcDateType	ConnectRequestDate	发出连接请求的日期
TThostFtdcTimeType	ConnectRequestTime	发出连接请求的时间
TThostFtdcDateType	LastReportDate	上次报告日期
TThostFtdcTimeType	LastReportTime	上次报告时间
TThostFtdcDateType	ConnectDate	完成连接日期
TThostFtdcTimeType	ConnectTime	完成连接时间
TThostFtdcDateType	StartDate	启动日期
TThostFtdcTimeType	StartTime	启动时间
TThostFtdcDateType	TradingDay	交易日
TThostFtdcBrokerIDType	BrokerID	经纪公司代码
TThostFtdcTradeIDType	MaxTradeID	本席位最大成交编号
TThostFtdcReturnCodeType	MaxOrderMessageReference	本席位最大报单备拷

pRspInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.60 OnRspQryInvestorProdSPBMDetail

功能：

请求查询投资者 SPBM 品种保证金明细响应；

函数原型：

```
virtual void OnRspQryInvestorProdSPBMDetail
(CThostFtdcInvestorProdSPBMDetailField *pInvestorProdSPBMDetail,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

参数：

pInvestorProdSPBMDetail 结构体如下：

字段类型	字段名称	含义
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcBrokerIDType	BrokerID	经纪公司代码
TThostFtdcInvestorIDType	InvestorID	投资者代码
TThostFtdcInstrumentIDType	ProdFamilyCode	品种代码
TThostFtdcMoneyType	IntralInstrMargin	合约内对锁保证金
TThostFtdcMoneyType	BCollectingMargin	买方向归集保证金
TThostFtdcMoneyType	SCollectingMargin	卖方向归集保证金

TThostFtdcMoneyType	IntraProdMargin	品种内合约间对锁保证金
TThostFtdcMoneyType	NetMargin	净保证金
TThostFtdcMoneyType	InterProdMargin	产品间对锁保证金
TThostFtdcMoneyType	SingleMargin	裸保证金
TThostFtdcMoneyType	AddOnMargin	附加保证金
TThostFtdcMoneyType	DeliveryMargin	交割保证金
TThostFtdcMoneyType	OptionMinRisk	卖方期权最低风险
TThostFtdcMoneyType	RealOptionValueOffset	价值冲抵
TThostFtdcMoneyType	Margin	保证金
TThostFtdcMoneyType	ExchMargin	交易所保证金

SPBM 保证金算法说明：

- 单个 SPBM 品种总保证金 = max (合约对冲保证金 + 品种内合约间对冲保证金 + 品种间对冲保证金 + 品种裸仓保证金 + 附加保证金, 卖方期权最低风险) + 交割保证金 - 价值冲抵
- 投资者 SPBM 总保证金等于各单一产品 SPBM 保证金加和

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.61 OnErrRtnOrderInsert

功能：

报单录入错单回报，当执行 ReqOrderInsert 后有交易所错误则通过此接口返回。

函数原型：

```
virtual void OnErrRtnOrderInsert(CThostFtdcInputOrderField *pInputOrder,
CThostFtdcRspInfoField *pRspInfo);
```

参数：

pInputOrder 报单操作错误回报信息；对应的结构各字段参见《ReqOrderInsert》章节。

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

3.2.62 OnRspQrySPMMInvestorCommodityGroupMargin

功能：

请求查询投资者 SPMM 商品群保证金明细响应；

函数原型：

```

    virtual void
OnRspQrySPMMInvestorCommodityGroupMargin( CThostFtdcSPMMInvestorCommodityGroupMarginField
                                            *pSPMMInvestorCommodityGroupMargin,
CThostFtdcRsplInfoField *pRsplInfo, int nRequestId, bool bIsLast ) {};

```

参数：

pSPMMInvestorCommodityGroupMargin 结构体如下：

字段类型	字段名称	含义
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcBrokerIDType	BrokerID	经纪公司代码
TThostFtdcInvestorIDType	InvestorID	投资者代码
TThostFtdcSPMMProductIDType	CommodityGroupID	商品群代码
TThostFtdcMoneyType	MarginBeforeDiscount	优惠仓位应收保证金
TThostFtdcMoneyType	MarginNoDiscount	不优惠仓位应收保证金
TThostFtdcMoneyType	LongRisk	多头风险
TThostFtdcMoneyType	ShortRisk	空头风险
TThostFtdcMoneyType	CloseFrozenMargin	商品群平仓冻结保证金
TThostFtdcMoneyType	InterCommodityRate	SPMM 跨品种优惠系数
TThostFtdcMoneyType	MinMarginRatio	商品群最小保证金比例
TThostFtdcMoneyType	AdjustRatio	投资者保证金和交易所保证金的比例
TThostFtdcMoneyType	IntraCommodityDiscount	SPMM 品种内优惠汇总
TThostFtdcMoneyType	InterCommodityDiscount	SPMM 跨品种优惠
TThostFtdcMoneyType	ExchMargin	交易所保证金
TThostFtdcMoneyType	InvestorMargin	投资者保证金
TThostFtdcMoneyType	FrozenCommission	冻结的手续费
TThostFtdcMoneyType	Commission	手续费
TThostFtdcMoneyType	FrozenCash	冻结的资金
TThostFtdcMoneyType	CashIn	资金差额
TThostFtdcMoneyType	StrikeFrozenMargin	行权冻结资金

SPMM 保证金算法说明：

- 单个 SPMM 商品群总保证金=MAX (商品群应收保证金-商品群跨期优惠保证金-商品群报单优惠保证金-商品群跨品种优惠保证金，商品群最小保证金)；
- 投资者 SPMM 总保证金=SUM (单个 SPMM 商品群总保证金)；

pRsplInfo：响应信息结构体，包含响应代码 ErrorCode、响应信息ErrorMsg；

nRequestId：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.63 OnRspQryRULEInvestorProdMargin

功能：

请求查询投资者 RULE 商品群保证金明细响应；

函数原型：

