

飞马开放柜台平台 TraderAPI 说明书



上海金融期货信息技术有限公司
Shanghai Financial Futures Information Technology Co., Ltd

文档标识

项目名称	飞马开放柜台 FIP
文档名称	Trader API
版本号	1.12
状况	<input type="radio"/> 草案 <input type="radio"/> 评审过的 <input checked="" type="radio"/> 更新过的 <input type="radio"/> 定为基线的

文档修订历史

版本	日期	描述	修订者
V0.8	20171102	初稿 1. 增加了飞马柜台直接给终端返回接收申报应答, XXXXAccept() 方法	李华华
V0.10	20180201	1. 增加了原油铁矿石业务的货币质押解质接口 ReqCurrencyPledge () /OnRspCurrencyPledge ()。 2. 增加了可提金额等字段。 CUSTpFtdcRspInvestorField 、 CUSTpFtdcRspUserInvestorField : 增加 InvestorType (投资者类型) 字段; CUSTpFtdcReqForQuoteField 、 CUSTpFtdcInputQuoteField 、 CUSTpFtdcInputOrderField 结构、CUSTpFtdcOrderActionField 结构中增加 ClientID 字段; CUSTpFtdcInvestorAccountChangeField 、CUSTpFtdcRspInvestorAccountField 结构中增加了 Currency (币种)、TradingStatus (账户交易状态)、MoneyStatus (帐户资金状态)、PreCurrencyPledge (上日货币质押金额)、CurrencyPledgeIn (货币质入金额)、CurrencyPledgeOut (货币质出金额)、CurrencyExchangeIn (外币汇入金额)、CurrencyExchangeOut (外币汇出金额)、FrozenMoney (冻结资金)、MinGuarantee (保底金额)、WithdrawRatioAllowed (可提比例)、DeliveryMargin (交割保证金)、WithdrawAllowed (可提资金) 字段; 3. 修改了消息确认接口 ReqMsgNotifyConfirm () /OnRspMsgNotifyConfirm ()。 4. 增加了对账单确认接口 ReqSettlementInfoConfirm () /OnRspSettlementInfoConfirm ()。 5. 修改了查询结算单确认接口 OnRspQrySettlementInfoConfirm ()。	李华华
V0.12	20180416	1. 兼容飞马多交易所版本的 API, 增加了接口 ReqUserDeposit () - 出入金, ReqTransferMoney () - 主次席出入金, ReqQrySystemTime () - 查询系统时间, 以及对应的 OnRspUserDeposit (), OnRspTransferMoney (), OnRtnInvestorAccountDeposit (), OnRtnUserDeposit (), OnRspQrySystemTime ()。只	李华华

		<p>是 API 接口支持，但飞马主柜台后台并不支持这些功能。</p> <p>2. 消息确认接口 ReqMsgNotifyConfirm () 及 OnRspMsgNotifyConfirm () 回报接口中增加消息号字段。</p> <p>3. 查询类接口增加 ClientId 字段。</p> <p>4. 资金表增加外币相关字段。</p>	
V0.14	20180607	<p>为增加铜期权行权对冲等新业务对 API 接口中的部分枚举值进行调整：</p> <p>1. 修改 TFtdcUstpHedgeFlagType 投机套保类型；</p> <p>2. 增加 TFtdcUstpArbiTypeType 策略类别类型；</p> <p>3. 增加 TFtdcUstpDeliveryFlagType 执行类型类型；</p> <p>4. 增加 TFtdcUstpCombActionStatusType 组合操作状态类型；</p> <p>5. 结构体中 CUstpFtdcInputMarginCombActionField 增加 3 个新字段；</p> <p>6. 接口未变；</p>	李华华
V1.01	20180921	<p>1. 增加错单查询接口 ReqQryOrderFailed () 和错单查询应答接口 OnRspQryOrderFailed (),</p> <p>2. 增加银期转账投资者资金账户变动通知 OnRtnTransferInvestorMoneyChanged ()</p> <p>3. 增加 TFtdcUstpTransferEventType 出入金通知类型</p> <p>4. 增加 TFtdcUstpTradeSyncSourceType 同步请求发起方类型</p> <p>5. TFtdcUstpDeliveryFlagType 期权执行类型增加枚举值</p> <p>6. 增加 TFtdcUstpReSendType 数据源类型</p>	李华华
V1.03	20181211	<p>1 增加查询最大可开/可平手数接口：OnRspQueryMaxOrderVolume () 和 ReqQueryMaxOrderVolume 方法</p>	李华华
V1.04	20190319	<p>1. 增加查询签约银行</p> <p>2. 修改消息通知推送，增加消息通知确认接口</p> <p>3. 修改投资者查询，投资者资金查询（浮盈浮亏算法），手续费率查询</p> <p>4. 穿透式监管相关接口：穿透监管客户端认证，穿透监管信息采集中继认证终端、穿透监管信息采集中继上传信息到飞马</p> <p>5. 大连六期相关接口：查询组合持仓明细、查询组合优惠接口</p> <p>6. 删除查询单腿持仓相关接口</p> <p>缺陷修正：</p> <p>1. 解决 API 实例启动时端口号冲突导致失败</p> <p>2. 解决订阅公共流异常问题，API 连接数扩展到 2048</p>	lihh
V1.05	20190724	<p>1. CUstpFtdcInvestorAccountChangeField 、CUstpFtdcRspInvestorAccountField 结构中增加了 2 个字段：ExchangeOutFrozenMoney（换汇出金冻结资金），ExchangeOpenFrozenMoney（换汇开仓冻结资金）</p> <p>2. 增加了 RegisterNameServer（注册飞马 Fens）方法</p> <p>3. 增加了 ReqProxyUserLogin（中继代理登录）方法</p>	Lihh
V1.05r1	20191008	ReqQueryXXX 方法补充必填字段说明	Lihh
V1.07	20191125	接口没有变化，有结构体字段修改	Lihh

V1.07r3	20200428	增加组合拆分接口、行权对冲设置接口、查询期权交易成本接口等； 增加总冻结资金、市值权益等字段； 增加浮动盈亏导致的资金、持仓变动通知接口等；	Lihh
V1.12	20210205	新增预埋单、条件单相关接口	
V1.12	20210205	新增报价查询、行权查询、历史转帐流水查询接口	
V1.12	20210205	删除用户出入金、主次席出入金相关接口	
V1.12	20210205	消息通知结构体增加消息主题属性	
V1.12	20210205	报单、报价、行权、组合申请、持仓查询等接口相关结构体增加客户号属性	
V1.12	20210205	报单操作接口相关结构体增加合约属性	
V1.12	20210205	报价、行权接口等相关结构体增加会话 ID 和前置 ID 属性	
V1.12	20210205	合约查询、合约状态通知、查询保证金优惠、查询标的物昨收盘价接口相关结构体增加产品类型属性	
V1.12	20210205	手续费查询接口相关结构体增加多空方向属性	
V1.12	20210205	资金变动通知结构体增加消息流水号、操作源、目的地、操作人属性	

此版本文档的正式核准

姓名	签字	日期

目录

1.	介绍	1
2.	体系结构	2
2.1	通讯模式	2
2.2	数据流	3
3.	接口模式	5
3.1	对话流、询价流和查询流编程接口	5
3.2	私有流编程接口	5
4.	运行模式	7
4.1	工作线程	7
4.2	本地文件	7
5.	业务和接口对照	8
6.	开发接口	13
6.1	通用规则	13
6.2	CUstpFtdcTraderSpi 接口	13
6.2.1	OnFrontConnected 方法	13
6.2.2	OnFrontDisconnected 方法	13
6.2.3	OnQryFrontConnected 方法	14
6.2.4	OnQryFrontDisconnected 方法	14
6.2.5	OnHeartBeatWarning 方法	14
6.2.6	OnPackageStart 方法	15
6.2.7	OnPackageEnd 方法	15
6.2.8	OnRspError 方法	15
6.2.9	OnRspUserLogin 方法	16
6.2.10	OnRspUserLogout 方法	17
6.2.11	OnRspUserPasswordUpdate 方法	18
6.2.12	OnRspSettlementInfoConfirm 方法	19
6.2.13	OnRspQuerySettlementInfo 方法	20
6.2.14	OnRspMsgNotifyConfirm 方法	21

6.2.15	OnRspOrderInsert 方法和 OnRspOrderInsertAccept 方法.....	22
6.2.16	OnRspOrderAction 方法和 OnRspOrderActionAccept 方法.....	24
6.2.17	OnRspQuoteInsert 方法和 OnRspQuoteInsertAccept 方法.....	25
6.2.18	OnRspQuoteAction 方法和 OnRspQuoteActionAccept 方法.....	27
6.2.19	OnRspForQuote 方法.....	28
6.2.20	OnRspMarginCombAction 方法.....	30
6.2.21	OnRtnTrade 方法.....	30
6.2.22	OnRtnOrder 方法.....	32
6.2.23	OnErrRtnOrderInsert 方法.....	35
6.2.24	OnErrRtnOrderAction 方法.....	37
6.2.25	OnRtnQuote 方法.....	38
6.2.26	OnErrRtnQuoteInsert 方法.....	40
6.2.27	OnErrRtnQuoteAction 方法.....	42
6.2.28	OnRtnForQuote 方法.....	43
6.2.29	OnRtnMarginCombAction 方法.....	44
6.2.30	OnRspQueryUserLogin 方法.....	44
6.2.31	OnRspQryOrder 方法.....	46
6.2.32	OnRspQryOrderFailed 方法.....	47
6.2.33	OnRspQryTrade 方法.....	49
6.2.34	OnRspQryUserInvestor 方法.....	52
6.2.35	OnRspQryTradingCode 方法.....	54
6.2.36	OnRspQryInvestorAccount 方法.....	55
6.2.37	OnRspQryExchange 方法.....	59
6.2.38	OnRspQryInstrument 方法.....	59
6.2.39	OnRspQryInvestorPosition 方法.....	62
6.2.40	OnRspQryInvestorFee 方法.....	65
6.2.41	OnRspQryInvestorMargin 方法.....	67
6.2.42	OnRspQryInvestorCombPosition 方法.....	68
6.2.43	OnRspQryInstrumentGroup 方法.....	70
6.2.44	OnRspQryClientMarginCombType 方法.....	71

6. 2. 45	OnRspExecOrderInsert 方法和 OnRspExecOrderInsertAccept 方法	72
6. 2. 46	OnRspExecOrderAction 方法和 OnRspExecOrderActionAccept 方法	73
6. 2. 47	OnRtnExecOrder 方法.....	74
6. 2. 48	OnErrRtnExecOrderInsert 方法.....	76
6. 2. 49	OnErrRtnExecOrderAction 方法.....	77
6. 2. 50	OnErrRtnExecSelfCloseInsert 方法.....	78
6. 2. 51	OnRspQrySettlementInfo 方法.....	80
6. 2. 52	OnRspTradingAccountPasswordUpdate 方法.....	81
6. 2. 53	OnRspQueryTransferSerialOffset 方法.....	82
6. 2. 54	OnRspQueryAccountRegister 方法.....	85
6. 2. 55	OnRspFromBankToFutureByFuture 方法.....	88
6. 2. 56	OnRtnFromBankToFutureByFuture 方法.....	91
6. 2. 57	OnRspFromFutureToBankByFuture 方法.....	94
6. 2. 58	OnRtnFromFutureToBankByFuture 方法.....	98
6. 2. 59	OnRtnFromFutureToBankByBank 方法.....	101
6. 2. 60	OnRtnFromBankToFutureByBank 方法.....	102
6. 2. 61	OnRspQueryBankAccountMoneyByFuture 方法.....	102
6. 2. 62	OnRtnQueryBankBalanceByFuture 方法.....	106
6. 2. 63	OnRtnOpenAccountByBank 方法.....	109
6. 2. 64	OnRtnCancelAccountByBank 方法.....	112
6. 2. 65	OnRspFromBankToFutureByBank 方法.....	112
6. 2. 66	OnRspFromFutureToBankByBank 方法.....	115
6. 2. 67	OnRtnFromBankToFutureByJSD.....	115
6. 2. 68	OnRtnFromFutureToBankByJSD.....	116
6. 2. 69	OnRtnTransferInvestorAccountChanged 方法.....	116
6. 2. 70	OnRspSignUpAccountByFuture 方法.....	116
6. 2. 71	OnRspCancelAccountByFuture 方法.....	119
6. 2. 72	OnRspQuerySignUpOrCancelAccountStatus 方法.....	120
6. 2. 73	OnRtnChangeAccountByBank 方法.....	122
6. 2. 74	OnRspOpenSimTradeAccount 方法.....	124

6. 2. 75	OnRspCheckOpenSimTradeAccount 方法.....	125
6. 2. 76	OnRspQryCFMMCTradingAccountKey 方法.....	126
6. 2. 77	OnRtnMsgNotify 方法.....	127
6. 2. 78	OnRtnClientPositionChange 方法.....	128
6. 2. 79	OnRtnClientPositionChangeMD 方法.....	131
6. 2. 80	OnRtnInvestorAccountChange 方法.....	132
6. 2. 81	OnRtnInvestorAccountChangeMD 方法.....	136
6. 2. 82	OnRspQryEnableRtnMoneyPositoinChange 方法.....	138
6. 2. 83	OnRspQueryHisOrder 方法.....	138
6. 2. 84	OnRspQueryHisTrade 方法.....	139
6. 2. 85	OnRtnInvestorAccountDeposit 方法.....	142
6. 2. 86	OnRspCurrencyPledge 方法.....	143
6. 2. 87	OnRtnTransferInvestorMoneyChanged 方法.....	144
6. 2. 88	OnRspQueryMaxOrderVolume 方法.....	145
6. 2. 89	OnRtnFlowMessageCancel 方法（未启用）	147
6. 2. 90	OnErrRtnFromFutureToBankByBank 方法（未启用）	147
6. 2. 91	OnErrRtnFromBankToFutureByFuture 方法（未启用）	150
6. 2. 92	OnErrRtnQueryBankBalanceByFuture 方法（未启用）	153
6. 2. 93	OnRtnRepealFromBankToFutureByBank 方法（未启用）	156
6. 2. 94	OnRtnRepealFromFutureToBankByBank 方法（未启用）	159
6. 2. 95	OnRtnRepealFromBankToFutureByFutureManual 方法（未启用） ..	162
6. 2. 96	OnErrRtnRepealBankToFutureByFutureManual 方法（未启用）	166
6. 2. 97	OnRtnRepealFromFutureToBankByFutureManual 方法（未启用） ..	169
6. 2. 98	OnErrRtnRepealFutureToBankByFutureManual 方法（未启用）	172
6. 2. 99	OnRtnRepealFromBankToFutureByFuture 方法（未启用）	176
6. 2. 100	OnRtnRepealFromFutureToBankByFuture 方法（未启用）	179
6. 2. 101	OnRspQueryContractBank 方法.....	183
6. 2. 102	OnRspQryInvestorPositionDetail 方法.....	183
6. 2. 103	OnRspQryInvestorPositionCombDetail 方法.....	186
6. 2. 104	OnRspQryMarginPrefParam 方法.....	188

6.2.105	OnRspDSUserCertification 方法.....	190
6.2.106	OnRspDSProxySubmitInfo 方法.....	191
6.2.107	OnRspQryExchangRate 方法（未启用）	191
6.2.108	OnRspQryEWarrantOffset 方法（未启用）	192
6.2.109	OnRspQryComplianceParam 方法（未启用）	193
6.2.110	OnRspQrySystemTime 方法（主柜台不支持）	194
6.2.111	OnErrRtnForQuote 方法.....	195
6.2.112	OnErrRtnMarginCombCancel 方法.....	195
6.2.113	OnErrRtnMarginCombAction 方法.....	196
6.2.114	OnRtnExecSelfClose 方法.....	198
6.2.115	OnRtnProductStatus 方法.....	199
6.2.116	OnRspQrySpotClosePrice 方法.....	200
6.2.117	OnRspQryOptionTradeCost 方法.....	201
6.2.118	OnRspQryProduct 方法.....	202
6.2.119	OnRspQryCombAction 方法.....	203
6.2.120	OnRspMarginCombCancelAccept 方法.....	204
6.2.121	OnRspMarginCombCancel 方法和 OnRspMarginCombCancelAccept 方法 204	
6.2.122	OnRspExecSelfClose 方法.....	205
6.2.123	OnRspExecSelfCloseInsert 方法.....	206
6.2.124	OnRspExecSelfCloseAction 方法.....	208
6.2.125	OnErrRtnExecSelfCloseAction 方法.....	210
6.2.126	OnRspExecSelfCloseInsertAccept 方法.....	211
6.2.127	OnRspExecSelfCloseActionAccept 方法.....	212
6.2.128	OnRspMarginCombActionAccept 方法.....	212
6.2.129	OnRspChangeCenter 方法.....	212
6.2.130	OnRspQryTradeCenter 方法.....	213
6.2.131	OnRspParkedOrderInsert 方法.....	214
6.2.132	OnRspParkedOrderAction 方法.....	216
6.2.133	OnRspQryParkedOrder 方法.....	217

6.2.134	OnRtnParkedOrder 方法.....	219
6.2.135	OnRspConditionalOrderInsert 方法.....	221
6.2.136	OnRspConditionalOrderAction 方法.....	223
6.2.137	OnRspQryConditionalOrder 方法.....	224
6.2.138	OnRtnConditionalOrder 方法.....	227
6.2.139	OnRspQryQuote 方法.....	229
6.2.140	OnRspQryExecOrder 方法.....	232
6.2.141	OnRspQueryHisTransferSerial 方法.....	234
6.3	CUSTPFtdcTraderApi 接口	236
6.3.1	CreateFtdcTraderApi 方法	236
6.3.2	GetVersion 方法	236
6.3.3	Release 方法	237
6.3.4	Init 方法	237
6.3.5	Join 方法	237
6.3.6	GetTradingDay 方法	237
6.3.7	RegisterSpi 方法	237
6.3.8	RegisterFront 方法	238
6.3.9	RegisterQryFront 方法	238
6.3.1	RegisterDSPProxyUserCert 方法	238
6.3.2	RegisterNameServer 方法	239
6.3.3	SubscribePrivateTopic 方法	240
6.3.4	SubscribePublicTopic 方法	240
6.3.5	SubscribeForQuote 方法	240
6.3.6	SetHeartbeatTimeout 方法	241
6.3.7	ReqUserLogin 方法	241
6.3.8	ReqUserLogout 方法	242
6.3.9	ReqUserPasswordUpdate 方法	243
6.3.10	ReqSettlementInfoConfirm 方法.....	243
6.3.11	ReqQuerySettlementInfo 方法.....	244
6.3.12	ReqMsgNotifyConfirm 方法.....	244

6.3.13	ReqOrderInsert 方法.....	245
6.3.14	ReqOrderAction 方法.....	250
6.3.15	ReqQuoteInsert 方法.....	252
6.3.16	ReqQuoteAction 方法.....	254
6.3.17	ReqForQuote 方法.....	255
6.3.18	ReqMarginCombAction 方法.....	256
6.3.19	ReqQryOrder 方法.....	258
6.3.20	ReqQryOrderFailed 方法.....	259
6.3.21	ReqQryTrade 方法.....	260
6.3.22	ReqQryUserInvestor 方法.....	260
6.3.23	ReqQryTradingCode 方法.....	261
6.3.24	ReqQryInvestorAccount 方法.....	261
6.3.25	ReqQryInstrument 方法.....	262
6.3.26	ReqQryExchange 方法.....	262
6.3.27	ReqQryInvestorPosition 方法.....	263
6.3.28	ReqQryInvestorFee 方法.....	263
6.3.29	ReqQryInvestorMargin 方法.....	264
6.3.30	ReqQryInvestorCombPosition 方法.....	265
6.3.31	ReqQryInstrumentGroup 方法.....	265
6.3.32	ReqQryClientMarginCombType 方法.....	266
6.3.33	ReqExecOrderInsert 方法.....	267
6.3.34	ReqExecOrderAction 方法.....	269
6.3.35	ReqTradingAccountPasswordUpdate 方法.....	270
6.3.36	ReqQueryTransferSeria 方法.....	271
6.3.37	ReqQueryAccountRgister 方法.....	272
6.3.38	ReqFromBankToFutureByFuture 方法.....	273
6.3.39	ReqFromFutureToBankByFuture 方法.....	277
6.3.40	ReqQueryBankAccountMoneyByFuture 方法.....	277
6.3.41	ReqSignUpAccountByFuture 方法.....	282
6.3.42	ReqCancleAccountByFuture 方法.....	286

6.3.43	ReqQuerySignUpOrCancleAccountStatus 方法.....	286
6.3.44	ReqChangeAccountByFuture 方法.....	287
6.3.45	ReqOpenSimTradeAccount 方法.....	292
6.3.46	ReqCheckOpenSimTradeAccount 方法.....	292
6.3.47	ReqCFMMCTradingAccountKey 方法.....	293
6.3.48	ReqQryEnableRtnMoneyPositoinChange 方法.....	294
6.3.49	ReqQueryHisOrder 方法.....	294
6.3.50	ReqQueryHisTrade 方法.....	295
6.3.51	ReqCurrencyPledge 方法.....	296
6.3.52	ReqSettlementInfoConfirm 方法.....	297
6.3.53	ReqQueryMaxOrderVolume 方法.....	297
6.3.54	ReqQueryContractBank 方法.....	298
6.3.55	ReqQryInvestorPositionDetail 方法.....	299
6.3.56	ReqQryInvestorPositionCombDetail 方法.....	299
6.3.57	ReqQryMarginPrefParam 方法.....	300
6.3.58	ReqDSUserCertification 方法.....	301
6.3.59	ReqDSProxySubmitInfo 方法.....	301
6.3.60	ReqQrySettlementInfoConfirm 方法.....	302
6.3.61	ReqQryExchangeRate 方法（未启用）	303
6.3.62	ReqQrySystemTime 方法（未启用）	303
6.3.63	ReqQrySettlementInfo 方法（未启用）	304
6.3.64	ReqQryEWarrantOffset 方法（未启用）	304
6.3.65	SubscribeUserTopic 方法（未启用）	304
6.3.66	ReqQryComplianceParam 方法（不启用）	304
6.3.67	ReqProxyUserLogin 方法.....	304
6.3.68	ReqQrySystemTime 方法（主柜台不支持）	307
6.3.69	ReqQrySpotPrevClosePrice 方法.....	307
6.3.70	ReqQryOptionTradeCost 方法.....	308
6.3.71	ReqQryProduct 方法.....	309
6.3.72	ReqQryInputCombAction 方法.....	309

6.3.73	ReqQryMarginCombCancel 方法.....	310
6.3.74	ReqMarginCombCancel 方法.....	311
6.3.75	ReqQryExecSelfClose 方法.....	312
6.3.76	ReqExecSelfCloseInsert 方法.....	313
6.3.77	ReqExecSelfCloseAction 方法.....	314
6.3.78	ReqChangeCenter 方法.....	315
6.3.79	ReqQryTradeCenter 方法.....	316
6.3.80	ReqParkedOrderInsert 方法.....	317
6.3.81	ReqParkedOrderAction 方法.....	319
6.3.82	ReqQryParkedOrder 方法.....	320
6.3.83	ReqConditionalOrderInsert 方法.....	321
6.3.84	ReqConditionalOrderAction 方法.....	323
6.3.85	ReqQryConditionalOrder 方法.....	324
6.3.86	ReqQryQuote 方法.....	325
6.3.87	ReqQryExecOrder 方法.....	325
6.3.88	ReqQueryHisTransferSerial 方法.....	326
7.	开发实例.....	327

1. 介绍

飞马平台 API 是一个基于 C++ 的类库，通过使用和扩展类库提供的接口来实现相关交易功能，包括报单录入、报单撤销、报单查询、成交单查询、投资者查询、投资者持仓查询、合约查询、交易日获取等。该类库包含以下 7 个文件：

文件名	版本	文件大小	文件描述
FtdcTraderApi.h			交易接口头文件
FtdcUserApiDataType.h			定义了 API 所需的一系列数据类型的头文件
FtdcUserApiStruct.h			定义了一系列业务相关的数据结构的头文件
USTPtraderapiFM.dll			动态链接库二进制文件
USTPtraderapiFM.lib			导入库文件
USTPmduserapiFM.dll			动态链接库二进制文件
USTPmduserapiFM.lib			导入库文件

注：库文件名称与多交易所版本（USTPtraderapiAF）动态库相近，使用时请注意区分不同版本。

支持 MS VC 6.0，MS VC.NET 2003 编译器。需要打开多线程编译选项/MT。

2. 体系结构

交易员 API 使用建立在 TCP 协议之上 FTD 协议与飞马平台进行通讯，飞马平台负责投资者的交易业务处理。

2.1 通讯模式

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

FTD 涉及的通讯模式共有三种：

- 对话通讯模式
- 私有通讯模式
- 广播通讯模式

对话通讯模式是指由用户端程序主动发起的通讯请求。该请求被飞马服务端接收和处理，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。

私有通讯模式是指飞马服务端主动向某个特定的用户发出的信息，例如委托回报、成交回报等。

广播通讯模式是指飞马服务端主动向系统中的所有用户都发出相同的信息，例如通知信息、市场公共信息等。

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。

无论哪种通讯模式，其通讯过程都如图 1 所示：

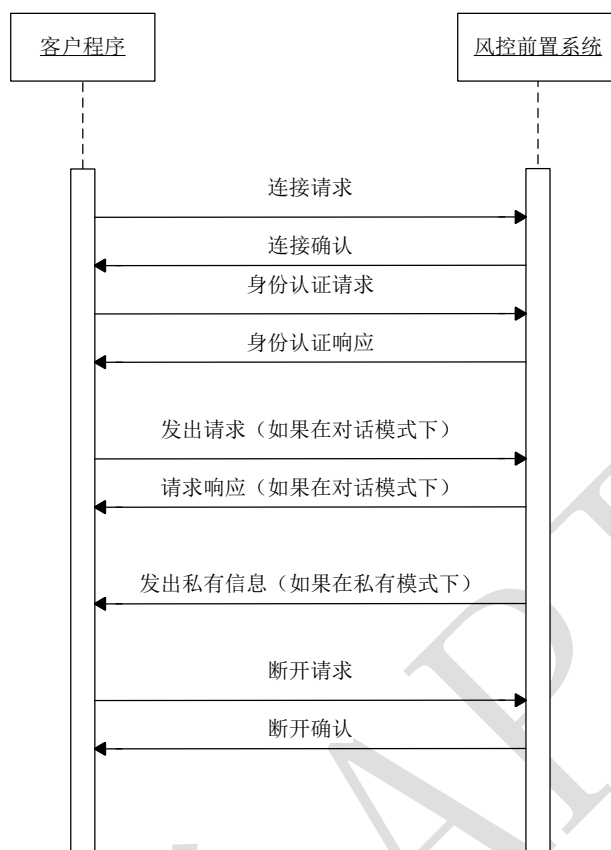


图 1-各种通讯模式的工作过程

本接口暂时没有使用广播通讯方式。

2.2 数据流

飞马平台支持对话通讯模式、私有通讯模式。

对话通讯模式下支持对话数据流、查询数据流和询价数据流：

对话数据流是一个双向数据流，客户端程序发送交易请求，飞马系统对该请求进行反馈应答。飞马系统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，客户端程序发送查询请求，飞马系统对该请求进行反馈应答。飞马系统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。

询价数据流是一个双向数据流，客户端程序发送查询请求，飞马系统对该请求进行反馈应答。飞马系统不维护询价流的状态。系统故障时，询价数据流会重置，通讯途中的数据可能会丢失。

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，由飞马系统发向客户端程序，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，飞马系统维护每个客户端用户的私有流，在一个交易日内，客户端断线后恢复连接时，可以请求飞马系统发送指定序号之后的私有流数据。私有数据流向客户端提供报单状态报告、成交回报等信息。

3. 接口模式

交易员 API 提供了二类接口,分别为 CUSTPFtdcTraderApi 和 CUSTPFtdcTraderSpi。这两个接口对 FTD 协议进行了封装,方便客户端应用程序的开发。

客户端应用程序可以通过 CUSTPFtdcTraderApi 发出操作请求,需要通过继承 CUSTPFtdcTraderSpi 并重载回调函数来处理后台服务的响应。

3.1 对话流、询价流和查询流编程接口

通过对话流进行通讯的编程接口通常如下:

```
请求: int CUSTPFtdcTraderApi::ReqXXX(  
    CUSTPFtdcXXXField *pReqXXX,  
    int nRequestID)  
响应: void CUSTPFtdcTraderSpi::OnRspXXX(  
    CUSTPFtdcXXXField *pRspXXX,  
    CUSTPFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

其中请求接口第一个参数为请求的内容,不能为空。

第二个参数为请求号。请求号由客户端应用程序负责维护,正常情况下每个请求的请求号是不能重复的。在接收飞马平台的响应时,可以得到当时发出请求时填写的请求号,从而可以将响应与请求对应起来。

当收到后台服务应答时, CUSTPFtdcTraderSpi 的回调函数会被调用。响应数据可以使一个或多个,如果响应数据不止一个,则回调函数会被多次调用。

回调函数的第一个参数为响应的具体数据,如果出错或没有结果有可能为 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。在发生多次回调时,除了第一次回调,其它的回调该参数都可能为 NULL。

第三个参数为请求号,即原来发出请求时填写的请求号。

第四个参数为响应结束标志,表明是否是本次响应的最后一次回调。

3.2 私有流编程接口

私有流中的数据中用户的私有信息,包括报单回报、成交回报等。

通过私有流接收回报的编程接口通常如下:

```
void CUSTPFtdcTraderSpi::OnRtnXXX(CUSTPFtdcXXXField *pXXX)
```

或

```
void CUSTPFtdcTraderSpi::OnErrRtnXXX(CUSTPFtdcXXXField *pXXX,  
    CUSTPFtdcRspInfoField *pRspInfo)
```

当收到飞马平台通过私有流发布的回报数据时，CUSTPFtdcTraderSpi 的回调函数会被调用。回调函数的参数为回报的具体内容。

4. 运行模式

4.1 工作线程

客户端应用程序至少由两个线程组成，一个是应用程序主线程，一个是交易员 API 工作线程。应用程序与交易系统的通讯是由 API 工作线程驱动的。

CUSTPFtdcTraderApi 提供的接口是线程安全的，可以有多个应用程序线程同时发出请求。

CUSTPFtdcTraderSpi 提供的接口回调是由 API 工作线程驱动，通过实现 SPI 中的接口方法，可以从飞马平台收取所需数据。

如果重载的某个回调函数阻塞，则等于阻塞了 API 工作线程，API 与交易系统的通讯会停止。因此，在 CUSTPFtdcTraderSpi 派生类的回调函数中，通常应迅速返回，可以利用将数据放入缓冲区或通过 Windows 的消息机制来实现。

4.2 本地文件

交易员 API 在运行过程中，会将一些数据写入本地文件中。调用 CreateFtdcTraderApi 函数，可以传递一个参数，指明存贮本地文件的路径。该路径必须在运行前已创建好。本地文件的扩展名都是“.con”。

5. 业务和接口对照

业务类型	业务	请求接口	响应接口	数据流
登录	登录	CUSTPFtdcTraderApi:: ReqUserLogin	CUSTPFtdcTraderSpi::OnRspUserLogin	对话流
	登出	CUSTPFtdcTraderApi::ReqUserLogout	CUSTPFtdcTraderSpi::OnRspUserLogout	对话流
	修改用户口令	CUSTPFtdcTraderApi::ReqUserPasswordUpdate	CUSTPFtdcTraderSpi::OnRspUserPasswordUpdate	对话流
报单	报单录入	CUSTPFtdcTraderApi::ReqOrderInsert	CUSTPFtdcTraderSpi::OnRspOrderInsert	对话流
	报单操作	CUSTPFtdcTraderApi::ReqOrderAction	CUSTPFtdcTraderSpi::OnRspOrderAction	对话流
报价	报价录入	CUSTPFtdcTraderApi::ReqQuoteInsert	CUSTPFtdcTraderSpi::OnRspQuoteInsert	对话流
	报价操作	CUSTPFtdcTraderApi::ReqQuoteAction	CUSTPFtdcTraderSpi::OnRspQuoteAction	对话流
行权	行权录入	CUSTPFtdcTraderApi:: ReqExecOrderInsert	CUSTPFtdcTraderSpi::OnRspExecOrderInsert	对话流
	行权操作	CUSTPFtdcTraderApi:: ReqExecOrderAction	CUSTPFtdcTraderSpi::OnRspExecOrderAction	对话流
询价	询价请求	CUSTPFtdcTraderApi::ReqForQuote	CUSTPFtdcTraderSpi:: OnRspForQuote	对话流
对账单	对账单确认请求	CUSTPFtdcTraderApi::ReqNotifyConfirm	CUSTPFtdcTraderSpi:: OnRspSettlementInfoConfirm	对话流

组保	组合拆分请求	CUSTPFtdcTraderApi::ReqMarginCombAction	CUSTPFtdcTraderSpi::OnRspMarginCombAction	对话流
出入金	用户出入金请求	CUSTPFtdcTraderApi::ReqUserDeposit	CUSTPFtdcTraderSpi::OnRspUserDeposit	对话流
私有回报	成交回报	N/A	CUSTPFtdcTraderSpi::OnRtnTrade	私有流
	报单回报	N/A	CUSTPFtdcTraderSpi::OnRtnOrder	私有流
	柜台出入金回报	N/A	CUSTPFtdcTraderSpi::OnRtnInvestorAccountDeposit	私有流
	报单录入错误回报	N/A	CUSTPFtdcTraderSpi::OnErrRtnOrderInsert	私有流
	报单操作错误回报	N/A	CUSTPFtdcTraderSpi::OnErrRtnOrderAction	私有流
	报价回报	N/A	CUSTPFtdcTraderSpi::OnRtnQuote	私有流
	报价录入错误回报	N/A	CUSTPFtdcTraderSpi::OnErrRtnQuoteInsert	私有流
	行权通知	N/A	CUSTPFtdcTraderSpi::OnRtnExecOrder	私有流
	行权录入错误回报	N/A	CUSTPFtdcTraderSpi::OnErrRtnExecOrderInsert	私有流
	行权操作错误回报	N/A	CUSTPFtdcTraderSpi::OnErrRtnExecOrderAction	私有流
	用户出入金回报	N/A	CUSTPFtdcTraderSpi::OnRtnUserDeposit	私有流
组保回报	组合拆分申请回报	N/A	CUSTPFtdcTraderSpi::OnRtnMarginCombAction	私有流

	组合规则回报	N/A	CUSTPFtdcTraderSpi::OnRtnMarginCombanationLe g	私有流
询价回报	询价回报	N/A	CUSTPFtdcTraderSpi::OnRtnForQuote	询价流
查询	报单查询	CUSTPFtdcTraderApi::ReqQryOrder	CUSTPFtdcTraderSpi::OnRspQryOrder	查询流
	成交查询	CUSTPFtdcTraderApi::ReqQryTrade	CUSTPFtdcTraderSpi::OnRspQryTrade	查询流
	合约查询	CUSTPFtdcTraderApi::ReqQryInstrument	CUSTPFtdcTraderSpi::OnRspQryInstrument	查询流
	可用投资者查询	CUSTPFtdcTraderApi::ReqQryUserInvestor	CUSTPFtdcTraderSpi::OnRspQryUserInvestor	查询流
	资金账户查询	CUSTPFtdcTraderApi::ReqQryInvestorAccount	CUSTPFtdcTraderSpi::OnRspQryInvestorAccount	查询流
	交易编码查询	CUSTPFtdcTraderApi::ReqQryTradingCode	CUSTPFtdcTraderSpi::OnRspQryTradingCode	查询流
	交易所查询	CUSTPFtdcTraderApi::ReqQryExchange	CUSTPFtdcTraderSpi::OnRspQryExchange	查询流
	投资者持仓查询	CUSTPFtdcTraderApi::ReqQryInvestorPositio n	CUSTPFtdcTraderSpi::OnRspQryInvestorPosition	查询流
	投资者组合持仓查 询	CUSTPFtdcTraderApi::ReqQryInvestorCombPos ition	CUSTPFtdcTraderSpi::OnRspQryInvestorCombPosi tion	查询流
	合规参数查询	CUSTPFtdcTraderApi:: ReqQryComplianceParam	CUSTPFtdcTraderSpi:: OnRspQryComplianceParam	查询流
	交叉汇率查询	CUSTPFtdcTraderApi:: ReqQryExchangeRate	CUSTPFtdcTraderSpi:: OnRspQryExchangeRate	查询流
	手续费率查询	CUSTPFtdcTraderApi:: ReqQryInvestorFee	CUSTPFtdcTraderSpi:: OnRspQryInvestorFee	查询流

	保证金率查询	CUSTPFtdcTraderApi:: ReqQryInvestorMargin	CUSTPFtdcTraderSpi:: OnRspQryInvestorMargin	查询流
	合约组信息查询	CUSTPFtdcTraderApi:: ReqQryInstrumentGroup	CUSTPFtdcTraderSpi::	查询流
	组合保证金类型查询	CUSTPFtdcTraderApi:: ReqQryClientMarginCombType	CUSTPFtdcTraderSpi::	查询流
公共信息	产品状态	N/A	CUSTPFtdcTraderSpi::OnRtnProductStatus	公共流
柜台应答	报单录入	CUSTPFtdcTraderApi::ReqOrderInsert	CUSTPFtdcTraderSpi::OnRspOrderInsertAccept	对话流
	报单操作	CUSTPFtdcTraderApi::ReqOrderAction	CUSTPFtdcTraderSpi::OnRspOrderActionAccept	对话流
	报价录入	CUSTPFtdcTraderApi::ReqQuoteInsert	CUSTPFtdcTraderSpi::OnRspQuoteInsertAccept	对话流
	报价操作	CUSTPFtdcTraderApi::ReqQuoteAction	CUSTPFtdcTraderSpi::OnRspQuoteActionAccept	对话流
	询价请求	CUSTPFtdcTraderApi::ReqForQuote	CUSTPFtdcTraderSpi::OnRspForQuoteAccept	对话流
	行权录入	CUSTPFtdcTraderApi:: ReqExecOrderInsert	CUSTPFtdcTraderSpi::OnRspExecOrderInsertAccept	对话流
	行权操作	CUSTPFtdcTraderApi:: ReqExecOrderAction	CUSTPFtdcTraderSpi::OnRspExecOrderActionAccept	对话流

交易接口和私有流接口会有相互关联，如用户报单录入 ReqOrderInsert，用户马上会收到飞马的柜台级的报单响应 OnRspOrderInsertAccept，说明飞马系统已经收到报单。报单进入飞马系统并通过飞马检查校验后会被报送到交易所，飞马收到交易所的应答后会推送给客户一个交易所报单响应 OnRspOrderInsert，如果报单的交易状态发生变化，飞马收到报单回报后会推送给用户一个报单回报 OnRtnOrder。如果报单被撮合(部分)成交，用户就会收到成交回报 OnRtnTrade。其中，一个用户(User)的报单回报和成交回报也会被所属投

资者下其他的用户接收到。

飞马 API

6. 开发接口

6.1 通用规则

客户端和飞马平台的通讯过程分为 2 个阶段：初始化阶段和功能调用阶段。

在初始化阶段，程序必须完成如下步骤（具体代码请参考开发实例）：

- 1, 产生一个 CUstpFtdcTraderApi 实例
- 2, 产生一个事件处理的实例
- 3, 注册一个事件处理的实例
- 4, 订阅私有流
- 5, 订阅公共流
- 6, 订阅询价流
- 7, 设置飞马平台服务的地址

在功能调用阶段, 程序可以任意调用交易接口中的请求方法, 如 ReqOrderInsert 等。同时按照需要响应回调接口中的应答方法。

其他注意事项：

- 1, API 请求的输入参数不能为 NULL。
- 2, API 请求的返回参数, 0 表示正确, 其他表示错误, 详细错误编码请查表。

6.2 CUstpFtdcTraderSpi 接口

CUstpFtdcTraderSpi 实现了事件通知接口。用户必需派生 CUstpFtdcTraderSpi 接口, 编写事件处理方法来处理感兴趣的事件。

6.2.1 OnFrontConnected 方法

当客户端与飞马平台建立起通信连接时（还未登录前），该方法被调用。

函数原形：

```
void OnFrontConnected();
```

本方法在完成初始化后调用，可以在其中完成用户登录任务。

6.2.2 OnFrontDisconnected 方法

当客户端与飞马平台通信连接断开时，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。客户端会自动重连地址。

函数原形:

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

参数:

nReason: 连接断开原因

0x1001 网络读失败

0x1002 网络写失败

0x2001 接收心跳超时

0x2002 发送心跳失败

0x2003 收到错误报文

6.2.3 OnQryFrontConnected 方法

当客户端与飞马平台查询前置建立起通信连接时（还未登录前），该方法被调用。

函数原形:

```
void OnQryFrontConnected ();
```

本方法在完成初始化后调用，可以在其中完成用户登录任务。

6.2.4 OnQryFrontDisconnected 方法

当客户端与飞马平台通信连接断开时，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。客户端会自动重连地址。

函数原形:

```
void OnQryFrontDisconnected (int nReason);
```

参数:

nReason: 连接断开原因

0x1001 网络读失败

0x1002 网络写失败

0x2001 接收心跳超时

0x2002 发送心跳失败

0x2003 收到错误报文

6.2.5 OnHeartBeatWarning 方法

心跳超时警告。当长时间未收到报文时，该方法被调用。

函数原形:

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

参数:

nTimeLapse: 距离上次接收报文的时间, 单位: 秒

6.2.6 OnPackageStart 方法

报文回调开始通知。当 API 收到一个报文后, 首先调用本方法, 然后是各数据域的回调, 最后是报文回调结束通知。

函数原形:

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

参数:

nTopicID: 主题代码 (如私有流、公共流、行情流等)

nSequenceNo: 报文序号

6.2.7 OnPackageEnd 方法

报文回调结束通知。当 API 收到一个报文后, 首先调用报文回调开始通知, 然后是各数据域的回调, 最后调用本方法。

函数原形:

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

参数:

nTopicID: 主题代码 (如私有流、公共流、行情流等)

nSequenceNo: 报文序号

6.2.8 OnRspError 方法

针对用户请求的出错通知。

函数原形:

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

参数:

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUSTPFTDCRSPINFOFIELD  
{  
    ///错误代码  
    TUSTPFTDCERRORIDTYPE ErrorID;  
    ///错误信息  
    TUSTPFTDCERRORMSGTYPE ErrorMessage;
```

```
};
```