```
virtual void
OnRspQryRULEInvestorProdMargin(CThostFtdcRULEInvestorProdMarginField*
pRULEInvestorProdMargin, CThostFtdcRsplInfoField* pRsplInfo, int nRequestID, bool
bIsLast) {};
```

参数：

pRULEInvestorProdMargin 结构体如下：

字段类型	字段名称	含义
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcBrokerIDType	BrokerID	经纪公司代码
TThostFtdcInvestorIDType	InvestorID	投资者代码
TThostFtdcProductIDType	ProdFamilyCode	品种代码
TThostFtdcRULEInstrClassType	InstrumentClass	合约类型
TThostFtdcRULECommodityGroupIDType	CommodityGroupID	商品群号
TThostFtdcMoneyType	IntraProdMargin	品种内对锁保证金
TThostFtdcMoneyType	InterProdMargin	品种间对锁保证金
TThostFtdcMoneyType	SingleMargin	跨品种单腿保证金
TThostFtdcMoneyType	NonCombMargin	非组合合约保证金
TThostFtdcMoneyType	AddOnMargin	附加保证金
TThostFtdcMoneyType	ExchMargin	交易所保证金
TThostFtdcMoneyType	OpenFrozenMargin	开仓冻结保证金
TThostFtdcMoneyType	CloseFrozenMargin	平仓冻结保证金
TThostFtdcMoneyType	Margin	品种保证金

TThostFtdcMoneyType

FrozenMargin;

冻结保证金

pRspInfo: 响应信息结构体, 包含响应代码 Errord、响应信息 ErrorMsg;

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包; false 表示还有包未传完, 继续传送;

3.2.64 OnRspQryRCAMSInvestorProdMargin

功能:

请求查询投资者 RCAMS 组合保证金响应;

函数原型:

```
virtual void OnRspQryRCAMSInvestorProdMargin(CThostFtdcRCAMSInvestorProdMarginField *pRCAMSInvestorProdMargin, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

参数:

pRCAMSInvestorProdMargin 结构体如下:

字段类型	字段名称	含义
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcBrokerIDType	BrokerID	经纪公司代码
TThostFtdcInvestorIDType	InvestorID	投资者代码
TThostFtdcInstrumentIDType	CombProductID	商品组代码
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志
TThostFtdcInstrumentIDType	ProductGroupID	商品群代码
TThostFtdcMoneyType	RiskBeforeDiscount	品种组合前风险
TThostFtdcMoneyType	IntraInstrRisk	同合约对冲风险
TThostFtdcMoneyType	BPosRisk	品种买持仓风险
TThostFtdcMoneyType	SPosRisk	品种卖持仓风险
TThostFtdcMoneyType	IntraProdRisk	品种内对冲风险
TThostFtdcMoneyType	NetRisk	品种净持仓风险
TThostFtdcMoneyType	InterProdRisk	品种间对冲风险
TThostFtdcMoneyType	OptionRoyalty	空头期权权利金

TThostFtdcMoneyType	DeliveryOpenFrozenMargin	交割月期货开仓冻结保证金
TThostFtdcMoneyType	OpenFrozenMargin	开仓冻结保证金
TThostFtdcMoneyType	UseFrozenMargin	投资者冻结保证金
TThostFtdcMoneyType	MMSAExchMargin	投资者冻结保证金
TThostFtdcMoneyType	DeliveryExchMargin	交割月期货交易所持仓保证金
TThostFtdcMoneyType	CombExchMargin	策略组合交易所保证金
TThostFtdcMoneyType	ExchMargin	交易所持仓保证金
TThostFtdcMoneyType	UseMargin	投资者持仓保证金

pRspInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.65 OnRspQryRCAMSInvestorCombPosition

功能：

请求查询 RCAMS 策略组合持仓响应

函数原型：

```
virtual void
OnRspQryRCAMSInvestorCombPosition(CThostFtdcRCAMSInvestorCombPositionField *pRCAMSInvestorCombPosition, CThostFtdcRspInfoField *pRspInfo, int nRequestID,
bool bIsLast) {};
```

参数：

pRCAMSInvestorCombPosition 结构体如下：

字段类型	字段名称	含义
TThostFtdcExchangeIDType	ExchangeID	交易所代码
TThostFtdcBrokerIDType	BrokerID	经纪公司代码
TThostFtdcInvestorIDType	InvestorID	投资者代码
TThostFtdcInstrumentIDType	InstrumentID	合约代码
TThostFtdcHedgeFlagType	HedgeFlag	投机套保标志
TThostFtdcPosiDirectionType	PosiDirection	持仓多空方向

TThostFtdcInstrumentIDType	CombiInstrumentID;	组合合约代码
TThostFtdcLegIDType	LegID	LegID
TThostFtdcInstrumentIDType	ExchangeInstID	交易所组合合约代码
TThostFtdcVolumeType	TotalAmt	数量
TThostFtdcMoneyType	ExchMargin	交易所持仓保证金
TThostFtdcMoneyType	Margin	投资者保证金

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.66 OnRspQryControlParam

功能：

请求查询交易开关设置响应。

相关参数返回值含义参见 [3.1.61 ReqQryControlParam](#)。

函数原型：

```
void OnRspQryControlParam(CThostFtdcControlParamField *pControlParam,
CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pControlParam 结构体返回如下：

字段类型	字段名称	含义	是否可作为过滤条件
TThostFtdcInvestorIDType	InvestorID	投资者代码	是
TThostFtdcControlParamIDType	ControlParamID	开关代码	是
TThostFtdcControlParamValueType	ControlParamValue	开关设置	是
TThostFtdcMemoType	Memo	备注说明	是

pRspInfo: 响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

3.2.67 OnRspSubscribeFundChange

功能：

订阅资金变动应答。

如后台订阅开关关闭，则返回错误“CTP：资金推送功能未开启”，请联系各家后台经纪公司。

函数原型：

```
virtual void OnRspSubscribeFundChange(CThostFtdcRspInfoField  
*pRspInfo, int nRequestId, bool bIsLast) {};
```

参数：

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestId：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.68 OnRspUnSubscribeFundChange

功能：

取消订阅资金变动应答。

如后台订阅开关关闭，则返回错误“CTP：资金推送功能未开启”，请联系各家后台经纪公司。

函数原型：

```
virtual void OnRspUnSubscribeFundChange(CThostFtdcRspInfoField  
*pRspInfo, int nRequestId, bool bIsLast) {};
```

参数：

pRspInfo：响应信息结构体，包含响应代码 Errord、响应信息 ErrorMsg；

nRequestId：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

3.2.69 OnRtnFundChange

功能：

订阅成功后接收资金变动推送。

函数原型：

```
virtual void OnRtnFundChange(CThostFtdcTradingAccountField  
*pTradingAccount) {};
```

参数：

pTradingAccount 结构体参见：[3.2.17 OnRspQryTradingAccount](#)

3.2.70 OnRspOffsetSetting

功能：

对冲设置请求响应，当执行 ReqOffsetSetting 后，该方法被调用。

函数原型：

```
virtual void OnRspOffsetSetting(CThostFtdcInputOffsetSettingField  
*pInputOffsetSetting, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast)  
{};
```

参数：

pInputOffsetSetting 结构体参见：[3.1.64 ReqOffsetSetting](#)

3.2.71 OnRspCancelOffsetSetting

功能：

撤销对冲设置请求响应，当执行 ReqCancelOffsetSetting 后，该方法被调用。

函数原型：

```
virtual void OnRspCancelOffsetSetting(CThostFtdcInputOffsetSettingField  
*pInputOffsetSetting, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast)  
{};
```

参数：

pInputOffsetSetting 结构体参见：[3.1.65 ReqCancelOffsetSetting](#)

3.2.72 OnRtnOffsetSetting

功能：

对冲设置通知，当执行 ReqOffsetSetting 、ReqCancelOffsetSetting 后，该方法被调用。

函数原型：

```
virtual void OnRtnOffsetSetting(CThostFtdcOffsetSettingField *pOffsetSetting)
{};
```

参数：

pOffsetSetting：对冲设置回报信息，相关字段信息参考下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInstrumentIDType	UnderlyingInstrID	标的期货合约代码	
TThostFtdcProductIDType	ProductID	产品代码	
TThostFtdcOffsetTypeType	OffsetType	对冲类型	
TThostFtdcVolumeType	Volume	申请对冲的合约数量	
TThostFtdcBoolType	IsOffset	是否对冲	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcExchangeInstIDType	ExchangeInstID	交易所合约代码	
TThostFtdcExchangeInstIDType	ExchangeSerialNo	交易所期权系列号	
TThostFtdcProductIDType	ExchangeProductID	交易所产品代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderSubmitStatusType	OrderSubmitStatus	对冲提交状态	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcDateType	InsertDate	报单日期	
TThostFtdcTimeType	InsertTime	插入时间	
TThostFtdcTimeType	CancelTime	撤销时间	
TThostFtdcExecResultType	ExecResult	对冲设置结果	
TThostFtdcSequenceNoType	SequenceNo	序号	
TThostFtdcFrontIDType	FrontID	前置编号	
TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcErrorMsgType	StatusMsg	状态信息	

TThostFtdcUserIDType	ActiveUserID	操作用户代码	
TThostFtdcSequenceNoType	BrokerOffsetSettingSeq	经纪公司报单编号	
TThostFtdcApplySrcType	ApplySrc	申请来源	

3.2.73 OnErrRtnCancelOffsetSetting

功能：

撤销对冲设置错误回报，当执行 ReqCancelOffsetSetting 后，该方法被调用。

函数原型：

```
virtual void OnErrRtnCancelOffsetSetting(CThostFtdcCancelOffsetSettingField
*pCancelOffsetSetting, CThostFtdcRspInfoField *pRspInfo) {};
```

参数：

pCancelOffsetSetting：对冲设置撤销错误回报信息，相关字段信息参考下表：

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInstrumentIDType	UnderlyingInstrID	标的期货合约代码	
TThostFtdcProductIDType	ProductID	产品代码	
TThostFtdcOffsetTypeType	OffsetType	对冲类型	
TThostFtdcVolumeType	Volume	申请对冲的合约数量	
TThostFtdcBoolType	IsOffset	是否对冲	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcExchangeInstIDType	ExchangeInstID	交易所合约代码	
TThostFtdcExchangeInstIDType	ExchangeSerialNo	交易所期权系列号	
TThostFtdcProductIDType	ExchangeProductID	交易所产品代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcOrderActionStatusType	OrderActionStatus	报单操作状态	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcOrderLocalIDType	ActionLocalID	操作本地编号	
TThostFtdcDateType	ActionDate	操作日期	
TThostFtdcTimeType	ActionTime	操作时间	

3.2.74 OnRspQryOffsetSetting

功能：

请求查询对冲设置响应。

函数原型：

```
virtual void OnRspQryOffsetSetting(CThostFtdcOffsetSettingField
*pOffsetSetting, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
bIsLast) {};
```

参数：

pOffsetSetting 查询对冲设置信息，相关字段信息参考章节：[3.2.72 OnRtnOffsetSetting](#)

字段类型	字段名称	字段说明	备注
TThostFtdcBrokerIDType	BrokerID	经纪公司代码	
TThostFtdcInvestorIDType	InvestorID	投资者代码	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcInstrumentIDType	UnderlyingInstrID	标的期货合约代码	
TThostFtdcProductIDType	ProductID	产品代码	
TThostFtdcOffsetTypeType	OffsetType	对冲类型	
TThostFtdcVolumeType	Volume	申请对冲的合约数量	
TThostFtdcBoolType	IsOffset	是否对冲	
TThostFtdcRequestIDType	RequestID	请求编号	
TThostFtdcUserIDType	UserID	用户代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcBusinessUnitType	BusinessUnit	业务单元	
TThostFtdcIPAddressType	IPAddress	IP 地址	
TThostFtdcMacAddressType	MacAddress	Mac 地址	
TThostFtdcExchangeInstIDType	ExchangeInstID	交易所合约代码	
TThostFtdcExchangeInstIDType	ExchangeSerialNo	交易所期权系列号	
TThostFtdcProductIDType	ExchangeProductID	交易所产品代码	
TThostFtdcParticipantIDType	ParticipantID	会员代码	
TThostFtdcClientIDType	ClientID	客户代码	
TThostFtdcTraderIDType	TraderID	交易所交易员代码	
TThostFtdcInstallIDType	InstallID	安装编号	
TThostFtdcOrderSubmitStatusType	OrderSubmitStatus	对冲提交状态	
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcSettlementIDType	SettlementID	结算编号	
TThostFtdcDateType	InsertDate	报单日期	
TThostFtdcTimeType	InsertTime	插入时间	
TThostFtdcTimeType	CancelTime	撤销时间	
TThostFtdcExecResultType	ExecResult	对冲设置结果	
TThostFtdcSequenceNoType	SequenceNo	序号	
TThostFtdcFrontIDType	FrontID	前置编号	

TThostFtdcSessionIDType	SessionID	会话编号	
TThostFtdcErrorMsgType	StatusMsg	状态信息	
TThostFtdcUserIDType	ActiveUserID	操作用户代码	
TThostFtdcSequenceNoType	BrokerOffsetSettingSeq	经纪公司报单编号	
TThostFtdcApplySrcType	ApplySrc	申请来源	

pRsplInfo: 响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestID: 返回终端原始请求编号

bIsLast: true 表示最后一个数据包； false 表示还有包未传完，继续传送；

4 行情接口

4.1 CThostFtdcMdApi

功能：

创建 MdApi

函数原型：