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

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

6.2.9 OnRspUserLogin 方法

当客户端发出登录请求之后, 飞马平台返回响应时, 该方法会被调用, 通知客户端登录是否成功。

函数原形:

```
void OnRspUserLogin(  
    CUstpFtdcRspUserLoginField *pRspUserLogin,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

pRspUserLogin: 返回用户登录信息的地址。

用户登录响应信息结构:

```
struct CUstpFtdcRspUserLoginField  
{  
    ///交易日  
    TUstpFtdcDateType    TradingDay;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType    BrokerID;  
    ///交易用户代码  
    TUstpFtdcUserIDType    UserID;  
    ///登录成功时间  
    TUstpFtdcTimeType    LoginTime;  
    ///登录成功时的交易所时间  
    TUstpFtdcTimeType    ExchangeTime;  
    ///用户最大本地报单号  
    TUstpFtdcUserOrderLocalIDType    MaxOrderLocalID;  
    ///交易系统名称  
    TUstpFtdcTradingSystemNameType    TradingSystemName;  
    ///数据中心代码  
    TUstpFtdcDataCenterIDType    DataCenterID;  
    ///会员私有流当前长度  
    TUstpFtdcSequenceNoType    PrivateFlowSize;  
    ///交易员私有流当前长度
```

```

TUstpFtdcSequenceNoType UserFlowSize;
///业务发生日期
TUstpFtdcDateType ActionDay;
///飞马后台版本号
TUstpFtdcFemasVersionType FemasVersion;
///飞马生命周期号
TUstpFtdcFemasLifeCycleType FemasLifeCycle;
/// 会话编号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
};

```

pRspInfo: 返回用户响应信息的地址。特别注意在有连续的成功的响应数据时，中间有可能返回 NULL，但第一次不会，以下同。错误代码为 0 时，表示操作成功，以下同。

响应信息结构：

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.10 OnRspUserLogout 方法

当客户端发出退出请求之后，飞马平台返回响应时，该方法会被调用，通知客户端退出是否成功。

函数原形：

```

void OnRspUserLogout(
    CUstpFtdcRspUserLogoutField *pRspUserLogout,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pRspUserLogout: 返回用户退出信息的地址。

用户登出信息结构：

```

struct CUstpFtdcRspUserLogoutField
{

```

```

    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易用户代码
    TUstpFtdcUserIDType UserID;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.11 OnRspUserPasswordUpdate 方法

用户密码修改应答。当客户端发出用户密码修改指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```

void OnRspUserPasswordUpdate(
    CUstpFtdcUserPasswordUpdateField *pUserPasswordUpdate,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

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

用户密码修改结构:

```

struct CUstpFtdcUserPasswordUpdateField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易用户代码
    TUstpFtdcUserIDType UserID;
    ///旧密码
    TUstpFtdcPasswordType OldPassword;
    ///新密码
    TUstpFtdcPasswordType NewPassword;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.12 OnRspSettlementInfoConfirm 方法

请求对账单确认响应。

函数原形:

```
void OnRspSettlementInfoConfirm(
    CUstpFtdcSettlementInfoConfirmField *pSettlementInfoConfirm,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pSettlementInfoConfirm: 指向对账单确认应答结构的地址

对账单确认应答结构:

```
struct CUstpFtdcSettlementInfoConfirmField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///软件产品和版本号
    TUstpFtdcProductInfoType UserProductInfo;
    ///确认日期
    TUstpFtdcDateType ConfirmDate;
    ///确认时间
    TUstpFtdcTimeType ConfirmTime;
    ///IP 地址
    TUstpFtdcIPAddressType IPAddress;
    ///Mac 地址
    TUstpFtdcMacAddressType MacAddress;
};
```


pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回查询前置登陆请求的 ID, 该 ID 由用户在查询前置登陆时指定。

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

6.2.13 OnRspQuerySettlementInfo 方法

请求查询对账单确认响应。

函数原型:

```
void OnRspQuerySettlementInfo (
    CUSTpFtdcSettlementRspField *pSettlementRsp,
    CUSTpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};
```

参数:

pSettlementInfoConfirm: 指向对账单确认结构地址。

```
struct CUSTpFtdcSettlementRspField
{
    ///交易日
    TUstpFtdcDateType TradingDay;
    ///结算编号
    TUstpFtdcSettlementIDType SettlementID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///序号
    TUstpFtdcSequenceNoType SequenceNo;
    ///消息内容
    TUstpFtdcLogStrType Content;
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUSTpFtdcRspInfoField
{
    ///错误代码
    TustpFtdcErrorIDType ErrorID;
    ///错误信息
    TustpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回查询前置登陆请求的 ID, 该 ID 由用户在查询前置登陆时指定。

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

6.2.14 OnRspMsgNotifyConfirm 方法

请求消息确认响应。

函数原形:

```
void OnRspMsgNotifyConfirm(CUSTpFtdcMsgNotifyConfirmField *pMsgNotifyConfirm,
    CUSTpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};
```

参数:

pMsgNotifyConfirm 指向消息确认结构地址消息。

///消息确认响应

```
struct CUSTpFtdcMsgNotifyConfirmField
{
    ///经纪公司编号
    TustpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TustpFtdcInvestorIDType InvestorID;
    ///消息号
    TustpFtdcMsgNoType MsgNumber;
    ///确认日期
    TustpFtdcDateType ConfirmDate;
    ///确认时间
    TustpFtdcTimeType ConfirmTime;
    ///IP 地址
    TustpFtdcIPAddressType IPAddress;
    ///Mac 地址
    TustpFtdcMacAddressType MacAddress;
```

```
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回查询前置登陆请求的 ID, 该 ID 由用户在前置登陆时指定。

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

6.2.15 OnRspOrderInsert 方法和 OnRspOrderInsertAccept 方法

报单录入应答。当客户端发出过报单录入指令后, 飞马平台进行检查后会调用 OnRspOrderInsertAccept 方法, 通知用户飞马已经收到该笔申报, 随即发送到交易所并得到交易所应答后, OnRspOrderInsert 方法会被调用。

函数原形:

```
void OnRspOrderInsert(
    CUstpFtdcInputOrderField *pInputOrder,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pInputOrder: 指向报单录入结构的地址, 包含了提交报单录入时的输入数据, 和后台返回的报单编号。

输入报单结构:

```
struct CUstpFtdcInputOrderField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///指定下单席位编号 (取值范围[1-N], N 为可用席位数目, 超出范围的随机分配席位)
```

```
TUstpFtdcSeatNoType SeatNo;
///合约代码/套利组合合约号
TUstpFtdcInstrumentIDType InstrumentID;
///用户本地报单号
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
///报单类型
TUstpFtdcOrderPriceTypeType OrderPriceType;
///买卖方向
TUstpFtdcDirectionType Direction;
///开平标志
TUstpFtdcOffsetFlagType OffsetFlag;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///价格
TUstpFtdcPriceType LimitPrice;
///数量
TUstpFtdcVolumeType Volume;
///有效期类型
TUstpFtdcTimeConditionType TimeCondition;
///GTD 日期（暂不支持，保留域）
TUstpFtdcDateType GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType VolumeCondition;
///最小成交量
TUstpFtdcVolumeType MinVolume;
///止损价/止盈价
TUstpFtdcPriceType StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志（暂不支持，保留域）
TUstpFtdcBoolType IsAutoSuspend;
///业务单元（暂不支持，保留域）
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域 64 字节
TUstpFtdcCustomType UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType ActionDay;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///客户号
TUstpFtdcClientIDType ClientID;
///会话编号
```

```

    TUstpFtdcSessionIDType SessionID;
    ///前置编号
    TUstpFtdcFrontIDType FrontID;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.16 OnRspOrderAction 方法和 OnRspOrderActionAccept 方法

报单的撤单应答。当客户端发出过撤单指令后, 飞马平台进行检查后会调用 OnRspOrderActionAccept 方法, 通知用户飞马已经收到该笔申报, 随即发送到交易所并得到交易所应答后, OnRspOrderAction 方法会被调用。

函数原形:

```

void OnRspOrderAction(
    CUstpFtdcOrderActionField * pOrderAction,
    CUstpFtdcRspInfoField * pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pOrderAction: 指向报单操作结构的地址, 包含了提交报单操作的输入数据, 和后台返回的报单编号。

报单操作结构:

```

struct CUstpFtdcOrderActionField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///交易所系统报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码

```

```

    TUstpFtdcUserIDType UserID;
    ///用户操作本地编号
    TUstpFtdcOrderLocalIDType    UserOrderActionLocalID;
    ///本地报单编号
    TUstpFtdcUserOrderLocalIDType    UserOrderLocalID;
    ///报单操作标志
    TUstpFtdcActionFlagType ActionFlag;
    ///价格 （暂不支持，保留域）
    TUstpFtdcPriceType    LimitPrice;
    ///数量变化（暂不支持，保留域）
    TUstpFtdcVolumeType VolumeChange;
    ///本地业务标识
    TUstpFtdcBusinessLocalIDType    BusinessLocalID;
    ///客户代码
    TUstpFtdcClientIDType    ClientID;
    ///会话编号
    TUstpFtdcSessionIDType    SessionID;
    ///前置编号
    TUstpFtdcFrontIDType    FrontID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.17 OnRspQuoteInsert 方法和 OnRspQuoteInsertAccept 方法

报价录入应答。当客户端发出报价录入指令后，飞马平台进行检查后会调用 OnRspQuoteInsertAccept 方法，通知用户飞马已经收到该笔申报，随即发送到交易所并得到交易所应答后，OnRspQuoteInsert 方法会被调用。

函数原形:

```
void OnRspQuoteInsert (
```

```
CUstpFtdcInputQuoteField *pInputQuote,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)
```

参数:

pInputQuote 指向报价结构的地址。

报价结构:

```
struct CUstpFtdcInputQuoteField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///指定通过此席位序号下单，取值范围[1-N]，N为可用席位数目，
    //超出范围的随机分配席位
    TUstpFtdcSeatNoType SeatNo;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///交易系统返回的系统报价编号
    TUstpFtdcQuoteSysIDType QuoteSysID;
    ///用户设定的本地报价编号
    TUstpFtdcUserQuoteLocalIDType UserQuoteLocalID;
    ///飞马向交易系统报的本地报价编号
    TUstpFtdcQuoteLocalIDType QuoteLocalID;
    ///买方买入数量
    TUstpFtdcVolumeType BidVolume;
    ///买方开平标志
    TUstpFtdcOffsetFlagType BidOffsetFlag;
    ///买方投机套保标志
    TUstpFtdcHedgeFlagType BidHedgeFlag;
    ///买方买入价格
    TUstpFtdcPriceType BidPrice;
    ///卖方卖出数量
    TUstpFtdcVolumeType AskVolume;
    ///卖方开平标志
    TUstpFtdcOffsetFlagType AskOffsetFlag;
    ///卖方投机套保标志
    TUstpFtdcHedgeFlagType AskHedgeFlag;
    ///卖方卖出价格
    TUstpFtdcPriceType AskPrice;
```

```

    ///业务单元
    TUstpFtdcBusinessUnitType BusinessUnit;
    ///用户自定义域
    TUstpFtdcCustomTypeUserCustom;
    ///拆分出来的买方用户本地报单编号
    TUstpFtdcUserOrderLocalIDType BidUserOrderLocalID;
    ///拆分出来的卖方用户本地报单编号
    TUstpFtdcUserOrderLocalIDType AskUserOrderLocalID;
    ///询价编号
    TUstpFtdcQuoteSysIDType ReqForQuoteID;
    ///报价停留时间(秒)
    TUstpFtdcMeasureSecType StandByTime;
    ///客户号
    TUstpFtdcClientIDType ClientID;
    ///会话编号
    TUstpFtdcSessionIDType SessionID;
    ///前置编号
    TUstpFtdcFrontIDType FrontID;
};
报价录入时必须满足: MaxOrderLocalID< UserQuoteLocalID< BidUserOrderLocalID<
AskUserOrderLocalID
pRspInfo: 指向响应信息结构的地址。
响应信息结构:
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
nRequestID: 返回用户报价请求的 ID, 该 ID 由用户在报价请求时指定。
bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

```

6.2.18 OnRspQuoteAction 方法和 OnRspQuoteActionAccept 方法

报价撤单应答。当客户端发出过报价撤单录入指令后, 飞马平台进行检查后会调用 OnRspQuoteAction Accept 方法, 通知用户飞马已经收到该笔申报, 随即发送到交易所并得到交易所应答后, OnRspQuoteAction 方法会被调用。

函数原形:

```

void OnRspQuoteAction (
    CustpFtdcQuoteActionField * pQuoteAction,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,

```



```
bool bIsLast)
```

参数:

pQuoteAction 指向报价操作结构的地址。

报价操作结构:

```
struct CUstpFtdcQuoteActionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///交易系统返回的系统报价编号
    TUstpFtdcQuoteSysIDType QuoteSysID;
    ///用户设定的被撤的本地报价编号
    TUstpFtdcUserQuoteLocalIDType UserQuoteLocalID;
    ///用户向飞马报的本地撤消报价编号
    TUstpFtdcUserQuoteLocalIDType UserQuoteActionLocalID;
    ///报单操作标志
    TUstpFtdcActionFlagType ActionFlag;
    ///业务单元
    TUstpFtdcBusinessUnitType BusinessUnit;
    ///用户自定义域
    TUstpFtdcCustomType UserCustom;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.19 OnRspForQuote 方法

询价应答。当客户端发出询价指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```
void OnRspForQuote(  
    CUstpFtdcReqForQuoteField *pReqForQuote  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数:

pReqForQuote 指向询价结构的地址。

询价结构:

```
struct CUstpFtdcReqForQuoteField  
{  
    ///询价编号  
    TUstpFtdcQuoteSysIDType ReqForQuoteID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///买卖方向  
    TUstpFtdcDirectionType Direction;  
    ///交易日  
    TUstpFtdcDateType TradingDay;  
    ///询价时间  
    TUstpFtdcTimeType ReqForQuoteTime;  
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField  
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

nRequestID: 返回询价请求的 ID, 该 ID 由用户在询价请求时指定。

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

6.2.20 OnRspMarginCombAction 方法

客户申请组合应答。当客户端发起组合操作请求 ReqMarginCombAction 时，该方法会被调用。

函数原形:

```
void OnRspMarginCombAction(  
    CUstpFtdcInputMarginCombActionField *pInputMarginCombAction,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

pInputMarginCombAction: 指向组合策略操作结构的地址，详见请求信息章节；

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField  
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

nRequestID: 返回组合操作请求的 ID，该 ID 由用户在组合操作请求时指定。

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

6.2.21 OnRtnTrade 方法

成交回报。当发生成交时飞马平台会通知客户端，该方法会被调用。

函数原形:

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

参数:

pTrade: 指向成交信息结构的地址。

成交信息结构:

```
struct CUstpFtdcTradeField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码
```

```
TUstpFtdcExchangeIDType ExchangeID;  
///交易日  
TUstpFtdcTradingDayType TradingDay;  
///会员编号  
TUstpFtdcParticipantIDType ParticipantID;  
///下单席位号  
TUstpFtdcSeatIDType SeatID;  
///投资者编号  
TUstpFtdcInvestorIDType InvestorID;  
///客户号  
TUstpFtdcClientIDType ClientID;  
///用户编号  
TUstpFtdcUserIDType UserID;  
///下单用户编号  
TUstpFtdcUserIDType OrderUserID;  
///成交编号  
TUstpFtdcTradeIDType TradeID;  
///报单编号  
TUstpFtdcOrderSysIDType OrderSysID;  
///本地报单编号  
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;  
///合约代码  
TUstpFtdcInstrumentIDType InstrumentID;  
///买卖方向  
TUstpFtdcDirectionType Direction;  
///开平标志  
TUstpFtdcOffsetFlagType OffsetFlag;  
///投机套保标志  
TUstpFtdcHedgeFlagType HedgeFlag;  
///成交价格  
TUstpFtdcPriceType TradePrice;  
///成交数量  
TUstpFtdcVolumeType TradeVolume;  
///成交时间  
TUstpFtdcTimeType TradeTime;
```

```
///清算会员编号
TUstpFtdcParticipantIDType ClearingPartID;
///本次成交手续费
TUstpFtdcMoneyType UsedFee;
///本次成交占用保证金
TUstpFtdcMoneyType UsedMargin;
///本次成交占用权利金
TUstpFtdcMoneyType Premium;
///持仓表今持仓量
TUstpFtdcVolumeType Position;
///持仓表今日持仓成本
TUstpFtdcPriceType PositionCost;
///资金表可用资金
TUstpFtdcMoneyType Available;
///资金表占用保证金
TUstpFtdcMoneyType Margin;
///资金表冻结的保证金
TUstpFtdcMoneyType FrozenMargin;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType ActionDay;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///合约编号(或者为套利组合合约号)
TUstpFtdcInstrumentIDType ArbiInstrumentID;
///平仓盈亏
TUstpFtdcMoneyType CloseProfit;
};
```

6.2.22 OnRtnOrder 方法

报单回报。当客户端进行报单录入、报单操作及其它原因（如部分成交）导致报单状态发生变化时，飞马平台会主动通知客户端，该方法会被调用。

函数原形：

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

参数:

pOrder: 指向报单信息结构的地址。

报单信息结构:

```
struct CUstpFtdcOrderField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///系统报单编号  
    TUstpFtdcOrderSysIDType OrderSysID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///指定下单席位编号  
    TUstpFtdcSeatNoType SeatNo;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///用户本地报单号  
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;  
    ///报单类型  
    TUstpFtdcOrderPriceTypeType OrderPriceType;  
    ///买卖方向  
    TUstpFtdcDirectionType Direction;  
    ///开平标志  
    TUstpFtdcOffsetFlagType OffsetFlag;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType HedgeFlag;  
    ///价格  
    TUstpFtdcPriceType LimitPrice;  
    ///数量  
    TUstpFtdcVolumeType Volume;  
    ///有效期类型  
    TUstpFtdcTimeConditionType TimeCondition;  
    ///GTD 日期（暂不支持，保留域）  
    TUstpFtdcDateType GTDDate;  
    ///成交量类型  
    TUstpFtdcVolumeConditionType VolumeCondition;  
    ///最小成交量  
    TUstpFtdcVolumeType MinVolume;  
    ///止损价
```

```
TUstpFtdcPriceType  StopPrice;
///强平原因（暂不支持，保留域）
TUstpFtdcForceCloseReasonType  ForceCloseReason;
///自动挂起标志（暂不支持，保留域）
TUstpFtdcBoolType  IsAutoSuspend;
///业务单元（暂不支持，保留域）
TUstpFtdcBusinessUnitType  BusinessUnit;
///用户自定义域
TUstpFtdcCustomType  UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType  BusinessLocalID;
///业务发生日期
TUstpFtdcDateType  ActionDay;
///策略类别
TUstpFtdcArbiTypeType  ArbiType;
///交易日
TUstpFtdcTradingDayType  TradingDay;
///会员编号
TUstpFtdcParticipantIDType  ParticipantID;
///最初下单用户编号
TUstpFtdcUserIDType  OrderUserID;
///客户号
TUstpFtdcClientIDType  ClientID;
///下单席位号
TUstpFtdcSeatIDType  SeatID;
///插入时间
TUstpFtdcTimeType  InsertTime;
///本地报单编号
TUstpFtdcOrderLocalIDType  OrderLocalID;
///报单来源
TUstpFtdcOrderSourceType  OrderSource;
///报单状态
TUstpFtdcOrderStatusType  OrderStatus;
///撤销时间
TUstpFtdcTimeType  CancelTime;
///撤单用户编号
TUstpFtdcUserIDType  CancelUserID;
///今成交数量
TUstpFtdcVolumeType  VolumeTraded;
///剩余数量
TUstpFtdcVolumeType  VolumeRemain;
///委托类型
```

```

    TUstpFtdcOrderTypeType OrderType;
    ///期权对冲标识
    TUstpFtdcDeliveryFlagType DeliveryFlag;
    ///交易所首次应答时间
    TUstpFtdcTimeType InsertTimeExchange;
};

```

6.2.23 OnErrRtnOrderInsert 方法

报单录入错误回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形：

```

void OnErrRtnOrderInsert(
    CUstpFtdcInputOrderField *pInputOrder,
    CUstpFtdcRspInfoField * pRspInfo);

```

参数：

pInputOrder：指向报单录入结构的地址，包含了提交报单录入时的输入数据，和后台返回的报单编号。

输入报单结构：

```

struct CUstpFtdcInputOrderField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///指定下单席位编号（取值范围[1-N]，N为可用席位数目，超出范围的随机分配席位）
    TUstpFtdcSeatNoType SeatNo;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///用户本地报单号
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
    ///报单类型
    TUstpFtdcOrderPriceTypeType OrderPriceType;
    ///买卖方向
    TUstpFtdcDirectionType Direction;
    ///开平标志
    TUstpFtdcOffsetFlagType OffsetFlag;

```



```

    ///投机套保标志
    TUstpFtdcHedgeFlagType HedgeFlag;
    ///价格
    TUstpFtdcPriceType LimitPrice;
    ///数量
    TUstpFtdcVolumeType Volume;
    ///有效期类型
    TUstpFtdcTimeConditionType TimeCondition;
    ///GTD 日期
    TUstpFtdcDateType GTDDate;
    ///成交量类型
    TUstpFtdcVolumeConditionType VolumeCondition;
    ///最小成交量
    TUstpFtdcVolumeType MinVolume;
    ///止损价
    TUstpFtdcPriceType StopPrice;
    ///强平原因
    TUstpFtdcForceCloseReasonType ForceCloseReason;
    ///自动挂起标志
    TUstpFtdcBoolType IsAutoSuspend;
    ///业务单元
    TUstpFtdcBusinessUnitType BusinessUnit;
    ///用户自定义域 64 字节
    TUstpFtdcCustomType UserCustom;
    ///本地业务标识
    TUstpFtdcBusinessLocalIDType BusinessLocalID;
    ///业务发生日期
    TUstpFtdcDateType ActionDay;
    ///策略类别
    TUstpFtdcArbiTypeType ArbiType;
    ///客户号
    TUstpFtdcClientIDType ClientID;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUSTpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

6.2.24 OnErrRtnOrderAction 方法

撤单错误回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形：

```
void OnErrRtnOrderAction (
    CUstpFtdcOrderActionField *pOrderAction,
    CUstpFtdcRspInfoField *pRspInfo);
```

参数：

pOrderAction: 指向报单操作结构的地址，包含了报单操作请求的输入数据。

报单操作结构：

```
struct CUstpFtdcOrderActionField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///本次撤单操作的本地编号
    TUstpFtdcUserOrderLocalIDType UserOrderActionLocalID;
    ///被撤订单的本地报单编号
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
    ///报单操作标志
    TUstpFtdcActionFlagType ActionFlag;
    ///价格
    TUstpFtdcPriceType LimitPrice;
    ///数量变化
    TUstpFtdcVolumeType VolumeChange;
    ///本地业务标识
    TUstpFtdcBusinessLocalIDType BusinessLocalID;
    ///客户代码
    TUstpFtdcClientIDType ClientID;
```

```
///会话编号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

6.2.25 OnRtnQuote 方法

报价回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形:

```
void OnRtnQuote (CustpFtdcRtnQuoteField *pRtnQuote);
```

参数:

pRtnQuote: 指向报价回报结构的地址，包含了报价操作请求的输入数据，和后台返回的报价编号。

报价回报结构:

```
struct CUstpFtdcRtnQuoteField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///买卖方向
    TUstpFtdcDirectionType Direction;
    ///交易系统返回的系统报价编号
    TUstpFtdcQuoteSysIDType QuoteSysID;
```

```
///用户设定的本地报价编号
TUstpFtdcUserQuoteLocalIDType  UserQuoteLocalID;
///飞马向交易系统报的本地报价编号
TUstpFtdcQuoteLocalIDType  QuoteLocalID;
///买方买入数量
TUstpFtdcVolumeType BidVolume;
///买方开平标志
TUstpFtdcOffsetFlagType BidOffsetFlag;
///买方投机套保标志
TUstpFtdcHedgeFlagType BidHedgeFlag;
///买方买入价格
TUstpFtdcPriceType BidPrice;
///卖方卖出数量
TUstpFtdcVolumeType AskVolume;
///卖方开平标志
TUstpFtdcOffsetFlagType AskOffsetFlag;
///卖方投机套保标志
TUstpFtdcHedgeFlagType AskHedgeFlag;
///卖方卖出价格
TUstpFtdcPriceType AskPrice;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///拆分出来的买方用户本地报单编号
TUstpFtdcUserOrderLocalIDType BidUserOrderLocalID;
///拆分出来的卖方用户本地报单编号
TUstpFtdcUserOrderLocalIDType AskUserOrderLocalID;
///拆分出来的买方本地报单编号
TUstpFtdcOrderLocalIDType BidOrderLocalID;
///拆分出来的卖方本地报单编号
TUstpFtdcOrderLocalIDType AskOrderLocalID;
///询价编号
TUstpFtdcQuoteSysIDType ReqForQuoteID;
///报价停留时间(秒)
TUstpFtdcMeasureSecType StandByTime;
///买方系统报单编号
TUstpFtdcQuoteSysIDType BidOrderSysID;
///卖方系统报单编号
TUstpFtdcQuoteSysIDType AskOrderSysID;
///报价单状态
TUstpFtdcQuoteStatusType QuoteStatus;
///插入时间
TUstpFtdcTimeType InsertTime;
```

```

    ///撤销时间
    TUstpFtdcTimeType   CancelTime;
    ///成交时间
    TUstpFtdcTimeType   TradeTime;
    ///客户号
    TUstpFtdcClientIDType ClientID;
    ///交易所首次应答时间
    TUstpFtdcTimeType   InsertTimeExchange;
    ///会话编号
    TUstpFtdcSessionIDType SessionID;
    ///前置编号
    TUstpFtdcFrontIDType FrontID;
};

```

6.2.26 OnErrRtnQuoteInsert 方法

报价录入错误回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形:

```

void OnErrRtnQuoteInsert (
    CUstpFtdcInputQuoteField *pInputQuote,
    CUstpFtdcRspInfoField *pRspInfo);

```

参数:

pInputQuote: 指向报价录入错误回报结构的地址，包含了报价操作请求的输入数据。

报价录入错误回报结构:

```

struct CUstpFtdcInputQuoteField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType   UserID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///买卖方向
    TUstpFtdcDirectionType Direction;
    ///交易系统返回的系统报价编号
    TUstpFtdcQuoteSysIDType QuoteSysID;
    ///用户设定的本地报价编号

```

```
TUstpFtdcUserQuoteLocalIDType  UserQuoteLocalID;
///飞马向交易系统报的本地报价编号
TUstpFtdcQuoteLocalIDType  QuoteLocalID;
///买方买入数量
TUstpFtdcVolumeType BidVolume;
///买方开平标志
TUstpFtdcOffsetFlagType BidOffsetFlag;
///买方投机套保标志
TUstpFtdcHedgeFlagType BidHedgeFlag;
///买方买入价格
TUstpFtdcPriceType BidPrice;
///卖方卖出数量
TUstpFtdcVolumeType AskVolume;
///卖方开平标志
TUstpFtdcOffsetFlagType AskOffsetFlag;
///卖方投机套保标志
TUstpFtdcHedgeFlagType AskHedgeFlag;
///卖方卖出价格
TUstpFtdcPriceType AskPrice;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///拆分出来的买方用户本地报单编号
TUstpFtdcUserOrderLocalIDType BidUserOrderLocalID;
///拆分出来的卖方用户本地报单编号
TUstpFtdcUserOrderLocalIDType AskUserOrderLocalID;
///拆分出来的买方本地报单编号
TUstpFtdcOrderLocalIDType BidOrderLocalID;
///拆分出来的卖方本地报单编号
TUstpFtdcOrderLocalIDType AskOrderLocalID;
///询价编号
TUstpFtdcQuoteSysIDType ReqForQuoteID;
///报价停留时间(秒)
TUstpFtdcMeasureSecType StandByTime;
///客户号
TUstpFtdcClientIDType ClientID;
///会话编号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

6.2.27 OnErrRtnQuoteAction 方法

报价操作错误回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形:

///报价撤单错误回报

```
void OnErrRtnQuoteAction(
    CUstpFtdcOrderActionField *pOrderAction,
    CUstpFtdcRspInfoField *pRspInfo)
```

参数:

pOrderAction: 指向报价操作结构的地址，包含了报价操作请求的输入数据。

报价操作结构:

```
struct CUstpFtdcQuoteActionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///交易系统返回的系统报价编号
    TUstpFtdcQuoteSysIDType QuoteSysID;
    ///用户设定的被撤的本地报价编号
    TUstpFtdcUserQuoteLocalIDType UserQuoteLocalID;
    ///用户向飞马报的本地撤消报价编号
    TUstpFtdcUserQuoteLocalIDType UserQuoteActionLocalID;
    ///报单操作标志
    TUstpFtdcActionFlagType ActionFlag;
    ///业务单元
    TUstpFtdcBusinessUnitType BusinessUnit;
    ///用户自定义域
    TUstpFtdcCustomType UserCustom;
```

```

    ///买卖方向
    TUstpFtdcDirectionType Direction;
    ///客户号
    TUstpFtdcClientIDType ClientID;
    ///会话编号
    TUstpFtdcSessionIDType SessionID;
    ///前置编号
    TUstpFtdcFrontIDType FrontID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

6.2.28 OnRtnForQuote 方法

询价回报通知。

函数原形:

```

void OnRtnForQuote(
    CUstpFtdcReqForQuoteField *pReqForQuote)

```

参数:

pReqForQuote: 指向询价请求结构的地址。

询价请求结构:

```

struct CUstpFtdcReqForQuoteField
{
    ///询价编号
    TUstpFtdcQuoteSysIDType ReqForQuoteID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
};

```



```
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///用户代码
TUstpFtdcUserIDType UserID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///买卖方向
TUstpFtdcDirectionType Direction;
///交易日
TUstpFtdcDateType TradingDay;
///询价时间
TUstpFtdcTimeType ReqForQuoteTime;
///客户号
TUstpFtdcClientIDType ClientID;

};
```

6.2.29 OnRtnMarginCombAction 方法

客户申请组合确认。

函数原形:

```
void OnRtnMarginCombAction(
    CUstpFtdcInputMarginCombActionField * pInputMarginCombAction);
```

参数:

pInputMarginCombAction: 指向组合操作请求结构的地址。

组合操作请求结构: 详见请求信息章节;

;

6.2.30 OnRspQueryUserLogin 方法

查询飞马系统用户登录应答。

函数原形:

```
void OnRspQueryUserLogin(
    CUstpFtdcRspUserLoginField *pRspUserLogin,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pRspUserLogin: 指向系统用户登录应答结构的地址

系统用户登录应答结构:

```
struct CUstpFtdcRspUserLoginField
{
    ///交易日
    TUstpFtdcDateType    TradingDay;
    ///经纪公司编号
    TUstpFtdcBrokerIDType    BrokerID;
    ///交易用户代码
    TUstpFtdcUserIDType    UserID;
    ///登录成功时间
    TUstpFtdcTimeType    LoginTime;
    ///登录成功时的交易所时间
    TUstpFtdcTimeType    ExchangeTime;
    ///用户最大本地报单号
    TUstpFtdcUserOrderLocalIDType    MaxOrderLocalID;
    ///交易系统名称
    TUstpFtdcTradingSystemNameType    TradingSystemName;
    ///数据中心代码
    TUstpFtdcDataCenterIDType    DataCenterID;
    ///会员私有流当前长度
    TUstpFtdcSequenceNoType    PrivateFlowSize;
    ///交易员私有流当前长度
    TUstpFtdcSequenceNoType    UserFlowSize;
    ///业务发生日期
    TUstpFtdcDateType    ActionDay;
    ///飞马版本号
    TUstpFtdcFemasVersionType    FemasVersion;
    ///飞马生命周期号
    TUstpFtdcFemasLifeCycleType    FemasLifeCycle;
    /// 会话编号
    TUstpFtdcSessionIDType    SessionID;
    ///前置编号
    TUstpFtdcFrontIDType    FrontID;
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUSTpFtdcRspInfoField
{
    ///错误代码
    TUSTpFtdcErrorIDType ErrorID;
    ///错误信息
    TUSTpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回查询前置登陆请求的 ID, 该 ID 由用户在查询前置登陆时指定。

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

6.2.31 OnRspQryOrder 方法

报单查询请求。当客户端发出报单查询指令后, 飞马平台返回响应时, 该方法会被调用。

注: 从 V1.01 版本后, 该方法仅返回正常的订单, 不包含错单。

函数原形:

```
void OnRspQryOrder(
    CUSTpFtdcOrderField *pOrder,
    CUSTpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pOrder: 指向报单查询的返回结构信息的地址。

报单信息结构: 参见请求信息章节。

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUSTpFtdcRspInfoField
{
    ///错误代码
    TUSTpFtdcErrorIDType ErrorID;
    ///错误信息
    TUSTpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.32 OnRspQryOrderFailed 方法

错单查询请求。当客户端发出错单查询指令后，飞马平台返回响应时，该方法会被调用。错单是指被交易所或者飞马柜台检查不通过的订单。

函数原形：

```
void OnRspQryOrderFailed(CUstpFtdcOrderInsertFailedField *pOrderInsertFailed,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pOrderInsertFailed：指向错单查询的返回结构信息的地址。

错单信息结构：

```
struct CUstpFtdcOrderInsertFailedField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///指定通过此席位序号下单
    TUstpFtdcSeatNoType SeatNo;
    ///合约代码/套利组合合约号
    TUstpFtdcInstrumentIDType InstrumentID;
    ///用户本地报单号
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
    ///报单类型
    TUstpFtdcOrderPriceTypeType OrderPriceType;
    ///买卖方向
    TUstpFtdcDirectionType Direction;
    ///开平标志
    TUstpFtdcOffsetFlagType OffsetFlag;
    ///投机套保标志
```

```
TUstpFtdcHedgeFlagType HedgeFlag;
///价格
TUstpFtdcPriceType LimitPrice;
///数量
TUstpFtdcVolumeType Volume;
///有效期类型
TUstpFtdcTimeConditionType TimeCondition;
///GTD 日期
TUstpFtdcDateType GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType VolumeCondition;
///最小成交量
TUstpFtdcVolumeType MinVolume;
///止损价止赢价
TUstpFtdcPriceType StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志
TUstpFtdcBoolType IsAutoSuspend;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType ActionDay;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///交易日
TUstpFtdcTradingDayType TradingDay;
///会员编号
TUstpFtdcParticipantIDType ParticipantID;
///最初下单用户编号
TUstpFtdcUserIDType OrderUserID;
```

```
///客户号
TUstpFtdcClientIDType ClientID;
///下单席位号
TUstpFtdcSeatIDType SeatID;
///插入时间
TUstpFtdcTimeType InsertTime;
///本地报单编号
TUstpFtdcOrderLocalIDType OrderLocalID;
///报单来源
TUstpFtdcOrderSourceType OrderSource;
///报单状态
TUstpFtdcOrderStatusType OrderStatus;
///撤销时间
TUstpFtdcTimeType CancelTime;
///撤单用户编号
TUstpFtdcUserIDType CancelUserID;
///今成交数量
TUstpFtdcVolumeType VolumeTraded;
///剩余数量
TUstpFtdcVolumeType VolumeRemain;
///委托类型
TUstpFtdcOrderTypeType OrderType;
///期权对冲标识
TUstpFtdcDeliveryFlagType DeliveryFlag;
///错误信息
TUstpFtdcErrorMsgType ErrorMsg;
///交易所首次应答时间
TUstpFtdcTimeType InsertTimeExchange;
};
```

6.2.33 OnRspQryTrade 方法

成交单查询应答。当客户端发出成交单查询指令后，飞马平台返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryTrade(  
    CUstpFtdcTradeField * pTrade,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

pTrade: 指向成交信息结构的地址。

成交信息结构:

```
struct CUstpFtdcTradeField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///交易日  
    TUstpFtdcTradingDayType TradingDay;  
    ///会员编号  
    TUstpFtdcParticipantIDType ParticipantID;  
    ///下单席位号  
    TUstpFtdcSeatIDType SeatID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///客户号  
    TUstpFtdcClientIDType ClientID;  
    ///用户编号  
    TUstpFtdcUserIDType UserID;  
    ///报单编号  
    TUstpFtdcOrderSysIDType OrderSysID;  
    ///下单用户编号  
    TUstpFtdcUserIDType OrderUserID;  
    ///成交编号  
    TUstpFtdcTradeIDType TradeID;  
    ///本地报单编号  
    TUstpFtdcOrderLocalIDType UserOrderLocalID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///买卖方向  
    TUstpFtdcDirectionType Direction;  
    ///开平标志  
    TUstpFtdcOffsetFlagType OffsetFlag;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType HedgeFlag;  
    ///成交价格
```

```

TUstpFtdcPriceType TradePrice;
///成交数量
TUstpFtdcVolumeType TradeVolume;
///成交时间
TUstpFtdcTimeType TradeTime;
///清算会员编号
TUstpFtdcParticipantIDType ClearingPartID;
///本次成交手续费
TUstpFtdcMoneyType UsedFee;
///本次成交占用保证金
TUstpFtdcMoneyType UsedMargin;
///本次成交占用权利金
TUstpFtdcMoneyType Premium;
///持仓表今持仓量
TUstpFtdcVolumeType Position;
///持仓表今日持仓成本
TUstpFtdcPriceType PositionCost;
///资金表可用资金
TUstpFtdcMoneyType Available;
///资金表占用保证金
TUstpFtdcMoneyType Margin;
///资金表冻结的保证金
TUstpFtdcMoneyType FrozenMargin;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType ActionDay;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///合约编号(或者为套利组合合约号)
TUstpFtdcInstrumentIDType ArbiInstrumentID;
///平仓盈亏
TUstpFtdcMoneyType CloseProfit;

```

```
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
```



```
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.34 OnRspQryUserInvestor 方法

可用投资者账户查询应答。当客户端发出可用投资者账户查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```
void OnRspQryUserInvestor(
    CUstpFtdcUserInvestorField *pUserInvestor,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pUserInvestor 指向用户可用投资者账户结构的地址。

投资者账户结构:

///可用投资者

```
struct CUstpFtdcRspUserInvestorField
```

```
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///授权功能集
    TUstpFtdcGrantFuncSetType GrantFuncSet;
    ///投资者名称
    TUstpFtdcInvestorNameType InvestorName;
    ///组织机构代码
    TUstpFtdcInstituteCodeType InstituteCode;
    ///营业部
    TUstpFtdcDepartmentType Department;
```

```
///客户状态
TUstpFtdcClientStatusType ClientStatus;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///证件类型
TUstpFtdcIdentifiedCardTypeType IdentifiedCardType;
///交易权限
TUstpFtdcTradingRightType TradingRight;
///投资者类型
TUstpFtdcInvestorTypeType InvestorType;
///委托密码
TUstpFtdcPasswordType DelegatePassword;
///交易密码
TUstpFtdcPasswordType TradePassword;
///手机
TUstpFtdcTransMobilePhoneType MobilePhone;
///开户日期
TUstpFtdcDateType OpenAccountDate;
///联系电话
TUstpFtdcTransMobilePhoneType Phone;
///是否活跃
TUstpFtdcBoolType IsActive;
///地址
TUstpFtdcTransAddressType Address;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.35 OnRspQryTradingCode 方法

交易编码查询。当客户端发出交易编码查询指令后，飞马平台返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryTradingCode(  
    CUstpFtdcRspTradingCodeField *pTradingCode,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数：

pTradingCode 指向交易编码查询返回信息的地址。

交易编码结构：

```
struct CUstpFtdcRspTradingCodeField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///客户代码  
    TUstpFtdcClientIDType ClientID;  
    ///客户代码权限  
    TUstpFtdcTradingRightType ClientRight;  
    ///客户保值类型  
    TUstpFtdcHedgeFlagType ClientHedgeFlag;  
    ///是否活跃  
    TUstpFtdcIsActiveType IsActive;  
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构：

```
struct CUstpFtdcRspInfoField  
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

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

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

6.2.36 OnRspQryInvestorAccount 方法

投资者资金账户查询。当客户端发出投资者资金账户查询指令后，飞马平台返回响应时，该方法会被调用。

函数原形：

```
void OnRspQryInvestorAccount(  
    CUstpFtdcRspInvestorAccountField *pRspInvestorAccount,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数：

pRspInvestorAccount 指向投资者资金账户信息的地址。

投资者资金账户结构