```
static CThostFtdcMdApi *CreateFtdcMdApi(const char *pszFlowPath = "", const  
bool bIsUsingUdp = false, const bool bIsMulticast = false);
```

参数：

pszFlowPath 存贮订阅信息文件的目录， 默认为当前目录

bIsUsingUdp 是否使用 udp,mini2 版本不再使用

bIsMulticast 是否使用多播,mini2 版本不再使用；

返回值：

创建出的 CThostFtdcMdApi 对象

示例：

```
void MyMDInit ()
```

```
{
```

```
//创建行情 API 实例
```

```
CThostFtdcMdApi *pUserMdApi = CThostFtdcMdApi::CreateFtdcMdApi();
```

```
//创建行情回调处理实例，投资者自行继承基类进行相关行情回调处理
```

```
CMyMdHandler ash(pUserMdApi);
```

```
//注册行情回调实例
```

```
pUserMdApi->RegisterSpi(&ash);  
//注册行情前置服务器  
pUserMdApi->RegisterFront( "tcp://127.0.0.1:41205" );  
//准备充分后正式启动行情实例初始化  
pUserMdApi->Init();  
//阻塞主线程，等待初始化完成；  
pUserMdApi->Join();  
}
```

4.1.2 GetApiVersion

功能：

获取 API 的版本信息

函数原型：

```
static const char *GetApiVersion();
```

参数：

无

返回值：

获取到的版本号字符串指针

4.1.3 Release

功能：

删除接口对象本身；不再使用本接口对象时，调用该函数删除接口对象

函数原型：

```
virtual void Release();
```

参数：

无

返回值：

无

4.1.4 Init

功能：

初始化运行环境,只有调用后,接口才开始工作

函数原型：

```
virtual void Init(bool bContinuous = false);
```

参数：

bContinuous 为 true 表示线程不休眠；
为 false 表示线程休眠；
缺省为 false；

返回值：

无

示例：

```
CThostFtdcMdApi *pUserMdApi = CThostFtdcMdApi::CreateFtdcMdApi();  
  
CMyMdHandler ash(pUserMdApi); //CMyMdHandler 继承了 CThostFtdcMdSpi 类  
  
pUserMdApi->RegisterSpi(&ash);  
  
pUserMdApi->RegisterFront( "tcp://127.0.0.1:41205" );  
  
pUserMdApi->Init();  
  
.....
```

4.1.5 Join

功能：

等待接口线程结束运行

函数原型：

```
virtual int Join();
```

参数：

无

返回值：

线程退出代码

示例：

略

4.1.6 GetTradingDay

功能：

获取当前交易日；只有登录成功后，才能得到正确的交易日

函数原型：

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

参数：

无

返回值：

交易日字符串指针

4.1.7 RegisterFront

功能：

注册前置机网络地址

函数原型：

```
virtual void RegisterFront(char *pszFrontAddress);
```

参数：

pszFrontAddress: 前置机网络地址, 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” 代表行情端口号。

返回值：

无

示例：

```
//注册前置  
pUserMdApi->RegisterFront( "tcp://192.168.0.1:41213" );
```

4.1.8 RegisterSpi

功能：

注册回调接口

函数原型：

```
virtual void RegisterSpi(CThostFtdcMdSpi *pSpi);
```

参数：

pSpi 派生自回调接口类 CThostFtdcMdSpi 的实例

返回值：

无

示例：

略

4.1.9 SubscribeMarketData

功能：

订阅行情

- 订阅全市场合约需要把全市场合约代码都赋值给 `pplInstrumentID`, 填空不能订阅全市场合约;
- 只推送最新行情, 不推送历史行情;

函数原型：

```
virtual int SubscribeMarketData(char *pplInstrumentID[], int nCount);
```

参数：

`pplInstrumentID`: 合约 ID 数组;

- 行情订阅中合约编码填写务必确保正确, 否则将收不到行情; 程序开发中可以通过复制查询回来的合约名方式确保填写正确;
 - 如各家交易所期货命名规则如下:
 - ◆ 上期/能源所: 小写+4 个数字 (rb2109 代表 rb 品种, 2021 年 9 月份到期)
 - ◆ 中金所: 大写+4 个数字
 - ◆ 郑商所: 大写+3 个数字(TA001 代表 TA 品种, 20 年 01 月份到期)
 - ◆ 大商所: 小写+4 个数字
- 过期合约行情订阅后将收不到行情;
- 交易系统提供的行情是各家交易所切片行情, 通常是 1 秒 2 笔。

`nCount`: 要订阅行情的合约个数;

返回值：

0: 成功

-1: 失败

示例：

```
char pplInstrumentID[2][100];
pplInstrumentID[0] = "T2102";
pplInstrumentID[1] = "T2112"
m_pUserMdApi->SubscribeMarketData(pplInstrumentID, 2);
```

4.1.10 UnSubscribeMarketData

功能：

取消行情订阅

函数原型：

```
virtual int UnSubscribeMarketData (char *ppInstrumentID[], int nCount);
```

参数：

ppInstrumentID: 合约 ID 数组;

nCount: 要退订行情的合约个数;

返回值：

0: 成功

-1: 失败

示例：

略

4.1.11 SubscribeForQuoteRsp

功能：

做市商订阅投资者询价

函数原型：

```
virtual int SubscribeForQuoteRsp(char *ppInstrumentID[], int nCount);
```

参数：

ppInstrumentID: 合约 ID 数组;

nCount: 要订阅询价的合约个数;

返回值：

0: 成功

-1: 失败

示例：

略

4.1.12 UnSubscribeForQuoteRsp

功能：

做市商退订投资者询价

函数原型：

```
virtual int UnSubscribeForQuoteRsp(char *ppInstrumentID[], int nCount);
```

参数：

ppInstrumentID：合约 ID 数组；

nCount：要退订询价的合约个数；

返回值：

0：成功

-1：失败

示例：

略

4.1.13 ReqUserLogin

功能：

用户登录请求

函数原型：

```
virtual int ReqUserLogin(CThostFtdcReqUserLoginField *pReqUserLoginField, int  
nRequestID);
```

参数：

pReqUserLoginField 登录用户的基本信息；

nRequestID：终端请求编号

返回值：

0：成功

-1：失败

示例：

参见前面章节《ReqUserLogin》

4.2 CThostFtdcMdSpi

4.2.1 OnFrontConnected

功能：

当客户端与交易后台建立起通信连接时（还未登录前），该方法被调用。

函数原型：

virtual void OnFrontConnected();

参数：

无；

示例：

略

4.2.2 OnFrontDisconnected

功能：

当客户端与交易后台通信连接断开时，该方法被调用。

当发生这个情况后，API 会自动重新连接，客户端可不做处理。

函数原型：

virtual void OnFrontDisconnected(int nReason);

参数：

nReason：错误原因，含义如下

- -3 关闭连接
- -4 网络读失败
- -5 网络写失败
- -6 读订阅流水请求出错
- -7 序列号错误
- -8 读心跳出错
- -9 错误的网络包大小

4.2.3 OnRspUserLogin

功能：

登录请求响应

函数原型：

**virtual void OnRspUserLogin(CThostFtdcRspUserLoginField *pRspUserLogin,
CThostFtdcRspInfoField *pRsplInfo, int nRequestId, bool bIsLast);**

参数：

pRspUserLogin：登录请求响应信息，对应的结构各字段参见《3.2.4
OnRspUserLogin》章节

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息ErrorMsg；

nRequestID：返回终端原始请求编号

bIsLast：true；

4.2.4 OnRspSubMarketData

功能：

订阅行情应答

函数原型：

```
virtual void OnRspSubMarketData(CThostFtdcSpecificInstrumentField  
*pSpecificInstrument, CThostFtdcRspInfoField *pRsplInfo, int nRequestId, bool  
bIsLast);
```

参数：

pSpecificInstrument： 订阅行情应答信息，对应的结构各字段如下表：

字段类型	字段名称	含义	备注
TThostFtdcInstrumentIDType	InstrumentID	合约代码	

pRsplInfo： 响应信息结构体，包含响应代码 ErrordID、响应信息 ErrorMsg；

nRequestId： 返回终端原始请求编号

bIsLast： true 表示最后一个数据包； false 表示还有包未传完，继续传送；

4.2.5 OnRspUnSubMarketData

功能：

取消订阅行情应答

函数原型：

```
virtual void OnRspUnSubMarketData(CThostFtdcSpecificInstrumentField  
*pSpecificInstrument, CThostFtdcRspInfoField *pRsplInfo, int nRequestId, bool  
bIsLast);
```

参数：

pSpecificInstrument： 取消订阅行情应答信息，对应的结构各字段参见《OnRspSubMarketData》章节

pRsplInfo： 响应信息结构体，包含响应代码 ErrordID、响应信息 ErrorMsg；

nRequestId： 返回终端原始请求编号

bIsLast： true 表示最后一个数据包； false 表示还有包未传完，继续传送；

4.2.6 OnRspSubForQuoteRsp

功能：

做市商订阅询价应答

函数原型：

```
virtual void OnRspSubForQuoteRsp(CThostFtdcSpecificInstrumentField  
*pSpecificInstrument, CThostFtdcRsplInfoField *pRsplInfo, int nRequestId, bool  
bIsLast);
```

参数：

pSpecificInstrument：订阅询价应答信息，对应的结构各字段参见《OnRspSubMarketData》章节

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestId：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

4.2.7 OnRspUnSubForQuoteRsp

功能：

做市商取消订阅询价应答

函数原型：

```
virtual void OnRspUnSubForQuoteRsp(CThostFtdcSpecificInstrumentField  
*pSpecificInstrument, CThostFtdcRsplInfoField *pRsplInfo, int nRequestId, bool  
bIsLast);
```

参数：

pSpecificInstrument：取消订阅询价应答信息，对应的结构各字段参见《OnRspSubMarketData》章节。

pRsplInfo：响应信息结构体，包含响应代码 ErrorID、响应信息 ErrorMsg；

nRequestId：返回终端原始请求编号

bIsLast：true 表示最后一个数据包；false 表示还有包未传完，继续传送；

4.2.8 OnRtnDepthMarketData

功能：

深度行情通知

函数原型：