```
struct CUstpFtdcRspInvestorAccountField
```

```
{
```

```
    ///经纪公司编号
```

```
    TUstpFtdcBrokerIDType   BrokerID;
```

```
    ///投资者编号
```

```
    TUstpFtdcInvestorIDType InvestorID;
```

```
    ///资金帐号
```

```
    TUstpFtdcAccountIDType  AccountID;
```

```
    ///上次结算准备金
```

```
    TUstpFtdcMoneyType      PreBalance;
```

```
    ///入金金额
```

```
    TUstpFtdcMoneyType      Deposit;
```

```
    ///出金金额
```

```
    TUstpFtdcMoneyType      Withdraw;
```

```
    ///冻结的保证金
```

```
    TUstpFtdcMoneyType      FrozenMargin;
```

```
    ///冻结手续费
```

```
    TUstpFtdcMoneyType      FrozenFee;
```

```
    ///冻结权利金
```

```
    TUstpFtdcMoneyType      FrozenPremium;
```

```
    ///手续费
```

```
    TUstpFtdcMoneyType      Fee;
```

```
    ///平仓盈亏
```

```
    TUstpFtdcMoneyType      CloseProfit;
```

```
///持仓盈亏
TUstpFtdcMoneyType  PositionProfit;
///可用资金
TUstpFtdcMoneyType  Available;
///多头冻结的保证金
TUstpFtdcMoneyType  LongFrozenMargin;
///空头冻结的保证金
TUstpFtdcMoneyType  ShortFrozenMargin;
///多头占用保证金
TUstpFtdcMoneyType  LongMargin;
///空头占用保证金
TUstpFtdcMoneyType  ShortMargin;
///当日释放保证金
TUstpFtdcMoneyType  ReleaseMargin;
///动态权益
TUstpFtdcMoneyType  DynamicRights;
///今日出入金
TUstpFtdcMoneyType  TodayInOut;
///占用保证金
TUstpFtdcMoneyType  Margin;
///期权权利金收支
TUstpFtdcMoneyType  Premium;
///风险度
TUstpFtdcMoneyType  Risk;
///上日可用资金
TUstpFtdcMoneyType  PreAvailable;
///总冻结持仓
TUstpFtdcVolumeType TotalFrozenPos;
///其他费用
TUstpFtdcMoneyType  OtherFee;
///结算准备金
TUstpFtdcMoneyType  Balance;
///质押金额
TUstpFtdcMoneyType  Mortgage;
///币种
```

```
TUstpFtdcCurrencyIDType Currency;  
///帐户交易状态  
TUstpFtdcAccountTradingStatusType TradingStatus;  
///帐户资金状态  
TUstpFtdcAccountMoneyStatusType MoneyStatus;  
///上日货币质押金额  
TUstpFtdcMoneyType PreCurrencyPledge;  
///货币质押入金额  
TUstpFtdcMoneyType CurrencyPledgeIn;  
///货币质押出金额  
TUstpFtdcMoneyType CurrencyPledgeOut;  
///外币汇入金额  
TUstpFtdcMoneyType CurrencyExchangeIn;  
///外币汇出金额  
TUstpFtdcMoneyType CurrencyExchangeOut;  
///冻结资金  
TUstpFtdcMoneyType FrozenMoney;  
///保底金额  
TUstpFtdcMoneyType MinGuarantee;  
///可提比例  
TUstpFtdcRatioType WithdrawRatioAllowed;  
///交割保证金  
TUstpFtdcMoneyType DeliveryMargin;  
///可提资金  
TUstpFtdcMoneyType WithdrawAllowed;  
///上日实有货币资金  
TUstpFtdcMoneyType PreDepositEnd;  
///上日本币质押为外币金额  
TUstpFtdcMoneyType PreCurrencyPledgeOut;  
///货币解质入金额  
TUstpFtdcMoneyType CurrencyUnPledgeIn;  
///货币解质出金额  
TUstpFtdcMoneyType CurrencyUnPledgeOut;  
///是否有持仓或成交  
TUstpFtdcCharBoolType HasPosOrTrade;
```

```

///逐笔对冲持仓盈亏
TUstpFtdcMoneyType  PositionProfitByDeal;
///逐笔对冲平仓盈亏
TUstpFtdcMoneyType  CloseProfitByDeal;
///资金计算配置串
TUstpFtdcMoneyCalcCfgType  MoneyCalcCfg;
///换汇出金冻结资金
TUstpFtdcMoneyType  ExchangeOutFrozenMoney;
///换汇开仓冻结资金
TUstpFtdcMoneyType  ExchangeOpenFrozenMoney;
    ///交易所占用保证金
    TUstpFtdcMoneyType  MarginExchange;
    ///交易所优惠前多头占用保证金
    TUstpFtdcMoneyType  LongMarginExchange;
    ///交易所优惠前空头占用保证金
    TUstpFtdcMoneyType  ShortMarginExchange;
    ///交易所当日释放保证金
    TUstpFtdcMoneyType  ReleaseMarginExchange;
    ///交易所交割保证金
    TUstpFtdcMoneyType  DeliveryMarginExchange;
    ///期权买方持仓市值
    TUstpFtdcMoneyType  LongOptionMarketValue;
    ///期权卖方持仓市值
    TUstpFtdcMoneyType  ShortOptionMarketValue;
    ///市值权益
    TUstpFtdcMoneyType  MarketValue;
    ///总冻结资金
    TUstpFtdcMoneyType  TotalFrozen;
    ///权利金收入
    TUstpFtdcMoneyType  PremiumIn;
    ///权利金支出
    TUstpFtdcMoneyType  PremiumOut;
    ///行权手续费冻结
    TUstpFtdcMoneyType  FrozenExecFee;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType  ErrorID;
    ///错误信息

```

```
TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.37 OnRspQryExchange 方法

交易所查询。当客户端发出交易所查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```
void OnRspQryExchange(
    CUstpFtdcRspExchangeField *pExchange,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pExchange 指向交易所结构的地址。

交易编码结构

```
struct CUstpFtdcRspExchangeField
{
```

```
    ///交易所代码
```

```
    TUstpFtdcExchangeIDType ExchangeID;
```

```
    ///交易所名称
```

```
    TUstpFtdcExchangeNameType ExchangeName;
```

```
    ///交易所状态
```

```
    TUstpFtdcExchangeStatusType ExchangeStatus;
```

```
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
```

```
    ///错误代码
```

```
    TUstpFtdcErrorIDType ErrorID;
```

```
    ///错误信息
```

```
    TUstpFtdcErrorMsgType ErrorMsg;
```

```
};
```

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

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

6.2.38 OnRspQryInstrument 方法

合约信息查询应答。当客户端发出合约信息查询指令后, 飞马平台返回响应时, 该

方法会被调用。

函数原形:

```
void OnRspQryUstpInstrument(  
    CUstpFtdcRspInstrumentField *pRspInstrument,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数:

pRspInstrument 指向合约信息结构的地址。

合约信息结构:

```
struct CUstpFtdcRspInstrumentField  
{  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///品种代码  
    TUstpFtdcProductIDType ProductID;  
    ///品种名称  
    TUstpFtdcProductNameType ProductName;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///合约名称  
    TUstpFtdcInstrumentNameType InstrumentName;  
    ///交割年份  
    TUstpFtdcDeliveryYearType DeliveryYear;  
    ///交割月  
    TUstpFtdcDeliveryMonthType DeliveryMonth;  
    ///限价单最大下单量  
    TUstpFtdcMaxLimitOrderVolumeType MaxLimitOrderVolume;  
    ///限价单最小下单量  
    TUstpFtdcMinLimitOrderVolumeType MinLimitOrderVolume;  
    ///市价单最大下单量  
    TUstpFtdcMaxMarketOrderVolumeType MaxMarketOrderVolume;  
    ///市价单最小下单量  
    TUstpFtdcMinMarketOrderVolumeType MinMarketOrderVolume;  
    ///数量乘数  
    TUstpFtdcVolumeMultipleType VolumeMultiple;  
    ///报价单位  
    TUstpFtdcPriceTickType PriceTick;  
    ///币种  
    TUstpFtdcCurrencyType Currency;  
    ///多头限仓  
    TUstpFtdcVolumeType LongPosLimit;  
    ///空头限仓  
    TUstpFtdcVolumeType ShortPosLimit;
```

```
///跌停板价
TUstpFtdcPriceType LowerLimitPrice;
///涨停板价
TUstpFtdcPriceType UpperLimitPrice;
///昨结算
TUstpFtdcPriceType PreSettlementPrice;
///合约交易状态
TUstpFtdcInstrumentStatusType InstrumentStatus;
///创建日
TUstpFtdcDateType CreateDate;
///上市日
TUstpFtdcDateType OpenDate;
///到期日
TUstpFtdcDateType ExpireDate;
///开始交割日
TUstpFtdcDateType StartDelivDate;
///最后交割日
TUstpFtdcDateType EndDelivDate;
///挂牌基准价
TUstpFtdcPriceType BasisPrice;
///当前是否交易
TUstpFtdcBoolType IsTrading;
///基础商品代码
TUstpFtdcInstrumentIDType UnderlyingInstrID;
///基础商品乘数
TUstpFtdcUnderlyingMultipleType UnderlyingMultiple;
///持仓类型
TUstpFtdcPositionTypeType PositionType;
///执行价
TUstpFtdcPriceType StrikePrice;
///期权类型
TUstpFtdcOptionsTypeType OptionsType;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///第一腿合约代码
TUstpFtdcInstrumentIDType InstrumentID_1;
///第一腿买卖方向
TUstpFtdcDirectionType Direction_1;
///第一腿数量比例
TUstpFtdcRatioType Ratio_1;
///第二腿合约代码
TUstpFtdcInstrumentIDType InstrumentID_2;
```

```

    ///第二腿买卖方向
    TUstpFtdcDirectionType  Direction_2;
    ///第二腿数量比例
    TUstpFtdcRatioType  Ratio_2;
    ///交割月自然人开仓
    TUstpFtdcAllowDelivPersonOpenType  AllowDelivPersonOpen;
    ///产品类型
    TUstpFtdcProductClassType  ProductClass;
    ///I02002
    TUstpFtdcSerialType  Serial;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType  ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType  ErrorMsg;
};

```

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

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

6.2.39 OnRspQryInvestorPosition 方法

投资者持仓查询应答。当客户端发出投资者持仓查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```

void OnRspQryInvestorPosition (
    CUstpFtdcRspInvestorPositionField *pInvestorPosition,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pInvestorPosition 指向投资者持仓结构的地址。

投资者持仓结构:

```

struct CUstpFtdcRspInvestorPositionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType  BrokerID;
    ///交易所代码

```

```
TUstpFtdcExchangeIDType ExchangeID;  
///投资者编号  
TUstpFtdcInvestorIDType InvestorID;  
///合约代码  
TUstpFtdcInstrumentIDType InstrumentID;  
///交易编码  
TUstpFtdcClientIDType ClientID;  
///多空方向  
TUstpFtdcDirectionType Direction;  
///投机套保标志  
TUstpFtdcHedgeFlagType HedgeFlag;  
///昨结算  
TUstpFtdcPriceType PreSettlementPrice;  
///初始持仓量  
TUstpFtdcVolumeType PrePosition;  
///总持仓量  
TUstpFtdcVolumeType Position;  
///开仓量  
TUstpFtdcVolumeType OpenPosition;  
///平仓量  
TUstpFtdcVolumeType ClosePosition;  
///单腿持仓量  
TUstpFtdcVolumeType LegPosition;  
///组合持仓量  
TUstpFtdcVolumeType CombPosition;  
///昨持仓剩余量  
TUstpFtdcVolumeType YdPositionRemain;  
///今持仓剩余量  
TUstpFtdcVolumeType TodayPositionRemain;  
///开仓冻结量  
TUstpFtdcVolumeType FrozenOpenPosition;  
///平仓冻结量  
TUstpFtdcVolumeType FrozenClosingPosition;  
///平昨冻结量  
TUstpFtdcVolumeType FrozenClosingYdPosition;
```

```
///平今冻结量
TUstpFtdcVolumeType FrozenClosingTodayPosition;
///执行冻结量
TUstpFtdcVolumeType FrozenExecPosition;
///放弃执行冻结量
TUstpFtdcVolumeType FrozenAbandonExecPosition;
///占用保证金
TUstpFtdcMoneyType UsedMargin;
///冻结保证金
TUstpFtdcMoneyType FrozenMargin;
///单腿占用保证金
TUstpFtdcMoneyType LegUsedMargin;
///单腿冻结保证金
TUstpFtdcMoneyType LegFrozenMargin;
///组合占用保证金
TUstpFtdcMoneyType CombUsedMargin;
///组合冻结保证金
TUstpFtdcMoneyType CombFrozenMargin;
///交易所占用保证金
TUstpFtdcMoneyType UsedMarginExchange;
///交易所单腿占用保证金
TUstpFtdcMoneyType LegUsedMarginExchange;
///交易所组合占用保证金
TUstpFtdcMoneyType CombUsedMarginExchange;
///开仓成本
TUstpFtdcMoneyType PositionCostByDeal;
///持仓成本
TUstpFtdcMoneyType PositionCostBySettl;
///持仓盈亏计算价
TUstpFtdcPriceType PositionProfitPrice;
///逐日盯市平仓盈亏
TUstpFtdcMoneyType CloseProfitBySettl;
///逐笔对冲平仓盈亏
TUstpFtdcMoneyType CloseProfitByDeal;
///逐日盯市持仓盈亏
```

```

TUstpFtdcMoneyType PositionProfitBySettl;
///逐笔对冲持仓盈亏
TUstpFtdcMoneyType PositionProfitByDeal;
///币种
TUstpFtdcCurrencyIDType Currency;
///期权持仓市值
TUstpFtdcMoneyType OptMarketValue;
///TAS 持仓成本
TUstpFtdcMoneyType TASPositionCost;
///TAS 持仓量
TUstpFtdcVolumeType TASPosition;
///实虚值额
TUstpFtdcMoneyType InOutTheMoneyValue;
///今执行冻结量
TUstpFtdcVolumeType TodayFrozenExecPosition;
///昨执行冻结量
TUstpFtdcVolumeType YdFrozenExecPosition;
///今放弃执行冻结量
TUstpFtdcVolumeType TodayFrozenAbandonExecPosition;
///昨放弃执行冻结量
TUstpFtdcVolumeType YdFrozenAbandonExecPosition;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.40 OnRspQryInvestorFee 方法

投资者手续费率查询应答。当客户端发出投资者手续费率查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```
void OnRspQryInvestorFee(  
    CUstpFtdcInvestorFeeField *pInvestorFee,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数:

pInvestorFee 指向投资者手续费率结构的地址。

投资者手续费率结构:

```
struct CUstpFtdcInvestorFeeField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType   BrokerID;  
    ///客户号  
    TUstpFtdcClientIDType   ClientID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///品种代码  
    TUstpFtdcProductIDType  ProductID;  
    ///开仓手续费按比例  
    TUstpFtdcRatioType   OpenFeeRate;  
    ///开仓手续费按手数  
    TUstpFtdcRatioType   OpenFeeAmt;  
    ///平仓手续费按比例  
    TUstpFtdcRatioType   OffsetFeeRate;  
    ///平仓手续费按手数  
    TUstpFtdcRatioType   OffsetFeeAmt;  
    ///平今仓手续费按比例  
    TUstpFtdcRatioType   OTFeeRate;  
    ///平今仓手续费按手数  
    TUstpFtdcRatioType   OTFeeAmt;  
    ///行权手续费按比例  
    TUstpFtdcRatioType   ExecFeeRate;  
    ///行权手续费按手数  
    TUstpFtdcRatioType   ExecFeeAmt;  
    ///每笔委托的申报费  
    TUstpFtdcRatioType   PerOrderAmt;  
    ///每笔撤单申报费  
    TUstpFtdcRatioType   PerCancelAmt;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType HedgeFlag;
```

```

    ///短线开仓手续费按比例
    TUstpFtdcRatioType ShortOpenFeeRate;
    ///短线开仓手续费按手数
    TUstpFtdcRatioType ShortOpenFeeAmt;
    ///多空方向
    TUstpFtdcDirectionType Direction;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.41 OnRspQryInvestorMargin 方法

投资者保证金率查询应答。当客户端发出投资者保证金率查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```

void OnRspQryInvestorMargin (
    CUstpFtdcInvestorMarginField *pInvestorMargin,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pInvestorMargin 指向投资者保证金率结构的地址。

投资者保证金率结构:

```

struct CUstpFtdcInvestorMarginField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///客户号
    TUstpFtdcClientIDType ClientID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;

```



```

    ///合约代码
    TUstpFtdcInstrumentIDType  InstrumentID;
    ///品种代码
    TUstpFtdcProductIDType  ProductID;
    ///多头占用保证金按比例
    TUstpFtdcRatioType  LongMarginRate;
    ///多头保证金按手数
    TUstpFtdcRatioType  LongMarginAmt;
    ///空头占用保证金按比例
    TUstpFtdcRatioType  ShortMarginRate;
    ///空头保证金按手数
    TUstpFtdcRatioType  ShortMarginAmt;
    ///投机套保标志
    TUstpFtdcHedgeFlagType  HedgeFlag;
    ///多头占用保证金按比例
    TUstpFtdcRatioType  LongMarginRateExchange;
    ///多头保证金按手数
    TUstpFtdcRatioType  LongMarginAmtExchange;
    ///空头占用保证金按比例
    TUstpFtdcRatioType  ShortMarginRateExchange;
    ///空头保证金按手数
    TUstpFtdcRatioType  ShortMarginAmtExchange;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType  ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType  ErrorMsg;
};

```

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

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

6.2.42 OnRspQryInvestorCombPosition 方法

投资者组合持仓查询应答。当客户端发出投资者组合持仓查询指令后, 飞马平台返

回响应时，该方法会被调用。

函数原形:

```
void OnRspQryInvestorCombPosition (
    CUstpFtdcRspInvestorCombPositionField *pInvestorPosition,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pInvestorPosition 指向投资者组合持仓结构的地址。

投资者组合持仓结构:

```
struct CUstpFtdcRspInvestorCombPositionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///持仓方向
    TUstpFtdcDirectionType  Direction;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType   HedgeFlag;
    ///客户代码
    TUstpFtdcClientIDType    ClientID;
    ///组合合约代码
    TUstpFtdcInstrumentIDType CombInstrumentID;
    ///单腿 1 合约代码
    TUstpFtdcInstrumentIDType Leg1InstrumentID;
    ///单腿 2 合约代码
    TUstpFtdcInstrumentIDType Leg2InstrumentID;
    ///组合持仓
    TUstpFtdcVolumeType      CombPosition;
    ///冻结组合持仓
    TUstpFtdcVolumeType      CombFrozenPosition;
    ///组合一手释放的保证金
    TUstpFtdcMoneyType       CombFreeMargin;
    ///开仓成本
```

```

    TUstpFtdcMoneyType  PositionCostByDeal;
    ///持仓成本
    TUstpFtdcMoneyType  PositionCostBySettl;
    ///优惠前客户保证金
    TUstpFtdcMoneyType  MarginToUse;
    ///优惠后客户保证金
    TUstpFtdcMoneyType  UseMargin;
    ///逐日盯市平仓盈亏
    TUstpFtdcMoneyType  CloseProfitBySettl;
    ///逐笔对冲平仓盈亏
    TUstpFtdcMoneyType  CloseProfitByDeal;
    ///逐日盯市持仓盈亏
    TUstpFtdcMoneyType  PositionProfitBySettl;
    ///逐笔对冲持仓盈亏
    TUstpFtdcMoneyType  PositionProfitByDeal;
    ///交易所组合一手释放的保证金
    TUstpFtdcMoneyType  CombFreeMarginExchange;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType  ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType  ErrorMsg;
};

```

nRequestID: 返回用户组合持仓查询请求的 ID, 该 ID 由用户在组合持仓查询时指定。

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

6.2.43 OnRspQryInstrumentGroup 方法

合约组信息查询应答。当客户端发出合约组查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```

void OnRspQryInstrumentGroup(
    CUstpFtdcRspInstrumentGroupField *pRspInstrumentGroup,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,

```

```
bool bIsLast);
```

参数:

pRspInstrumentGroup 指向合约组信息查询应答结构的地址。

合约组信息查询应答结构:

```
struct CUstpFtdcRspInstrumentGroupField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///合约组代码
    TUstpFtdcInstrumentGroupIDType InstrumentGroupID;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回合约组查询请求的 ID, 该 ID 由用户在合约组查询时指定。

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

6.2.44 OnRspQryClientMarginCombType 方法

客户组合保证金收取模式的查询应答。当客户端发出组合保证金类型查询指令后, 飞马平台返回响应时, 该方法会被调用。

函数原形:

```
void OnRspQryClientMarginCombType(
    CUstpFtdcRspClientMarginCombTypeField *pRspClientMarginCombType,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pRspClientMarginCombType 指向组合保证金类型查询应答结构的地址。

组合保证金类型查询应答结构:

```
struct CUstpFtdcRspClientMarginCombTypeField
```

```
{
```

```
    ///交易所代码
```

```
    TUstpFtdcExchangeIDType ExchangeID;
```

```
    ///经纪公司编号
```

```
    TUstpFtdcBrokerIDType BrokerID;
```

```
    ///会员代码
```

```
    TUstpFtdcParticipantIDType ParticipantID;
```

```
    ///客户代码
```

```
    TUstpFtdcClientIDType ClientID;
```

```
    ///合约组代码
```

```
    TUstpFtdcInstrumentGroupIDType InstrumentGroupID;
```

```
    ///保证金组合类型
```

```
    TUstpFtdcClientMarginCombTypeType MarginCombType;
```

```
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
```

```
{
```

```
    ///错误代码
```

```
    TUstpFtdcErrorIDType ErrorID;
```

```
    ///错误信息
```

```
    TUstpFtdcErrorMsgType ErrorMsg;
```

```
};
```

nRequestID: 返回用户保证金收取类型查询请求的 ID, 该 ID 由用户在保证金收取类型查询时指定。

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

6.2.45 OnRspExecOrderInsert 方法和 OnRspExecOrderInsertAccept 方法

行权录入应答。当客户端发出过行权录入指令后, 飞马平台进行检查后会调用 OnRspExecOrderInsertAccept 方法, 通知用户飞马已经收到该笔申报, 随即发送到交易所并得到交易所应答后, OnRspExecOrderInsert 方法会被调用。

函数原形:

```
void OnRspExecOrderInsert(
    CUstpFtdcInputExecOrderField *pInputExecOrder,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
```

```
bool bIsLast)
```

参数:

pInputExecOrder: 指向行权录入结构的地址。详见请求信息章节;

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回询价请求的 ID, 该 ID 由用户在询价请求时指定。

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

6.2.46 OnRspExecOrderAction 方法和 OnRspExecOrderActionAccept 方法

行权录入撤单/放弃行权应答。当客户端发出过行权录入撤单/放弃行权录入指令后, 飞马平台进行检查后会调用 OnRspExecOrderActionAccept 方法, 通知用户飞马已经收到该笔申报, 随即发送到交易所并得到交易所应答后, OnRspExecOrderAction 方法会被调用。

函数原形:

```
void OnRspExecOrderAction(
    CUstpFtdcInputExecOrderActionField * pInputExecOrderAction,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pInputExecOrderAction: 指向行权操作结构的地址。

```
struct CUstpFtdcInputExecOrderActionField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
```

```

    ///本次撤单操作的本地编号
    TUstpFtdcUserOrderLocalIDType    UserOrderActionLocalID;
    ///被撤订单的本地报单编号
    TUstpFtdcUserOrderLocalIDType    UserOrderLocalID;
    ///报单操作标志
    TUstpFtdcActionFlagType    ActionFlag;
    ///数量变化
    TUstpFtdcVolumeType    VolumeChange;
    ///本地业务标识
    TUstpFtdcBusinessLocalIDType    BusinessLocalID;
    ///委托类型
    TUstpFtdcOrderTypeType    OrderType;
    ///会话编号
    TUstpFtdcSessionIDType    SessionID;
    ///前置编号
    TUstpFtdcFrontIDType    FrontID;
    ///交易编码
    TUstpFtdcClientIDType    ClientID;
    ///合约代码
    TUstpFtdcInstrumentIDType    InstrumentID;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType    ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType    ErrorMsg;
};

```

nRequestID: 返回询价请求的 ID, 该 ID 由用户在询价请求时指定。

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

6.2.47 OnRtnExecOrder 方法

行权回报通知。与报单通知类似。

函数原形:

```
void OnRtnExecOrder(CUstpFtdcExecOrderField *pExecOrder)
```

参数:

pExecOrder: 指向行权通知结构的地址。

行权通知结构:

```
struct CUstpFtdcExecOrderField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType   UserID;
    ///合约代码
    TUstpFtdcInstrumentIDType   InstrumentID;
    ///用户本地报单号
    TUstpFtdcUserOrderLocalIDType   UserOrderLocalID;
    ///委托类型
    TUstpFtdcOrderTypeType   OrderType;
    ///期权对冲标识
    TUstpFtdcDeliveryFlagType   DeliveryFlag;
    ///投机套保标志
    TUstpFtdcHedgeFlagType   HedgeFlag;
    ///数量
    TUstpFtdcVolumeType   Volume;
    ///用户自定义域
    TUstpFtdcCustomType   UserCustom;
    ///业务发生日期
    TUstpFtdcDateType   ActionDay;
    ///本地业务标识
    TUstpFtdcBusinessLocalIDType   BusinessLocalID;
    ///业务单元
    TUstpFtdcBusinessUnitType   BusinessUnit;
    ///交易日
    TUstpFtdcTradingDayType   TradingDay;
    ///会员编号
    TUstpFtdcParticipantIDType   ParticipantID;
    ///最初下单用户编号
    TUstpFtdcUserIDType   OrderUserID;
    ///客户号
    TUstpFtdcClientIDType   ClientID;
```



```

    ///下单席位号
    TUstpFtdcSeatIDType SeatID;
    ///插入时间
    TUstpFtdcTimeType InsertTime;
    ///本地报单编号
    TUstpFtdcOrderLocalIDType OrderLocalID;
    ///报单来源
    TUstpFtdcOrderSourceType OrderSource;
    ///报单状态
    TUstpFtdcOrderStatusType OrderStatus;
    ///撤销时间
    TUstpFtdcTimeType CancelTime;
    ///撤单用户编号
    TUstpFtdcUserIDType CancelUserID;
    ///交易所首次应答时间
    TUstpFtdcTimeType InsertTimeExchange;
    ///今昨仓标志
    TUstpFtdcOffsetFlagType OffsetFlag;
    ///会话编号
    TUstpFtdcSessionIDType SessionID;
    ///前置编号
};

```

6.2.48 OnErrRtnExecOrderInsert 方法

行权录入错误回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形:

```

void OnErrRtnExecOrderInsert(
    CUstpFtdcExecInputOrderField *pInputExecOrder,
    CUstpFtdcRspInfoField *pRspInfo);

```

参数:

pInputExecOrder: 指向行权录入结构的地址，包含了提交行权录入时的输入数据，和后台返回的相关。详见请求信息章节；

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;

```

```
};
```

6.2.49 OnErrRtnExecOrderAction 方法

行权撤单/放弃行权请求的错误回报。由飞马平台主动通知客户端，该方法会被调用。

函数原形:

```
void OnErrRtnExecOrderAction (  
    CUstpFtdcExecOrderActionField *pExecOrderAction,  
    CUstpFtdcRspInfoField *pRspInfo);
```

参数:

pExecOrderAction: 指向行权操作结构的地址，包含了行权操作请求的输入数据。

报单操作结构:

```
struct CUstpFtdcInputExecOrderActionField  
{  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///报单编号  
    TUstpFtdcOrderSysIDType OrderSysID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///本次撤单操作的本地编号  
    TUstpFtdcUserOrderLocalIDType UserOrderActionLocalID;  
    ///被撤订单的本地报单编号  
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;  
    ///报单操作标志  
    TUstpFtdcActionFlagType ActionFlag;  
    ///数量变化  
    TUstpFtdcVolumeType VolumeChange;  
    ///本地业务标识  
    TUstpFtdcBusinessLocalIDType BusinessLocalID;  
    ///委托类型  
    TUstpFtdcOrderTypeType OrderType;  
    ///会话编号  
    TUstpFtdcSessionIDType SessionID;  
    ///前置编号  
    TUstpFtdcFrontIDType FrontID;  
    ///交易编码  
    TUstpFtdcClientIDType ClientID;
```

```

        ///合约代码
        TUstpFtdcInstrumentIDType InstrumentID;
};
pRspInfo: 返回用户响应信息的地址。
响应信息结构:
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

6.2.50 OnErrRtnExecSelfCloseInsert 方法

期权对冲设置请求的错误回报。由飞马平台主动通知客户端。

函数原形:

```

void OnErrRtnExecSelfCloseInsert(
    CUstpFtdcExecSelfCloseField *pExecSelfClose,
    CUstpFtdcRspInfoField *pRspInfo) {};

```

参数:

pExecSelfClose: 指向期权对冲设置结构的地址。

期权对冲设置结构:

```

struct CUstpFtdcExecSelfCloseField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///本地期权自对冲编号
    TUstpFtdcUserSelfCloseLocalIDType UserSelfCloseLocalID;
    ///履约头寸自对冲标志

```

```
TUstpFtdcSelfCloseFlagType SelfCloseFlag;  
///投机套保标志  
TUstpFtdcHedgeFlagType HedgeFlag;  
///数量  
TUstpFtdcVolumeType Volume;  
///业务单元  
TUstpFtdcBusinessUnitType BusinessUnit;  
///币种  
TUstpFtdcCurrencyIDType Currency;  
///会话号  
TUstpFtdcSessionIDType SessionID;  
///前置编号  
TUstpFtdcFrontIDType FrontID;  
///自对冲提交状态  
TUstpFtdcSubmitStatusType SubmitStatus;  
///交易日  
TUstpFtdcTradingDayType TradingDay;  
///自对冲编号  
TUstpFtdcSelfCloseSysIDType SelfCloseSysID;  
///申报日期  
TUstpFtdcDateType InsertDate;  
///插入时间  
TUstpFtdcTimeType InsertTime;  
///交易所应答时间  
TUstpFtdcTimeType InsertTimeExchange;  
///撤销时间  
TUstpFtdcTimeType CancelTime;  
///自对冲结果  
TUstpFtdcExecResultType ExecResult;  
///是否有效  
TUstpFtdcIsActiveType ActiveFlag;  
///交易编码  
TUstpFtdcClientIDType ClientID;  
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

6.2.51 OnRspQrySettlementInfo 方法

请求查询投资者结算结果响应。

函数原形:

```
void OnRspQrySettlementInfo(
    CUstpFtdcSettlementRspField *pSettlementRsp,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pSettlementRsp 指向查询投资者结算结果应答结构的地址。

查询投资者结算结果应答结构:

```
struct CUstpFtdcSettlementRspField
{
    ///交易日
    TUstpFtdcDateType TradingDay;
    ///结算编号
    TUstpFtdcSettlementIDType SettlementID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///序号
    TUstpFtdcSequenceNoType SequenceNo;
    ///消息内容
    TUstpFtdcLogStrType Content;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
```

```
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回用户结算单查询请求的 ID, 该 ID 由用户在结算单查询时指定。

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

6.2.52 OnRspTradingAccountPasswordUpdate 方法

投资者资金账户口令更新请求响应。区别于 OnRspUserPasswordUpdate 方法（交易密码的更新请求）。

函数原形:

```
void OnRspTradingAccountPasswordUpdate(
    CUstpFtdcTradingAccountPasswordUpdateRspField *pTradingAccountPasswordUpdateRsp,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pTradingAccountPasswordUpdateRsp 指向资金账户口令更新请求响应结构的地址。

资金账户口令更新请求响应结构:

```
struct CUstpFtdcTradingAccountPasswordUpdateRspField
```

```
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID; ///不验证该字段
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///旧密码
    TUstpFtdcPasswordType OldPassword;
    ///新密码
    TUstpFtdcPasswordType NewPassword;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
```

```
TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回资金帐号密码修改请求的 ID, 该 ID 由用户在资金帐号密码修改时指定。

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

6.2.53 OnRspQueryTransferSerialOffset 方法

请求查询转帐流水响应。

函数原形:

```
void OnRspQryTransferSerialOffset(
    CUstpFtdcTransferSerialFieldRspField *pTransferSerialFieldRsp,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pTransferSerialFieldRsp 指向查询转帐流水响应响应结构的地址。

查询转帐流水响应响应结构:

```
struct CUstpFtdcTransferSerialFieldRspField
{
    ///平台流水号
    TUstpFtdcPlateSerialType PlateSerial;
    ///交易发起方日期
    TUstpFtdcDateType TradeDate;
    ///交易日期
    TUstpFtdcDateType TradingDay;
    ///交易时间
    TUstpFtdcTimeType TradeTime;
    ///交易代码
    TUstpFtdcTradeCodeType TradeCode;
    ///会话编号
    TUstpFtdcSessionIDType SessionID;
    ///银行代码
    TUstpFtdcBankIDType BankID;
    ///银行分中心代码
    TUstpFtdcInvestorIDType BankBrchID;
    ///银行帐号类型
    TUstpFtdcBankAccType BankAccType;
    ///银行帐号
```

```
TUstpFtdcBankAccountType    BankAccount;
///银行流水号
TUstpFtdcBankSerialType    BankSerial;
///经纪公司编号
TUstpFtdcBrokerIDType    BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType    BrokerBranchID;
///期货公司帐号类型
TUstpFtdcFutureAccType    FutureAccType;
///资金帐号
TUstpFtdcAccountIDType    AccountID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///期货公司流水号
TUstpFtdcFutureSerialType    FutureSerial;
///交易发起方
TUstpFtdcTransferSourceType    TransferSource;
///出入金方向
TUstpFtdcTransferDirectionType    TransferDirection;
///证件类型
TUstpFtdcIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType    IdentifiedCardNo;
///币种代码
TUstpFtdcCurrencyIDType    CurrencyID;
///金额
TUstpFtdcMoneyType    TradeAmount;
///应收客户费用
TUstpFtdcMoneyType    CustFee;
///应收期货公司费用
TUstpFtdcMoneyType    BrokerFee;
///有效标志
TUstpFtdcAvailabilityFlagTypeType    AvailabilityFlag;
///操作员
TUstpFtdcOperatorCodeType    OperatorCode;
```



```
///新银行帐号
TUstpFtdcBankAccountType    BankNewAccount;
///流水返回码
TUstpFtdcTransferRtnCodeType    RtnCode;
///流水返回信息
TUstpFtdcTransferRtnMsgType    RtnMsg;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType    ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType    ErrorMsg;
};
```

nRequestID: 返回用户转账流水查询请求的 ID, 该 ID 由用户在转账流水查询时指定。

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

备注:

如果当日没有转账流水, 回调函数中的指针为 NULL, 注意判断。

示例:

```
if (pRspInfo!=NULL&& pRspInfo->ErrorID!=0)
{
    printf("-----\n");
    printf("查询期货流水失败 错误原因: %s\n", pRspInfo->ErrorMsg);
    printf("-----\n");
    return;
}
if (NULL == pTransferSerialFieldRsp)
{
    printf("没有期货流水查询数据\n");
    return;
}
printf("-----\n");
printf("查询期货流水应答\n");
printf("BrokerID =[%s]\n", pTransferSerialFieldRsp->BrokerID);
printf("AccountID =[%s]\n", pTransferSerialFieldRsp->AccountID);
printf("BankID =[%s]\n", pTransferSerialFieldRsp->BankID);
printf("BankAccount =[%s]\n", pTransferSerialFieldRsp->BankAccount);
printf("FutureSerial =[%s]\n", pTransferSerialFieldRsp->FutureSerial);
printf("BankSerial =[%s]\n", pTransferSerialFieldRsp->BankSerial);
printf("TradeAmount =[%lf]\n", pTransferSerialFieldRsp->TradeAmount);
printf("TradingDay =[%s]\n", pTransferSerialFieldRsp->TradingDay);
printf("TradeTime =[%s]\n", pTransferSerialFieldRsp->TradeTime);
printf("TradeCode =[%s]\n", pTransferSerialFieldRsp->TradeCode);
printf("TransferSource =[%s]\n", pTransferSerialFieldRsp->TransferSource);
printf("TransferDirection\n[%s]\n", pTransferSerialFieldRsp->TransferDirection);
printf("RtnCode =[%s]\n", pTransferSerialFieldRsp->RtnCode);
printf("RtnMsg =[%s]\n", pTransferSerialFieldRsp->RtnMsg);
printf("-----\n");
return ;
```

6.2.54 OnRspQueryAccountRegister 方法

请求查询银期签约关系响应。

函数原形:

```
void OnRspQueryAccountregister(
    CUstpFtdcAccountregisterRspField *pAccountregisterRsp,
    CUstpFtdcRspInfoField *pRspInfo,
```

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

参数:

pAccountregisterRsp 指向查询银期签约关系响应结构的地址。

查询银期签约关系响应结构:

```
struct CUstpFtdcAccountregisterRspField  
{  
    ///交易日期  
    TUstpFtdcDateType    TradeDate;  
    ///银行代码  
    TUstpFtdcBrokerIDType    BankID;  
    ///银行分中心代码  
    TUstpFtdcInvestorIDType    BankBrchID;  
    ///银行帐号  
    TUstpFtdcBankAccountType    BankAccount;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType    BrokerID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType    InvestorID;  
    ///资金帐号  
    TUstpFtdcAccountIDType    AccountID;  
    ///证件类型  
    TUstpFtdcIdCardTypeType    IdCardType;  
    ///证件号码  
    TUstpFtdcIdentifiedCardNoType    IdentifiedCardNo;  
    ///客户姓名  
    TUstpFtdcIndividualNameType    CustomerName;  
    ///币种代码  
    TUstpFtdcCurrencyIDType    CurrencyID;  
    ///开销户类别  
    TUstpFtdcOpenOrDestroyTypeType    OpenOrDestroy;  
    ///签约日期  
    TUstpFtdcDateType    RegDate;  
    ///解约日期  
    TUstpFtdcDateType    OutDate;  
    ///交易 ID
```

```

TUstpFtdcTIDType    TID;
///客户类型
TUstpFtdcCustTypeType    CustType;
///银行帐号类型
TUstpFtdcBankAccType    BankAccType;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

nRequestID: 返回用户签约查询请求的 ID, 该 ID 由用户在签约关系查询时指定。

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

备注:

如果没有签约记录, 回调函数中的指针为 NULL, 注意判断。

示例:

```

if (pRspInfo!=NULL&& pRspInfo->ErrorID!=0) {
    printf("签约失败 错误原因: %d, %s\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
    return;
}
if (NULL == pSignUpOrCancleAccountRspField) {
    printf("没有签约应答数据\n");
    return;
}
printf("签约应答\n");
printf("BankAccount =[%s]\n", pSignUpOrCancleAccountRspField->BankAccount);
printf("TradeCode =[%s]\n", pSignUpOrCancleAccountRspField->TradeCode);
...
printf("UserID =[%s]\n", pSignUpOrCancleAccountRspField->UserID);
return ;

```

6.2.55 OnRspFromBankToFutureByFuture 方法

期货端发起银行资金转期货应答。

函数原形：

```
void OnRspFromBankToFutureByFuture(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数：

pTransferFieldReq 指向转账请求结构的地址。

转账请求结构：

```
struct CUstpFtdcTransferFieldReqField  
{  
    ///业务功能码  
    TUSTPFTDCTRANS_TRADE_CODE_TYPE TradeCode;  
    ///业务发起方  
    TUSTPFTDCTRANS_TRADE_SOURCE_TYPE TradeSource;  
    ///银行类型  
    TUSTPFTDCTRANS_BANK_TYPE_TYPE BankType;  
    ///银行代码  
    TUSTPFTDCTRANS_BANK_ID_TYPE BankID;  
    ///银行分中心代码  
    TUSTPFTDCTRANS_BANK_BRCH_ID_TYPE BankBrchID;  
    ///经纪公司类型  
    TUSTPFTDCTRANS_BROKER_TYPE_TYPE BrokerType;  
    ///经纪公司编号  
    TUSTPFTDCTRANS_BROKER_ID_TYPE BrokerID;  
    ///期商分支机构代码  
    TUSTPFTDCTRANS_FUTURE_BRANCH_ID_TYPE BrokerBranchID;  
    ///交易日期  
    TUSTPFTDCTRANS_DATE_TYPE TradeDate;  
    ///成交时间  
    TUSTPFTDCTRANS_TIME_TYPE TradeTime;  
    ///发起方流水号  
    TUSTPFTDCTRANS_SEND_SOURCE_SERIAL_TYPE SendSourceSerial;  
    ///发起方关联流水号
```

```
TUstpFtdcTransSendSourceReletedSerialType    SendSourceReletedSerial;
///重发标志
TUstpFtdcTransResendFlagType    ResendFlag;
///出入金方向
TUstpFtdcTransTransferDirFlagType    TransferDirFlag;
///币种
TUstpFtdcTransCurrencyType    CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType    CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType    IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType    CustType;
///机构客户号
TUstpFtdcOrganCustNoType    OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType    BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType    BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType    BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType    BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType    BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType    BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType    BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType    BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType    BankAccountPwd2;
```

```
///资金帐号
TUstpFtdcTransAccountIDType AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType AccountPwd2;
///摘要
TUstpFtdcTransDigestType Digest;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///当前金额
TUstpFtdcMoneyType CurrentBalance;
///可用金额
TUstpFtdcMoneyType UsableBalance;
///可取金额
TUstpFtdcMoneyType WithdrawableBalance;
///交易日
TUstpFtdcTransDateType TradingDay;
///用户编号
TUstpFtdcUserIDType UserID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///待冲正流水号
TUstpFtdcTransSendSourceSerialType RevReference;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUSTpFtdcRspInfoField
{
    ///错误代码
    TUSTpFtdcErrorIDType ErrorID;
    ///错误信息
    TUSTpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回用户银转期请求的 ID, 该 ID 由用户在银转期请求时指定。

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

备注:

TradeCode 字典项为: 15 入金, 16 出金, 17 入金冲正, 18 出金冲正, 该回调接口只会收到 15 和 17 两个。

示例:

参考出金回调接口 OnRspFromFutureToBankByFuture

6.2.56 OnRtnFromBankToFutureByFuture 方法

期货端发起银行资金转期货回报通知。

函数原形:

```
void OnRtnFromBankToFutureByFuture(
    CUSTpFtdcTransferFieldReqField *pTransferFieldReq);
```

参数:

pTransferFieldRtn: 指向期货段发起银行资金转期货通知结构的地址。

期货段发起银行资金转期货通知结构:

```
struct CUSTpFtdcTransferFieldReqField
{
    ///业务功能码
    TUSTpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
    TUSTpFtdcTransTradeSourceType TradeSource;
    ///银行类型
    TUSTpFtdcTransBankTypeType BankType;
    ///银行代码
    TUSTpFtdcTransBankIDType BankID;
    ///银行分中心代码
    TUSTpFtdcTransBankBrchIDType BankBrchID;
```



```
///经纪公司类型
TUstpFtdcTransBrokerTypeType    BrokerType;
///经纪公司编号
TUstpFtdcTransBrokerIDType    BrokerID;
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType    BrokerBranchID;
///交易日期
TUstpFtdcTransDateType    TradeDate;
///成交时间
TUstpFtdcTransTimeType    TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType    SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType    SendSourceReletedSerial;
///重发标志
TUstpFtdcTransResendFlagType    ResendFlag;
///出入金方向
TUstpFtdcTransTransferDirFlagType    TransferDirFlag;
///币种
TUstpFtdcTransCurrencyType    CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType    CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType    IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType    CustType;
///机构客户号
TUstpFtdcOrganCustNoType    OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType    BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType    BankAccountType;
///开户分行号
```

```
TUstpFtdcTransBankBranchIDType  BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType  BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType  BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType  BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType  BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType  BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType  BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType  AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType  AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType  AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType  AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType  AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType  AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType  AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType  AccountPwd2;
///摘要
TUstpFtdcTransDigestType  Digest;
///转帐金额
TUstpFtdcMoneyType  TradeAmount;
///当前金额
TUstpFtdcMoneyType  CurrentBalance;
```

```
///可用金额
TUstpFtdcMoneyType UsableBalance;
///可取金额
TUstpFtdcMoneyType WithdrawableBalance;
///交易日
TUstpFtdcTransDateType TradingDay;
///用户编号
TUstpFtdcUserIDType UserID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///待冲正流水号
TUstpFtdcTransSendSourceSerialType RevReference;
}
```

6.2.57 OnRspFromFutureToBankByFuture 方法

期货端发起期货资金转入银行应答。

函数原形:

```
void OnRspFromFutureToBankByFuture(
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pTransferFieldReq: 指向转账请求结构的地址。

转账请求结构:

```
struct CUstpFtdcTransferFieldReqField
{
    ///业务功能码
    TUstpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
    TUstpFtdcTransTradeSourceType TradeSource;
    ///银行类型
    TUstpFtdcTransBankTypeType BankType;
    ///银行代码
    TUstpFtdcTransBankIDType BankID;
    ///银行分中心代码
```

TUstpFtdcTransBankBrchIDType BankBrchID;
///经纪公司类型

TUstpFtdcTransBrokerTypeType BrokerType;
///经纪公司编号

TUstpFtdcTransBrokerIDType BrokerID;
///期商分支机构代码

TUstpFtdcTransFutureBranchIDType BrokerBranchID;
///交易日期

TUstpFtdcTransDateType TradeDate;
///成交时间

TUstpFtdcTransTimeType TradeTime;
///发起方流水号

TUstpFtdcTransSendSourceSerialType SendSourceSerial;
///发起方关联流水号

TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;
///重发标志

TUstpFtdcTransResendFlagType ResendFlag;
///出入金方向

TUstpFtdcTransTransferDirFlagType TransferDirFlag;
///币种

TUstpFtdcTransCurrencyType CurrencyID;
///客户姓名

TUstpFtdcTransIndividualNameType CustomerName;
///证件类型

TUstpFtdcTransIdCardTypeType IdCardType;
///证件号码

TUstpFtdcTransIdentifiedCardNoType IdentifiedCardNo;
///客户类型

TUstpFtdcTransCustTypeType CustType;
///机构客户号

TUstpFtdcOrganCustNoType OrganCustNo;
///银行帐号

TUstpFtdcTransBankAccountType BankAccount;
///银行帐号类型

TUstpFtdcTransAccountTypeType BankAccountType;

```
///开户分行号
TUstpFtdcTransBankBranchIDType  BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType  BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType  BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType  BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType  BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType  BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType  BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType  AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType  AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType  AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType  AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType  AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType  AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType  AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType  AccountPwd2;
///摘要
TUstpFtdcTransDigestType  Digest;
///转帐金额
TUstpFtdcMoneyType  TradeAmount;
///当前金额
```

```
TUstpFtdcMoneyType CurrentBalance;  
///可用金额  
TUstpFtdcMoneyType UsableBalance;  
///可取金额  
TUstpFtdcMoneyType WithdrawableBalance;  
///交易日  
TUstpFtdcTransDateType TradingDay;  
///用户编号  
TUstpFtdcUserIDType UserID;  
///投资者编号  
TUstpFtdcInvestorIDType InvestorID;  
///待冲正流水号  
TUstpFtdcTransSendSourceSerialType RevReference;  
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField  
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

nRequestID: 返回用户期转银请求的 ID, 该 ID 由用户在期转银请求时指定。

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

备注:

TradeCode 字典项为: 15 入金, 16 出金, 17 入金冲正, 18 出金冲正, 该回调接口只会收到 16 和 18 两个。

示例:

```

if(strcmp(pTransferFieldReq->TradeCode,"16") == 0) {
    printf("期货出金应答\n");
    if (pRspInfo!=NULL&& pRspInfo->ErrorID!=0) {
        printf("期货发起出金失败 错误原因: %d, %s\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        return;
    }
    if (NULL == pTransferFieldReq) {
        return;
    }
    printf("-----成功-----\n");
    printf("BankAccount =[%s]\n", pTransferFieldReq->BankAccount);
    printf("TradeCode =[%s]\n", pTransferFieldReq->TradeCode);
    printf("TradeSource =[%s]\n", pTransferFieldReq->TradeSource);
    printf("tradeAmount =[%lf]\n", pTransferFieldReq->TradeAmount);
    printf("-----\n");
} else {
    printf("期货出金自动冲正应答\n");
    if (pRspInfo!=NULL&& pRspInfo->ErrorID!=0) {
        printf("期货发起出金自动冲正失败 错误原因: %d, %s\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        return;
    }
    if (NULL == pTransferFieldReq) {
        return;
    }
    printf("-----成功-----\n");
    printf("BankAccount =[%s]\n", pTransferFieldReq->BankAccount);
    printf("TradeCode =[%s]\n", pTransferFieldReq->TradeCode);
    printf("TradeSource =[%s]\n", pTransferFieldReq->TradeSource);
    ...
    printf("tradeAmount =[%lf]\n", pTransferFieldReq->TradeAmount);
    printf("-----\n");
}

```

6.2.58 OnRtnFromFutureToBankByFuture 方法

期货端发起期货资金转银行回报通知。

函数原形:

```
void OnRtnFromFutureToBankByFuture(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq);
```

参数:

pTransferFieldRtn: 指向发起资金转银行的通知结构的地址。

期货端发起期货资金转银行通知结构:

```
struct CUstpFtdcTransferFieldReqField  
{  
    ///业务功能码  
    TUstpFtdcTransTradeCodeType TradeCode;  
    ///业务发起方  
    TUstpFtdcTransTradeSourceType TradeSource;  
    ///银行类型  
    TUstpFtdcTransBankTypeType BankType;  
    ///银行代码  
    TUstpFtdcTransBankIDType BankID;  
    ///银行分中心代码  
    TUstpFtdcTransBankBrchIDType BankBrchID;  
    ///经纪公司类型  
    TUstpFtdcTransBrokerTypeType BrokerType;  
    ///经纪公司编号  
    TUstpFtdcTransBrokerIDType BrokerID;  
    ///期商分支机构代码  
    TUstpFtdcTransFutureBranchIDType BrokerBranchID;  
    ///交易日期  
    TUstpFtdcTransDateType TradeDate;  
    ///成交时间  
    TUstpFtdcTransTimeType TradeTime;  
    ///发起方流水号  
    TUstpFtdcTransSendSourceSerialType SendSourceSerial;  
    ///发起方关联流水号  
    TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;  
    ///重发标志  
    TUstpFtdcTransResendFlagType ResendFlag;  
    ///出入金方向  
    TUstpFtdcTransTransferDirFlagType TransferDirFlag;  
    ///币种
```



```
TUstpFtdcTransCurrencyType  CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType  CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType  IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType  IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType  CustType;
///机构客户号
TUstpFtdcOrganCustNoType  OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType  BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType  BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType  BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType  BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType  BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType  BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType  BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType  BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType  BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType  AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType  AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType  AccountPwdType1;
```

```
///资金账户加密方式 1
TUstpFtdcTransPwdEncType    AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType    AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType    AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType    AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType    AccountPwd2;
///摘要
TUstpFtdcTransDigestType    Digest;
///转帐金额
TUstpFtdcMoneyType    TradeAmount;
///当前金额
TUstpFtdcMoneyType    CurrentBalance;
///可用金额
TUstpFtdcMoneyType    UsableBalance;
///可取金额
TUstpFtdcMoneyType    WithdrawableBalance;
///交易日
TUstpFtdcTransDateType    TradingDay;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///待冲正流水号
TUstpFtdcTransSendSourceSerialType    RevReference;
}
```

6.2.59 OnRtnFromFutureToBankByBank 方法

银行端发起期货资金转银行通知。

函数原形:

```
void OnRtnFromFutureToBankByBank(
    CUstpFtdcTransferFieldReqField *pTransferFieldReq)
```

参数:

pTransferFieldRtn: 指向银行端发起期货资金转银行通知结构的地址。

银行端发起期货资金转银行通知结构:

参考 *OnRtnFromFutureToBankByFuture* 方法。

6.2.60 OnRtnFromBankToFutureByBank 方法

银行端发起银行资金转期货通知。

函数原形:

```
void OnRtnFromBankToFutureByBank(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq)
```

参数:

pTransferFieldRtn: 指向转账请求通知结构的地址。

银行端发起银行资金转期货通知结构:

参考 *OnRtnFromFutureToBankByFuture* 方法。

6.2.61 OnRspQueryBankAccountMoneyByFuture 方法

期货端发起查询银行余额应答。

函数原形:

```
void OnRspQueryBankAccountMoneyByFuture(  
    CUstpFtdcAccountFieldRspField * pAccountFieldRsp,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数:

pAccountFieldRsp: 指向查询账户信息请求结构的地址。

查询账户信息请求结构:

```
struct CUstpFtdcAccountFieldRspField  
{  
    ///业务功能码  
    TUstpFtdcTransTradeCodeType TradeCode;  
    ///业务发起方  
    TUstpFtdcTransTradeSourceType TradeSource;  
    ///银行类型  
    TUstpFtdcTransBankTypeType BankType;  
    ///银行代码
```

```
TUstpFtdcTransBankIDType    BankID;
///银行分中心代码
TUstpFtdcTransBankBrchIDType    BankBrchID;
///经纪公司类型
TUstpFtdcTransBrokerTypeType    BrokerType;
///经纪公司编号
TUstpFtdcTransBrokerIDType    BrokerID;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType    BrokerBranchID;
///交易日期
TUstpFtdcTransDateType    TradeDate;
///成交时间
TUstpFtdcTransTimeType    TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType    SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType    SendSourceReletedSerial;
///币种
TUstpFtdcTransCurrencyType    CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType    CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType    IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType    CustType;
///机构客户号
TUstpFtdcOrganCustNoType    OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType    BankAccount;
```

```
///银行帐号类型
TUstpFtdcTransAccountTypeType   BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType   BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType   BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType   BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType   BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType   BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType   BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType   BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType   AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType   AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType   AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType   AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType   AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType   AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType   AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType   AccountPwd2;
///摘要
TUstpFtdcTransDigestType   Digest;
///交易日
```

```
TUstpFtdcTransDateType TradingDay;  
///登录经纪公司编号  
TUstpFtdcBrokerIDType LogBrokerID;  
///登录用户代码  
TUstpFtdcUserIDType LogUserID;  
///当前金额  
TUstpFtdcMoneyType CurrentBalance;  
///可用金额  
TUstpFtdcMoneyType UsableBalance;  
///可取金额  
TUstpFtdcMoneyType WithdrawableBalance;  
///转帐金额  
TUstpFtdcMoneyType FrozenBalance;  
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField  
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

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

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

备注:

如果界面上只显示一个余额, 请使用 CurrentBalance

示例:

```

        if (pRspInfo!=NULL&& pRspInfo->ErrorID!=0) {
            printf("查询银行余额失败 错误原因: %d, %s\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
            return;
        }
        if (NULL == pAccountFieldRsp) {
            printf("没有银行余额查询数据\n");
            return;
        }
        printf("查询银行余额应答\n");
        printf("BrokerID =[%s]\n", pAccountFieldRsp->BrokerID);
        printf("BankAccount =[%s]\n", pAccountFieldRsp->BankAccount);
        printf("AccountID =[%s]\n", pAccountFieldRsp->AccountID);
        printf("CurrentBalance =[%lf]\n", pAccountFieldRsp->CurrentBalance);
        printf("UsableBalance =[%lf]\n", pAccountFieldRsp->UsableBalance);
        printf("WithdrawableBalance =[%lf]\n", pAccountFieldRsp->WithdrawableBalance);
        return ;
    }
}

```

6.2.62 OnRtnQueryBankBalanceByFuture 方法

期货端发起查询银行余额回报通知。

函数原形:

```
void OnRtnQueryBankBalanceByFuture(
    CUstpFtdcAccountFieldRtnField *pAccountFieldRtn)

```

参数:

pAccountFieldRtn: 指向查询账户信息通知结构的地址。

查询账户信息通知结构:

```

struct CUstpFtdcAccountFieldRtnField
{
    ///业务功能码
    TustpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
    TustpFtdcTransTradeSourceType TradeSource;
    ///银行类型
    TustpFtdcTransBankTypeType BankType;
}

```

```
///银行代码
TUstpFtdcTransBankIDType    BankID;
///银行分中心代码
TUstpFtdcTransBankBrchIDType    BankBrchID;
///经纪公司类型
TUstpFtdcTransBrokerTypeType    BrokerType;
///经纪公司编号
TUstpFtdcTransBrokerIDType    BrokerID;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType    BrokerBranchID;
///交易日期
TUstpFtdcTransDateType    TradeDate;
///成交时间
TUstpFtdcTransTimeType    TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType    SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType    SendSourceReletedSerial;
///币种
TUstpFtdcTransCurrencyType    CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType    CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType    IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType    CustType;
///机构客户号
TUstpFtdcOrganCustNoType    OrganCustNo;
///银行帐号
```



```
TUstpFtdcTransBankAccountType    BankAccount;
///银行帐号类型

TUstpFtdcTransAccountTypeType    BankAccountType;
///开户分行号

TUstpFtdcTransBankBranchIDType    BankBranchID;
///银行账户密码类型 1

TUstpFtdcTransPwdTypeType    BankAccountPwdType1;
///银行账户加密方式 1

TUstpFtdcTransPwdEncType    BankAccountPwdEnc1;
///银行账户密码 1

TUstpFtdcTransPwdType    BankAccountPwd1;
///银行账户密码类型 2

TUstpFtdcTransPwdTypeType    BankAccountPwdType2;
///银行账户加密方式 2

TUstpFtdcTransPwdEncType    BankAccountPwdEnc2;
///银行账户密码 2

TUstpFtdcTransPwdType    BankAccountPwd2;
///资金帐号

TUstpFtdcTransAccountIDType    AccountID;
///资金帐号类型

TUstpFtdcTransAccountTypeType    AccountType;
///资金账户密码类型 1

TUstpFtdcTransPwdTypeType    AccountPwdType1;
///资金账户加密方式 1

TUstpFtdcTransPwdEncType    AccountPwdEnc1;
///资金账户密码 1

TUstpFtdcTransPwdType    AccountPwd1;
///资金账户密码类型 2

TUstpFtdcTransPwdTypeType    AccountPwdType2;
///资金账户加密方式 2

TUstpFtdcTransPwdEncType    AccountPwdEnc2;
///资金账户密码 2

TUstpFtdcTransPwdType    AccountPwd2;
///摘要

TUstpFtdcTransDigestType    Digest;
```

```
///交易日
TUstpFtdcTransDateType TradingDay;
///登录经纪公司编号
TUstpFtdcBrokerIDType LogBrokerID;
///登录用户代码
TUstpFtdcUserIDType LogUserID;
///银行可用金额
TUstpFtdcMoneyType BankUseAmount;
///银行可取金额
TUstpFtdcMoneyType BankFetchAmount;
}
```

6.2.63 OnRtnOpenAccountByBank 方法

银行端发起的银期开户回报通知。

函数原形:

```
void OnRtnOpenAccountByBank(
    CUstpFtdcSignUpOrCancleAccountRspFieldField * pSignUpOrCancleAccountRspField)
```

参数:

pSignUpOrCancleAccountRspField: 指向银期开销户信息结构的地址。

银期开销户信息结构:

```
struct CUstpFtdcSignUpOrCancleAccountRspFieldField
{
    ///业务功能码
    TUstpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
    TUstpFtdcTransTradeSourceType TradeSource;
    ///银行类型
    TUstpFtdcTransBankTypeType BankType;
    ///银行代码
    TUstpFtdcTransBankIDType BankID;
    ///银行分中心代码
    TUstpFtdcTransBankBrchIDType BankBrchID;
    ///经纪公司类型
    TUstpFtdcTransBrokerTypeType BrokerType;
```

```
///经纪公司编号
TUstpFtdcTransBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcTransDateType TradeDate;
///成交时间
TUstpFtdcTransTimeType TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;
///币种
TUstpFtdcTransCurrencyType CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType CustType;
///机构客户号
TUstpFtdcOrganCustNoType OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType BankAccountPwdEnc1;
///银行账户密码 1
```

```
TUstpFtdcTransPwdType    BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType    BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType    BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType    BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType    AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType    AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType    AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType    AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType    AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType    AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType    AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType    AccountPwd2;
///摘要
TUstpFtdcTransDigestType    Digest;
///交易日
TUstpFtdcTransDateType    TradingDay;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///汇钞标志
TUstpFtdcTransCashExCodeType    CashExCode;
///当前余额
TUstpFtdcMoneyType    CurrentBalance;
};
```

6.2.64 OnRtnCancelAccountByBank 方法

银行端发起银期销户回报通知。

函数原形:

```
void OnRtnCancelAccountByBank(  
    CUstpFtdcSignUpOrCancleAccountRspFieldField *pSignUpOrCancleAccountRspField)
```

参数:

pAccountField: 指向银期开销户信息结构的地址。

银期开销户信息结构:

参考 *OnRtnOpenAccountByBank* 方法。

6.2.65 OnRspFromBankToFutureByBank 方法

银行端发起银行资金转期货应答。

函数原形:

```
void OnRspFromBankToFutureByBank(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast) {};
```

参数:

pTransferFieldReq: 指向资金转账信息结构的地址。

资金转账信息结构:

```
struct CUstpFtdcTransferFieldReqField  
{  
    ///业务功能码  
    TUstpFtdcTransTradeCodeType TradeCode;  
    ///业务发起方  
    TUstpFtdcTransTradeSourceType TradeSource;  
    ///银行类型  
    TUstpFtdcTransBankTypeType BankType;  
    ///银行代码  
    TUstpFtdcTransBankIDType BankID;  
    ///银行分中心代码  
    TUstpFtdcTransBankBrchIDType BankBrchID;  
    ///经纪公司类型  
    TUstpFtdcTransBrokerTypeType BrokerType;  
    ///经纪公司编号  
    TUstpFtdcTransBrokerIDType BrokerID;
```

```
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcTransDateType TradeDate;
///成交时间
TUstpFtdcTransTimeType TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;
///重发标志
TUstpFtdcTransResendFlagType ResendFlag;
///出入金方向
TUstpFtdcTransTransferDirFlagType TransferDirFlag;
///币种
TUstpFtdcTransCurrencyType CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType CustType;
///机构客户号
TUstpFtdcOrganCustNoType OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType BankAccountPwd2;
```

```

    ///资金帐号
    TUstpFtdcTransAccountIDType AccountID;
    ///资金帐号类型
    TUstpFtdcTransAccountTypeType AccountType;
    ///资金账户密码类型 1
    TUstpFtdcTransPwdTypeType AccountPwdType1;
    ///资金账户加密方式 1
    TUstpFtdcTransPwdEncType AccountPwdEnc1;
    ///资金账户密码 1
    TUstpFtdcTransPwdType AccountPwd1;
    ///资金账户密码类型 2
    TUstpFtdcTransPwdTypeType AccountPwdType2;
    ///资金账户加密方式 2
    TUstpFtdcTransPwdEncType AccountPwdEnc2;
    ///资金账户密码 2
    TUstpFtdcTransPwdType AccountPwd2;
    ///摘要
    TUstpFtdcTransDigestType Digest;
    ///转帐金额
    TUstpFtdcMoneyType TradeAmount;
    ///当前金额
    TUstpFtdcMoneyType CurrentBalance;
    ///可用金额
    TUstpFtdcMoneyType UsableBalance;
    ///可取金额
    TUstpFtdcMoneyType WithdrawableBalance;
    ///交易日
    TUstpFtdcTransDateType TradingDay;
    ///用户编号
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///待冲正流水号
    TUstpFtdcTransSendSourceSerialType RevReference;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUSTPFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.66 OnRspFromFutureToBankByBank 方法

银行端发起期货资金转银行应答。

函数原形:

```
void OnRspFromFutureToBankByBank(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast) {};
```

参数:

pTransferFieldReq: 指向资金转账信息结构的地址。

资金转账信息结构:

参考 *OnRspFromBankToFutureByBank* 方法。

6.2.67 OnRtnFromBankToFutureByJSD

金仕达端发起银行资金转期货回报通知。

函数原形:

```
void OnRtnFromBankToFutureByJSD(CUstpFtdcJSDMoneyField *pJSDMoney) {};
```

参数:

pJSDMoney: 指向金仕达资金转账信息结构的地址。

金仕达资金转账信息结构:

```
struct CUstpFtdcJSDMoneyField  
{  
    ///用户编号  
    TustpFtdcUserIDType UserID;  
    ///经纪公司编号  
    TustpFtdcBrokerIDType BrokerID;  
    ///投资者编号  
    TustpFtdcInvestorIDType InvestorID;  
    ///资金帐号  
    TustpFtdcAccountIDType AccountID;  
    ///银行代码
```



```
TUstpFtdcBankIDType BankID;  
///转帐金额  
TUstpFtdcMoneyType TradeAmount;  
///银期冻结资金流水号  
TUstpFtdcFrozenRefrenceType FrozenRefrence;  
///金仕达流水号  
TUstpFtdcJSDSerialType JSDSerial;  
};
```

6.2.68 OnRtnFromFutureToBankByJSD

金仕达端发起期货资金转银行通知。

函数原形:

```
void OnRtnFromFutureToBankByJSD(CUstpFtdcJSDMoneyField *pJSDMoney) {};
```

参数:

pJSDMoney: 指向金仕达资金转账信息结构的地址。

金仕达资金转账信息结构:

参考 *OnRtnFromBankToFutureByJSD* 方法。

6.2.69 OnRtnTransferInvestorAccountChanged 方法

银期转账投资者资金账户变动回报通知。

函数原形:

```
void OnRtnTransferInvestorAccountChanged(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq) {};
```

参数:

pTransferFieldReq: 指向资金转账信息结构的地址。

资金转账信息结构:

参考 *OnRspFromBankToFutureByBank* 方法。

6.2.70 OnRspSignUpAccountByFuture 方法

期货发起银行账户签约应答。

函数原形:

```
void OnRspSignUpAccountByFuture(  
    CUstpFtdcSignUpOrCancleAccountRspFieldField *pSignUpOrCancleAccountRspField,
```

```
CUstpFtdcRspInfoField *pRspInfo,  
int nRequestID,  
bool bIsLast) {};
```

参数:

pSignUpOrCancleAccountRspField: 指向银行账户结构的地址。

银行账户信息结构:

```
struct CUstpFtdcSignUpOrCancleAccountRspFieldField  
{  
    ///业务功能码  
    TUSTPFTDC_TRANS_TRADE_CODE_TYPE TradeCode;  
    ///业务发起方  
    TUSTPFTDC_TRANS_TRADE_SOURCE_TYPE TradeSource;  
    ///银行类型  
    TUSTPFTDC_TRANS_BANK_TYPE_TYPE BankType;  
    ///银行代码  
    TUSTPFTDC_TRANS_BANK_ID_TYPE BankID;  
    ///银行分中心代码  
    TUSTPFTDC_TRANS_BANK_BRCH_ID_TYPE BankBrchID;  
    ///经纪公司类型  
    TUSTPFTDC_TRANS_BROKER_TYPE_TYPE BrokerType;  
    ///经纪公司编号  
    TUSTPFTDC_TRANS_BROKER_ID_TYPE BrokerID;  
    ///期商分支机构代码  
    TUSTPFTDC_TRANS_FUTURE_BRANCH_ID_TYPE BrokerBranchID;  
    ///交易日期  
    TUSTPFTDC_TRANS_DATE_TYPE TradeDate;  
    ///成交时间  
    TUSTPFTDC_TRANS_TIME_TYPE TradeTime;  
    ///发起方流水号  
    TUSTPFTDC_TRANS_SEND_SOURCE_SERIAL_TYPE SendSourceSerial;  
    ///发起方关联流水号  
    TUSTPFTDC_TRANS_SEND_SOURCE_RELATED_SERIAL_TYPE SendSourceRelatedSerial;  
    ///币种  
    TUSTPFTDC_TRANS_CURRENCY_TYPE CurrencyID;  
    ///客户姓名  
    TUSTPFTDC_TRANS_INDIVIDUAL_NAME_TYPE CustomerName;
```

```
///证件类型
TUstpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType  IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType    CustType;
///机构客户号
TUstpFtdcOrganCustNoType    OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType    BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType    BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType    BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType    BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType    BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType    BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType    BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType    BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType    BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType    AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType    AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType    AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType    AccountPwdEnc1;
///资金账户密码 1
```

```

TUstpFtdcTransPwdType    AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType    AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType    AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType    AccountPwd2;
///摘要
TUstpFtdcTransDigestType    Digest;
///交易日
TUstpFtdcTransDateType    TradingDay;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///汇钞标志
TUstpFtdcTransCashExCodeType    CashExCode;
///当前余额
TUstpFtdcMoneyType    CurrentBalance;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType    ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType    ErrorMsg;
};

```

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

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

6.2.71 OnRspCancleAccountByFuture 方法

期货发起银行账户解约应答。

函数原形:

```
void OnRspCancleAccountByFuture(
```

```
CUstpFtdcSignUpOrCancleAccountRspFieldField *pSignUpOrCancleAccountRspField,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast) {};
```

参数:

pSignUpOrCancleAccountRspField: 指向银行账户信息结构的地址。

银行账户信息结构:

参考 *OnRspSignUpAccountByFuture* 方法。

6.2.72 OnRspQuerySignUpOrCancleAccountStatus 方法

请求查询银行签约状态响应。

函数原形:

```
void OnRspQuerySignUpOrCancleAccountStatus(
    CUstpFtdcQuerySignUpOrCancleAccountStatusRspFieldField
    *pQuerySignUpOrCancleAccountStatusRspField,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};
```

参数:

pQuerySignUpOrCancleAccountStatusRspField: 指向银行签约状态结构的地址。

银行签约状态信息结构:

```
struct CUstpFtdcQuerySignUpOrCancleAccountStatusRspFieldField
{
    ///用户编号
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///经纪公司编号
    TUstpFtdcTransBrokerIDType BrokerID;
    ///资金帐号
    TUstpFtdcTransAccountIDType AccountID;
    ///银行代码
    TUstpFtdcTransBankIDType BankID;
    ///银行帐号
```

```
TUstpFtdcTransBankAccountType  BankAccount;
///签约日期
TUstpFtdcTransDateType  OpenAccountDate;
///解约日期
TUstpFtdcTransDateType  CanceAccountDate;
///签约状态
TUstpFtdcTransSignUPOrCanceStatusType  SignUPOrCanceStatus;
///签约返回码
TUstpFtdcTransSignUPOrCanceRtnCodeType  SignUPOrCanceRtnCode;
///签约返回信息
TUstpFtdcTransSignUPOrCanceRtnMsgType  SignUPOrCanceRtnMsg;
///币种
TUstpFtdcTransCurrencyType  CurrencyID;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType  ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType  ErrorMsg;
};
```

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

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

备注:

签约状态, a 为已签约, 其他为未签约

示例:

```

        if (pRspInfo!=NULL&& pRspInfo->ErrorID!=0) {
            printf("查询签约状态 错误原因: %s\n", pRspInfo->ErrorMsg);
            return;
        }
        if (NULL == pQuerySignUpOrCancleAccountStatusRspField) {
            return;
        }
        printf("查询签约状态应答\n");
        printf("UserID
= [%s]\n", pQuerySignUpOrCancleAccountStatusRspField->UserID);
        printf("SignUPOrCancleStatus
= [%s]\n", pQuerySignUpOrCancleAccountStatusRspField->SignUPOrCancleStatus);
        ...
        return ;

```

6.2.73 OnRtnChangeAccountByBank 方法

银行发起变更银行账号通知。

函数原形:

```

void OnRtnChangeAccountByBank (
    CUstpFtdcChangeAccountFieldField *pChangeAccountField)

```

参数:

pChangeAccountField: 指向银期变更银行账号信息结构的地址。

银期变更银行账号信息结构:

```

struct CUstpFtdcChangeAccountRspFieldField
{
    ///业务功能码
    TustpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
    TustpFtdcTransTradeSourceType TradeSource;
    ///银行类型
    TustpFtdcTransBankTypeType BankType;
    ///银行代码

```

```
TUstpFtdcTransBankIDType    BankID;
///银行分中心代码
TUstpFtdcTransBankBrchIDType    BankBrchID;
///经纪公司类型
TUstpFtdcTransBrokerTypeType    BrokerType;
///经纪公司编号
TUstpFtdcTransBrokerIDType    BrokerID;
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType    BrokerBranchID;
///交易日期
TUstpFtdcTransDateType    TradeDate;
///成交时间
TUstpFtdcTransTimeType    TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType    SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType    SendSourceReletedSerial;
///币种
TUstpFtdcTransCurrencyType    CurrencyID;
///银行帐号
TUstpFtdcTransBankAccountType    BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType    BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType    BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType    BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType    BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType    BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType    BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType    BankAccountPwdEnc2;
```



```
///银行账户密码 2
TUstpFtdcTransPwdType    BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType AccountPwdType1;
///资金账户加密方式 1
TUstpFtdcTransPwdEncType AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType AccountPwd2;
///摘要
TUstpFtdcTransDigestType Digest;
///交易日
TUstpFtdcTransDateType TradingDay;
///用户编号
TUstpFtdcUserIDType UserID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
};
```

6.2.74 OnRspOpenSimTradeAccount 方法

仿真交易帐户申请开户的应答。

函数原形:

```
void OnRspOpenSimTradeAccount (CUstpFtdcSimTradeAccountInfoField
*pSimTradeAccountInfo, CUstpFtdcRspInfoField *pRspInfo, int nRequestID, bool
bIsLast) {};
```

参数:

pSignUpOrCancleAccountRspField: 指向银行账户信息结构的地址。

银行账户信息结构:

```
struct CUstpFtdcSimTradeAccountInfoField
{
    ///客户姓名
    TUstpFtdcIndividualNameType CustomerName;
    ///证件号码
    TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///手机
    TUstpFtdcMobilePhoneType MobilePhone;
    ///电子邮件
    TUstpFtdcEMailType EMail;
    ///开户密码
    TUstpFtdcPasswordType PassWord;
    ///用户代码
    TUstpFtdcUserIDType UserID;
};
```

6.2.75 OnRspCheckOpenSimTradeAccount 方法

仿真交易帐户查询响应。

函数原形:

```
void OnRspCheckOpenSimTradeAccount(
    CUstpFtdcSimTradeAccountInfoField *pSimTradeAccountInfo,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};
```

参数:

pSimTradeAccountInfo: 指向仿真交易帐户信息结构的地址。

仿真交易帐户信息结构:

```
struct CUstpFtdcSimTradeAccountInfoField
{
    ///客户姓名
    TUstpFtdcIndividualNameType CustomerName;
```

```
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///手机
TUstpFtdcMobilePhoneType MobilePhone;
///电子邮件
TUstpFtdcEmailType Email;
///开户密码
TUstpFtdcPasswordType PassWord;
///用户代码
TUstpFtdcUserIDType UserID;
};
```

6.2.76 OnRspQryCFMMCTradingAccountKey 方法

查询保证金监管系统经纪公司资金账户密钥响应。

函数原形:

```
void OnRspQryCFMMCTradingAccountKey(
    CUstpFtdcCFMMCTradingAccountKeyRspField *pCFMMCTradingAccountKeyRsp,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pCFMMCTradingAccountKeyRsp 指向保证金监管系统经纪公司资金账户密钥应答结构的地址。

保证金监管系统经纪公司资金账户密钥应答结构:

```
struct CUstpFtdcCFMMCTradingAccountKeyRspField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///会员编号
    TUstpFtdcParticipantIDType ParticipantID;
    ///资金帐号
    TUstpFtdcAccountIDType AccountID;
    ///密钥编号
    TUstpFtdcSequenceNoType KeyID;
    ///动态密钥
    TUstpFtdcCFMMCKeyType CurrentKey;
```

```
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回用户查询密钥请求的 ID, 该 ID 由用户在密钥查询时指定。

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

6.2.77 OnRtnMsgNotify 方法

用户消息通知。

函数原形:

```
void OnRtnMsgNotify(CUstpFtdcMsgNotifyField *pMsgNotify) {};
```

参数:

pMsgNotify: 指向用户消息结构的地址。

用户消息信息结构:

```
struct CUstpFtdcMsgNotifyField
{
    ///通知范围
    TUstpFtdcNotifyRangeType Range;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///是否需要重发
    TUstpFtdcReSendType ReSend;
    ///通知来源
    TUstpFtdcNotifySourceType Source;
    ///通知内容
    TUstpFtdcMsgType MsgContent;
    ///消息号
```

```
TUstpFtdcMsgNoType  MsgNumber;
///日期
TUstpFtdcDateType   CommandDate;
///时间
TUstpFtdcTimeType   CommandTime;
///消息主题
TUstpFtdcMsgSubjectType MsgSubject;
};
```

6.2.78 OnRtnClientPositionChange 方法

持仓变化通知。

函数原形:

```
void OnRtnClientPositionChange(
    CUstpFtdcClientPositionChangeField *pClientPositionChange) {};
```

参数:

pClientPositionChange: 指向持仓变化消息结构的地址。

持仓变化消息结构:

```
struct CUstpFtdcClientPositionChangeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType  BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///交易编码
    TUstpFtdcClientIDType  ClientID;
    ///多空方向
    TUstpFtdcDirectionType Direction;
    ///投机套保标志
    TUstpFtdcHedgeFlagType HedgeFlag;
    ///昨结算
    TUstpFtdcPriceType  PreSettlementPrice;
```

```
///初始持仓量
TUstpFtdcVolumeType PrePosition;
///总持仓量
TUstpFtdcVolumeType Position;
///开仓量
TUstpFtdcVolumeType OpenPosition;
///平仓量
TUstpFtdcVolumeType ClosePosition;
///单腿持仓量
TUstpFtdcVolumeType LegPosition;
///组合持仓量
TUstpFtdcVolumeType CombPosition;
///昨持仓剩余量
TUstpFtdcVolumeType YdPositionRemain;
///今持仓剩余量
TUstpFtdcVolumeType TodayPositionRemain;
///开仓冻结量
TUstpFtdcVolumeType FrozenOpenPosition;
///平仓冻结量
TUstpFtdcVolumeType FrozenClosingPosition;
///平昨冻结量
TUstpFtdcVolumeType FrozenClosingYdPosition;
///平今冻结量
TUstpFtdcVolumeType FrozenClosingTodayPosition;
///执行冻结量
TUstpFtdcVolumeType FrozenExecPosition;
///放弃执行冻结量
TUstpFtdcVolumeType FrozenAbandonExecPosition;
///占用保证金
TUstpFtdcMoneyType UsedMargin;
///冻结保证金
TUstpFtdcMoneyType FrozenMargin;
///单腿占用保证金
TUstpFtdcMoneyType LegUsedMargin;
///单腿冻结保证金
```

TUstpFtdcMoneyType LegFrozenMargin;
///组合占用保证金

TUstpFtdcMoneyType CombUsedMargin;
///组合冻结保证金

TUstpFtdcMoneyType CombFrozenMargin;
///交易所占用保证金

TUstpFtdcMoneyType UsedMarginExchange;
///交易所单腿占用保证金

TUstpFtdcMoneyType LegUsedMarginExchange;
///交易所组合占用保证金

TUstpFtdcMoneyType CombUsedMarginExchange;
///开仓成本

TUstpFtdcMoneyType PositionCostByDeal;
///持仓成本

TUstpFtdcMoneyType PositionCostBySettl;
///持仓盈亏计算价

TUstpFtdcPriceType PositionProfitPrice;
///逐日盯市平仓盈亏

TUstpFtdcMoneyType CloseProfitBySettl;
///逐笔对冲平仓盈亏

TUstpFtdcMoneyType CloseProfitByDeal;
///逐日盯市持仓盈亏

TUstpFtdcMoneyType PositionProfitBySettl;
///逐笔对冲持仓盈亏

TUstpFtdcMoneyType PositionProfitByDeal;
///币种

TUstpFtdcCurrencyIDType Currency;
///期权持仓市值

TUstpFtdcMoneyType OptMarketValue;
///TAS 持仓成本

TUstpFtdcMoneyType TASPositionCost;
///TAS 持仓量

TUstpFtdcVolumeType TASPosition;
///实虚值额

TUstpFtdcMoneyType InOutTheMoneyValue;

```
///用户代码
TUstpFtdcUserIDType UserID;
///今执行冻结量
TUstpFtdcVolumeType TodayFrozenExecPosition;
///昨执行冻结量
TUstpFtdcVolumeType YdFrozenExecPosition;
///今放弃执行冻结量
TUstpFtdcVolumeType TodayFrozenAbandonExecPosition;
///昨放弃执行冻结量
TUstpFtdcVolumeType YdFrozenAbandonExecPosition;
};
```

6.2.79 OnRtnClientPositionChangeMD 方法

浮动盈亏导致的持仓变化通知。

函数原形:

```
void OnRtnClientPositionChangeMD(
    CUstpFtdcClientPositionProfitChangeField *pClientPositionProfitChange) {};
```

参数:

pClientPositionProfitChange: 指向浮动盈亏持仓变化消息结构的地址。

浮动盈亏持仓变化消息结构:

```
///持仓浮动盈亏变化
struct CUstpFtdcClientPositionProfitChangeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///交易编码
    TUstpFtdcClientIDType ClientID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///多空方向
    TUstpFtdcDirectionType Direction;
    ///投机套保标志
```



```
TUstpFtdcHedgeFlagType HedgeFlag;
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///币种
TUstpFtdcCurrencyIDType Currency;
///占用保证金
TUstpFtdcMoneyType UsedMargin;
///单腿占用保证金
TUstpFtdcMoneyType LegUsedMargin;
///组合占用保证金
TUstpFtdcMoneyType CombUsedMargin;
///逐日盯市持仓盈亏
TUstpFtdcMoneyType PositionProfitBySettl;
///逐笔对冲持仓盈亏
TUstpFtdcMoneyType PositionProfitByDeal;
///期权持仓市值
TUstpFtdcMoneyType OptMarketValue;
///TAS 持仓成本
TUstpFtdcMoneyType TASPositionCost;
///TAS 持仓量
TUstpFtdcVolumeType TASPosition;
///实虚值额
TUstpFtdcMoneyType InOutTheMoneyValue;
///用户代码
TUstpFtdcUserIDType UserID;
};
```

6.2.80 OnRtnInvestorAccountChange 方法

资金变化通知。

函数原形：

```
void OnRtnInvestorAccountChange(
    CUstpFtdcInvestorAccountChangeField *pInvestorAccountChange) {}
```

参数：

pInvestorAccountChange：指向资金变化结构的地址。

资金变化结构：

```
struct CUstpFtdcInvestorAccountChangeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///资金帐号
    TUstpFtdcAccountIDType  AccountID;
    ///上次结算准备金
    TUstpFtdcMoneyType   PreBalance;
    ///入金金额
    TUstpFtdcMoneyType   Deposit;
    ///出金金额
    TUstpFtdcMoneyType   Withdraw;
    ///冻结的保证金
    TUstpFtdcMoneyType   FrozenMargin;
    ///冻结手续费
    TUstpFtdcMoneyType   FrozenFee;
    ///冻结权利金
    TUstpFtdcMoneyType   FrozenPremium;
    ///手续费
    TUstpFtdcMoneyType   Fee;
    ///平仓盈亏
    TUstpFtdcMoneyType   CloseProfit;
    ///持仓盈亏
    TUstpFtdcMoneyType   PositionProfit;
    ///可用资金
    TUstpFtdcMoneyType   Available;
    ///多头冻结的保证金
    TUstpFtdcMoneyType   LongFrozenMargin;
    ///空头冻结的保证金
    TUstpFtdcMoneyType   ShortFrozenMargin;
    ///多头占用保证金
    TUstpFtdcMoneyType   LongMargin;
    ///空头占用保证金
```

TUstpFtdcMoneyType ShortMargin;
///当日释放保证金

TUstpFtdcMoneyType ReleaseMargin;
///动态权益

TUstpFtdcMoneyType DynamicRights;
///今日出入金

TUstpFtdcMoneyType TodayInOut;
///占用保证金

TUstpFtdcMoneyType Margin;
///期权权利金收支

TUstpFtdcMoneyType Premium;
///风险度

TUstpFtdcMoneyType Risk;
///上日可用资金

TUstpFtdcMoneyType PreAvailable;
///总冻结持仓

TUstpFtdcVolumeType TotalFrozenPos;
///其他费用

TUstpFtdcMoneyType OtherFee;
///结算准备金

TUstpFtdcMoneyType Balance;
///质押金额

TUstpFtdcMoneyType Mortgage;
///币种

TUstpFtdcCurrencyIDType Currency;
///帐户交易状态

TUstpFtdcAccountTradingStatusType TradingStatus;
///帐户资金状态

TUstpFtdcAccountMoneyStatusType MoneyStatus;
///上日货币质押金额

TUstpFtdcMoneyType PreCurrencyPledge;
///货币质押入金额

TUstpFtdcMoneyType CurrencyPledgeIn;
///货币质押出金额

TUstpFtdcMoneyType CurrencyPledgeOut;

///外币汇入金额
TUstpFtdcMoneyType CurrencyExchangeIn;
///外币汇出金额
TUstpFtdcMoneyType CurrencyExchangeOut;
///冻结资金
TUstpFtdcMoneyType FrozenMoney;
///保底金额
TUstpFtdcMoneyType MinGuarantee;
///可提比例
TUstpFtdcRatioType WithdrawRatioAllowed;
///交割保证金
TUstpFtdcMoneyType DeliveryMargin;
///可提资金
TUstpFtdcMoneyType WithdrawAllowed;
///上日实有货币资金
TUstpFtdcMoneyType PreDepositEnd;
///上日本币质押为外币金额
TUstpFtdcMoneyType PreCurrencyPledgeOut;
///货币解质入金额
TUstpFtdcMoneyType CurrencyUnPledgeIn;
///货币解质出金额
TUstpFtdcMoneyType CurrencyUnPledgeOut;
///用户代码
TUstpFtdcUserIDType UserID;
///是否有持仓或成交
TUstpFtdcCharBoolType HasPosOrTrade;
///逐笔对冲持仓盈亏
TUstpFtdcMoneyType PositionProfitByDeal;
///逐笔对冲平仓盈亏
TUstpFtdcMoneyType CloseProfitByDeal;
///资金计算配置串
TUstpFtdcMoneyCalcCfgType MoneyCalcCfg;
///换汇出金冻结资金
TUstpFtdcMoneyType ExchangeOutFrozenMoney;
///换汇开仓冻结资金

```
TUstpFtdcMoneyType  ExchangeOpenFrozenMoney;  
///交易所占用保证金  
TUstpFtdcMoneyType  MarginExchange;  
///交易所优惠前多头占用保证金  
TUstpFtdcMoneyType  LongMarginExchange;  
///交易所优惠前空头占用保证金  
TUstpFtdcMoneyType  ShortMarginExchange;  
///交易所当日释放保证金  
TUstpFtdcMoneyType  ReleaseMarginExchange;  
///交易所交割保证金  
TUstpFtdcMoneyType  DeliveryMarginExchange;  
///期权买方持仓市值  
TUstpFtdcMoneyType  LongOptionMarketValue;  
///期权卖方持仓市值  
TUstpFtdcMoneyType  ShortOptionMarketValue;  
///市值权益  
TUstpFtdcMoneyType  MarketValue;  
///总冻结资金  
TUstpFtdcMoneyType  TotalFrozen;  
///权利金收入  
TUstpFtdcMoneyType  PremiumIn;  
///权利金支出  
TUstpFtdcMoneyType  PremiumOut;  
///行权手续费冻结  
TUstpFtdcMoneyType  FrozenExecFee;  
};
```

6.2.81 OnRtnInvestorAccountChangeMD 方法

浮动盈亏导致的资金变化通知。

函数原形:

```
void OnRtnInvestorAccountChangeMD(  
    CUstpFtdcInvestorAccountProfitChangeField *pInvestorAccountProfitChange) {};
```

参数:

pInvestorAccountProfitChange: 指向浮动盈亏资金变化消息结构的地址。

浮动盈亏资金变化消息结构:

///资金浮动盈亏变化

```
struct CUstpFtdcInvestorAccountProfitChangeField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///投资者编号

```
TUstpFtdcInvestorIDType InvestorID;
```

///资金帐号

```
TUstpFtdcAccountIDType AccountID;
```

///币种

```
TUstpFtdcCurrencyIDType Currency;
```

///占用保证金

```
TUstpFtdcMoneyType Margin;
```

///优惠前多头占用保证金

```
TUstpFtdcMoneyType LongMargin;
```

///优惠前空头占用保证金

```
TUstpFtdcMoneyType ShortMargin;
```

///期权买方持仓市值

```
TUstpFtdcMoneyType LongOptionMarketValue;
```

///期权卖方持仓市值

```
TUstpFtdcMoneyType ShortOptionMarketValue;
```

///市值权益

```
TUstpFtdcMoneyType MarketValue;
```

///可提资金

```
TUstpFtdcMoneyType WithdrawAllowed;
```

///风险度

```
TUstpFtdcMoneyType Risk;
```

///动态权益

```
TUstpFtdcMoneyType DynamicRights;
```

///可用资金

```
TUstpFtdcMoneyType Available;
```

///持仓盈亏

```
TUstpFtdcMoneyType PositionProfit;
```

///逐笔对冲持仓盈亏

```

TUstpFtdcMoneyType PositionProfitByDeal;
///用户代码
TUstpFtdcUserIDType UserID;
};

```

6.2.82 OnRspQryEnableRtnMoneyPositoinChange 方法

允许资金持仓变化通知应答。

函数原形:

```

void OnRspQryEnableRtnMoneyPositoinChange(
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};

```

参数:

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

nRequestID: 返回用户请求的 ID, 该 ID 由用户在查询时指定。

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

6.2.83 OnRspQueryHisOrder 方法

历史订单查询应答。

函数原形:

```

void OnRspQueryHisOrder(
    CUstpFtdcOrderField *pOrder,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};

```

参数:

pOrder: 指向订单结构的地址。

订单信息结构:

参考 *OnRspQryOrder* 方法。

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回用户请求的 ID, 该 ID 由用户在查询时指定。

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

6.2.84 OnRspQueryHisTrade 方法

历史成交单查询应答。

函数原形:

```
void OnRspQueryHisTrade(
    CUstpFtdcTradeField *pTrade,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) {};
```

参数:

pTrade: 指向成交结构的地址。

成交信息结构:

```
struct CUstpFtdcTradeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///交易日
    TUstpFtdcTradingDayType TradingDay;
    ///会员编号
    TUstpFtdcParticipantIDType ParticipantID;
    ///下单席位号
```



```
TUstpFtdcSeatIDType SeatID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///客户号
TUstpFtdcClientIDType ClientID;
///用户编号
TUstpFtdcUserIDType UserID;
///下单用户编号
TUstpFtdcUserIDType OrderUserID;
///成交编号
TUstpFtdcTradeIDType TradeID;
///系统报单编号
TUstpFtdcOrderSysIDType OrderSysID;
///本地报单编号
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///买卖方向
TUstpFtdcDirectionType Direction;
///开平标志
TUstpFtdcOffsetFlagType OffsetFlag;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///成交价格
TUstpFtdcPriceType TradePrice;
///成交数量
TUstpFtdcVolumeType TradeVolume;
///成交时间
TUstpFtdcTimeType TradeTime;
///清算会员编号
TUstpFtdcParticipantIDType ClearingPartID;
///本次成交手续费
TUstpFtdcMoneyType UsedFee;
///本次成交占用保证金
TUstpFtdcMoneyType UsedMargin;
```

```
///本次成交占用权利金
TUstpFtdcMoneyType  Premium;
///持仓表今持仓量
TUstpFtdcVolumeType  Position;
///持仓表今日持仓成本
TUstpFtdcPriceType  PositionCost;
///资金表可用资金
TUstpFtdcMoneyType  Available;
///资金表占用保证金
TUstpFtdcMoneyType  Margin;
///资金表冻结的保证金
TUstpFtdcMoneyType  FrozenMargin;
///本地业务标识
TUstpFtdcBusinessLocalIDType  BusinessLocalID;
///业务发生日期
TUstpFtdcDateType  ActionDay;
///策略类别
TUstpFtdcArbiTypeType  ArbiType;
///合约编号(或者为套利组合合约号)
TUstpFtdcInstrumentIDType  ArbiInstrumentID;
///平仓盈亏
TUstpFtdcMoneyType  CloseProfit;
///投机逐笔对冲平仓盈亏
TUstpFtdcMoneyType  CloseProfitByDeal;
///交易类型
TUstpFtdcTradeTypeType  TradeType;
///资金帐号
TUstpFtdcAccountIDType  AccountID;
///本地报单编号
TUstpFtdcOrderLocalIDType  OrderLocalID;
///本次交易所占用保证金
TUstpFtdcMoneyType  UsedMarginExchange;
///资金表交易所占用保证金
TUstpFtdcMoneyType  MarginExchange;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

nRequestID: 返回用户请求的 ID, 该 ID 由用户在查询时指定。

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

6.2.85 OnRtnInvestorAccountDeposit 方法

出入金结果回报。当柜台端发起了出入金时, 飞马平台主动通知客户端, 该方法会被调用。

函数原形:

```
void OnRtnInvestorAccountDeposit(
    CUstpFtdcInvestorAccountDepositResField* pInvestorAccountDepositRes)
```

参数:

pInvestorAccountDepositRes: 指向出入金结果回报结构的地址。

出入金结果回报结构:

```
struct CUstpFtdcInvestorAccountDepositResField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///资金帐号
    TUstpFtdcAccountIDType AccountID;
    ///资金流水号
    TUstpFtdcAccountSeqNoType AccountSeqNo;
    ///金额
    TUstpFtdcMoneyType Amount;
    ///出入金方向
    TUstpFtdcAccountDirectionType AmountDirection;
```

```
///可用资金
TUstpFtdcMoneyType Available;
///结算准备金
TUstpFtdcMoneyType Balance;
};
```

6.2.86 OnRspCurrencyPledge 方法

请求货币质押/解质响应。

函数原形:

```
void OnRspCurrencyPledge(CUstpFtdcCurrencyPledgeField *pCurrencyPledge,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pCurrencyPledge 指向货币质押解质响应结构的地址。

货币质押解质响应结构:

```
struct CUstpFtdcCurrencyPledgeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///资金帐号
    TUstpFtdcAccountIDType AccountID;
    ///资金流水号
    TUstpFtdcAccountSeqNoType AccountSeqNo;
    ///币种
    TUstpFtdcCurrencyIDType SrcCurrency;
    ///金额
    TUstpFtdcMoneyType SrcAmount;
    ///币种
    TUstpFtdcCurrencyIDType DestCurrency;
    ///金额
    TUstpFtdcMoneyType DestAmount;
    ///质押动作类型
    TUstpFtdcPledgeActionTypeType Action;
    ///交易日
    TUstpFtdcTradingDayType TradingDay;
```

```

        ///交易时间
        TUstpFtdcTimeType   TradingTime;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

nRequestID: 返回查询转账银行请求的 ID, 该 ID 由用户在转账银行查询时指定。

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

6.2.87 OnRtnTransferInvestorMoneyChanged 方法

银期转账投资者资金账户变动通知。

函数原形:

```

void OnRtnTransferInvestorMoneyChanged(CUstpFtdcSyncMoneyTransferField
*pSyncMoneyTransfer)

```

参数:

pSyncMoneyTransfer 指向投资者资金账户结构的地址。

投资者资金账户结构:

```

struct CUstpFtdcSyncMoneyTransferField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///资金帐号
    TUstpFtdcAccountIDType  AccountID;
    ///银行代码
    TUstpFtdcTransBankIDType BankID;
    ///银行帐号
    TUstpFtdcTransBankAccountType BankAccount;
    ///币种
    TUstpFtdcCurrencyIDType Currency;
    ///金额
    TUstpFtdcMoneyType   Amount;
};

```

```

///出入金事件通知
TUstpFtdcTransferEventType SyncTransMoneyEvent;
///银期业务功能码
TUstpFtdcTransTradeCodeType TradeCode;
///业务发起方
TUstpFtdcTradeSyncSourceType TradeSource;
///主席流水号
TUstpFtdcTransSerialType TransSerialNo;
///次席流水号
TUstpFtdcTransSerialType SecondSerialNo;
///用户编号
TUstpFtdcUserIDType UserID;
///DXP 消息流水号
TUstpFtdcDxpSeqNoType SeqNo;
///操作源(清算, 交易等)
TUstpFtdcOperateSourceType OperateSource;
///操作源目的地(清算, 交易等)
TUstpFtdcOperateSourceType OperateDest;
///操作者
TUstpFtdcIndividualNameType Operator;
///日期
TUstpFtdcDateType CommandDate;
TUstpFtdcDateType Date;
///时间
TUstpFtdcTimeType CommandTime;
TUstpFtdcTimeType Time;
};

```

6.2.88 OnRspQueryMaxOrderVolume 方法

查询最大报单数量响应。当客户端发出过查询最大报单数量指令后，飞马平台返回响应时，该方法会被调用。

函数原形:

```

void OnRspQueryMaxOrderVolume(
    CUstpFtdcRspQueryMaxOrderVolumeField *pRspQueryMaxOrderVolume,
    CUstpFtdcRspInfoField *pRspInfo,

```

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

参数:

pRspQueryMaxOrderVolume : 指向最大报单数量响应结构的地址。

最大可开/可平数量响应结构:

```
struct CUstpFtdcRspQueryMaxOrderVolumeField
```

```
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType   BrokerID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///客户代码  
    TUstpFtdcClientIDType   ClientID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///买卖方向  
    TUstpFtdcDirectionType  Direction;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType  HedgeFlag;  
    ///开平标志  
    TUstpFtdcOffsetFlagType OffsetFlag;  
    ///价格  
    TUstpFtdcPriceType      Price;  
    ///最大允许报单数量  
    TUstpFtdcVolumeType     MaxVolume;  
};
```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
```

```
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

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

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

注：

1. 在查询可开手数时，考虑了手续费和保证金的占用。组合、大边、对锁等保证金优惠暂不考虑，套利组合、展期互换定单暂不考虑。

2. ClientID 只适用用于计算境外客户的可平数量，对于境外客户该字段必填，对于境内客户若填写该字段飞马会忽略。

3. 可平手数：郑商所不区分投机套保属性，可以任填一项；中金所、上期所、能源中心、大商所区分投机套保属性，须准确填入。

4. 若 Price 不填，则按最新价计算；若 Price 超过涨跌停价或者是错误价格，则按最新价计算。

5. 若是平仓，则区分可平/可平昨/可平今最大数量。若是开仓，则是最大开仓数量。

6. 计算公式：

期货：可开手数=向下取整[可用资金/（每手所需的保证金+每手所需的手续费）]

买期权：可开手数=向下取整[可用资金/（每手权利金+每手所需的手续费）]

卖期权：可开手数=向下取整[可用资金/（每手所需的保证金-每手权利金+每手所需的手续费）]

6.2.89 OnRtnFlowMessageCancel 方法（未启用）

数据流回退通知。

函数原形：

```
void OnRtnFlowMessageCancel(  
    CUstpFtdcFlowMessageCancelField *pFlowMessageCancel)
```

6.2.90 OnErrRtnFromFutureToBankByBank 方法（未启用）

期货发起期货资金转银行错误回报。

函数原形：

```
void OnErrRtnFromFutureToBankByBank(  
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,  
    CUstpFtdcRspInfoField *pRspInfo)
```

参数：

pTransferFieldReq：指向转账请求结构的地址。

转账请求结构：

```
struct CUstpFtdcTransferFieldReqField  
{  
    ///业务功能码  
    TUstpFtdcTradeCodeType TradeCode;  
    ///银行代码
```



```
TUstpFtdcBrokerIDType   BankID;
///银行分中心代码

TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号

TUstpFtdcBrokerIDType   BrokerID;
///期商分支机构代码

TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期

TUstpFtdcDateType       TradeDate;
///成交时间

TUstpFtdcTimeType       TradeTime;
///银行流水号

TUstpFtdcBankSerialType BankSerial;
///交易日

TUstpFtdcDateType       TradingDay;
///平台流水号

TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志

TUstpFtdcBoolType       LastFragment;
///登录的 SessionID

TUstpFtdcSessionIDType  SessionID;
///客户姓名

TUstpFtdcIndividualNameType CustomerName;
///证件类型

TUstpFtdcIdCardTypeType IdCardType;
///证件号码

TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型

TUstpFtdcCustTypeType    CustType;
///银行帐号

TUstpFtdcBankAccountType BankAccount;
///银行密码

TUstpFtdcPasswordType    BankPassWord;
///资金帐号

TUstpFtdcAccountIDType   AccountID;
```

```
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType Message;
///摘要
TUstpFtdcMessageType Digest;
///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///渠道标志
TUstpFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
```

```

TUstpFtdcBankAccountType    BankSecuAcc;
///银行密码标志

TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志

TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员

TUstpFtdcOperNoType    OperNo;
///请求编号

TUstpFtdcRequestIDType    RequestID;
///交易 ID

TUstpFtdcTIDType    TID;
///转账交易状态

TUstpFtdcTransferStatusType    TransferStatus;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType    ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType    ErrorMsg;
};

```

6.2.91 OnErrRtnFromBankToFutureByFuture 方法（未启用）

期货发起银行资金转期货错误回报。

函数原形:

```

void OnErrRtnFromBankToFutureByFuture(
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,
    CUstpFtdcRspInfoField *pRspInfo)

```

参数:

pTransferFieldReq: 指向转账请求结构的地址。

转账请求结构:

```

struct CUstpFtdcTransferFieldReqField
{
    ///业务功能码
    TUstpFtdcTradeCodeType    TradeCode;

```

```
///银行代码
TUstpFtdcBrokerIDType   BankID;
///银行分中心代码
TUstpFtdcInvestorIDType  BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType   BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType       TradeDate;
///成交时间
TUstpFtdcTimeType       TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType       TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType       LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType  SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType    CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType    BankPassWord;
///资金帐号
```

```
TUstpFtdcAccountIDType  AccountID;
///期货密码
TUstpFtdcPasswordType   Password;
///安装编号
TUstpFtdcInstallIDType  InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType     UserID;
///客户证件号码标志
TUstpFtdcBoolType       VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType      TradeAmount;
///期货可取金额
TUstpFtdcMoneyType      FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType      CustFee;
///应收期货公司费用
TUstpFtdcMoneyType      BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType    Message;
///摘要
TUstpFtdcMessageType    Digest;
///银行帐号类型
TUstpFtdcBankAccType    BankAccType;
///渠道标志
TUstpFtdcDeviceIdType   DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType    BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType BrokerIDByBank;
```

```

///期货单位帐号
TUstpFtdcBankAccountType    BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType    OperNo;
///请求编号
TUstpFtdcRequestIDType    RequestID;
///交易 ID
TUstpFtdcTIDType    TID;
///转账交易状态
TUstpFtdcTransferStatusType    TransferStatus;
};

```

pRspInfo: 返回用户响应信息的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType    ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType    ErrorMsg;
};

```

6.2.92 OnErrRtnQueryBankBalanceByFuture 方法（未启用）

期货发起查询银行余额错误回报。

函数原形:

```

void OnErrRtnQueryBankBalanceByFuture(
    CUstpFtdcAccountFieldReqField *pAccountFieldReq,
    CUstpFtdcRspInfoField *pRspInfo)

```

参数:

pAccountFieldReq: 指向查询账户信息请求结构的地址。

查询账户信息请求

```

struct CUstpFtdcAccountFieldReqField
{
    ///业务功能码

```

```
TUstpFtdcTradeCodeType TradeCode;
///银行代码
TUstpFtdcBrokerIDType BankID;
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType TradeDate;
///成交时间
TUstpFtdcTimeType TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType BankPassWord;
```

```
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///摘要
TUstpFtdcMessageType Digest;
///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///渠道标志
TUstpFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType OperNo;
///请求编号
TUstpFtdcRequestIDType RequestID;
///交易 ID
```



```
TUstpFtdcTIDType    TID;
};
pRspInfo: 返回用户响应信息的地址。
响应信息结构:
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

6.2.93 OnRtnRepealFromBankToFutureByBank 方法（未启用）

银行发起冲正银行转期货通知。

函数原形:

```
void OnRtnRepealFromBankToFutureByBank(
    CUstpFtdcRepealFieldRtnField *pRepealFieldRtn)
```

参数:

pRepealFieldRtn: 指向冲正通知结构的地址。

冲正通知结构:

```
struct CUstpFtdcRepealFieldRtnField
{
    ///冲正时间间隔
    TUstpFtdcRepealTimeIntervalType RepealTimeInterval;
    ///已经冲正次数
    TUstpFtdcRepealedTimesType RepealedTimes;
    ///银行冲正标志
    TUstpFtdcBankRepealFlagType BankRepealFlag;
    ///已经冲正次数
    TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;
    ///被冲正平台流水号
    TUstpFtdcPlateSerialType PlateRepealSerial;
    ///被冲正银行流水号
    TUstpFtdcBankSerialType BankRepealSerial;
    ///被冲正期货公司流水号
    TUstpFtdcFutureSerialType FutureRepealSerial;
    ///业务功能码
```

```
TUstpFtdcTradeCodeType  TradeCode;
///银行代码
TUstpFtdcBrokerIDType   BankID;
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType   BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType       TradeDate;
///成交时间
TUstpFtdcTimeType       TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType       TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType       LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType  SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType    CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType    BankPassWord;
```

```
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType Message;
///摘要
TUstpFtdcMessageType Digest;
///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///渠道标志
TUstpFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType BankSecuAccType;
///期货公司银行编码
```

```

TUstpFtdcBankCodingForFutureType    BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType    BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType    OperNo;
///请求编号
TUstpFtdcRequestIDType    RequestID;
///交易 ID
TUstpFtdcTIDType    TID;
///转账交易状态
TUstpFtdcTransferStatusType    TransferStatus;
///错误代码
TUstpFtdcErrorIDType    ErrorID;
///错误信息
TUstpFtdcErrorMsgType    ErrorMsg;
};

```

6.2.94 OnRtnRepealFromFutureToBankByBank 方法（未启用）

银行发起冲正期货转银行通知。

函数原形:

```

void OnRtnRepealFromFutureToBankByBank(
    CUstpFtdcRepealFieldRtnField *pRepealFieldRtn)

```

参数:

pRepealFieldRtn: 指向冲正通知结构的地址。

冲正通知结构:

```

struct CUstpFtdcRepealFieldRtnField
{
    ///冲正时间间隔
    TUstpFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TUstpFtdcRepealedTimesType    RepealedTimes;

```

```
///银行冲正标志
TUstpFtdcBankRepealFlagType BankRepealFlag;
///已经冲正次数
TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;
///被冲正平台流水号
TUstpFtdcPlateSerialType PlateRepealSerial;
///被冲正银行流水号
TUstpFtdcBankSerialType BankRepealSerial;
///被冲正期货公司流水号
TUstpFtdcFutureSerialType FutureRepealSerial;
///业务功能码
TUstpFtdcTradeCodeType TradeCode;
///银行代码
TUstpFtdcBrokerIDType BankID;
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType TradeDate;
///成交时间
TUstpFtdcTimeType TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
```

```
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType BankPassWord;
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
```

```
///发送方给接收方的消息
TUstpFtdcMessageType    Message;
///摘要
TUstpFtdcMessageType    Digest;
///银行帐号类型
TUstpFtdcBankAccType    BankAccType;
///渠道标志
TUstpFtdcDeviceIdType    DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType    BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType    BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType    BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType    OperNo;
///请求编号
TUstpFtdcRequestIDType    RequestID;
///交易 ID
TUstpFtdcTIDType    TID;
///转账交易状态
TUstpFtdcTransferStatusType    TransferStatus;
///错误代码
TUstpFtdcErrorIDType    ErrorID;
///错误信息
TUstpFtdcErrorMsgType    ErrorMsg;
};
```

6.2.95 OnRtnRepealFromBankToFutureByFutureManual 方法（未启用）

系统运行时期货端手工发起冲正银行转期货请求，银行处理完毕后报盘发回的通知。

函数原形：

```
void OnRtnRepealFromBankToFutureByFutureManual(  
    CUstpFtdcRepealFieldRtnField *pRepealFieldRtn)
```

参数:

pRepealFieldRtn: 指向冲正通知结构的地址。

冲正通知结构:

```
struct CUstpFtdcRepealFieldRtnField  
{  
    ///冲正时间间隔  
    TUstpFtdcRepealTimeIntervalType RepealTimeInterval;  
    ///已经冲正次数  
    TUstpFtdcRepealedTimesType RepealedTimes;  
    ///银行冲正标志  
    TUstpFtdcBankRepealFlagType BankRepealFlag;  
    ///已经冲正次数  
    TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;  
    ///被冲正平台流水号  
    TUstpFtdcPlateSerialType PlateRepealSerial;  
    ///被冲正银行流水号  
    TUstpFtdcBankSerialType BankRepealSerial;  
    ///被冲正期货公司流水号  
    TUstpFtdcFutureSerialType FutureRepealSerial;  
    ///业务功能码  
    TUstpFtdcTradeCodeType TradeCode;  
    ///银行代码  
    TUstpFtdcBrokerIDType BankID;  
    ///银行分中心代码  
    TUstpFtdcInvestorIDType BankBrchID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///期商分支机构代码  
    TUstpFtdcFutureBranchIDType BrokerBranchID;  
    ///交易日期  
    TUstpFtdcDateType TradeDate;  
    ///成交时间  
    TUstpFtdcTimeType TradeTime;  
    ///银行流水号
```



```
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType BankPassWord;
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
```

///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType Message;
///摘要
TUstpFtdcMessageType Digest;
///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///渠道标志
TUstpFtdcDeviceIdType DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType OperNo;
///请求编号
TUstpFtdcRequestIDType RequestID;
///交易 ID
TUstpFtdcTIDType TID;
///转账交易状态

```
TUstpFtdcTransferStatusType TransferStatus;  
///错误代码  
TUstpFtdcErrorIDType ErrorID;  
///错误信息  
TUstpFtdcErrorMsgType ErrorMsg;  
};
```

6.2.96 OnErrRtnRepealBankToFutureByFutureManual 方法（未启用）

系统运行时期货端手工发起冲正银行转期货错误回报。

函数原形：

```
void OnErrRtnRepealBankToFutureByFutureManual(  
    CUstpFtdcRepealFieldRspField *pRepealFieldRsp,  
    CUstpFtdcRspInfoField *pRspInfo)
```

参数：

pRepealFieldRsp: 指向冲正信息结构的地址。

冲正信息结构：

```
struct CUstpFtdcRepealFieldRspField  
{  
    ///冲正时间间隔  
    TUstpFtdcRepealTimeIntervalType RepealTimeInterval;  
    ///已经冲正次数  
    TUstpFtdcRepealedTimesType RepealedTimes;  
    ///银行冲正标志  
    TUstpFtdcBankRepealFlagType BankRepealFlag;  
    ///已经冲正次数  
    TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;  
    ///被冲正平台流水号  
    TUstpFtdcPlateSerialType PlateRepealSerial;  
    ///被冲正银行流水号  
    TUstpFtdcBankSerialType BankRepealSerial;  
    ///被冲正期货公司流水号  
    TUstpFtdcFutureSerialType FutureRepealSerial;  
    ///业务功能码  
    TUstpFtdcTradeCodeType TradeCode;  
    ///银行代码  
    TUstpFtdcBrokerIDType BankID;
```

```
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType TradeDate;
///成交时间
TUstpFtdcTimeType TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType BankPassWord;
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
```

```
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType Message;
///摘要
TUstpFtdcMessageType Digest;
///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///渠道标志
TUstpFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType BankSecuAcc;
```

```

///银行密码标志
TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType    OperNo;
///请求编号
TUstpFtdcRequestIDType    RequestID;
///交易 ID
TUstpFtdcTIDType    TID;
///转账交易状态
TUstpFtdcTransferStatusType    TransferStatus;
};
pRspInfo: 返回用户响应信息的地址。
响应信息结构:
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType    ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType    ErrorMsg;
};

```

6.2.97 OnRtnRepealFromFutureToBankByFutureManual 方法（未启用）

系统运行时期货端手工发起冲正期货转银行请求，银行处理完毕后报盘发回的通知。

函数原形：

```

void OnRtnRepealFromFutureToBankByFutureManual (
    CUstpFtdcRepealFieldRtnField *pRepealFieldRtn)

```

参数：

pRepealFieldRtn: 指向冲正通知结构的地址。

冲正通知结构：

```

struct CUstpFtdcRepealFieldRtnField
{
    ///冲正时间间隔
    TUstpFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TUstpFtdcRepealedTimesType    RepealedTimes;

```

```
///银行冲正标志
TUstpFtdcBankRepealFlagType BankRepealFlag;
///已经冲正次数
TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;
///被冲正平台流水号
TUstpFtdcPlateSerialType PlateRepealSerial;
///被冲正银行流水号
TUstpFtdcBankSerialType BankRepealSerial;
///被冲正期货公司流水号
TUstpFtdcFutureSerialType FutureRepealSerial;
///业务功能码
TUstpFtdcTradeCodeType TradeCode;
///银行代码
TUstpFtdcBrokerIDType BankID;
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType TradeDate;
///成交时间
TUstpFtdcTimeType TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
```

```
TUstpFtdcIndividualNameType CustomerName;  
///证件类型  
TUstpFtdcIdCardTypeType IdCardType;  
///证件号码  
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;  
///客户类型  
TUstpFtdcCustTypeType CustType;  
///银行帐号  
TUstpFtdcBankAccountType BankAccount;  
///银行密码  
TUstpFtdcPasswordType BankPassWord;  
///资金帐号  
TUstpFtdcAccountIDType AccountID;  
///期货密码  
TUstpFtdcPasswordType Password;  
///安装编号  
TUstpFtdcInstallIDType InstallID;  
///期货公司流水号  
TUstpFtdcFutureSerialType FutureSerial;  
///用户标识  
TUstpFtdcUserIDType UserID;  
///客户证件号码标志  
TUstpFtdcBoolType VerifyCertNoFlag;  
///币种代码  
TUstpFtdcCurrencyIDType CurrencyID;  
///转帐金额  
TUstpFtdcMoneyType TradeAmount;  
///期货可取金额  
TUstpFtdcMoneyType FutureFetchAmount;  
///费用支付标志  
TUstpFtdcFeePayFlagType FeePayFlag;  
///应收客户费用  
TUstpFtdcMoneyType CustFee;  
///应收期货公司费用  
TUstpFtdcMoneyType BrokerFee;
```



```
///发送方给接收方的消息
TUstpFtdcMessageType    Message;
///摘要
TUstpFtdcMessageType    Digest;
///银行帐号类型
TUstpFtdcBankAccType    BankAccType;
///渠道标志
TUstpFtdcDeviceIdType    DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType    BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType    BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType    BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType    OperNo;
///请求编号
TUstpFtdcRequestIDType    RequestID;
///交易 ID
TUstpFtdcTIDType    TID;
///转账交易状态
TUstpFtdcTransferStatusType    TransferStatus;
///错误代码
TUstpFtdcErrorIDType    ErrorID;
///错误信息
TUstpFtdcErrorMsgType    ErrorMsg;
};
```

6.2.98 OnErrRtnRepealFutureToBankByFutureManual 方法（未启用）

系统运行时期货端手工发起冲正期货转银行错误回报。

函数原形：

```
void OnErrRtnRepealFutureToBankByFutureManual(  
    CUstpFtdcRepealFieldRspField *pRepealFieldRsp,  
    CUstpFtdcRspInfoField *pRspInfo)
```

参数:

pRepealFieldRsp: 指向冲正信息结构的地址。

冲正信息结构:

```
struct CUstpFtdcRepealFieldRspField  
{  
    ///冲正时间间隔  
    TUstpFtdcRepealTimeIntervalType RepealTimeInterval;  
    ///已经冲正次数  
    TUstpFtdcRepealedTimesType RepealedTimes;  
    ///银行冲正标志  
    TUstpFtdcBankRepealFlagType BankRepealFlag;  
    ///已经冲正次数  
    TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;  
    ///被冲正平台流水号  
    TUstpFtdcPlateSerialType PlateRepealSerial;  
    ///被冲正银行流水号  
    TUstpFtdcBankSerialType BankRepealSerial;  
    ///被冲正期货公司流水号  
    TUstpFtdcFutureSerialType FutureRepealSerial;  
    ///业务功能码  
    TUstpFtdcTradeCodeType TradeCode;  
    ///银行代码  
    TUstpFtdcBrokerIDType BankID;  
    ///银行分中心代码  
    TUstpFtdcInvestorIDType BankBrchID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///期商分支机构代码  
    TUstpFtdcFutureBranchIDType BrokerBranchID;  
    ///交易日期  
    TUstpFtdcDateType TradeDate;  
    ///成交时间  
    TUstpFtdcTimeType TradeTime;
```

```
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType BankPassWord;
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
```

```
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType Message;
///摘要
TUstpFtdcMessageType Digest;
///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///渠道标志
TUstpFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType OperNo;
///请求编号
TUstpFtdcRequestIDType RequestID;
///交易 ID
TUstpFtdcTIDType TID;
```

```

///转账交易状态
TUstpFtdcTransferStatusType TransferStatus;
};
pRspInfo: 返回用户响应信息的地址。
响应信息结构:
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

6.2.99 OnRtnRepealFromBankToFutureByFuture 方法（未启用）

期货发起冲正银行转期货请求，银行处理完毕后报盘发回的通知。

函数原形:

```

void OnRtnRepealFromBankToFutureByFuture(
    CUstpFtdcRepealFieldRtnField *pRepealFieldRtn)

```

参数:

pRepealFieldRtn: 指向冲正通知结构的地址。

冲正通知结构:

```

struct CUstpFtdcRepealFieldRtnField
{
    ///冲正时间间隔
    TUstpFtdcRepealTimeIntervalType RepealTimeInterval;
    ///已经冲正次数
    TUstpFtdcRepealedTimesType RepealedTimes;
    ///银行冲正标志
    TUstpFtdcBankRepealFlagType BankRepealFlag;
    ///已经冲正次数
    TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;
    ///被冲正平台流水号
    TUstpFtdcPlateSerialType PlateRepealSerial;
    ///被冲正银行流水号
    TUstpFtdcBankSerialType BankRepealSerial;
    ///被冲正期货公司流水号
    TUstpFtdcFutureSerialType FutureRepealSerial;
}

```

```
///业务功能码
TUstpFtdcTradeCodeType TradeCode;
///银行代码
TUstpFtdcBrokerIDType BankID;
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType TradeDate;
///成交时间
TUstpFtdcTimeType TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
```

```
TUstpFtdcPasswordType    BankPassWord;
///资金帐号
TUstpFtdcAccountIDType    AccountID;
///期货密码
TUstpFtdcPasswordType    Password;
///安装编号
TUstpFtdcInstallIDType    InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType    FutureSerial;
///用户标识
TUstpFtdcUserIDType    UserID;
///客户证件号码标志
TUstpFtdcBoolType    VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType    CurrencyID;
///转帐金额
TUstpFtdcMoneyType    TradeAmount;
///期货可取金额
TUstpFtdcMoneyType    FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType    FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType    CustFee;
///应收期货公司费用
TUstpFtdcMoneyType    BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType    Message;
///摘要
TUstpFtdcMessageType    Digest;
///银行帐号类型
TUstpFtdcBankAccType    BankAccType;
///渠道标志
TUstpFtdcDeviceIdType    DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType    BankSecuAccType;
```

```

///期货公司银行编码
TUstpFtdcBankCodingForFutureType    BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType    BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType    BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType    SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType    OperNo;
///请求编号
TUstpFtdcRequestIDType    RequestID;
///交易 ID
TUstpFtdcTIDType    TID;
///转账交易状态
TUstpFtdcTransferStatusType    TransferStatus;
///错误代码
TUstpFtdcErrorIDType    ErrorID;
///错误信息
TUstpFtdcErrorMsgType    ErrorMsg;
};

```

6.2. 1000OnRtnRepealFromFutureToBankByFuture 方法（未启用）

期货发起冲正期货转银行请求，银行处理完毕后报盘发回的通知。

函数原形：

```

void OnRtnRepealFromFutureToBankByFuture(
    CUstpFtdcRepealFieldRtnField *pRepealFieldRtn)

```

参数：

pRepealFieldRtn: 指向冲正通知结构的地址。

冲正通知结构：

```

struct CUstpFtdcRepealFieldRtnField
{
    ///冲正时间间隔
    TUstpFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数

```



```
TUstpFtdcRepealedTimesType RepealedTimes;
///银行冲正标志
TUstpFtdcBankRepealFlagType BankRepealFlag;
///已经冲正次数
TUstpFtdcBrokerRepealFlagType BrokerRepealFlag;
///被冲正平台流水号
TUstpFtdcPlateSerialType PlateRepealSerial;
///被冲正银行流水号
TUstpFtdcBankSerialType BankRepealSerial;
///被冲正期货公司流水号
TUstpFtdcFutureSerialType FutureRepealSerial;
///业务功能码
TUstpFtdcTradeCodeType TradeCode;
///银行代码
TUstpFtdcBrokerIDType BankID;
///银行分中心代码
TUstpFtdcInvestorIDType BankBrchID;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcDateType TradeDate;
///成交时间
TUstpFtdcTimeType TradeTime;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///交易日
TUstpFtdcDateType TradingDay;
///平台流水号
TUstpFtdcPlateSerialType PlateSerial;
///最后分片标志
TUstpFtdcBoolType LastFragment;
///登录的 SessionID
TUstpFtdcSessionIDType SessionID;
```

```
///客户姓名
TUstpFtdcIndividualNameType CustomerName;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcCustTypeType CustType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行密码
TUstpFtdcPasswordType BankPassWord;
///资金帐号
TUstpFtdcAccountIDType AccountID;
///期货密码
TUstpFtdcPasswordType Password;
///安装编号
TUstpFtdcInstallIDType InstallID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///用户标识
TUstpFtdcUserIDType UserID;
///客户证件号码标志
TUstpFtdcBoolType VerifyCertNoFlag;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///转帐金额
TUstpFtdcMoneyType TradeAmount;
///期货可取金额
TUstpFtdcMoneyType FutureFetchAmount;
///费用支付标志
TUstpFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
```

```
TUstpFtdcMoneyType  BrokerFee;
///发送方给接收方的消息
TUstpFtdcMessageType  Message;
///摘要
TUstpFtdcMessageType  Digest;
///银行帐号类型
TUstpFtdcBankAccType  BankAccType;
///渠道标志
TUstpFtdcDeviceIdType  DeviceID;
///期货单位帐号类型
TUstpFtdcBankAccType  BankSecuAccType;
///期货公司银行编码
TUstpFtdcBankCodingForFutureType  BrokerIDByBank;
///期货单位帐号
TUstpFtdcBankAccountType  BankSecuAcc;
///银行密码标志
TUstpFtdcPwdFlagTypeType  BankPwdFlag;
///期货资金密码核对标志
TUstpFtdcPwdFlagTypeType  SecuPwdFlag;
///交易柜员
TUstpFtdcOperNoType  OperNo;
///请求编号
TUstpFtdcRequestIDType  RequestID;
///交易 ID
TUstpFtdcTIDType  TID;
///转账交易状态
TUstpFtdcTransferStatusType  TransferStatus;
///错误代码
TUstpFtdcErrorIDType  ErrorID;
///错误信息
TUstpFtdcErrorMsgType  ErrorMsg;
};
```

6.2.101 OnRspQueryContractBank 方法

请求查询签约银行响应。

函数原形:

```
void OnRspQueryContractBank(  
    CUstpFtdcQryContractBankRspField *pQryContractBankRsp,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

pQryContractBankRsp 指向查询签约银行响应结构的地址。

查询签约银行响应结构:

```
struct CUstpFtdcQryContractBankRspField  
{  
    ///经纪公司编号  
    TUstpFtdcTransBrokerIDType BrokerID;  
    ///银行代码  
    TUstpFtdcTransBankIDType BankID;  
    ///银行名称  
    TUstpFtdcTransNameType BankName;  
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField  
{  
    ///错误代码  
    TUstpFtdcErrorIDType ErrorID;  
    ///错误信息  
    TUstpFtdcErrorMsgType ErrorMsg;  
};
```

nRequestID: 返回查询转账银行请求的 ID, 该 ID 由用户在转账银行查询时指定。

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

6.2.102 OnRspQryInvestorPositionDetail 方法

请求查询投资者持仓明细的响应, 包含单腿的和组合持仓。

函数原形:

```
void OnRspQryInvestorPositionDetail(  
    CUstpFtdcRspQryInvestorPositionDetailField *pRspQryInvestorPositionDetail,
```

```
CUstpFtdcRspInfoField *pRspInfo,  
int nRequestID,  
bool bIsLast);
```

参数:

pRspQryInvestorPositionDetail 指向查询签约银行响应结构的地址。

///请求查询持仓明细应答

```
struct CUstpFtdcRspQryInvestorPositionDetailField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType ExchangeID;
```