```
virtual void OnRtnDepthMarketData(CThostFtdcDepthMarketDataField *pDepthMarketData);
```

参数：

pDepthMarketData: 深度行情通知信息，对应的结构各字段如下表：

字段类型	字段名称	字段说明	备注
TThostFtdcDateType	TradingDay	交易日	
TThostFtdcInstrumentIDType	InstrumentID	合约代码	
TThostFtdcExchangeIDType	ExchangeID	交易所代码	
TThostFtdcExchangeInstIDType	ExchangeInstID	合约在交易所的代码	
TThostFtdcPriceType	LastPrice	最新价	
TThostFtdcPriceType	PreSettlementPrice	上次结算价	
TThostFtdcPriceType	PreClosePrice	昨收盘	
TThostFtdcLargeVolumeType	PreOpenInterest	昨持仓量	
TThostFtdcPriceType	OpenPrice	今开盘	
TThostFtdcPriceType	HighestPrice	最高价	
TThostFtdcPriceType	LowestPrice	最低价	
TThostFtdcVolumeType	Volume	数量	
TThostFtdcMoneyType	Turnover	成交金额，郑商所行情中，不推送成交金额，可以自行计算，成交金额=成交均价*成交量。	
TThostFtdcLargeVolumeType	OpenInterest	持仓量	
TThostFtdcPriceType	ClosePrice	今收盘	
TThostFtdcPriceType	SettlementPrice	本次结算价	
TThostFtdcPriceType	UpperLimitPrice	涨停板价	
TThostFtdcPriceType	LowerLimitPrice	跌停板价	
TThostFtdcRatioType	PreDelta	昨虚实度	
TThostFtdcRatioType	CurrDelta	今虚实度	
TThostFtdcTimeType	UpdateTime	最后修改时间	
TThostFtdcMillisecondType	UpdateMillisecond	最后修改毫秒	
TThostFtdcPriceType	BidPrice1	申买价一	
TThostFtdcVolumeType	BidVolume1	申买量一	
TThostFtdcPriceType	AskPrice1	申卖价一	
TThostFtdcVolumeType	AskVolume1	申卖量一	
TThostFtdcPriceType	BidPrice2	申买价二	
TThostFtdcVolumeType	BidVolume2	申买量二	
TThostFtdcPriceType	AskPrice2	申卖价二	
TThostFtdcVolumeType	AskVolume2	申卖量二	
TThostFtdcPriceType	BidPrice3	申买价三	
TThostFtdcVolumeType	BidVolume3	申买量三	
TThostFtdcPriceType	AskPrice3	申卖价三	
TThostFtdcVolumeType	AskVolume3	申卖量三	
TThostFtdcPriceType	BidPrice4	申买价四	
TThostFtdcVolumeType	BidVolume4	申买量四	

TThostFtdcPriceType	AskPrice4	申卖价四	
TThostFtdcVolumeType	AskVolume4	申卖量四	
TThostFtdcPriceType	BidPrice5	申买价五	
TThostFtdcVolumeType	BidVolume5	申买量五	
TThostFtdcPriceType	AskPrice5	申卖价五	
TThostFtdcVolumeType	AskVolume5	申卖量五	
TThostFtdcPriceType	AveragePrice	当日均价	
TThostFtdcDateType	ActionDay	业务日期	
TThostFtdcPriceType	BandingUpperPrice	上带价	V1.5.7 新增
TThostFtdcPriceType	BandingLowerPrice	下带价	V1.5.7 新增

- 只提供实时行情，不提供历史行情；
- 大商所和郑商所有组合合约行情，其他交易所没有组合合约行情。V1.5.8 之前版本的 Mini 交易系统只提供单腿行情，不支持组合行情；V1.5.8 版本增加了大商所组合合约行情、V1.5.9 版本增加了郑商所组合合约行情；
- level2 行情是指深度行情，不同于 level1 只有 1 档，level2 有多档；
- 对于 TAS 合约行情，其中合约代码、合约状态、买卖方向、报单量、成交量、日期、时间等字段正常发布，其余字段无意义，均为空。盘中 TAS 成交量、成交额不计入标的合约的成交量、成交额；

4.2.9 OnRtnForQuoteRsp

功能：

做市商接收询价通知

- 询价录入成功后会推送，但是事先要调用行情 api 里的 SubscribeForQuoteRsp 去订阅该消息；

函数原型：