///客户交易编码

```
TUstpFtdcClientIDType ClientID;
```

///单腿合约代码

```
TUstpFtdcInstrumentIDType InstrumentID;
```

///持仓类型

```
TUstpFtdcPosLegCombTypeType PositionType;
```

///组合合约代码

```
TUstpFtdcCombInstrumentIDType CombInstrumentID;
```

///组合类型

```
TUstpFtdcCombTypeType CombType;
```

///优先级

```
TUstpFtdcPriorityType Priority;
```

///买卖方向

```
TUstpFtdcDirectionType Direction;
```

///投机套保标志

```
TUstpFtdcHedgeFlagType HedgeFlag;
```

///成交编号

```
TUstpFtdcTradeIDType TradeID;
```

///开仓日期

```
TUstpFtdcDateType OpenDate;
```

///开仓时间

```
TUstpFtdcTimeType OpenTime;
```

///开仓价

```
TUstpFtdcPriceType DealPrice;
```

```

///昨结算价
TUstpFtdcPriceType  PreSettlementPrice;
///昨仓类型
TUstpFtdcYestPosType  YestodayPosType;
///持仓量
TUstpFtdcVolumeType  Position;
///优惠后客户单腿占用保证金
TUstpFtdcMoneyType  LegUsedMargin;
///优惠前客户单腿占用保证金
TUstpFtdcMoneyType  LegMarginToUse;
///持仓腿号
TUstpFtdcLegIDType  LegID;
///组合成交配对标识
TUstpFtdcTradeIDType  CombTradeID;
///平仓量
TUstpFtdcVolumeType  ClosePosition;
///平仓金额
TUstpFtdcMoneyType  CloseAmount;
///优惠后交易所单腿占用保证金
TUstpFtdcMoneyType  LegUsedMarginExchange;
///优惠前交易所单腿占用保证金
TUstpFtdcMoneyType  LegMarginToUseExchange;
///逐日盯市平仓盈亏
TUstpFtdcMoneyType  CloseProfitBySettl;
///逐笔对冲平仓盈亏
TUstpFtdcMoneyType  CloseProfitByDeal;
///逐日盯市持仓盈亏
TUstpFtdcMoneyType  PositionProfitBySettl;
///逐笔对冲持仓盈亏
TUstpFtdcMoneyType  PositionProfitByDeal;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType  ErrorID;

```

```

    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.103 OnRspQryInvestorPositionCombDetail 方法

请求查询投资者组合持仓=明细响应。

函数原形:

```

void OnRspQryInvestorPositionCombDetail(
    CUstpFtdcRspQryInvestorPositionCombDetailField *pRspQryInvestorPositionCombDetail,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pRspQryInvestorPositionCombDetail 指向查询签约银行响应结构的地址。

///请求查询持仓组合明细应答

```

struct CUstpFtdcRspQryInvestorPositionCombDetailField
{

```

///经纪公司编号

```
TUstpFtdcBrokerIDType    BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType  ExchangeID;
```

///客户交易编码

```
TUstpFtdcClientIDType    ClientID;
```

///单腿合约代码

```
TUstpFtdcInstrumentIDType InstrumentID;
```

///组合合约代码

```
TUstpFtdcCombInstrumentIDType CombInstrumentID;
```

///持仓腿号

```
TUstpFtdcLegIDType    LegID;
```

///买卖方向

```
TUstpFtdcDirectionType  Direction;
```

///投机套保标志

```
TUstpFtdcHedgeFlagType  HedgeFlag;
```

```
///成交编号
TUstpFtdcTradeIDType    TradeID;
///开仓日期
TUstpFtdcDateType    OpenDate;
///开仓时间
TUstpFtdcTimeType    OpenTime;
///开仓价
TUstpFtdcPriceType    DealPrice;
///昨结算价
TUstpFtdcPriceType    PreSettlementPrice;
///组合成交配对标识
TUstpFtdcTradeIDType    CombTradeID;
///组合类型
TUstpFtdcCombTypeType    CombType;
///优先级
TUstpFtdcPriorityType    Priority;
///组合持仓量
TUstpFtdcVolumeType    CombPosition;
///优惠后客户组合占用保证金
TUstpFtdcMoneyType    CombUsedMargin;
///优惠前客户组合占用保证金
TUstpFtdcMoneyType    CombMarginToUse;
///优惠后交易所组合占用保证金
TUstpFtdcMoneyType    CombUsedMarginExchange;
///优惠前交易所组合占用保证金
TUstpFtdcMoneyType    CombMarginToUseExchange;
///逐日盯市平仓盈亏
TUstpFtdcMoneyType    CloseProfitBySettl;
///逐笔对冲平仓盈亏
TUstpFtdcMoneyType    CloseProfitByDeal;
///逐日盯市持仓盈亏
TUstpFtdcMoneyType    PositionProfitBySettl;
///逐笔对冲持仓盈亏
TUstpFtdcMoneyType    PositionProfitByDeal;

};
pRspInfo: 指向响应信息结构的地址。
```


响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.104 OnRspQryMarginPrefParam 方法

请求查询大商所保证金优惠参数信息响应。

函数原形:

```
void OnRspQryMarginPrefParam(
    CUstpFtdcRspQryMarginPrefParamField *pRspQryMarginPrefParam,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pRspQryMarginPrefParam 指向查询保证金优惠参数响应结构的地址。

///查询保证金优惠参数应答

```
struct CUstpFtdcRspQryMarginPrefParamField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType HedgeFlag;
    ///组合合约代码
    TUstpFtdcCombInstrumentIDType CombInstrumentID;
    ///组合合约名称
    TUstpFtdcCombInstrumentIDType CombInstrumentName;
    ///组合类型
    TUstpFtdcCombTypeType CombType;
    ///腿 1 合约代码
```

```

TUstpFtdcInstrumentIDType  Leg1InstrumentID;
///腿 1 品种代码
TUstpFtdcProductIDType  Leg1ProductID;
///腿 1 方向
TUstpFtdcDirectionType  Leg1Direction;
///腿 1 投机套保标志
TUstpFtdcHedgeFlagType  Leg1HedgeFlag;
///腿 1 昨结算价
TUstpFtdcPriceType  Leg1SettlementPrice;
///腿 2 合约代码
TUstpFtdcInstrumentIDType  Leg2InstrumentID;
///腿 2 品种代码
TUstpFtdcProductIDType  Leg2ProductID;
///腿 2 方向
TUstpFtdcDirectionType  Leg2Direction;
///腿 2 投机套保标志
TUstpFtdcHedgeFlagType  Leg2HedgeFlag;
///腿 2 昨结算价
TUstpFtdcPriceType  Leg2SettlementPrice;
///优先级
TUstpFtdcPriorityType  Priority;
///是否交易期间优惠
TUstpFtdcBoolFlagType  TradeEna;
///单腿乘数
TUstpFtdcLegMultipleType  LegMultiple;
///期权组合保证金比例
TUstpFtdcRatioType  MarginAdjRatio;
///腿 1 产品类型
TUstpFtdcProductClassType  Leg1ProductClass;
///腿 2 产品类型
TUstpFtdcProductClassType  Leg2ProductClass;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{

```

```

    ///错误代码

```

```

    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.105 OnRspDSUserCertification 方法

请求穿透监管客户认证应答。

函数原形:

```

void OnRspDSUserCertification(
    CUstpFtdcDSUserCertRspDataField *pDSUserCertRspData,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pDSUserCertRspData 指向穿透监管客户认证响应结构的地址。

///穿透监管客户认证响应信息

```

struct CUstpFtdcDSUserCertRspDataField
{
    ///AppID
    TUstpFtdcDSAppIDType AppID;
    ///AppID 类型
    TUstpFtdcDSAppIDTypeType AppIDType;
    ///用户认证返回信息
    TUstpFtdcDSStreamDataType UserCertRspData;
    ///用户证书返回信息分片的总片数
    TUstpFtdcDSSeqeneceType TotalNum;
    ///用户证书返回信息分片的第几片
    TUstpFtdcDSSeqeneceType CurrentNum;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

```
};
```

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

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

6.2.106 OnRspDSProxySubmitInfo 方法

请求穿透监管信息采集中继上传信息响应。

函数原形:

```
void OnRspDSProxySubmitInfo(  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast);
```

参数:

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUSTPFTDCRSPINFOFIELD  
{  
    ///错误代码  
    TUSTPFTDCERRORIDTYPE ErrorID;  
    ///错误信息  
    TUSTPFTDCERRORMSGTYPE ErrorMessage;  
};
```

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

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

6.2.107 OnRspQryExchangeRate 方法（未启用）

交叉汇率查询应答。当客户端发出交叉汇率查询指令后，飞马平台返回响应时，该方法会被调用。

函数原形:

```
void OnRspQryExchangeRate (  
    CUSTPFTDCRSPEXCHANGERATEFIELD *pRspExchangeRate,  
    CUSTPFTDCRSPINFOFIELD *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数:

pExchange 指向交叉汇率结构的地址。

交叉汇率结构

```
struct CUSTPFTDCRSPEXCHANGERATEFIELD
```

```
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///品种代码
    TUstpFtdcProductIDType ProductID;
    ///折算汇率
    TUstpFtdcExchangeRateType ExchangeRate;
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};
```

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

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

6.2.108 OnRspQryEWarrantOffset 方法 (未启用)

请求查询仓单折抵信息响应。

函数原形:

```
void OnRspQryEWarrantOffset(
    CUstpFtdcEWarrantOffsetFieldRspField *pEWarrantOffsetFieldRsp,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pEWarrantOffsetFieldRsp 指向查询仓单折抵信息响应结构的地址。

查询仓单折抵信息响应结构:

```
struct CUstpFtdcEWarrantOffsetFieldRspField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///交易所代码
```

```

TUstpFtdcExchangeIDType ExchangeID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///买卖方向
TUstpFtdcDirectionType Direction;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///数量
TUstpFtdcVolumeType Volume;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

nRequestID: 返回用户仓单查询请求的 ID, 该 ID 由用户在仓单查询时指定。

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

6.2. 1090OnRspQryComplianceParam 方法（未启用）

合规参数查询应答。当客户端发出合规参数查询指令后，飞马平台返回响应时，该方法会被调用。

函数原形:

```

void OnRspQryComplianceParam (
    CUstpFtdcRspComplianceParamField *pRspComplianceParam,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pRspComplianceParam 指向合规参数结构的地址。

合规参数结构:

```

struct CUstpFtdcRspComplianceParamField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号

```

```

    TUstpFtdcInvestorIDType InvestorID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///客户号
    TUstpFtdcClientIDType ClientID;
    ///每日最大报单笔
    TUstpFtdcVolumeType DailyMaxOrder;
    ///每日最大撤单笔
    TUstpFtdcVolumeType DailyMaxOrderAction;
    ///每日最大错单笔
    TUstpFtdcVolumeType DailyMaxErrorOrder;
    ///每日最大报单手
    TUstpFtdcVolumeType DailyMaxOrderVolume;
    ///每日最大撤单手
    TUstpFtdcVolumeType DailyMaxOrderActionVolume;
};

```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```

struct CUstpFtdcRspInfoField
{
    ///错误代码
    TUstpFtdcErrorIDType ErrorID;
    ///错误信息
    TUstpFtdcErrorMsgType ErrorMsg;
};

```

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

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

6.2.1100OnRspQrySystemTime 方法（主柜台不支持）

系统时间查询应答。

函数原形:

```

void OnRspQrySystemTime(
    CUstpFtdcRspQrySystemTimeField *pRspQrySystemTime,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pRspQrySystemTime 指向系统时间结构的地址。

系统时间结构

///系统时间

```
struct CUSTpFtdcRspQrySystemTimeField
```

```
{
```

///交易所代码

```
TUstpFtdcExchangeIDType ExchangeID;
```

///系统时间

```
TUstpFtdcTimeType SystemTime;
```

```
};
```

pRspInfo: 指向响应信息结构的地址。

响应信息结构:

```
struct CUSTpFtdcRspInfoField
```

```
{
```

///错误代码

```
TUstpFtdcErrorIDType ErrorID;
```

///错误信息

```
TUstpFtdcErrorMsgType ErrorMsg;
```

```
};
```

nRequestID: 返回系统时间查询请求的 ID, 该 ID 由用户在系统时间查询时指定。

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

6.2.111 OnErrRtnForQuote 方法

报价错误回报通知。

函数原形:

```
virtual void OnErrRtnForQuote(CUSTpFtdcReqForQuoteField *pReqForQuote,
CUSTpFtdcRspInfoField *pRspInfo) {};
```

参数:

CUSTpFtdcReqForQuoteField: 指向报价回报结构的地址。

报价结果回报结构请参考询价请求;

6.2.112 OnErrRtnMarginCombCancel 方法

客户申请组合拆分撤单的错误回报。

函数原形:

```
void OnErrRtnMarginCombCancel(CUSTpFtdcAPIMarginCombCancelField
*pAPIMarginCombCancel, CUSTpFtdcRspInfoField *pRspInfo);
```

参数:

CUSTpFtdcAPIMarginCombCancelField: 指向组合拆分回报结构的地址。

组合拆分回报结构:

///组合申请撤销应答

```
struct CUstpFtdcAPIMarginCombCancelField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType    BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType  ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType  OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
    ///用户代码
    TUstpFtdcUserIDType      UserID;
    ///用户本地编号
    TUstpFtdcUserOrderLocalIDType  CancelUserActionLocalID;
    ///组合合约代码
    TUstpFtdcCombInstrumentIDType  CombInstrumentID;
    ///客户代码
    TUstpFtdcClientIDType      ClientID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType     HedgeFlag;
    ///组合方向
    TUstpFtdcDirectionType     Direction;
    ///组合动作方向
    TUstpFtdcCombDirectionType  CombDirection;
    ///组合数量
    TUstpFtdcVolumeType        CombVolume;
    ///组合操作状态
    TUstpFtdcCombActionStatusType  CombActionStatus;
};;
```

6.2.113OnErrRtnMarginCombAction 方法

客户申请组合拆分错误回报。

函数原形:

```
void OnErrRtnMarginCombAction(  
    CUstpFtdcInputMarginCombActionField          *pInputMarginCombAction,  
    CUstpFtdcRspInfoField *pRspInfo);
```

参数:

CUstpFtdcInputMarginCombActionField: 指向组合回报结构的地址。

组合结果回报结构:

///申请组合

```
struct CUstpFtdcInputMarginCombActionField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType   BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///交易用户代码  
    TUstpFtdcUserIDType   UserID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType HedgeFlag;  
    ///用户本地编号  
    TUstpFtdcUserOrderLocalIDType   UserActionLocalID;  
    ///组合合约代码  
    TUstpFtdcCombInstrumentIDType   CombInstrumentID;  
    ///组合数量  
    TUstpFtdcVolumeType   CombVolume;  
    ///组合动作方向  
    TUstpFtdcCombDirectionType   CombDirection;  
    ///本地编号  
    TUstpFtdcOrderLocalIDType   ActionLocalID;  
    ///系统编号  
    TUstpFtdcOrderSysIDType   OrderSysID;  
    ///组合操作状态  
    TUstpFtdcCombActionStatusType   CombActionStatus;  
    ///交易编码  
    TUstpFtdcClientIDType   ClientID;  
};;
```

6.2. 1140nRtnExecSelfClose 方法

期权对冲设置私有流推送接口。

函数原形:

```
virtual void OnRtnExecSelfClose(CUstpFtdcExecSelfCloseField *pExecSelfClose) {};
```

参数:

CUstpFtdcExecSelfCloseField: 指向对冲设置结果回报结构的地址。

对冲设置结果回报结构:

///期权对冲设置录入应答

```
struct CUstpFtdcExecSelfCloseField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType ExchangeID;
```

///用户代码

```
TUstpFtdcUserIDType UserID;
```

///投资者编号

```
TUstpFtdcInvestorIDType InvestorID;
```

///合约代码

```
TUstpFtdcInstrumentIDType InstrumentID;
```

///本地期权自对冲编号

```
TUstpFtdcUserSelfCloseLocalIDType UserSelfCloseLocalID;
```

///履约头寸自对冲标志

```
TUstpFtdcSelfCloseFlagType SelfCloseFlag;
```

///投机套保标志

```
TUstpFtdcHedgeFlagType HedgeFlag;
```

///数量

```
TUstpFtdcVolumeType Volume;
```

///业务单元

```
TUstpFtdcBusinessUnitType BusinessUnit;
```

///币种

```
TUstpFtdcCurrencyIDType Currency;
```

///会话号

```
TUstpFtdcSessionIDType SessionID;
```

```
///前置编号
TUstpFtdcFrontIDType    FrontID;
///自对冲提交状态
TUstpFtdcSubmitStatusType    SubmitStatus;
///交易日
TUstpFtdcTradingDayType    TradingDay;
///自对冲编号
TUstpFtdcSelfCloseSysIDType    SelfCloseSysID;
///申报日期
TUstpFtdcDateType    InsertDate;
///插入时间
TUstpFtdcTimeType    InsertTime;
///交易所应答时间
TUstpFtdcTimeType    InsertTimeExchange;
///撤销时间
TUstpFtdcTimeType    CancelTime;
///自对冲结果
TUstpFtdcExecResultType    ExecResult;
///是否有效
TUstpFtdcIsActiveType    ActiveFlag;
///交易编码
TUstpFtdcClientIDType    ClientID;
};;
```

6.2.115OnRtnProductStatus 方法

产品状态推送接口。

函数原形:

```
virtual void OnRtnProductStatus(CUstpFtdcProductStatusField *pProductStatus) {};
```

参数:

CUstpFtdcProductStatusField: 指向产品状态回报结构的地址。
产品状态推送接口回报结构:

```
struct CUstpFtdcProductStatusField
{
```

```
///经纪公司编号
TUstpFtdcBrokerIDType   BrokerID;
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///品种代码
TUstpFtdcProductIDType   ProductID;
///品种名称
TUstpFtdcProduct_nameType   ProductName;
///产品状态
TUstpFtdcInstrumentStatusType   ProductStatus;
///币种
TUstpFtdcCurrencyIDType   Currency;
///产品类型
TUstpFtdcProductClassType   ProductClass;
///产品乘数
TUstpFtdcVolumeMultipleType   VolumeMultiple;
///特殊品种标志
TUstpFtdcProductTypeType   ProductType;
};;
```

6.2. 1160nRspQrySpotClosePrice 方法

查询现货标的昨日收盘价应答

函数原形:

```
void OnRspQrySpotClosePrice(
    CUstpFtdcSportClosePriceField           *pSportClosePrice,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

///查询现货标的昨日收盘价应答

```
struct CUstpFtdcSportClosePriceField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///交易所代码
```

```
TUstpFtdcExchangeIDType  ExchangeID;  
///品种代码  
TUstpFtdcProductIDType   ProductID;  
///交易日  
TUstpFtdcTradingDayType  TradingDay;  
///现货标的  
TUstpFtdcInstrumentIDType UnderlyingInstrID;  
///昨收盘  
TUstpFtdcPriceType       PreClosePrice;  
///产品类型  
TUstpFtdcProductClassType ProductClass;  
};;
```

6.2.1170nRspQryOptionTradeCost 方法

期权交易成本查询应答

函数原形:

```
int ReqQueryContractBank(  
    CUstpFtdcQryContractBankReqField *pQryContractBankReq,  
    int nRequestID)
```

参数:

```
///期权成交成本查询应答  
struct CUstpFtdcOptionTradeCostField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType  BrokerID;  
    ///用户代码  
    TUstpFtdcUserIDType    UserID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType  HedgeFlag;
```

```
///期权合约报价
TUstpFtdcPriceType    InputPrice;
///标的价格
TUstpFtdcPriceType    UnderlyingPrice;
///期权合约保证金不变部分
TUstpFtdcMoneyType    FixedMargin;
///期权合约最小保证金
TUstpFtdcMoneyType    MiniMargin;
///期权合约权利金
TUstpFtdcMoneyType    Premium;
///交易所期权合约保证金不变部分
TUstpFtdcMoneyType    FixedMarginExchange;
///交易所期权合约最小保证金
TUstpFtdcMoneyType    MiniMarginExchange;
};
```

6.2.1180nRspQryProduct 方法

产品信息查询应答。

函数原形:

```
void OnRspQryProduct(
    CUstpFtdcProductField *pProduct,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID, bool bIsLast);
```

参数:

```
///产品信息查询应答
struct CUstpFtdcProductField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType    BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType  ExchangeID;
    ///品种代码
    TUstpFtdcProductIDType    ProductID;
    ///产品类型
    TUstpFtdcProductClassType  ProductClass;
```

```
///特殊品种标志
TUstpFtdcProductTypeType ProductType;
///品种名称
TUstpFtdcProductNameType ProductName;
///产品状态
TUstpFtdcInstrumentStatusType ProductStatus;
///币种
TUstpFtdcCurrencyIDType Currency;
///产品乘数
TUstpFtdcVolumeMultipleType VolumeMultiple;
};
```

6.2. 1190nRspQryCombAction 方法

组合拆分查询应答。

函数原形：

```
void OnRspQryCombAction(
    CUstpFtdcQryCombActionField *pQryCombAction,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

```
///组合拆分申请查询应答
struct CUstpFtdcQryCombActionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType HedgeFlag;
    ///组合合约代码
```



```

TUstpFtdcCombInstrumentIDType  CombInstrumentID;
///组合动作方向
TUstpFtdcDirectionType  CombDirection;
///数量
TUstpFtdcVolumeType  CombVolume;
///组合请求的本地编号
TUstpFtdcOrderLocalIDType  ActionLocalID;
///系统报单编号
TUstpFtdcOrderSysIDType  OrderSysID;
///组合操作状态
TUstpFtdcCombActionStatusType  CombActionStatus;
};

```

6.2.120 OnRspMarginCombCancelAccept 方法

飞马收到组合申请撤销请求应答

```

virtual void
OnRspMarginCombCancelAccept(CUstpFtdcAPIInputMarginCombCancelField
*pAPIInputMarginCombCancel, CUstpFtdcRspInfoField *pRspInfo, int nRequestID,
bool bIsLast) {};

```

6.2.121 OnRspMarginCombCancel 方法和 OnRspMarginCombCancelAccept 方法

组合申请撤销应答。

当客户端发出指令后，飞马平台进行检查正确后会调用 OnRspMarginCombCancelAccept 方法，通知用户飞马已经收到该笔申报，随即发送到交易所并得到交易所应答后，OnRspMarginCombCancel 方法会被调用。

函数原形：

```

void OnRspMarginCombCancel(
    CUstpFtdcstpMarginCombCancelField *pstpMarginCombCancel,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

详见申请信息章节；

6.2. 1220nRspExecSelfClose 方法

期权对冲设置查询应答。

函数原形：

```
void OnRspExecSelfClose(  
    CUstpFtdcExecSelfCloseField *pExecSelfClose,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数：

```
    ///期权对冲设置录入应答  
struct CUstpFtdcExecSelfCloseField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType   BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///用户代码  
    TUstpFtdcUserIDType    UserID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///本地期权自对冲编号  
    TUstpFtdcUserSelfCloseLocalIDType UserSelfCloseLocalID;  
    ///履约头寸自对冲标志  
    TUstpFtdcSelfCloseFlagType SelfCloseFlag;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType   HedgeFlag;  
    ///数量  
    TUstpFtdcVolumeType     Volume;  
    ///业务单元  
    TUstpFtdcBusinessUnitType BusinessUnit;  
    ///币种  
    TUstpFtdcCurrencyIDType Currency;
```

```
///会话号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
///自对冲提交状态
TUstpFtdcSubmitStatusType SubmitStatus;
///交易日
TUstpFtdcTradingDayType TradingDay;
///自对冲编号
TUstpFtdcSelfCloseSysIDType SelfCloseSysID;
///申报日期
TUstpFtdcDateType InsertDate;
///插入时间
TUstpFtdcTimeType InsertTime;
///交易所应答时间
TUstpFtdcTimeType InsertTimeExchange;
///撤销时间
TUstpFtdcTimeType CancelTime;
///自对冲结果
TUstpFtdcExecResultType ExecResult;
///是否有效
TUstpFtdcIsActiveType ActiveFlag;
///交易编码
TUstpFtdcClientIDType ClientID;
};;
```

6.2. 1230nRspExecSelfCloseInsert 方法

期权对冲设置录入应答。

函数原形:

```
void OnRspExecSelfCloseInsert(
    CUstpFtdcExecSelfCloseField *pExecSelfClose,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

```
    ///期权对冲设置录入应答
struct CUstpFtdcExecSelfCloseField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType    BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType  ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType    UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
    ///合约代码
    TUstpFtdcInstrumentIDType  InstrumentID;
    ///本地期权自对冲编号
    TUstpFtdcUserSelfCloseLocalIDType  UserSelfCloseLocalID;
    ///履约头寸自对冲标志
    TUstpFtdcSelfCloseFlagType  SelfCloseFlag;
    ///投机套保标志
    TUstpFtdcHedgeFlagType    HedgeFlag;
    ///数量
    TUstpFtdcVolumeType    Volume;
    ///业务单元
    TUstpFtdcBusinessUnitType  BusinessUnit;
    ///币种
    TUstpFtdcCurrencyIDType  Currency;
    ///会话号
    TUstpFtdcSessionIDType  SessionID;
    ///前置编号
    TUstpFtdcFrontIDType  FrontID;
    ///自对冲提交状态
    TUstpFtdcSubmitStatusType  SubmitStatus;
    ///交易日
    TUstpFtdcTradingDayType  TradingDay;
    ///自对冲编号
    TUstpFtdcSelfCloseSysIDType  SelfCloseSysID;
```

```
///申报日期
TUstpFtdcDateType InsertDate;
///插入时间
TUstpFtdcTimeType InsertTime;
///交易所应答时间
TUstpFtdcTimeType InsertTimeExchange;
///撤销时间
TUstpFtdcTimeType CancelTime;
///自对冲结果
TUstpFtdcExecResultType ExecResult;
///是否有效
TUstpFtdcIsActiveType ActiveFlag;
///交易编码
TUstpFtdcClientIDType ClientID;
};;
```

6.2. 1240nRspExecSelfCloseAction 方法

期权对冲设置撤单录入应答。

函数原形：

```
void OnRspExecSelfCloseAction(
    CUstpFtdcExecSelfCloseActionField          *pExecSelfCloseAction,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

///期权对冲设置录入撤销应答

```
struct CUstpFtdcExecSelfCloseActionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
```

```
TUstpFtdcInvestorIDType InvestorID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///本地期权自对冲撤单编号
TUstpFtdcUserSelfCloseActionLocalIDType UserSelfCloseActionLocalID;
///自对冲编号
TUstpFtdcSelfCloseSysIDType SelfCloseSysID;
///会话号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
///本地期权自对冲编号
TUstpFtdcUserSelfCloseLocalIDType UserSelfCloseLocalID;
///履约头寸自对冲标志
TUstpFtdcSelfCloseFlagType SelfCloseFlag;
///数量
TUstpFtdcVolumeType Volume;
///自对冲提交状态
TUstpFtdcSubmitStatusType SubmitStatus;
///交易日
TUstpFtdcTradingDayType TradingDay;
///申报日期
TUstpFtdcDateType InsertDate;
///插入时间
TUstpFtdcTimeType InsertTime;
///交易所应答时间
TUstpFtdcTimeType InsertTimeExchange;
///撤销时间
TUstpFtdcTimeType CancelTime;
///自对冲结果
TUstpFtdcExecResultType ExecResult;
///是否有效
TUstpFtdcIsActiveType ActiveFlag;
```

```
///交易编码
TUstpFtdcClientIDType ClientID;
};
```

6.2. 1250nErrRtnExecSelfCloseAction 方法

期权对冲设置录入操作错误应答。

函数原形:

```
void OnErrRtnExecSelfCloseAction(
CUstpFtdcExecSelfCloseActionField *pExecSelfCloseAction,
CUstpFtdcRspInfoField *pRspInfo)
```

参数:

```
    ///期权对冲设置录入撤销应答
struct CUstpFtdcExecSelfCloseActionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType    UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType  HedgeFlag;
    ///本地期权自对冲撤单编号
    TUstpFtdcUserSelfCloseActionLocalIDType UserSelfCloseActionLocalID;
    ///自对冲编号
    TUstpFtdcSelfCloseSysIDType SelfCloseSysID;
    ///会话号
    TUstpFtdcSessionIDType   SessionID;
    ///前置编号
    TUstpFtdcFrontIDType     FrontID;
    ///本地期权自对冲编号
```

```
TUstpFtdcUserSelfCloseLocalIDType  UserSelfCloseLocalID;
///履约头寸自对冲标志
TUstpFtdcSelfCloseFlagType  SelfCloseFlag;
///数量
TUstpFtdcVolumeType  Volume;
///自对冲提交状态
TUstpFtdcSubmitStatusType  SubmitStatus;
///交易日
TUstpFtdcTradingDayType  TradingDay;
///申报日期
TUstpFtdcDateType  InsertDate;
///插入时间
TUstpFtdcTimeType  InsertTime;
///交易所应答时间
TUstpFtdcTimeType  InsertTimeExchange;
///撤销时间
TUstpFtdcTimeType  CancelTime;
///自对冲结果
TUstpFtdcExecResultType  ExecResult;
///是否有效
TUstpFtdcIsActiveType  ActiveFlag;
///交易编码
TUstpFtdcClientIDType  ClientID;
};;
```

6.2.1260nRspExecSelfCloseInsertAccept 方法

飞马收到期权对冲设置录入应答。

函数原形：

```
void OnRspExecSelfCloseInsertAccept(
    CUstpFtdcExecSelfCloseField *pExecSelfClose,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

详见申请信息章节；

6.2. 1270nRspExecSelfCloseActionAccept 方法

飞马收到期权对冲撤销录入应答。

函数原形：

```
void OnRspExecSelfCloseActionAccept(  
    CUstpFtdcExecSelfCloseActionField *pExecSelfCloseAction,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数：

详见申请信息章节；

6.2. 1280nRspMarginCombActionAccept 方法

飞马收到客户申请组合应答。

函数原形：

```
void OnRspMarginCombActionAccept(  
    CUstpFtdcInputMarginCombActionField *pInputMarginCombAction,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数：

详见申请信息章节；

6.2. 1290nRspChangeCenter 方法

切换交易中心应答。

函数原形：

```
void OnRspChangeCenter(  
    CUstpFtdcRspChangeCenterField *pRspChangeCenter,  
    CUstpFtdcRspInfoField *pRspInfo,  
    int nRequestID,  
    bool bIsLast)
```

参数：

///切换交易中心应答

```
struct CUstpFtdcRspChangeCenterField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType          BrokerID;
    ///用户代码
    TUstpFtdcUserIDType            UserID;
    ///新所属交易中心
    TUstpFtdcDataCenterIDType      NewCenterID;
    ///原所属交易中心
    TUstpFtdcDataCenterIDType      OldCenterID;
    ///投资者编号
    TUstpFtdcInvestorIDType        InvestorID;
};
```

6.2. 1300nRspQryTradeCenter 方法

交易中心查询应答。

函数原形:

```
void OnRspQryTradeCenter(
    CUstpFtdcRspTradeCenterField *pRspTradeCenter,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

///查询交易中心应答

```
struct CUstpFtdcRspTradeCenterField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType          BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType        InvestorID;
    ///用户代码
    TUstpFtdcUserIDType            UserID;
    ///中心号
```

```

TUstpFtdcDataCenterIDType          CenterID;
///是否所属交易中心
TUstpFtdcBoolType                   IsBelonged;
///是否当前登录的交易中心
TUstpFtdcBoolType                   IsLoggedin;
///中心信息描述
TUstpFtdcInfoType                   CenterInfo;
};

```

6.2. 1310nRspParkedOrderInsert 方法

预埋单报单录入应答。

函数原形:

```

void OnRspParkedOrderInsert(
CUstpFtdcParkedOrderInsertField *pParkedOrderInsert,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)

```

参数:

///预埋单报单

```

struct CUstpFtdcParkedOrderInsertField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType          BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType        ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType        InvestorID;
    ///用户代码
    TUstpFtdcUserIDType            UserID;
    ///指定通过此席位序号下单
    TUstpFtdcSeatNoType            SeatNo;
    ///合约代码/套利组合合约号
    TUstpFtdcInstrumentIDType      InstrumentID;
}

```

///预埋单报单编号
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
///报单类型
TUstpFtdcOrderPriceTypeType OrderPriceType;
///买卖方向
TUstpFtdcDirectionType Direction;
///开平标志
TUstpFtdcOffsetFlagType OffsetFlag;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///价格
TUstpFtdcPriceType LimitPrice;
///数量
TUstpFtdcVolumeType Volume;
///有效期类型
TUstpFtdcTimeConditionType TimeCondition;
///GTD 日期
TUstpFtdcDateType GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType VolumeCondition;
///最小成交量
TUstpFtdcVolumeType MinVolume;
///止损价止赢价
TUstpFtdcPriceType StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志
TUstpFtdcBoolType IsAutoSuspend;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期

```
TUstpFtdcDateType          ActionDay;
///策略类别
TUstpFtdcArbiTypeType      ArbiType;
///客户号
TUstpFtdcClientIDType      ClientID;
///会话编号
TUstpFtdcSessionIDType     SessionID;
///前置编号
TUstpFtdcFrontIDType       FrontID;
///预埋单状态
TUstpFtdcParkedOrderStatusType  ParkedOrderStatus;
};
```

6.2. 1320nRspParkedOrderAction 方法

预埋单撤单应答。

函数原形:

```
void OnRspParkedOrderAction(
CUstpFtdcParkedOrderActionField *pParkedOrderAction,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)
```

参数:

```
///预埋单撤单
struct CUstpFtdcParkedOrderActionField
{
    ///交易所代码
    TUstpFtdcExchangeIDType      ExchangeID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType        BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType      InvestorID;
    ///用户代码
    TUstpFtdcUserIDType          UserID;
    ///本次撤单操作的本地编号
    TUstpFtdcUserOrderLocalIDType  UserOrderActionLocalID;
```

```
///被撤预埋单报单编号
TUstpFtdcUserOrderLocalIDType      UserOrderLocalID;
///本地业务标识
TUstpFtdcBusinessLocalIDType        BusinessLocalID;
///客户代码
TUstpFtdcClientIDType               ClientID;
///会话编号
TUstpFtdcSessionIDType              SessionID;
///前置编号
TUstpFtdcFrontIDType                FrontID;
///合约代码
TUstpFtdcInstrumentIDType           InstrumentID;
};
```

6. 2. 1330nRspQryParkedOrder 方法

预埋单查询应答。

函数原形:

```
void OnRspQryParkedOrder(
CUstpFtdcParkedOrderField *pParkedOrder,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)
```

参数:

///预埋单查询响应

```
struct CUstpFtdcParkedOrderField
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType      BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType    ExchangeID;
```

///投资者编号

```
TUstpFtdcInvestorIDType    InvestorID;
```

///用户代码

```
TUstpFtdcUserIDType        UserID;
```

///指定通过此席位序号下单

TUstpFtdcSeatNoType ///合约代码/套利组合合约号	SeatNo;
TUstpFtdcInstrumentIDType ///预埋单报单编号	InstrumentID;
TUstpFtdcUserOrderLocalIDType ///报单类型	UserOrderLocalID;
TUstpFtdcOrderPriceTypeType ///买卖方向	OrderPriceType;
TUstpFtdcDirectionType ///开平标志	Direction;
TUstpFtdcOffsetFlagType ///投机套保标志	OffsetFlag;
TUstpFtdcHedgeFlagType ///价格	HedgeFlag;
TUstpFtdcPriceType ///数量	LimitPrice;
TUstpFtdcVolumeType ///有效期类型	Volume;
TUstpFtdcTimeConditionType ///GTD 日期	TimeCondition;
TUstpFtdcDateType ///成交量类型	GTDDate;
TUstpFtdcVolumeConditionType ///最小成交量	VolumeCondition;
TUstpFtdcVolumeType ///止损价止赢价	MinVolume;
TUstpFtdcPriceType ///强平原因	StopPrice;
TUstpFtdcForceCloseReasonType ///自动挂起标志	ForceCloseReason;
TUstpFtdcBoolType ///业务单元	IsAutoSuspend;
TUstpFtdcBusinessUnitType ///用户自定义域	BusinessUnit;
TUstpFtdcCustomType	UserCustom;

```

///本地业务标识
TUstpFtdcBusinessLocalIDType      BusinessLocalID;
///业务发生日期
TUstpFtdcDateType                  ActionDay;
///策略类别
TUstpFtdcArbiTypeType              ArbiType;
///客户号
TUstpFtdcClientIDType              ClientID;
///会话编号
TUstpFtdcSessionIDType             SessionID;
///前置编号
TUstpFtdcFrontIDType               FrontID;
///预埋单状态
TUstpFtdcParkedOrderStatusType     ParkedOrderStatus;
///本次撤单操作的本地编号
TUstpFtdcUserOrderLocalIDType      UserOrderActionLocalID;
///插入时间
TUstpFtdcTimeType                   InsertTime;
///撤销时间
TUstpFtdcTimeType                   CancelTime;
};

```

6.2. 1340nRtnParkedOrder 方法

预埋单状态推送。

函数原形:

```

void OnRtnParkedOrder(
CUstpFtdcParkedOrderField *pParkedOrder)

```

参数:

///预埋单查询响应

```

struct CUstpFtdcParkedOrderField

```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType      BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType    ExchangeID;
```


///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///用户代码
TUstpFtdcUserIDType UserID;
///指定通过此席位序号下单
TUstpFtdcSeatNoType SeatNo;
///合约代码/套利组合合约号
TUstpFtdcInstrumentIDType InstrumentID;
///预埋单报单编号
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
///报单类型
TUstpFtdcOrderPriceTypeType OrderPriceType;
///买卖方向
TUstpFtdcDirectionType Direction;
///开平标志
TUstpFtdcOffsetFlagType OffsetFlag;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///价格
TUstpFtdcPriceType LimitPrice;
///数量
TUstpFtdcVolumeType Volume;
///有效期类型
TUstpFtdcTimeConditionType TimeCondition;
///GTD 日期
TUstpFtdcDateType GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType VolumeCondition;
///最小成交量
TUstpFtdcVolumeType MinVolume;
///止损价止赢价
TUstpFtdcPriceType StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志

```

TUstpFtdcBoolType          IsAutoSuspend;
///业务单元
TUstpFtdcBusinessUnitType   BusinessUnit;
///用户自定义域
TUstpFtdcCustomType         UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType           ActionDay;
///策略类别
TUstpFtdcArbiTypeType       ArbiType;
///客户号
TUstpFtdcClientIDType        ClientID;
///会话编号
TUstpFtdcSessionIDType       SessionID;
///前置编号
TUstpFtdcFrontIDType         FrontID;
///预埋单状态
TUstpFtdcParkedOrderStatusType ParkedOrderStatus;
///本次撤单操作的本地编号
TUstpFtdcUserOrderLocalIDType UserOrderActionLocalID;
///插入时间
TUstpFtdcTimeType            InsertTime;
///撤销时间
TUstpFtdcTimeType            CancelTime;
};

```

6.2.1350nRspConditionalOrderInsert 方法

条件单报单录入应答。

函数原形:

```

void OnRspConditionalOrderInsert(
CUstpFtdcConditionalOrderInsertField *pConditionalOrderInsert,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)

```

参数:

///条件单报单

struct CUstpFtdcConditionalOrderInsertField

{

///经纪公司编号

TUstpFtdcBrokerIDType BrokerID;

///交易所代码

TUstpFtdcExchangeIDType ExchangeID;

///投资者编号

TUstpFtdcInvestorIDType InvestorID;

///用户代码

TUstpFtdcUserIDType UserID;

///指定通过此席位序号下单

TUstpFtdcSeatNoType SeatNo;

///合约代码/套利组合合约号

TUstpFtdcInstrumentIDType InstrumentID;

///条件单报单编号

TUstpFtdcUserOrderLocalIDType UserOrderLocalID;

///报单类型

TUstpFtdcOrderPriceTypeType OrderPriceType;

///买卖方向

TUstpFtdcDirectionType Direction;

///开平标志

TUstpFtdcOffsetFlagType OffsetFlag;

///投机套保标志

TUstpFtdcHedgeFlagType HedgeFlag;

///价格

TUstpFtdcPriceType LimitPrice;

///数量

TUstpFtdcVolumeType Volume;

///有效期类型

TUstpFtdcTimeConditionType TimeCondition;

///GTD 日期

TUstpFtdcDateType GTDDate;

///成交量类型

```

TUstpFtdcVolumeConditionType      VolumeCondition;
///最小成交量

TUstpFtdcVolumeType                MinVolume;
///触发价格

TUstpFtdcPriceType                 StopPrice;
///强平原因

TUstpFtdcForceCloseReasonType      ForceCloseReason;
///自动挂起标志

TUstpFtdcBoolType                  IsAutoSuspend;
///业务单元

TUstpFtdcBusinessUnitType          BusinessUnit;
///用户自定义域

TUstpFtdcCustomType                UserCustom;
///本地业务标识

TUstpFtdcBusinessLocalIDType       BusinessLocalID;
///业务发生日期

TUstpFtdcDateType                  ActionDay;
///策略类别

TUstpFtdcArbiTypeType              ArbiType;
///客户号

TUstpFtdcClientIDType              ClientID;
///会话编号

TUstpFtdcSessionIDType             SessionID;
///前置编号

TUstpFtdcFrontIDType               FrontID;
///条件单状态

TUstpFtdcConditionalOrderStatusType ConditionalOrderStatus;
};

```

6.2. 1360nRspConditionalOrderAction 方法

条件单撤单应答。

函数原形:

```

void OnRspConditionalOrderAction(
CUstpFtdcConditionalOrderActionField *pConditionalOrderAction,
CUstpFtdcRspInfoField *pRspInfo,