```
virtual void OnRtnForQuoteRsp(CThostFtdcForQuoteRspField
*pForQuoteRsp);
```

参数：

pForQuoteRsp: 询价通知信息，对应的结构各字段参考《[OnRtnForQuoteRsp](#)》章节

5 错误码对应表

错误 ID 宏变量	错误 ID	错误描述
ERROR_NONE	0	CTP:正确



ERROR_INVALID_DATA_SYNC_STATUS	1	CTP:不在已同步状态
ERROR_INCONSISTENT_INFORMATION	2	CTP:会话信息不一致
ERROR_INVALID_LOGIN	3	CTP:不合法的登录
ERROR_USER_NOT_ACTIVE	4	CTP:用户不活跃
ERROR_DUPLICATE_LOGIN	5	CTP:重复的登录
ERROR_NOT_LOGIN_YET	6	CTP:还没有登录
ERROR_NOT_INITED	7	CTP:还没有初始化
ERROR_FRONT_NOT_ACTIVE	8	CTP:前置不活跃
ERROR_NO_PRIVILEGE	9	CTP:无此权限
ERROR_CHANGE_OTHER_PASSWORD	10	CTP:修改别人的口令
ERROR_USER_NOT_FOUND	11	CTP:找不到该用户
ERROR_BROKER_NOT_FOUND	12	CTP:找不到该经纪公司
ERROR_INVESTOR_NOT_FOUND	13	CTP:找不到投资者
ERROR_OLD_PASSWORD_MISMATCH	14	CTP:原口令不匹配
ERROR_BAD_FIELD	15	CTP:报单字段有误
ERROR_INSTRUMENT_NOT_FOUND	16	CTP:找不到合约
ERROR_INSTRUMENT_NOT_TRADING	17	CTP:合约不能交易
ERROR_NOT_EXCHANGE_PARTICIPANT	18	CTP:经纪公司不是交易所的会员
ERROR_INVESTOR_NOT_ACTIVE	19	CTP:投资者不活跃
ERROR_NOT_EXCHANGE_CLIENT	20	CTP:投资者未在交易所开户
ERROR_NO_VALID_TRADER_AVAILABLE	21	CTP:无可用交易席位
ERROR_DUPLICATE_ORDER_REF	22	CTP:报单错误：不允许重复报单
ERROR_BAD_ORDER_ACTION_FIELD	23	CTP:错误的报单操作字段
ERROR_DUPLICATE_ORDER_ACTION_REF	24	CTP:撤单已报送，不允许重复撤单
ERROR_ORDER_NOT_FOUND	25	CTP:撤单找不到相应报单
ERROR_INSUITABLE_ORDER_STATUS	26	CTP:报单已全成交或已撤销，或不支持撤未知单
ERROR_UNSUPPORTED_FUNCTION	27	CTP:不支持的功能
ERROR_NO_TRADING_RIGHT	28	CTP:没有报单交易权限
ERROR_CLOSE_ONLY	29	CTP:只能平仓
ERROR_OVER_CLOSE_POSITION	30	CTP:平仓量超过最大可平数量
ERROR_INSUFFICIENT MONEY	31	CTP:资金不足
ERROR_DUPLICATE_PK	32	CTP:主键重复
ERROR_CANNOT_FIND_PK	33	CTP:找不到主键
ERROR_CAN_NOT_INACTIVE_BROKER	34	CTP:设置经纪公司不活跃状态失败
ERROR_BROKER_SYNCHRONIZING	35	CTP:经纪公司正在同步
ERROR_BROKER_SYNCHRONIZED	36	CTP:经纪公司已同步
ERROR_SHORT_SELL	37	CTP:现货交易不能卖空
ERROR_INVALID_SETTLEMENT_REF	38	CTP:不合法的结算引用
ERROR_CFFEX_NETWORK_ERROR	39	CTP:交易所网络连接失败
ERROR_CFFEX_OVER_REQUEST	40	CTP:交易所未处理请求超过许可数
ERROR_CFFEX_OVER_REQUEST_PER_SECOND	41	CTP:交易所每秒发送请求数超过许可数



ERROR_SETTLEMENT_INFO_NOT_CONFIRMED	42	CTP:结算结果未确认
ERROR_DEPOSIT_NOT_FOUND	43	CTP:没有对应的入金记录
ERROR_EXCHANG_TRADING	44	CTP:交易所已经进入连续交易状态
ERROR_PARKEDORDER_NOT_FOUND	45	CTP:找不到预埋(撤单)单
ERROR_PARKEDORDER_HASSENDED	46	CTP:预埋(撤单)单已经发送
ERROR_PARKEDORDER_HASDELETE	47	CTP:预埋(撤单)单已经删除
ERROR_INVALID_INVESTORIDORPASSWORD	48	CTP:无效的投资者或者密码
ERROR_INVALID_LOGIN_IPADDRESS	49	CTP:不合法的登录 IP 地址
ERROR_OVER_CLOSETODAY_POSITION	50	CTP:平今仓位不足
ERROR_OVER_CLOSEYESTERDAY_POSITION	51	CTP:平昨仓位不足
ERROR_BROKER_NOT_ENOUGH_CONDORDER	52	CTP:经纪公司没有足够可用的条件单数量
ERROR_INVESTOR_NOT_ENOUGH_CONDORDER	53	CTP:投资者没有足够可用的条件单数量
ERROR_BROKER_NOT_SUPPORT_CONDORDER	54	CTP:经纪公司不支持条件单
ERROR_RESEND_ORDER_BROKERINVESTOR_NOTMATCH	55	CTP:重发未知单经济公司/投资者不匹配
ERROR_SYC OTP FAILED	56	CTP:同步动态令牌失败
ERROR OTP MISMATCH	57	CTP:动态令牌校验错误
ERROR_OTPPARAM_NOT_FOUND	58	CTP:找不到动态令牌配置信息
ERROR_UNSUPPORTED_OTPTYPE	59	CTP:不支持的动态令牌类型
ERROR_SINGLEUSERSESSION_EXCEED_LIMIT	60	CTP:用户在线会话超出上限
ERROR_EXCHANGE_UNSUPPORTED_ARBITRAGE	61	CTP:该交易所不支持套利类型报单
ERROR_NO_CONDITIONAL_ORDER_RIGHT	62	CTP:没有条件单交易权限
ERROR_AUTH_FAILED	63	CTP:客户端认证失败
ERROR_NOT_AUTHENT	64	CTP:客户端未认证
ERROR_SWAPORDER_UNSUPPORTED	65	CTP:该合约不支持互换类型报单
ERROR_OPTIONS_ONLY_SUPPORT_SPEC	66	CTP:该期权合约只支持投机类型报单
ERROR_DUPLICATE_EXECORDER_REF	67	CTP:执行宣告错误, 不允许重复执行
ERROR_RESEND_EXECORDER_BROKERINVESTOR_NOTMATCH	68	CTP:重发未知执行宣告经纪公司/投资者不匹配
ERROR_EXECORDER_NOTOPTIONS	69	CTP:只有期权合约可执行
ERROR_OPTIONS_NOT_SUPPORT_EXEC	70	CTP:该期权合约不支持执行
ERROR_BAD_EXECORDER_ACTION_FIELD	71	CTP:执行宣告字段有误
ERROR_DUPLICATE_EXECORDER_ACTION_REF	72	CTP:执行宣告撤单已报送, 不允许重复撤单
ERROR_EXECORDER_NOT_FOUND	73	CTP:执行宣告撤单找不到相应执行宣告
ERROR_OVER_EXECUTE_POSITION	74	CTP:执行仓位不足
ERROR_LOGIN_FORBIDDEN	75	CTP:连续登录失败次数超限, 登录被禁止
ERROR_INVALID_TRANSFER_AGENT	76	CTP:非法银期代理关系
ERROR_NO_FOUND_FUNCTION	77	CTP:无此功能
ERROR_SEND_EXCHANGEORDER_FAILED	78	CTP:发送报单失败
ERROR_SEND_EXCHANGEORDERACTION_FAILED	79	CTP:发送报单操作失败
ERROR_PRICETYPE_NOTSUPPORT_BYEXCHANGE	80	CTP:交易所不支持的价格类型
ERROR_BAD_EXECUTE_TYPE	81	CTP:错误的执行类型
ERROR_BAD_OPTION_INSTR	82	CTP:无效的组合合约



ERROR_INSTR_NOTSUPPORT_FORQUOTE	83	CTP:该合约不支持询价
ERROR_RESEND_QUOTE_BROKERINVESTOR_NOTMATCH	84	CTP:重发未知报价经纪公司/投资者不匹配
ERROR_INSTR_NOTSUPPORT_QUOTE	85	CTP:该合约不支持报价
ERROR_QUOTE_NOT_FOUND	86	CTP:报价撤单找不到相应报价
ERROR_OPTIONS_NOT_SUPPORT_ABANDON	87	CTP:该期权合约不支持放弃执行
ERROR_COMBOPTIONS_SUPPORT_IOC_ONLY	88	CTP:该组合期权合约只支持 IOC
ERROR_OPEN_FILE_FAILED	89	CTP:打开文件失败
ERROR_NEED_RETRY	90	综合交易平台：查询未就绪，请稍后重试
ERROR_OVER_INFO_CNT_LIMIT	94	CTP:超过信息量限制笔数，禁止交易
ERROR_NO_TRADING_RIGHT_IN_SEPC_DR	101	CTP:用户在本系统没有报单权限
ERROR_NO_DR_NO	102	CTP:系统缺少灾备标示号
ERROR_BATCHACTION_NOSUPPORT	103	CTP:该交易所不支持批量撤单
ERROR_API_VERSION	104	CTP:API 版本错误
ERROR_INSTRUMENT_INVALID	105	CTP:无效的合约 ID
ERROR_DIRECTION	120	CTP:买卖方向错误
ERROR_OFFSETFLAG	121	CTP:开平标志错误
ERROR_EXCHANGEID	122	CTP:交易所代码错误
ERROR_PRICE_OR_VOLUME	123	CTP:价格或数量错误
ERROR_NO_MARKETDATA	124	CTP:找不到行情
ERROR_NO_MARGINRATE	125	CTP:找不到保证金率
ERROR_NO_COMMISSIONRATE	126	CTP:找不到手续费率
ERROR_NO_ACCTBYPRODUCT	127	CTP:找不到产品资金映射表
ERROR_NO_TRADINGACCOUNT	128	CTP:找不到资金账户
ERROR_NO_PRODUCTGROUP	129	CTP:找不到产品组
ERROR_OPTIONSELFCLOSE_NOT_FOUND	130	CTP:找不到期权自对冲申请
ERROR_OPTIONS_NO_SUPPORT_SPEC	131	CTP:该期权合约只支持投机和套保类型报单
ERROR_OPTIONS_SELFCLOSE_NO_CANCLE	132	CTP:期权自对冲、履约对冲不能重复设置，请先撤销
ERROR_OPTIONS_SELFCLOSE_NO_INSTRUMENT	133	CTP: 期权自对冲、履约对冲合约不存在
ERROR_ORDERTYPE	134	CTP:不支持报单类型
ERROR_ASKPRICE_BIDPRICE_ERROR	135	CTP:双边报价卖价要大于买价
ERROR_OPTION_SPE_ONLY	136	CTP:CTP Mini 期权暂时只支持投机
ERROR_OPTION_AUTO_EXEC_ABANDAND	137	CTP:先撤销再报入
ERROR_NO_CFFEXINSTRUMENTORDERCOMM RATE	138	CTP:找不到中金所申报费率表
ERROR_DRMT_SYNC_DEPOSIT_FORBIDDEN	139	CTP:该投资者没有出入金或修改信用额度权限
ERROR_DUPLICATE_OPTION_SELFCLOSE_REF	140	CTP:期权自对冲、履约对冲报单引用重复
ERROR_OPTION_SELFCLOSE_NOTOPTIONS	141	CTP:只有期权可以申请期权自对冲和履约对冲
ERROR_ORDER_REF	142	CTP:错误的报单引用
ERROR_NO_SUPPORT_ARBI	143	CTP:该合约只支持投机和套保类型报单
ERROR_OUT_OF_TIMEINTERVAL	144	CTP:询价频率超限
ERROR_COMBINATION_LEG_NOT_FOUND	145	CTP:找不到组合腿信息

ERROR_COMBACTION_NOT_FOUND	146	CTP:找不到合法的申请组合/拆分单
ERROR_HEDGEFLAG	147	CTP:投机套保标志错误
ERROR_OUT_OF_PRICEINTERVAL	148	CTP:当前价差禁止询价
ERROR_NO_PRODUCTID	149	CTP:找不到产品代码
ERROR_NO_LIMIT_INFO	150	CTP:批量撤单 OrderRef 模式找不到上下限条件
ERROR_NOT_ALLOW_SELFTRADE	151	CTP:当前报单存在自成交风险
ERROR_PARAMETER_NOT_FOUND	152	CTP:找不到参数信息
ERROR_EXCEED_RETRY_COUNT	153	CTP:重试次数超限，账户已锁定
ERROR_RCAMS_PARA_NOT_FOUND	154	CTP:找不到 RCAMS 合约参数
ERROR_VOLUME_INVALID	155	CTP:无效的交易手数
ERROR_NO_PRODUCTCLASS	156	CTP:不支持的产品类型
ERROR_NO_INVESTOR_POSITION	157	CTP:找不到投资者持仓
ERROR_RULE_INSTR PARA NOT FOUND	160	CTP:找不到 RULE 合约参数
ERROR_PASSWORD_MISMATCH	2001	CTP:资金密码错误
ERROR_DUPLATION_OFFER_SERIAL	2005	CTP:报盘流水号重复
ERROR_IdentifiedCardNo_MISMATCH	2009	CTP:证件号码或类型错误
ERROR_ACCOUNT_NOT_FUND	2011	CTP:资金账户不存在
ERROR_ACCOUNT_NOT_ACTIVE	2012	CTP:资金账户已经销户
ERROR_INVESTOR_NOT_ENOUGH_POSITION	3039	CTP:投资者持仓不足
ERROR_NO_INFOCOMM_STAGE	3040	CTP:没有对应的申报费阶梯配置
ERROR_IS_CALINFOCOMM	3041	CTP:是否收取申报费标记错误
ERROR_NO_VALID_OFFER_AVAILABLE	3042	CTP:报盘未连接
ERROR_AUTHUSER_CHECK_ERR	3043	CTP:用户与客户端授权验证失败
ERROR_OFFSET_SETTING_DOUBLE_SET	3044	CTP:对冲设置不能重复提交，需要先撤销再设置
ERROR_OFFSET_INSTR_PROD	3045	CTP:对冲的合约/产品设置不合法
ERROR_OFFSET_VOLUME	3046	CTP:对冲的数量设置不合法
ERROR_OFFSET_SETTING_REC_NOT_FOUND	3047	CTP:撤销对冲设置未找到记录
ERROR_OFFSET_SETTING_STATUS_ERR	3048	CTP:对冲状态不对，不能撤销
ERROR_OFFSET_TYPE_ERR	3049	CTP:对冲的类型设置不合法
ERROR_OTHER	3200	其它错误
登录错误 1	-1	交易或查询会话没有创建
登录错误 2	-2	用户没有登录
各类查询返回 QRY_ERROR_REQUEST	-6	前一个查询还没有结束，不能提供新查询
通讯包乱序断开连接	-7	发到后台的通讯包出现乱序，无法处理，断开连接，需要终端重连