```

```
int nRequestID,
```

```
bool bIsLast)
```

参数:

```
///条件单撤单
```

```
struct CUstpFtdcConditionalOrderActionField
```

```
{
```

```
///交易所代码
```

```
TUstpFtdcExchangeIDType          ExchangeID;
```

```
///经纪公司编号
```

```
TUstpFtdcBrokerIDType            BrokerID;
```

```
///投资者编号
```

```
TUstpFtdcInvestorIDType          InvestorID;
```

```
///用户代码
```

```
TUstpFtdcUserIDType              UserID;
```

```
///本次撤单操作的本地编号
```

```
TUstpFtdcUserOrderLocalIDType    UserOrderActionLocalID;
```

```
///被撤条件单报单编号
```

```
TUstpFtdcUserOrderLocalIDType    UserOrderLocalID;
```

```
///本地业务标识
```

```
TUstpFtdcBusinessLocalIDType     BusinessLocalID;
```

```
///客户代码
```

```
TUstpFtdcClientIDType            ClientID;
```

```
///会话编号
```

```
TUstpFtdcSessionIDType           SessionID;
```

```
///前置编号
```

```
TUstpFtdcFrontIDType             FrontID;
```

```
///合约代码
```

```
TUstpFtdcInstrumentIDType        InstrumentID;
```

```
};
```

6.2. 1370nRspQryConditionalOrder 方法

条件单查询应答。

函数原形:

```
void OnRspQryConditionalOrder(
```

```
CUstpFtdcConditionalOrderField *pConditionalOrder,
```

```
CUstpFtdcRspInfoField *pRspInfo,  
int nRequestID,  
bool bIsLast)
```

参数:

///条件单查询响应

```
struct CUstpFtdcConditionalOrderField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType          BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType        ExchangeID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType        InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType            UserID;  
    ///指定通过此席位序号下单  
    TUstpFtdcSeatNoType            SeatNo;  
    ///合约代码/套利组合合约号  
    TUstpFtdcInstrumentIDType      InstrumentID;  
    ///条件单报单编号  
    TUstpFtdcUserOrderLocalIDType  UserOrderLocalID;  
    ///报单类型  
    TUstpFtdcOrderPriceTypeType    OrderPriceType;  
    ///买卖方向  
    TUstpFtdcDirectionType         Direction;  
    ///开平标志  
    TUstpFtdcOffsetFlagType        OffsetFlag;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType         HedgeFlag;  
    ///价格  
    TUstpFtdcPriceType             LimitPrice;  
    ///数量  
    TUstpFtdcVolumeType            Volume;  
    ///有效期类型  
    TUstpFtdcTimeConditionType      TimeCondition;
```

///GTD 日期
TUstpFtdcDateType GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType VolumeCondition;
///最小成交量
TUstpFtdcVolumeType MinVolume;
///触发价格
TUstpFtdcPriceType StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志
TUstpFtdcBoolType IsAutoSuspend;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType ActionDay;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///客户号
TUstpFtdcClientIDType ClientID;
///会话编号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
///条件单状态
TUstpFtdcConditionalOrderStatusType ConditionalOrderStatus;
///本次撤单操作的本地编号
TUstpFtdcUserOrderLocalIDType UserOrderActionLocalID;
///插入时间
TUstpFtdcTimeType InsertTime;
///撤销时间

```
TUstpFtdcTimeType          CancelTime;
};
```

6.2. 1380nRtnConditionalOrder 方法

条件单状态推送。

函数原形:

```
OnRtnConditionalOrder(
CUstpFtdcConditionalOrderField *pConditionalOrder)
```

参数:

///条件单查询响应

```
struct CUstpFtdcConditionalOrderField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType          BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType        ExchangeID;
```

///投资者编号

```
TUstpFtdcInvestorIDType        InvestorID;
```

///用户代码

```
TUstpFtdcUserIDType            UserID;
```

///指定通过此席位序号下单

```
TUstpFtdcSeatNoType            SeatNo;
```

///合约代码/套利组合合约号

```
TUstpFtdcInstrumentIDType      InstrumentID;
```

///条件单报单编号

```
TUstpFtdcUserOrderLocalIDType  UserOrderLocalID;
```

///报单类型

```
TUstpFtdcOrderPriceTypeType    OrderPriceType;
```

///买卖方向

```
TUstpFtdcDirectionType        Direction;
```

///开平标志

```
TUstpFtdcOffsetFlagType        OffsetFlag;
```

///投机套保标志

```
TUstpFtdcHedgeFlagType         HedgeFlag;
```

///价格

TUstpFtdcPriceType ///数量	LimitPrice;
TUstpFtdcVolumeType ///有效期类型	Volume;
TUstpFtdcTimeConditionType ///GTD 日期	TimeCondition;
TUstpFtdcDateType ///成交量类型	GTDDate;
TUstpFtdcVolumeConditionType ///最小成交量	VolumeCondition;
TUstpFtdcVolumeType ///触发价格	MinVolume;
TUstpFtdcPriceType ///强平原因	StopPrice;
TUstpFtdcForceCloseReasonType ///自动挂起标志	ForceCloseReason;
TUstpFtdcBoolType ///业务单元	IsAutoSuspend;
TUstpFtdcBusinessUnitType ///用户自定义域	BusinessUnit;
TUstpFtdcCustomType ///本地业务标识	UserCustom;
TUstpFtdcBusinessLocalIDType ///业务发生日期	BusinessLocalID;
TUstpFtdcDateType ///策略类别	ActionDay;
TUstpFtdcArbiTypeType ///客户号	ArbiType;
TUstpFtdcClientIDType ///会话编号	ClientID;
TUstpFtdcSessionIDType ///前置编号	SessionID;
TUstpFtdcFrontIDType ///条件单状态	FrontID;
TUstpFtdcConditionalOrderStatusType	ConditionalOrderStatus;

```

///本次撤单操作的本地编号
TUstpFtdcUserOrderLocalIDType      UserOrderActionLocalID;
///插入时间
TUstpFtdcTimeType                    InsertTime;
///撤销时间
TUstpFtdcTimeType                    CancelTime;
};

```

6.2. 1390nRspQryQuote 方法

报价查询应答。

函数原形:

```

void OnRspQryQuote(
CUstpFtdcQuoteField *pQuote,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)

```

参数:

///报价查询应答

```

struct CUstpFtdcQuoteField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType      BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType    ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType    InvestorID;
    ///用户代码
    TUstpFtdcUserIDType        UserID;
    ///合约代码
    TUstpFtdcInstrumentIDType  InstrumentID;
    ///买卖方向
    TUstpFtdcDirectionType     Direction;
    ///交易系统返回的系统报价编号
    TUstpFtdcQuoteSysIDType    QuoteSysID;
    ///用户设定的本地报价编号

```

TUstpFtdcUserQuoteLocalIDType ///飞马向交易系统报的本地报价编号	UserQuoteLocalID;
TUstpFtdcQuoteLocalIDType ///买方买入数量	QuoteLocalID;
TUstpFtdcVolumeType ///买方开平标志	BidVolume;
TUstpFtdcOffsetFlagType ///买方投机套保标志	BidOffsetFlag;
TUstpFtdcHedgeFlagType ///买方买入价格	BidHedgeFlag;
TUstpFtdcPriceType ///卖方卖出数量	BidPrice;
TUstpFtdcVolumeType ///卖方开平标志	AskVolume;
TUstpFtdcOffsetFlagType ///卖方投机套保标志	AskOffsetFlag;
TUstpFtdcHedgeFlagType ///卖方卖出价格	AskHedgeFlag;
TUstpFtdcPriceType ///业务单元	AskPrice;
TUstpFtdcBusinessUnitType ///用户自定义域	BusinessUnit;
TUstpFtdcCustomType ///拆分出来的买方用户本地报单编号	UserCustom;
TUstpFtdcUserOrderLocalIDType ///拆分出来的卖方用户本地报单编号	BidUserOrderLocalID;
TUstpFtdcUserOrderLocalIDType ///拆分出来的买方本地报单编号	AskUserOrderLocalID;
TUstpFtdcOrderLocalIDType ///拆分出来的卖方本地报单编号	BidOrderLocalID;
TUstpFtdcOrderLocalIDType ///询价编号	AskOrderLocalID;
TUstpFtdcQuoteSysIDType ///报价停留时间(秒)	ReqForQuoteID;
TUstpFtdcMeasureSecType	StandByTime;

///买方系统报单编号	
TUstpFtdcQuoteSysIDType	BidOrderSysID;
///卖方系统报单编号	
TUstpFtdcQuoteSysIDType	AskOrderSysID;
///报价单状态	
TUstpFtdcQuoteStatusType	QuoteStatus;
///插入时间	
TUstpFtdcTimeType	InsertTime;
///撤销时间	
TUstpFtdcTimeType	CancelTime;
///成交时间	
TUstpFtdcTimeType	TradeTime;
///客户号	
TUstpFtdcClientIDType	ClientID;
///交易所首次应答时间	
TUstpFtdcTimeType	InsertTimeExchange;
///会员编号	
TUstpFtdcParticipantIDType	ParticipantID;
///下单席位号	
TUstpFtdcSeatIDType	SeatID;
///资金帐号	
TUstpFtdcAccountIDType	AccountID;
///交易日	
TUstpFtdcTradingDayType	TradingDay;
///结算会员编号	
TUstpFtdcParticipantIDType	ClearingPartID;
///IP 地址	
TUstpFtdcIPAddressType	IPAddress;
///Mac 地址	
TUstpFtdcMacAddressType	MacAddress;
///用户端产品信息	
TUstpFtdcProductInfoType	UserProductInfo;
///会话编号	
TUstpFtdcSessionIDType	SessionID;
///前置编号	

```
TUstpFtdcFrontIDType          FrontID;
};
```

6.2. 1400nRspQryExecOrder 方法

行权查询应答。

函数原形:

```
void OnRspQryExecOrder(
CUstpFtdcExecOrderField *pExecOrder,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)
```

参数:

///行权应答

```
struct CUstpFtdcExecOrderField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType          BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType        ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType        OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType        InvestorID;
    ///用户代码
    TUstpFtdcUserIDType            UserID;
    ///合约代码
    TUstpFtdcInstrumentIDType      InstrumentID;
    ///用户本地报单号
    TUstpFtdcUserOrderLocalIDType  UserOrderLocalID;
    ///委托类型
    TUstpFtdcOrderTypeType         OrderType;
    ///期权对冲标识
    TUstpFtdcDeliveryFlagType      DeliveryFlag;
    ///投机套保标志
    TUstpFtdcHedgeFlagType         HedgeFlag;
```

///数量
TUstpFtdcVolumeType Volume;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///业务发生日期
TUstpFtdcDateType ActionDay;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///交易日
TUstpFtdcTradingDayType TradingDay;
///会员编号
TUstpFtdcParticipantIDType ParticipantID;
///最初下单用户编号
TUstpFtdcUserIDType OrderUserID;
///客户号
TUstpFtdcClientIDType ClientID;
///下单席位号
TUstpFtdcSeatIDType SeatID;
///插入时间
TUstpFtdcTimeType InsertTime;
///本地报单编号
TUstpFtdcOrderLocalIDType OrderLocalID;
///报单来源
TUstpFtdcOrderSourceType OrderSource;
///报单状态
TUstpFtdcOrderStatusType OrderStatus;
///撤销时间
TUstpFtdcTimeType CancelTime;
///撤单用户编号
TUstpFtdcUserIDType CancelUserID;
///交易所首次应答时间
TUstpFtdcTimeType InsertTimeExchange;
///今昨仓标志

```

TUstpFtdcOffsetFlagType      OffsetFlag;
///会话编号
TUstpFtdcSessionIDType       SessionID;
///前置编号
TUstpFtdcFrontIDType         FrontID;
};

```

6.2. 1410nRspQueryHisTransferSerial 方法

查询历史转帐流水应答。

函数原形：

```

void OnRspQueryHisTransferSerial(
CUstpFtdcHisTransferSerialFieldRspField *pHisTransferSerialFieldRsp,
CUstpFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast)

```

参数：

///查询历史转帐流水响应响应

```

struct CUstpFtdcHisTransferSerialFieldRspField
{
    ///平台流水号
    TUstpFtdcPlateSerialType      PlateSerial;
    ///交易发起方日期
    TUstpFtdcDateType             TradeDate;
    ///交易日期
    TUstpFtdcDateType             TradingDay;
    ///交易时间
    TUstpFtdcTimeType             TradeTime;
    ///交易代码
    TUstpFtdcTradeCodeType        TradeCode;
    ///会话编号
    TUstpFtdcSessionIDType        SessionID;
    ///银行代码
    TUstpFtdcBankIDType           BankID;
    ///银行分中心代码
    TUstpFtdcInvestorIDType       BankBrchID;

```

///银行帐号类型
TUstpFtdcBankAccType BankAccType;
///银行帐号
TUstpFtdcBankAccountType BankAccount;
///银行流水号
TUstpFtdcBankSerialType BankSerial;
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcFutureBranchIDType BrokerBranchID;
///期货公司帐号类型
TUstpFtdcFutureAccType FutureAccType;
///资金帐号
TUstpFtdcAccountIDType AccountID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///期货公司流水号
TUstpFtdcFutureSerialType FutureSerial;
///交易发起方
TUstpFtdcTransferSourceType TransferSource;
///出入金方向
TUstpFtdcTransferDirectionType TransferDirection;
///证件类型
TUstpFtdcIdCardTypeType IdCardType;
///证件号码
TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
///币种代码
TUstpFtdcCurrencyIDType CurrencyID;
///金额
TUstpFtdcMoneyType TradeAmount;
///应收客户费用
TUstpFtdcMoneyType CustFee;
///应收期货公司费用
TUstpFtdcMoneyType BrokerFee;
///有效标志


```

TUstpFtdcAvailabilityFlagTypeType      AvailabilityFlag;
///操作员
TUstpFtdcOperatorCodeType              OperatorCode;
///新银行帐号
TUstpFtdcBankAccountType               BankNewAccount;
///流水返回码
TUstpFtdcTransferRtnCodeType           RtnCode;
///流水返回信息
TUstpFtdcTransferRtnMsgType            RtnMsg;
};

```

6.3 CUSTPFtdcTraderApi 接口

CUstpFtdcTraderApi 接口提供给用户的功能包括，报单的录入、报单的撤销、报单的查询、成交单查询、客户持仓查询、合约查询、合约交易状态查询、交易所公告查询等功能。

6.3.1 CreateFtdcTraderApi 方法

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

函数原形：

```
static CUstpFtdcTradeApi *CreateFtdcTradeApi(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飞马平台发布消息的状态。默认值代表当前目录。

返回值：

返回一个指向 CUstpFtdcTradeApi 实例的指针。

6.3.2 GetVersion 方法

获取飞马 API 版本号。

函数原形：

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

参数：

nMajorVersion: 主版本号。

nMinorVersion: 子版本号

返回值：

返回一个飞马 API 版本描述字符串。

6.3.3 Release 方法

释放一个 CUstpFtdcTradeApi 实例。不能使用 delete 方法

函数原形:

```
void Release();
```

6.3.4 Init 方法

使客户端开始与飞马平台建立连接，连接成功后可以进行登陆。

函数原形:

```
void Init();
```

6.3.5 Join 方法

客户端等待一个接口实例线程的结束。

函数原形:

```
void Join();
```

6.3.6 GetTradingDay 方法

获得当前交易日。只有当与飞马平台连接建立后才会取到正确的值。

函数原形:

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

返回值:

返回一个指向日期信息字符串的常量指针。

6.3.7 RegisterSpi 方法

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

函数原形:

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

参数:

pSpi: 实现了 CUstpFtdcTraderSpi 接口的实例指针。

6.3.8 RegisterFront 方法

设置飞马平台的网络通讯地址，飞马平台拥有多个通信地址，但用户只需要选择一个通信地址。

函数原形：

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

参数：

pszFrontAddress：指向后台服务器地址的指针。

服务器地址的格式为：“protocol://ipaddress:port”，如：“tcp://127.0.0.1:17001”。“tcp”代表传输协议，“127.0.0.1”代表服务器地址。”17001”代表服务器端口号。

6.3.9 RegisterQryFront 方法

设置飞马平台的前置机网络通讯地址，飞马平台拥有多个通信地址，但用户只需要选择一个通信地址。

函数原形：

```
void RegisterQryFront(char *pszFrontAddress);
```

参数：

pszFrontAddress：指向后台服务器地址的指针。

服务器地址的格式为：“protocol://ipaddress:port”，如：“tcp://127.0.0.1:17001”。“tcp”代表传输协议，“127.0.0.1”代表服务器地址。”17001”代表服务器端口号。

6.3.1 RegisterDSProxyUserCert 方法

穿透监管中继处用户证书认证请求。

函数原形：

```
int RegisterDSProxyUserCert(  
    CUstpFtdcDSProxyCheckUserInfoField *pDSProxyUserUserInfo,  
    CUstpFtdcDSProxyUserCertInField *pDSProxyUserCertIn,  
    CUstpFtdcDSProxyUserCertOutField *pDProxyUserCertOut,  
    int nRequestID);
```

参数：

pDSProxyUserUserInfo 指向穿透监管中继验证客户信息结构的地址。

pDSProxyUserCertIn 指向穿透监管中继处接收到的终端认证信息结构的地址。

pDProxyUserCertOut 指向穿透监管中继处接终端认证返回信息结构的地址。

///穿透监管中继验证客户信息

```
struct CUstpFtdcDSProxyCheckUserInfoField
```

```

{
    ///用户 AppID
    TUstpFtdcDSAppIDType    AppID;

    ///用户授权码
    TUstpFtdcDSAAuthorizationCodeType    AuthCode;

    ///密钥加密类型
    TUstpFtdcDSKeyEncryptType    EncryptType;
};

///穿透监管中继处接收到的终端认证信息
struct CUstpFtdcDSProxyUserCertInField
{
    ///穿透监管中继处接收到的终端认证信息
    TUstpFtdcDSProxyCertStreamDataType    UserCertReqInfo;
};

///穿透监管中继处接终端认证返回信息
struct CUstpFtdcDSProxyUserCertOutField
{
    ///穿透监管中继处证书认证的返回结果
    TUstpFtdcDSProxyCertStreamDataType    UserCertRspInfo;

    ///中继处返回数据已使用长度信息
    TUstpFtdcDSBufLenType    UserCertRspInfoLen;
};

```

6.3.2 RegisterNameServer 方法

设置飞马 NameServer 的网络通讯地址，用于获取行情服务列表。交易系统拥有多个 NameServer，用户可以同时注册多个 NameServer 的网络通讯地址。

该方法要在 Init 方法之前调用。

函数原型：

```
void RegisterNameServer (char *pszNsAddress, char *pszInvestorID);
```

参数:

pszNsAddress: 指向飞马服务端 NameServer 网络通讯地址的指针。网络地址的格式为: “protocol://ipaddress:port”, 如: ” tcp://127.0.0.1:17001”。 “tcp” 代表传输协议, “127.0.0.1” 代表服务器地址。 ” 17001” 代表服务器端口号。

pszInvestorID: 字符串, 代表用户的投资者号 (客户号)。

注意: 此接口 1.05_r1 版本启用!

6.3.3 SubscribePrivateTopic 方法

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的数据。

函数原形:

```
void SubscribePrivateTopic(USTP_TE_RESUME_TYPE nResumeType);
```

参数:

nResumeType: 私有流重传方式

USTP_TERT_RESTART: 从本交易日开始重传

USTP_TERT_RESUME: 从上次收到的续传

USTP_TERT_QUICK: 只传送登录后私有流的内容

6.3.4 SubscribePublicTopic 方法

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。

函数原形:

```
void SubscribePublicTopic(USTP_TE_RESUME_TYPE nResumeType);
```

参数:

nResumeType: 公共流重传方式

USTP_TERT_RESTART: 从本交易日开始重传

USTP_TERT_RESUME: 从上次收到的续传

USTP_TERT_QUICK: 只传送登录后公共流的内容

6.3.5 SubscribeForQuote 方法

订阅询价流。该方法要在 Init 方法前调用。若不调用则不会收到询价流的数据。

函数原形:

```
void SubscribeForQuote(USTP_TE_RESUME_TYPE nResumeType);
```

参数:

nResumeType: 公共流重传方式

USTP_TERT_RESTART: 从本交易日开始重传

USTP_TERT_RESUME: 从上次收到的续传

USTP_TERT_QUICK: 只传送登录后公共流的内容

6.3.6 SetHeartbeatTimeout 方法

设置心跳超时时间。

函数原形:

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

参数:

timeout: 心跳超时时间, 单位秒。

6.3.7 ReqUserLogin 方法

用户发出登陆请求。

函数原形:

```
int ReqUserLogin (  
    CUstpFtdcReqUserLoginField *pReqUserLogin,  
    int nRequestID);
```

参数:

pReqUserLogin: 指向用户登录请求结构的地址。

用户登录请求结构:

```
struct CUstpFtdcReqUserLoginField  
{  
    ///交易日  
    TUstpFtdcDateType    TradingDay;  
    ///交易用户代码  
    TUstpFtdcUserIDType  UserID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///密码  
    TUstpFtdcPasswordType Password;  
    ///用户端产品信息  
    TUstpFtdcProductInfoType UserProductInfo;  
    ///接口端产品信息
```

```

TUstpFtdcProductInfoType    InterfaceProductInfo;
///协议信息
TUstpFtdcProtocolInfoType    ProtocolInfo;
///IP 地址
TUstpFtdcIPAddressType    IPAddress;
///Mac 地址
TUstpFtdcMacAddressType    MacAddress;
///数据中心代码
TUstpFtdcDataCenterIDType    DataCenterID;
///客户端运行文件大小
TUstpFtdcFileSizeType    UserProductFileSize;
///客户认证类型
TUstpFtdcAuthenticate2TypeType    Authenticate2Type;
///客户认证密码
TUstpFtdcAuthenticate2PasswordType    Authenticate2Password;
///开发厂商终端编码
TUstpFtdcTerminalCodeType    TerminalCode;
};

```

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

用户需要填写 UserProductInfo 字段, 即客户端的产品信息, 如软件开发商、版本号等。

InterfaceProductInfo 和 ProtocolInfo 只须占位, 不必有效赋值。

返回值:

0, 代表成功。

-1, 表示网络连接失败;

-2, 表示未处理请求超过许可数;

-3, 表示每秒发送请求数超过许可数。

6.3.8 ReqUserLogout 方法

用户发出登出请求。

函数原形:

```

int ReqUserLogout (
    CUstpFtdcReqUserLogoutField * pReqUserLogout,
    int nRequestID);

```

参数:

pReqUserLogout: 指向用户登出请求结构的地址。

用户登出请求结构:

```

struct CUstpFtdcReqUserLogoutField
{

```

```

        ///经纪公司编号
        TUstpFtdcBrokerIDType BrokerID;
        ///交易用户代码
        TUstpFtdcUserIDType UserID;
};

```

6.3.9 ReqUserPasswordUpdate 方法

用户密码修改请求。

函数原形:

```

int ReqUserPasswordUpdate (
    CUstpFtdcUserPasswordUpdateField *pUserPasswordUpdate,
    int nRequestID);

```

参数:

pUserPasswordUpdate: 指向用户口令修改结构的地址。

用户口令修改结构:

```

struct CUstpFtdcUserPasswordUpdateField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易用户代码
    TUstpFtdcUserIDType UserID;
    ///旧密码
    TUstpFtdcPasswordType OldPassword;
    ///新密码
    TUstpFtdcPasswordType NewPassword;
};

```

6.3.10 ReqSettlementInfoConfirm 方法

请求对账单确认。

函数原形:

```

int ReqSettlementInfoConfirm (
    CUstpFtdcInputSettlementInfoConfirmField *pInputSettlementInfoConfirm,
    int nRequestID)

```

参数:

pInputSettlementInfoConfirm: 指向请求对账单结构的地址。

请求对账单结构:

///结算单确认请求

```

struct CUstpFtdcInputSettlementInfoConfirmField

```



```
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
};
```

6.3.11 ReqQuerySettlementInfo 方法

请求查询对账单确认。

函数原形:

```
int ReqQuerySettlementInfo (
    CUstpFtdcSettlementQryReqField *pSettlementQryReq,
    int nRequestID)
```

参数:

pSettlementQryReq: 指向请求查询对账单结构的地址。

请求查询对账单结构:

///查询对账单结构请求

```
struct CUstpFtdcSettlementQryReqField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID; //必填
    ///交易日
    TUstpFtdcDateType        TradingDay;
    ///结算单类型
    TUstpFtdcSettlementTypeType SettlementType;
    ///交易日
    TUstpFtdcDateType        StartDay;
    ///交易日
    TUstpFtdcDateType        EndDay;
};
```

6.3.12 ReqMsgNotifyConfirm 方法

请求通知消息的确认。

函数原形:

```
int ReqMsgNotifyConfirm (  
    CUstpFtdcInputMsgNotifyConfirmField* pInputMsgNotifyConfirm,  
    int nRequestID)
```

参数:

pInputMsgNotifyConfirm: 指向请求消息确认结构的地址。

请求消息确认结构:

///消息确认请求

```
struct CUstpFtdcInputMsgNotifyConfirmField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType   BrokerID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///消息号  
    TUstpFtdcMsgNoType   MsgNumber;  
};
```

6.3.13 ReqOrderInsert 方法

客户端发出报单录入请求。

函数原形:

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

参数:

pInputOrder: 指向输入报单结构的地址。

输入报单结构:

```
struct CUstpFtdcInputOrderField  
{  
    ///经纪公司编号, 必填字段  
    TUstpFtdcBrokerIDType   BrokerID;  
    ///交易所代码, 必填字段 (支持中金所、上期所、大商所、郑商所)  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///系统报单编号  
    TUstpFtdcOrderSysIDType OrderSysID;  
    ///投资者编号, 必填字段 (客户资金账户)  
    TUstpFtdcInvestorIDType InvestorID;
```

```
///用户代码, 必填字段 (飞马用户代码)
TUstpFtdcUserIDType UserID;

///指定下单席位编号 (取值范围[1-N], N 为可用席位数目, 超出范围的随机分配席位-暂
不支持)
TUstpFtdcSeatNoType SeatNo;

///合约代码, 必填字段
TUstpFtdcInstrumentIDType InstrumentID;

///用户本地报单号, 必填字段
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;

///报单价格条件, 必填字段 (支持市价、限价、五档价和最优价)
TUstpFtdc OrderPriceTypeType OrderPriceType;

///买卖方向, 必填字段
TUstpFtdcDirectionType Direction;

///开平标志, 必填字段 (中金所支持开仓和平仓, 上期所支持开仓、平今和平昨)
TUstpFtdcOffsetFlagType OffsetFlag;

///投机套保标志, 必填字段 (支持投机、套保、套利、做市商编码)
TUstpFtdcHedgeFlagType HedgeFlag;

///价格, 必填字段
TUstpFtdcLimitPriceType LimitPrice;

///数量, 必填字段
TUstpFtdcVolumeType Volume;

///有效期类型, 必填字段 (“立即完成, 否则撤销” 1 OR “当日有效” 3)
TUstpFtdcTimeConditionType TimeCondition;

///GTD 日期 (暂不支持, 保留域)
TUstpFtdcDateType GTDDate;

///成交量类型, 必填字段 (支持 “任意数量” 和 “全部数量”)
TUstpFtdcVolumeConditionType VolumeCondition;

///最小成交量 (暂不支持, 保留域)
TUstpFtdcVolumeType MinVolume;

///止损价 (暂不支持, 保留域)
TUstpFtdcPriceType StopPrice;

///强平原因, 只支持” 非强平”
TUstpFtdcForceCloseReasonType ForceCloseReason;

///自动挂起标志 (暂不支持, 保留域)
TUstpFtdcBoolType IsAutoSuspend;
```

```

    ///业务单元（暂不支持，保留域）
    TUstpFtdcBusinessUnitType    BusinessUnit;
    ///用户自定义域 64 字节
    TUstpFtdcCustomType    UserCustom;
    ///本地业务标识（暂不支持，保留域）
    TUstpFtdcBusinessLocalIDType    BusinessLocalID;
    ///业务发生日期
    TUstpFtdcDateType    ActionDay;
    ///策略类别（套利组合业务必填字段）
    TUstpFtdcArbiTypeType    ArbiType;
    ///客户号
    TUstpFtdcClientIDType    ClientID;
};

```

注意:

用户本地报单号 *UserOrderLocalID* 是一个 21 位的字符串，下一笔报单的本地报单编号需要比前一笔报单的本地报单编号大（不一定需要连续），其比较方式为字符串的比较。

MaxOrderLocalID < UserOrderLocalID

普通限价单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) ExchangeID, 交易所代码, “CFFEX/SHFE/DCE/ZCE”
- 3) InvestorID, 投资者编号, 形如 “10000029”;
- 4) UserID, 用户代码, 形如 “test1”;
- 5) InstrumentID, 合约代码, 形如 “IF1109/cu1206/m1301”;
- 6) **OrderPriceType**, 报单价格条件, 限价单 **USTP_FTDC_OPT_LimitPrice**;
- 7) Direction, 买卖方向, **USTP_FTDC_D_Buy** 表示买, **USTP_FTDC_D_Sell** 表示卖;
- 8) OffsetFlag, 开平标志, “0” 表示开仓, “1” 表示平仓;
- 9) HedgeFlag, 投机套保标志, “1” 为投机, “2” 为套保, “3” 为套利, “4” 为做市商;
- 10) LimitPrice, 价格, 形如 3500.00;
- 11) Volume, 数量, 例如 5 表示 5 手;
- 12) **TimeCondition**, 有效期类型, 为 **USTP_FTDC_TC_GFD** (“当日有效”);
- 13) **VolumeCondition**, 成交量类型, 为 **USTP_FTDC_VC_AV** (“任意数量”);
- 14) **ArbiType**, 套利类型, 为 **USTP_FTDC_AT_Basic** (“普通单”)
- 15) ForceCloseReason, 强平原因, 为 **USTP_FTDC_FCC_NotForceClose** (“非强平”);

16) UserOrderLocalID, 用户本地报单编号, 形如"00000025"。

普通市价单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如 "2008";
- 2) ExchangeID, 交易所代码, "CFFEX/SHFE/DCE/ZCE"
- 3) InvestorID, 客户号, 形如 "10000029";
- 4) UserID, 交易用户代码, 形如 "test1";
- 5) InstrumentID, 合约代码, 形如 "IF1109";
- 6) **OrderPriceType, 报单价格条件, 市价单为 USTP_FTDC_OPT_AnyPrice;**
- 7) Direction, 买卖方向, USTP_FTDC_D_Buy 表示买, USTP_FTDC_D_Sell 表示卖;
- 8) OffsetFlag, 开平标志, "0" 表示开仓, "1" 表示平仓;
- 9) HedgeFlag, 投机套保标志, "1" 为投机, "2" 为套保, "3" 为套利, "4" 为做市商;
- 10) Volume, 数量, 例如 5 表示 5 手;
- 11) **TimeCondition, 有效期类型, 为 USTP_FTDC_TC_IOC ("立即成交, 否则撤销");**
- 12) VolumeCondition, 成交量类型, 为 USTP_FTDC_VC_AV ("任意数量");
- 13) **ArbiType, 套利类型, 为 USTP_FTDC_AT_Basic ("普通单")**
- 14) ForceCloseReason, 强平原因, 只能为 USTP_FTDC_FCC_NotForceClose ("非强平");
- 15) UserOrderLocalID, 本地报单编号, 形如"00000025"。

期货套利组合单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如 "2008";
- 2) **ExchangeID, 交易所代码, "DCE/ZCE"**
- 3) InvestorID, 客户号, 形如 "10000029";
- 4) UserID, 交易用户代码, 形如 "test1";
- 5) **InstrumentID, 合约代码, 形如 "SP jm1301&jm1303/SPC a1301&m1301/SPD SR609&SR611";**
- 6) **OrderPriceType, 报单价格条件, 只能为 USTP_FTDC_OPT_AnyPrice;**
- 7) Direction, 买卖方向, USTP_FTDC_D_Buy 表示买, USTP_FTDC_D_Sell 表示卖;
- 8) OffsetFlag, 开平标志, "0" 表示开仓, "1" 表示平仓;
- 9) **HedgeFlag, 投机套保标志, 套利组合单类型只能为 "3" - 套利;**
- 10) Volume, 数量, 例如 5 表示 5 手;
- 11) **TimeCondition, 有效期类型, 为 USTP_FTDC_TC_GFD ("当日有效");**
- 12) VolumeCondition, 成交量类型, 为 USTP_FTDC_VC_AV ("任意数量");

13) **ArbiType**, 套利类型, 只支持 **USTP_FTDC_AT_SP** (“跨期套利”) 或者 **USTP_FTDC_AT_SPC** (“跨品种套利”)

14) **ForceCloseReason**, 强平原因, 只能为 **USTP_FTDC_FCC_NotForceClose** (“非强平”);

15) **UserOrderLocalID**, 本地报单编号, 形如“00000025”。

期权组合单必须填写的字段包括:

1) **BrokerID**, 会员号, 形如 “2008”;

2) **ExchangeID**, 交易所代码, “ZCE”

3) **InvestorID**, 客户号, 形如 “10000029”;

4) **UserID**, 交易用户代码, 形如 “test1”;

5) **InstrumentID**, 合约代码, 形如 “SPjm1301&jm1303/SPC a1301&m1301/SPD SR609&SR611”;

6) **OrderPriceType**, 报单价格条件, 只能为 **USTP_FTDC_OPT_AnyPrice**;

7) **Direction**, 买卖方向, **USTP_FTDC_D_Buy** 表示买, **USTP_FTDC_D_Sell** 表示卖;

8) **OffsetFlag**, 开平标志, “0” 表示开仓, “1” 表示平仓;

9) **HedgeFlag**, 投机套保标志, 套利组合单类型只能为 “3” - 套利;

10) **Volume**, 数量, 例如 5 表示 5 手;

11) **TimeCondition**, 有效期类型, 为 **USTP_FTDC_TC_GFD** (“当日有效”);

12) **VolumeCondition**, 成交量类型, 为 **USTP_FTDC_VC_AV** (“任意数量”);

13) **ArbiType**, 套利类型, 只支持 **USTP_FTDC_AT_SP** (“跨期套利”) 或者 **USTP_FTDC_AT_SPC** (“跨品种套利”)

14) **ForceCloseReason**, 强平原因, 只能为 **USTP_FTDC_FCC_NotForceClose** (“非强平”);

15) **UserOrderLocalID**, 本地报单编号, 形如“00000025”。

中金所支持 9 种订单类型, 报单时 **ExchangeID** 和 **InstrumentID** 应填写中金所交易所代码和合约代码, 相关详细说明如下:

订单类型	OrderPriceType	TimeCondition	VolumeCondition	备注
限价单	USTP_FTDC_OPT_LimitPrice	USTP_FTDC_TC_GFD	USTP_FTDC_VC_AV	当日有效, 限价成交
FOK	USTP_FTDC_OPT_LimitPrice	USTP_FTDC_TC_IOC	USTP_FTDC_VC_CV	限价立即全部成交否则自动撤销
FAK	USTP_FTDC_OPT_LimitPrice	USTP_FTDC_TC_IOC	USTP_FTDC_VC_AV	限价立即成交剩余自动撤销

市价单	USTP_FTDC_OPT_An yPrice	USTP_FTDC_TC_IOC	USTP_FTDC_VC_AV	市价立即成交否则撤单
市价转限价	USTP_FTDC_OPT_An yPrice	USTP_FTDC_TC_GFD	USTP_FTDC_VC_AV	市价单未成交部分转为最新限价单
五档市价	USTP_FTDC_OPT_Fi veLevelPrice	USTP_FTDC_TC_IOC	USTP_FTDC_VC_AV	市价单与对手方五档价格报单尝试成交， 剩余未成交部分撤销
五档市价转 限价	USTP_FTDC_OPT_Fi veLevelPrice	USTP_FTDC_TC_GFD	USTP_FTDC_VC_AV	市价单与对手方五档价格报单尝试成交， 剩余未成交部分转为最新限价单
最优价	USTP_FTDC_OPT_Be stPrice	USTP_FTDC_TC_IOC	USTP_FTDC_VC_AV	市价单与对手方最优一档价格报单尝试成 交，剩余未成交部分撤销
最优价转限 价	USTP_FTDC_OPT_Be stPrice	USTP_FTDC_TC_GFD	USTP_FTDC_VC_AV	市价单与对手方最优一档价格报单尝试成 交，剩余未成交部分转为最新限价单

*FOK 单和 FAK 单为限价单，除这 3 个字段外的其他字段填充方式参照限价单。
其余 5 种订单为市价单，除这 3 个字段外的其他字段填充方式参照市价单。*

6.3.14 ReqOrderAction 方法

客户端发出报单操作请求，包括报单的撤销、报单的挂起（暂不支持）、报单的激活（暂不支持）、报单的修改（暂不支持）。

函数原形：

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

参数：

pOrderAction：指向报单操作结构的地址。

报单操作结构：

```
struct CUstpFtdcOrderActionField
{
```

///交易所代码, 必填字段

TUstpFtdcExchangeIDType ExchangeID;

///交易所系统报单编号（不为空按该字段操作，为空按本地报单编号操作）

TUstpFtdcOrderSysIDType OrderSysID;

///经纪公司编号, 必填字段

TUstpFtdcBrokerIDType BrokerID;

///投资者编号, 必填字段

TUstpFtdcInvestorIDType InvestorID;

///用户代码, 必填字段

TUstpFtdcUserIDType UserID;

~~///指定下单席位编号（取值范围[1-N]，N为可用席位数目，超出范围的随机分配席位）~~

~~TUstpFtdcSeatNoType SeatNo;~~

```

    ///本次撤单 Req 的本地编号，必填字段    （需要保证按字典序递增）
    TustpFtdcUserOrderLocalIDType    UserOrderActionLocalID;
    ///被撤订单的本地报单编号
    TustpFtdcUserOrderLocalIDType    UserOrderLocalID;
    ///报单操作标志，必填字段（只支持删除）
    TustpFtdcActionFlagType    ActionFlag;
    ///价格 （暂不支持，保留域）
    TustpFtdcPriceType    LimitPrice;
    ///数量变化（暂不支持，保留域）
    TustpFtdcVolumeType    VolumeChange;
    ///本地业务标识
    TustpFtdcBusinessLocalIDType    BusinessLocalID;
    ///客户代码
    TustpFtdcClientIDType    ClientID;
    ///会话编号
    TustpFtdcSessionIDType    SessionID;
    ///前置编号
    TustpFtdcFrontIDType    FrontID;
    ///合约代码
    TustpFtdcInstrumentIDType    InstrumentID;
};

```

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

返回值:

0，代表成功。

-1，表示网络连接失败；

-2，表示未处理请求超过许可数；

-3，表示每秒发送请求数超过许可数。

撤单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) ExchangeID, 交易所代码, 形如 “CFEX”;
- 3) InvestorID, 投资者编号, 形如 “10000029”;
- 4) UserID, 用户代码, 形如 “test1”;
- 5) OrderSysID 或者 UserOrderLocalID 任选其一, 被撤订单的订单编号, 优先按照 OrderSysID 定位, 形如 “00000025”;
- 6) UserOrderActionLocalID, 撤单请求的本地编号, 与 UserOrderLocalID 同一个序列按字典序增加, 形如 “00000025”;
- 7) ActionFlag, 操作标志, 只能填 **USTP_FTDC_AF_Delete** (“删除”);

6.3.15 ReqQuoteInsert 方法

报价录入请求

做市商进行双边报价的接口

函数原形:

```
int ReqQuoteInsert (  
    CUstpFtdcInputQuoteField *pInputQuote,  
    int nRequestID)
```

参数:

pInputQuote 指报价录入结构的地址。

报价录入结构

```
struct CUstpFtdcInputQuoteField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///买卖方向  
    TUstpFtdcDirectionType Direction;  
    ///交易系统返回的系统报价编号  
    TUstpFtdcQuoteSysIDType QuoteSysID;  
    ///用户设定的本地报价编号  
    TUstpFtdcUserQuoteLocalIDType UserQuoteLocalID;  
    ///飞马向交易系统报的本地报价编号  
    TUstpFtdcQuoteLocalIDType QuoteLocalID;  
    ///买方买入数量  
    TUstpFtdcVolumeType BidVolume;  
    ///买方开平标志  
    TUstpFtdcOffsetFlagType BidOffsetFlag;  
    ///买方投机套保标志  
    TUstpFtdcHedgeFlagType BidHedgeFlag;  
    ///买方买入价格  
    TUstpFtdcPriceType BidPrice;  
    ///卖方卖出数量  
    TUstpFtdcVolumeType AskVolume;  
    ///卖方开平标志
```

```

TustpFtdcOffsetFlagType AskOffsetFlag;
///卖方投机套保标志
TustpFtdcHedgeFlagType AskHedgeFlag;
///卖方卖出价格
TustpFtdcPriceType AskPrice;
///业务单元
TustpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TustpFtdcCustomType UserCustom;
///拆分出来的买方用户本地报单编号
TustpFtdcUserOrderLocalIDType BidUserOrderLocalID;
///拆分出来的卖方用户本地报单编号
TustpFtdcUserOrderLocalIDType AskUserOrderLocalID;
///拆分出来的买方本地报单编号
TustpFtdcOrderLocalIDType BidOrderLocalID;
///拆分出来的卖方本地报单编号
TustpFtdcOrderLocalIDType AskOrderLocalID;
///询价编号
TustpFtdcQuoteSysIDType ReqForQuoteID;
///报价停留时间(秒)
TustpFtdcMeasureSecType StandByTime;
///客户号
TustpFtdcClientIDType ClientID;
///会话编号
TustpFtdcSessionIDType SessionID;
///前置编号
TustpFtdcFrontIDType FrontID;
};

```

报价单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如“2008”;
- 2) ExchangeID, 交易所代码, 只支持“CFFEX”
- 3) InvestorID, 投资者编号, 形如“10000029”;
- 4) UserID, 用户代码, 形如“test1”;
- 5) InstrumentID, 合约代码, 形如“IF1109”;
- 6) UserQuoteLocalID, 用户本地报价单号, 形如“a8703bf4000000000101”;
- 7) Direction, 买卖方向, 取值范围【0,1,9】, 9: 代表双边报价, 0: 代表只有买方报价, 1: 代表只有
卖方报价
- 8) BidVolume, 买方买入数量, 例如 5 表示 5 手;

- 9) BidOffsetFlag, 买方开平标志, “0” 表示开仓, “1” 表示平仓;
- 10) BidHedgeFlag, 买方投机套保标志;
- 11) BidPrice, 价格, 形如 3510.00;
- 12) AskVolume, 买方买入数量, 例如 5 表示 5 手;
- 13) AskOffsetFlag, 买方开平标志, “0” 表示开仓, “1” 表示平仓;
- 14) AskHedgeFlag, 买方投机套保标志, 必须填 “4”, 表示用做市商编码;
- 15) AskPrice, 价格, 形如 3500.00;
- 16) ReqForQuoteID, 询价编号, 大商所、郑商所期权应价必须指定是针对哪笔询价的应价

6.3.16 ReqQuoteAction 方法

报价操作请求

撤销某一笔报价, 仅做市商有权限进行此操作。

函数原形:

```
int ReqQuoteAction (  
    CUstpFtdcQuoteActionField *pQuoteAction,  
    int nRequestID)
```

参数:

pQuoteAction 指报价操作结构的地址。

报价操作结构

```
struct CUstpFtdcQuoteActionField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///交易系统返回的系统报价编号  
    TUstpFtdcQuoteSysIDType QuoteSysID;  
    ///用户设定的被撤的本地报价编号  
    TUstpFtdcUserQuoteLocalIDType UserQuoteLocalID;  
    ///用户向飞马报的本地撤消报价编号  
    TUstpFtdcUserQuoteLocalIDType UserQuoteActionLocalID;  
    ///报单操作标志  
    TUstpFtdcActionFlagType ActionFlag;  
    ///业务单元
```

```

TUstpFtdcBusinessUnitType  BusinessUnit;
///用户自定义域
TUstpFtdcCustomType  UserCustom;
///买卖方向
TUstpFtdcDirectionType  Direction;
};

```

撤销报价单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) ExchangeID, 交易所代码, 只支持 “CFEX”
- 3) InvestorID, 投资者编号, 形如 “10000029”;
- 4) UserID, 用户代码, 形如 “test1”;
- 5) InstrumentID, 合约代码, 形如 “IF1109”;
- 6) QuoteSysID /UserQuoteLocalID, 要撤销报价单的系统报价单号/用户本地报价单号, 形如 “000000101”。二者选其一即可, 优先按照系统报价单号撤销;
- 7) UserQuoteActionLocalID, 撤销报价操作的用户本地操作编号, 形如 “000000101”;

注意: MaxOrderLocalID < UserQuoteActionLocalID

6.3.17 ReqForQuote 方法

询价请求

由客户发起的询价请求。针对某一个合约询价后, 做市商的报价在该合约的行情上反映。

函数原形:

```

int ReqForQuote (
    CUstpFtdcReqForQuoteField *pReqForQuote,
    int nRequestID)

```

参数:

pReqForQuote 指向询价结构的地址。

询价结构

```

struct CUstpFtdcReqForQuoteField
{
    ///询价编号
    TUstpFtdcQuoteSysIDType ReqForQuoteID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType  BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号

```

```

TUstpFtdcInvestorIDType InvestorID;
///用户代码
TUstpFtdcUserIDType UserID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///买卖方向
TUstpFtdcDirectionType Direction;
///客户号
TUstpFtdcClientIDType ClientID;
///交易日
TUstpFtdcDateType TradingDay;
///询价时间
TUstpFtdcTimeType ReqForQuoteTime;
};

```

询价单必须填充的字段包括:

- 1) BrokerID, 会员号, 形如“2008”;
- 2) ExchangeID, 交易所代码, 只支持“CFFEX”
- 3) InvestorID, 投资者编号, 形如“10000029”;
- 4) UserID, 用户代码, 形如“test1”;
- 5) InstrumentID, 合约代码, 形如“IF1109”;

6.3.18 ReqMarginCombAction 方法

策略组合/策略拆分请求

此接口用于对已有持仓进行组合, 或者对已有组合进行拆分。该接口只能对中金所公布的策略组合规则表中的组合合约进行组合。

函数原形:

```

int ReqMarginCombAction(
    CUstpFtdcInputMarginCombActionField *pInputMarginCombAction,
    int nRequestID)

```

参数:

pInputMarginCombAction 指向策略组合结构的地址。

询价结构: 详见应答信息;

```

struct CUstpFtdcInputMarginCombActionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码

```

```
TUstpFtdcExchangeIDType ExchangeID;  
///交易用户代码  
TUstpFtdcUserIDType UserID;  
///投资者编号  
TUstpFtdcInvestorIDType InvestorID;  
///投机套保标志  
TUstpFtdcHedgeFlagType HedgeFlag;  
///用户本地编号  
TUstpFtdcUserOrderLocalIDType UserActionLocalID;  
///组合合约代码  
TUstpFtdcInstrumentIDType CombInstrumentID;  
///组合数量  
TUstpFtdcVolumeType CombVolume;  
///组合动作方向  
TUstpFtdcCombInstrumentIDType CombDirection;  
///本地编号  
TUstpFtdcOrderLocalIDType ActionLocalID;  
///系统编号  
TUstpFtdcOrderSysIDType OrderSysID;  
///组合操作状态  
TUstpFtdcCombActionStatusType CombActionStatus;  
///组合方向  
TUstpFtdcDirectionType Direction;  
///交易编码  
TUstpFtdcClientIDType ClientID;  
};
```

策略组合/策略拆分必须填充的字段包括：

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) ExchangeID, 交易所代码, 只支持 “CFFEX”
- 3) InvestorID, 投资者编号, 形如 “10000029”;
- 4) UserID, 用户代码, 形如 “test1”;
- 5) HedgeFlag, 合约代码, 形如 “IF1109”;
- 6) UserActionLocalID, 用户本地报价单号, 形如 “000000101”;
- 7) CombInstrumentID, 组合合约代码, 形如 “2213”;

- 8) CombVolume, 组合数量, 形如 “1”;
- 9) CombDirection, 组合方向。“USTP_FTDC_CA_Combine” 为组合, “USTP_FTDC_CA_Uncombine” 为拆分组合

6.3.19 ReqQryOrder 方法

报单查询请求。

函数原形:

```
int ReqQryUstpOrder (
    CUstpFtdcQryOrderField *pQryOrder,
    int nRequestID);
```

参数:

pQryOrder: 指向报单查询结构的地址。

报单查询结构:

```
struct CUstpFtdcQryOrderField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType  BrokerID;
    ///用户代码
    TUstpFtdcUserIDType  UserID;
    ///交易所代码
    TUstpFtdcExchangeIDType  ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
    ///报单编号
    TUstpFtdcOrderSysIDType  OrderSysID;
    ///合约代码
    TUstpFtdcInstrumentIDType  InstrumentID;
    ///报单状态
    TUstpFtdcOrderStatusType  OrderStatus;
    ///委托类型
    TUstpFtdcOrderTypeType  OrderType;
    ///客户号
    TUstpFtdcClientIDType  ClientID;
};
```

按投资者查询至少应该填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) InvestorID, 投资者编号, 形如 “10000029”;
- 3) OrderStatus, 填写 ‘ ‘ (空格), 查询所有满足条件订单

按用户查询至少应该填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) UserID, 用户代码 (登陆用户, 不能查询其他用户), 形如 “test1”;
- 3) OrderStatus, 订单状态, 填写 ‘ ’ (空格), 查询所有满足条件订单

按订单状态查询至少应该填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) UserID, 用户代码 (登陆用户, 不能查询其他用户), 形如 “test1”;
- 3) OrderStatus, 订单状态, 填写 ” 或者 null, 查询错单, 填写 ‘ ’ (空格), 查询所有订单 (包括错单)

按委托类型查询, 可以区分普通单 (基础单和组合单)、行权单至少应该填充的字段包括:

- 1) BrokerID, 会员号, 形如 “2008”;
- 2) UserID, 用户代码 (登陆用户, 不能查询其他用户), 形如 “test1”;
- 3) OrderStatus, 订单状态, 填写 ‘ ’ (空格), 查询所有订单 (包括错单)
- 4) OrderType, 填写相应的类型, 填写 ‘USTP_FTDC_OT_Common’ 可查所有普通单 (包括组合单), 填写 ‘USTP_FTDC_OT_OptExec’ 可查所有行权单, ‘USTP_FTDC_OT_OptAbandon’ 可查所有放弃行权单。

6.3.20 ReqQryOrderFailed 方法

错单查询请求。

函数原形:

```
Int ReqQryOrderFailed(CUstpFtdcQryOrderFailedField *pQryOrderFailed,  
    int nRequestID);
```

参数:

pQryOrderFailed: 指向错单查询结构的地址。

错单查询结构:

```
struct CUstpFtdcQryOrderFailedField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///本地报单编号
```



```

TUstpFtdcUserOrderLocalIDType   UserOrderLocalID;
///合约代码
TUstpFtdcInstrumentIDType       InstrumentID;
///委托类型
TUstpFtdcOrderTypeType          OrderType;
///客户号
TUstpFtdcClientIDType           ClientID;
};

```

6.3.21 ReqQryTrade 方法

成交单查询请求。

函数原形：

```

int ReqQryUstpTrade (
    CUstpFtdcQryTradeField *pQryTrade,
    int nRequestID);

```

参数：

pQryTrade: 指向成交查询结构的地址。

成交查询结构：

```

struct CUstpFtdcQryTradeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///用户代码
    TUstpFtdcUserIDType     UserID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///成交编号
    TUstpFtdcTradeIDType    TradeID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///客户号
    TUstpFtdcClientIDType   ClientID;
};

```

6.3.22 ReqQryUserInvestor 方法

可用投资者账户查询。

函数原形：

```

int ReqQryUserInvestor(

```

```
CUstpFtdcQryUserInvestorField *pQryUserInvestor,
int nRequestID)
```

参数:

pQryUserInvestor 指向可用投资者账户查询结构的地址。

可用投资者查询结构

```
struct CUstpFtdcQryUserInvestorField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
};
```

6.3.23 ReqQryTradingCode 方法

交易编码查询。

函数原形:

```
int ReqQryTradingCode (
    CUstpFtdcQryTradingCodeField *pQryTradingCode,
    int nRequestID)
```

参数:

pQryTradingCode 指向交易编码查询结构的地址。

交易编码结构:

```
struct CUstpFtdcQryTradingCodeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///客户号
    TUstpFtdcClientIDType ClientID;
};
```

6.3.24 ReqQryInvestorAccount 方法

投资者资金账户查询。

函数原形:

```
int ReqQryInvestorAccount (
```

```
CUstpFtdcQryInvestorAccountField *pQryInvestorAccount,  
int nRequestID)
```

参数:

pQryInvestorAccount 指向投资者账户查询结构的地址。

投资者账户查询结构:

```
struct CUstpFtdcQryInvestorAccountField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
};
```

6.3.25 ReqQryInstrument 方法

合约信息查询。

函数原形:

```
int ReqQryInstrument (  
    CUstpFtdcQryInstrumentField *pQryInstrument,  
    int nRequestID)
```

参数:

pQryInstrument 指向合约信息查询结构的地址。

合约查询结构

```
struct CUstpFtdcQryInstrumentField  
{  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///产品代码  
    TUstpFtdcProductIDType ProductID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///产品类型  
    TUstpFtdcProductClassType ProductClass;  
};
```

6.3.26 ReqQryExchange 方法

交易所查询。

函数原形:

```
int ReqQryExchange(  

```

```
CUstpFtdcQryExchangeField *pQryExchange,
int nRequestID)
```

参数:

pQryExchange 指向交易所查询结构的地址。

交易编码结构:

```
struct CUstpFtdcQryExchangeField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
};
```

6.3.27 ReqQryInvestorPosition 方法

投资者持仓查询。

函数原形:

```
int ReqQryInvestorPosition (
    CUstpFtdcQryInvestorPositionField *pQryUserInvestorPosition,
    int nRequestID)
```

参数:

pQryUserInvestorPosition 指向投资者持仓查询结构的地址。

投资者持仓查询结构

```
struct CUstpFtdcQryInvestorPositionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///交易编码

    TUstpFtdcClientIDType ClientID;
};
```

6.3.28 ReqQryInvestorFee 方法

投资者手续费率查询。

函数原形:

```
int ReqQryInvestorFee (
```

```
CUstpFtdcQryInvestorFeeField *pQryInvestorFee,  
int nRequestID)
```

参数:

pQryInvestorFee 指向投资者手续费率查询结构的地址。

投资者手续费率查询结构

```
struct CUstpFtdcQryInvestorFeeField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///客户号  
    TUstpFtdcClientIDType ClientID;  
    ///投机套保标志  
    TUstpFtdcHedgeFlagType HedgeFlag;  
    ///多空方向  
    TUstpFtdcDirectionType Direction;  
};
```

6.3.29 ReqQryInvestorMargin 方法

投资者保证金率查询

函数原形:

```
int ReqQryInvestorMargin (  
    CUstpFtdcQryInvestorMarginField *pQryInvestorMargin,  
    int nRequestID)
```

参数:

pQryInvestorMargin 指向投资者保证金率查询结构的地址。

投资者保证金率查询结构

```
struct CUstpFtdcQryInvestorMarginField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///投资者编号
```

```

    TUstpFtdcInvestorIDType InvestorID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///客户号
    TUstpFtdcClientIDType ClientID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType HedgeFlag;
};

```

6.3.30 ReqQryInvestorCombPosition 方法

投资者组合持仓查询。

函数原形:

```

int ReqQryInvestorCombPosition(
    CUstpFtdcQryInvestorCombPositionField *pQryInvestorCombPosition,
    int nRequestID)

```

参数:

pQryUserInvestorPosition 指向投资者组合持仓查询结构的地址。

投资者组合持仓查询结构

```

struct CUstpFtdcQryInvestorCombPositionField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType HedgeFlag;
    ///组合合约代码
    TUstpFtdcInstrumentIDType CombInstrumentID;
    ///交易编码
    TUstpFtdcClientIDType ClientID;
};

```

6.3.31 ReqQryInstrumentGroup 方法

查询合约组信息请求，属于同一个合约组的合约可以在合约组内采用大边的方式收取保证金。

函数原形:

```
int ReqQryInstrumentGroup(  
    CUstpFtdcQryInstrumentGroupField *pQryInstrumentGroup,  
    int nRequestID)
```

参数:

pQryInstrumentGroup: 指向合约组信息结构的地址。

合约组信息结构:

```
struct CUstpFtdcQryUstpInstrumentGroupField  
{  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
};
```

6.3.32 ReqQryClientMarginCombType 方法

查询客户的保证金收取类型信息，指定当前客户的保证金收取类型，包含双边，大边（也称单向大边或单边），手动策略组合，手动策略大边。

函数原形:

```
int ReqQryClientMarginCombType (  
    CUstpFtdcQryClientMarginCombTypeField *pQryClientMarginCombType,  
    int nRequestID)
```

参数:

pQryClientMarginCombType: 指向客户的保证金收取类型信息结构的地址。

客户的保证金收取类型信息结构:

```
struct CUstpFtdcQryClientMarginCombTypeField  
{  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///投资者编号
```

```
TUstpFtdcInvestorIDType InvestorID;  
///投机套保标志  
TUstpFtdcHedgeFlagType HedgeFlag;  
///合约组代码  
TUstpFtdcInstrumentGroupIDType InstrumentGroupID;  
};
```

6.3.33 ReqExecOrderInsert 方法

行权录入请求

此接口用于发起行权请求。使用方法与报单接口类似。通过 OrderType 选择是行权还是放弃行权，通过 DeliveryFlag 选择是否需要对冲。

函数原形:

```
int ReqExecOrderInsert (  
    CUstpFtdcInputExecOrderField *pInputExecOrder,,  
    int nRequestID)
```

参数:

pInputExecOrder: 指向行权录入结构的地址。

行权录入信息结构:

```
struct CUstpFtdcInputExecOrderField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///交易所代码  
    TUstpFtdcExchangeIDType ExchangeID;  
    ///系统报单编号  
    TUstpFtdcOrderSysIDType OrderSysID;  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///用户代码  
    TUstpFtdcUserIDType UserID;  
    ///合约代码  
    TUstpFtdcInstrumentIDType InstrumentID;  
    ///用户本地报单号  
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
```



```

    ///委托类型
    TustpFtdcOrderTypeType OrderType;

    ///期权对冲标识
    TustpFtdcDeliveryFlagType DeliveryFlag;

    ///投机套保标志
    TustpFtdcHedgeFlagType HedgeFlag;

    ///数量
    TustpFtdcVolumeType Volume;

    ///用户自定义域
    TustpFtdcCustomType UserCustom;

    ///业务发生日期
    TustpFtdcDateType ActionDay;

    ///本地业务标识
    TustpFtdcBusinessLocalIDType BusinessLocalID;

    ///业务单元
    TustpFtdcBusinessUnitType BusinessUnit;

    ///今昨仓标志
    TustpFtdcOffsetFlagType OffsetFlag;

    ///会话编号
    TustpFtdcSessionIDType SessionID;

    ///前置编号
    TustpFtdcFrontIDType FrontID;

    ///交易编码
    TustpFtdcClientIDType ClientID;
};

```

字段说明:

字段	说明
ExchangeID	支持郑商所、大商所、上期所、能源中心
OrderType	///普通委托#define USTP_FTDC_OT_Common '0'（本接口不使用） ///期权执行#define USTP_FTDC_OT_OptExec '1'（行权。使用该值时，DeliveryFlag只能为USTP_FTDC_DF_ExecOpen或USTP_FTDC_DF_Delivery） ///放弃执行#define USTP_FTDC_OT_OptAbandon '2'（放弃行权。

	<p>使用该值时，DeliveryFlag 只能为 USTP_FTDC_DF_ExecOpen)</p> <p>///OTC 衍生#define USTP_FTDC_OT_OTCDerived '3' (本接口不使用)</p> <p>///期转现衍生#define USTP_FTDC_OT_EFPDerived '4' (本接口不使用)</p>
DeliveryFlag	<p>///不对冲#define USTP_FTDC_DF_ExecOpen '0' (OrderType 为 USTP_FTDC_OT_OptExec 和 USTP_FTDC_OT_OptAbandon 时使用)</p> <p>///对冲(期货)#define USTP_FTDC_DF_Delivery '1' (OrderType 为 USTP_FTDC_OT_OptExec 时使用)</p> <p>///对冲(期权)#define USTP_FTDC_DF_OptDelivery '2' (本接口不使用)</p> <p>///保留持仓#define USTP_FTDC_DF_OptHold '3' (本接口不使用)</p> <p>///履约后对冲#define USTP_FTDC_DF_SellDelivery '4' (本接口不使用)</p> <p>///履约后不对冲#define USTP_FTDC_DF_SellNoDelivery '5' (本接口不使用)</p>
OffsetFlag	<p>上期所和能源所区分昨持仓和今持仓,需要使用对应的平今与平昨(平昨与平仓同效);郑商、大商使用平仓</p> <p>///开仓#define USTP_FTDC_OF_Open '0'</p> <p>///平仓#define USTP_FTDC_OF_Close '1'</p> <p>///强平#define USTP_FTDC_OF_ForceClose '2'</p> <p>///平今#define USTP_FTDC_OF_CloseToday '3'</p> <p>///平昨#define USTP_FTDC_OF_CloseYesterday '4'</p> <p>///组合开平#define USTP_FTDC_OF_CombOpenClose '5'</p> <p>///组合平开#define USTP_FTDC_OF_CombCloseOpen '6'</p> <p>///ALL#define USTP_FTDC_OF_ALL '9'</p>
Volume	大商所 OrderType 为放弃执行时, 该值为 0

6.3.34 ReqExecOrderAction 方法

行权操作请求

此接口用于对已经发起的行权请求进行撤销。

函数原形:

```
int ReqExecOrderAction (
    CUSTpFtdcInputExecOrderActionField *pInputExecOrderAction,
    int nRequestID)
```

参数:

pInputExecOrderAction: 指向行权撤单结构的地址。

行权撤单信息结构:

```
struct CUstpFtdcInputExecOrderActionField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///本次撤单操作的本地编号
    TUstpFtdcUserOrderLocalIDType UserOrderActionLocalID;
    ///被撤订单的本地报单编号
    TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
    ///报单操作标志
    TUstpFtdcActionFlagType ActionFlag;
    ///数量变化
    TUstpFtdcVolumeType VolumeChange;
    ///本地业务标识
    TUstpFtdcBusinessLocalIDType BusinessLocalID;
    ///委托类型
    TUstpFtdcOrderTypeType OrderType;
    ///会话编号
    TUstpFtdcSessionIDType SessionID;
    ///前置编号
    TUstpFtdcFrontIDType FrontID;
    ///交易编码
    TUstpFtdcClientIDType ClientID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
};
```

6.3.35 ReqTradingAccountPasswordUpdate 方法

资金账户口令更新请求。

函数原形:

```
int ReqTradingAccountPasswordUpdate(  
    CUstpFtdcTradingAccountPasswordUpdateReqField *pTradingAccountPasswordUpdateReq,  
    int nRequestID)
```

参数:

pTradingAccountPasswordUpdateReq: 指向资金账户口令结构的地址。

///资金账户口令更新请求

```
struct CUstpFtdcTradingAccountPasswordUpdateReqField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID; //不验证该字段  
    ///投资者编号  
    TUstpFtdcInvestorIDType InvestorID;  
    ///旧密码  
    TUstpFtdcPasswordType OldPassword;  
    ///新密码  
    TUstpFtdcPasswordType NewPassword;  
    ///登录经纪公司编号  
    TUstpFtdcBrokerIDType LogBrokerID;  
    ///登录用户代码  
    TUstpFtdcUserIDType LogUserID;  
};
```

6.3.36 ReqQueryTransferSerial 方法

请求查询转帐流水接口。

函数原形:

```
int ReqQueryTransferSerial (  
    CUstpFtdcTransferSerialFieldReqField *pTransferSerialFieldReq,  
    int nRequestID)
```

参数:

///查询转帐流水请求

```
struct CUstpFtdcTransferSerialFieldReqField  
{  
    ///经纪公司编号  
    TUstpFtdcBrokerIDType BrokerID;  
    ///用户编号  
    TUstpFtdcUserIDType UserID;
```

```
///资金帐号
TUstpFtdcAccountIDType  AccountID; //必填
///银行代码
TUstpFtdcBankIDType  BankID;
///开始日期
TUstpFtdcDateType StartDay;
///结束日期
TUstpFtdcDateType EndDay;
};
```

备注:

开始日期和结束日期指交易日

目前银行代码未使用，默认查询所有银行流水

示例:

```
printf("查询期货流水\n");
CUstpFtdcTransferSerialFieldReqField qef;
memset(&qef, 0, sizeof(CUstpFtdcTransferSerialFieldReqField));
strcpy(qef.BrokerID, "0001");
strcpy(qef.AccountID, "2008002");
strcpy(qef.UserID, "2008002");
strcpy(qef.StartDay, "20190103");
strcpy(qef.EndDay, "20190103");
```

6.3.37 ReqQueryAccountRegister 方法

请求查询银期签约关系。

函数原形:

```
int ReqQueryAccountregister(
    CUstpFtdcAccountregisterReqField *pAccountregisterReq,
    int nRequestID)
```

参数:

///查询银期签约关系

```
struct CUstpFtdcAccountregisterReqField
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType  BrokerID;
```

```
///用户编号
TUstpFtdcUserIDType  UserID;
///投资者编号
TUstpFtdcInvestorIDType  InvestorID; //必填
///银行代码
TUstpFtdcBankIDType  BankID;
};
```

备注:

银行号非必须，默认查询所有银行

示例:

```
printf("查询关联帐号\n");
CUstpFtdcAccountregisterReqField qef;
memset(&qef, 0, sizeof(CUstpFtdcAccountregisterReqField));
strcpy(qef.BrokerID, "0001");
strcpy(qef.InvestorID, "2008002");
strcpy(qef.UserID, "2008002");
```

6.3.38 ReqFromBankToFutureByFuture 方法

期货发起银行资金转期货请求。

函数原形:

```
int ReqFromBankToFutureByFuture (
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,
    int nRequestID)
```

参数:

```
///转账请求
struct CUstpFtdcTransferFieldReqField
{
    ///业务功能码
    TUstpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
```

```
TUstpFtdcTransTradeSourceType TradeSource;
///银行类型
TUstpFtdcTransBankTypeType BankType;
///银行代码
TUstpFtdcTransBankIDType BankID;
///银行分中心代码
TUstpFtdcTransBankBrchIDType BankBrchID;
///经纪公司类型
TUstpFtdcTransBrokerTypeType BrokerType;
///经纪公司编号
TUstpFtdcTransBrokerIDType BrokerID;
///期商分支机构代码
TUstpFtdcTransFutureBranchIDType BrokerBranchID;
///交易日期
TUstpFtdcTransDateType TradeDate;
///成交时间
TUstpFtdcTransTimeType TradeTime;
///发起方流水号
TUstpFtdcTransSendSourceSerialType SendSourceSerial;
///发起方关联流水号
TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;
///重发标志
TUstpFtdcTransResendFlagType ResendFlag;
///出入金方向
TUstpFtdcTransTransferDirFlagType TransferDirFlag;
///币种
TUstpFtdcTransCurrencyType CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType CustType;
```

```
///机构客户号
TustpFtdcOrganCustNoType OrganCustNo;
///银行帐号
TustpFtdcTransBankAccountType BankAccount;
///银行帐号类型
TustpFtdcTransAccountTypeType BankAccountType;
///开户分行号
TustpFtdcTransBankBranchIDType BankBranchID;
///银行账户密码类型 1
TustpFtdcTransPwdTypeType BankAccountPwdType1;
///银行账户加密方式 1
TustpFtdcTransPwdEncType BankAccountPwdEnc1;
///银行账户密码 1
TustpFtdcTransPwdType BankAccountPwd1;
///银行账户密码类型 2
TustpFtdcTransPwdTypeType BankAccountPwdType2;
///银行账户加密方式 2
TustpFtdcTransPwdEncType BankAccountPwdEnc2;
///银行账户密码 2
TustpFtdcTransPwdType BankAccountPwd2;
///资金帐号
TustpFtdcTransAccountIDType AccountID;
///资金帐号类型
TustpFtdcTransAccountTypeType AccountType;
///资金账户密码类型 1
TustpFtdcTransPwdTypeType AccountPwdType1;
///资金账户加密方式 1
TustpFtdcTransPwdEncType AccountPwdEnc1;
///资金账户密码 1
TustpFtdcTransPwdType AccountPwd1;
///资金账户密码类型 2
TustpFtdcTransPwdTypeType AccountPwdType2;
///资金账户加密方式 2
TustpFtdcTransPwdEncType AccountPwdEnc2;
///资金账户密码 2
```



```
TUstpFtdcTransPwdType    AccountPwd2;
///摘要
TUstpFtdcTransDigestType Digest;
///转帐金额
TUstpFtdcMoneyType    TradeAmount;
///当前金额
TUstpFtdcMoneyType    CurrentBalance;
///可用金额
TUstpFtdcMoneyType    UsableBalance;
///可取金额
TUstpFtdcMoneyType    WithdrawableBalance;
///交易日
TUstpFtdcTransDateType    TradingDay;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///待冲正流水号
TUstpFtdcTransSendSourceSerialType    RevReference;
};
```

备注:

填写项见示例代码

示例:

```
printf("期货发起入金请求\n");
CUstpFtdcTransferFieldReqField qef;
memset(&qef, 0, sizeof(CUstpFtdcTransferFieldReqField));
strcpy(qef.BrokerID, "0013");
strcpy(qef.BrokerBranchID, "001");
strcpy(qef.CurrencyID, "CNY");
strcpy(qef.BankAccount, "NRA03427900048400014");
strcpy(qef.AccountID, "314159272");
strcpy(qef.UserID, "314159272");
strcpy(qef.InvestorID, "314159272");
strcpy(qef.BankID, "02");
strcpy(qef.AccountPwd1, "123456");
strcpy(qef.BankAccountPwd1, "147258");
strcpy(qef.BankAccountPwdEnc1, "01");
qef.TradeAmount = 5000;
strcpy(qef.OrganCustNo, "001");
```

6.3.39 ReqFromFutureToBankByFuture 方法

期货发起期货资金转银行请求。

函数原形:

```
int ReqFromFutureToBankByFuture (
    CUstpFtdcTransferFieldReqField *pTransferFieldReq,
    int nRequestID)
```

参数:

参考 *ReqFromBankToFutureByFuture* 方法。

6.3.40 ReqQueryBankAccountMoneyByFuture 方法

期货发起查询银行余额请求。

函数原形:

```
int ReqQueryBankAccountMoneyByFuture (
    CUstpFtdcAccountFieldReqField *pAccountFieldReq,
    int nRequestID)
```

参数:

///查询账户信息请求应答

```
struct CUstpFtdcAccountFieldRspField
{
    ///业务功能码
    TustpFtdcTransTradeCodeType TradeCode;
    ///业务发起方
    TustpFtdcTransTradeSourceType TradeSource;
    ///银行类型
    TustpFtdcTransBankTypeType BankType;
    ///银行代码
    TustpFtdcTransBankIDType BankID;
    ///银行分中心代码
    TustpFtdcTransBankBrchIDType BankBrchID;
    ///经纪公司类型
    TustpFtdcTransBrokerTypeType BrokerType;
    ///经纪公司编号
    TustpFtdcTransBrokerIDType BrokerID;
    ///用户编号
    TustpFtdcUserIDType UserID;
    ///投资者编号
    TustpFtdcInvestorIDType InvestorID; //必填
    ///期商分支机构代码
    TustpFtdcTransFutureBranchIDType BrokerBranchID;
    ///交易日期
    TustpFtdcTransDateType TradeDate;
    ///成交时间
    TustpFtdcTransTimeType TradeTime;
    ///发起方流水号
    TustpFtdcTransSendSourceSerialType SendSourceSerial;
    ///发起方关联流水号
    TustpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;
    ///币种
    TustpFtdcTransCurrencyType CurrencyID;
    ///客户姓名
    TustpFtdcTransIndividualNameType CustomerName;
```

```
///证件类型
TustpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TustpFtdcTransIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TustpFtdcTransCustTypeType    CustType;
///机构客户号
TustpFtdcOrganCustNoType OrganCustNo;
///银行帐号
TustpFtdcTransBankAccountType    BankAccount;
///银行帐号类型
TustpFtdcTransAccountTypeType    BankAccountType;
///开户分行号
TustpFtdcTransBankBranchIDType    BankBranchID;
///银行账户密码类型 1
TustpFtdcTransPwdTypeType    BankAccountPwdType1;
///银行账户加密方式 1
TustpFtdcTransPwdEncType BankAccountPwdEnc1;
///银行账户密码 1
TustpFtdcTransPwdType    BankAccountPwd1;
///银行账户密码类型 2
TustpFtdcTransPwdTypeType    BankAccountPwdType2;
///银行账户加密方式 2
TustpFtdcTransPwdEncType BankAccountPwdEnc2;
///银行账户密码 2
TustpFtdcTransPwdType    BankAccountPwd2;
///资金帐号
TustpFtdcTransAccountIDType AccountID;
///资金帐号类型
TustpFtdcTransAccountTypeType    AccountType;
///资金账户密码类型 1
TustpFtdcTransPwdTypeType    AccountPwdType1;
///资金账户加密方式 1
TustpFtdcTransPwdEncType AccountPwdEnc1;
///资金账户密码 1
```

```
TUstpFtdcTransPwdType    AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType    AccountPwd2;
///摘要
TUstpFtdcTransDigestType Digest;
///交易日
TUstpFtdcTransDateType    TradingDay;
///登录经纪公司编号
TUstpFtdcBrokerIDType    LogBrokerID;
///登录用户代码
TUstpFtdcUserIDType    LogUserID;
///当前金额
TUstpFtdcMoneyType    CurrentBalance;
///可用金额
TUstpFtdcMoneyType    UsableBalance;
///可取金额
TUstpFtdcMoneyType    WithdrawableBalance;
///转帐金额
TUstpFtdcMoneyType    FrozenBalance;
};
```

备注:

填写项见示例代码

示例:

```
printf("查询银行余额\n");
CUstpFtdcAccountFieldReqField qef;
memset(&qef, 0, sizeof(CUstpFtdcAccountFieldReqField));
strcpy(qef.BrokerID, "0001");
strcpy(qef.BrokerBranchID, "001");
strcpy(qef.CurrencyID, "CNY");
strcpy(qef.BankAccount, "62284100500000000216");
strcpy(qef.AccountID, "2008002");
strcpy(qef.UserID, "2008002");
strcpy(qef.BankID, "02");
strcpy(qef.AccountPwd1, "123456");
strcpy(qef.BankAccountPwd1, "147258");
strcpy(qef.BankAccountPwdEnc1, "01");
strcpy(qef.OrganCustNo, "001");
```

6.3.41 ReqSignUpAccountByFuture 方法

期货发起银行账户签约请求。

函数原形:

```
int ReqSignUpAccountByFuture(  
    CUstpFtdcSignUpOrCancleAccountReqFieldField *pSignUpOrCancleAccountReqField,  
    int nRequestID)
```

参数: pSignUpOrCancleAccountReqField 指向银期开销户请求结构的地址。

银期开销户请求结构:

```
struct CUstpFtdcSignUpOrCancleAccountReqFieldField  
{  
    ///业务功能码  
    TUstpFtdcTransTradeCodeType TradeCode;  
    ///业务发起方  
    TUstpFtdcTransTradeSourceType TradeSource;  
    ///银行类型  
    TUstpFtdcTransBankTypeType BankType;  
    ///银行代码  
    TUstpFtdcTransBankIDType BankID;  
    ///银行分中心代码  
    TUstpFtdcTransBankBrchIDType BankBrchID;  
    ///经纪公司类型  
    TUstpFtdcTransBrokerTypeType BrokerType;  
    ///经纪公司编号  
    TUstpFtdcTransBrokerIDType BrokerID;  
    ///期商分支机构代码  
    TUstpFtdcTransFutureBranchIDType BrokerBranchID;  
    ///交易日期  
    TUstpFtdcTransDateType TradeDate;  
    ///成交时间  
    TUstpFtdcTransTimeType TradeTime;  
    ///发起方流水号  
    TUstpFtdcTransSendSourceSerialType SendSourceSerial;  
    ///发起方关联流水号  
    TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;  
    ///币种
```

```
TUstpFtdcTransCurrencyType  CurrencyID;
///客户姓名
TUstpFtdcTransIndividualNameType  CustomerName;
///证件类型
TUstpFtdcTransIdCardTypeType  IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType  IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType  CustType;
///机构客户号
TUstpFtdcOrganCustNoType  OrganCustNo;
///银行帐号
TUstpFtdcTransBankAccountType  BankAccount;
///银行帐号类型
TUstpFtdcTransAccountTypeType  BankAccountType;
///开户分行号
TUstpFtdcTransBankBranchIDType  BankBranchID;
///银行账户密码类型 1
TUstpFtdcTransPwdTypeType  BankAccountPwdType1;
///银行账户加密方式 1
TUstpFtdcTransPwdEncType  BankAccountPwdEnc1;
///银行账户密码 1
TUstpFtdcTransPwdType  BankAccountPwd1;
///银行账户密码类型 2
TUstpFtdcTransPwdTypeType  BankAccountPwdType2;
///银行账户加密方式 2
TUstpFtdcTransPwdEncType  BankAccountPwdEnc2;
///银行账户密码 2
TUstpFtdcTransPwdType  BankAccountPwd2;
///资金帐号
TUstpFtdcTransAccountIDType  AccountID;
///资金帐号类型
TUstpFtdcTransAccountTypeType  AccountType;
///资金账户密码类型 1
TUstpFtdcTransPwdTypeType  AccountPwdType1;
```



```
///资金账户加密方式 1
TUstpFtdcTransPwdEncType    AccountPwdEnc1;
///资金账户密码 1
TUstpFtdcTransPwdType    AccountPwd1;
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType    AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType    AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType    AccountPwd2;
///摘要
TUstpFtdcTransDigestType    Digest;
///交易日
TUstpFtdcTransDateType    TradingDay;
///用户编号
TUstpFtdcUserIDType    UserID;
///投资者编号
TUstpFtdcInvestorIDType    InvestorID;
///性别
TUstpFtdcTransGenderType    Gender;
///国家代码
TUstpFtdcTransCountryCodeType    CountryCode;
///地址
TUstpFtdcTransAddressType    Address;
///邮政编码类型
TUstpFtdcTransZipCodeType    ZipCode;
///电话号码
TUstpFtdcTransTelephoneType    Telephone;
///手机
TUstpFtdcTransMobilePhoneType    MobilePhone;
///传真
TUstpFtdcTransFaxType    Fax;
///电子邮件
TUstpFtdcTransEMailType    EMail;
///传呼
```

```
TUstpFtdcTransBpType    Bp;
///银行账户户主名称
TUstpFtdcTransNameType  BankAccountName;
///银行账户状态
TUstpFtdcTransStatusType BankAccountStatus;
///银行账户开户日期
TUstpFtdcTransRegisterDateType BankAccountRegisterDate;
///银行账户生效日期
TUstpFtdcTransValidDateType BankAccountValidDate;
///资金账户户主名称
TUstpFtdcTransNameType  AccountName;
///资金账户状态
TUstpFtdcTransStatusType AccountStatus;
///资金账户开户日期
TUstpFtdcTransDateType  AccountRegisterDate;
///资金账户生效日期
TUstpFtdcTransDateType  AccountValidDate;
///汇钞标志
TUstpFtdcTransCashExCodeType CashExCode;
///预约流水号
TUstpFtdcTransBookSeqType  BookSeq;
///当前余额
TUstpFtdcMoneyType  CurrentBalance;
};
```

备注:

填写项见示例代码

示例:

```
printf("签约\n");
CUstpFtdcSignUpOrCancleAccountReqFieldField qef;
memset(&qef, 0, sizeof(CUstpFtdcSignUpOrCancleAccountReqFieldField));
strcpy(qef.BrokerID, "0001");
strcpy(qef.BrokerBranchID, "001");
strcpy(qef.CurrencyID, "CNY");
strcpy(qef.BankAccount, "6228410050000000216");
strcpy(qef.AccountID, "2008002");
strcpy(qef.UserID, "2008002");
strcpy(qef.InvestorID, "2008002");
strcpy(qef.BankID, "02");
strcpy(qef.BankAccountPwd1, "147258");
strcpy(qef.BankAccountPwdEnc1, "01");
strcpy(qef.AccountPwd1, "123456");
strcpy(qef.OrganCustNo, "001");
```

6.3.42 ReqCancleAccountByFuture 方法

期货发起银行账户解约请求

函数原形:

```
int ReqCancleAccountByFuture(
    CUstpFtdcSignUpOrCancleAccountReqFieldField *pSignUpOrCancleAccountReqField,
    int nRequestID);
```

参数: pSignUpOrCancleAccountReqField 指向银期开销户请求结构的地址。

银期开销户请求结构:

参考 *ReqSignUpAccountByFuttrue* 方法

6.3.43 ReqQuerySignUpOrCancleAccountStatus 方法

请求查询银行签约状态

函数原形:

```
int ReqQuerySignUpOrCancleAccountStatus(
    CUstpFtdcQuerySignUpOrCancleAccountStatusReqFieldField
    *pQuerySignUpOrCancleAccountStatusReqField,
    int nRequestID)
```

参数: pQuerySignUpOrCancleAccountStatusReqField 指向银行签约状态结构的地址。

银行签约状态结构

```

struct CUstpFtdcQuerySignUpOrCancleAccountStatusReqFieldField
{
    ///用户编号
    TUSTPFTDCUserIDType UserID;
    ///投资者编号
    TUSTPFTDCInvestorIDType InvestorID;
    ///经纪公司编号
    TUSTPFTDCTransBrokerIDType BrokerID;
    ///资金帐号
    TUSTPFTDCTransAccountIDType AccountID; //必填
    ///银行代码
    TUSTPFTDCTransBankIDType BankID;
    ///银行帐号
    TUSTPFTDCTransBankAccountType BankAccount;
    ///币种
    TUSTPFTDCTransCurrencyType CurrencyID;
};

```

备注:

填写项见示例代码

示例:

```

printf("查询银行签约状态\n");
CUstpFtdcQuerySignUpOrCancleAccountStatusReqFieldField qef;
memset(&qef, 0, sizeof(CUstpFtdcQuerySignUpOrCancleAccountStatusReqFieldField));
strcpy(qef.BrokerID, "0013");
strcpy(qef.BankAccount, "NRA03427900048400014");
strcpy(qef.AccountID, "314159272");
strcpy(qef.UserID, "314159272");
strcpy(qef.InvestorID, "314159272");
strcpy(qef.BankID, "02");
strcpy(qef.CurrencyID, "CNY");

```

6.3.44 ReqChangeAccountByFuture 方法

期货发起银行账户变更请求。

函数原形:

```
int ReqChangeAccountByFuture(  
    CUstpFtdcChangeAccountReqFieldField *pChangeAccountReqField,  
    int nRequestID)
```

参数: pChangeAccountReqField 指向银期变更银行账号结构的地址。

银期变更银行账号信息:

```
struct CUstpFtdcChangeAccountReqFieldField  
{  
    ///业务功能码  
    TUstpFtdcTransTradeCodeType TradeCode;  
    ///业务发起方  
    TUstpFtdcTransTradeSourceType TradeSource;  
    ///银行类型  
    TUstpFtdcTransBankTypeType BankType;  
    ///银行代码  
    TUstpFtdcTransBankIDType BankID;  
    ///银行分中心代码  
    TUstpFtdcTransBankBrchIDType BankBrchID;  
    ///经纪公司类型  
    TUstpFtdcTransBrokerTypeType BrokerType;  
    ///经纪公司编号  
    TUstpFtdcTransBrokerIDType BrokerID;  
    ///期商分支机构代码  
    TUstpFtdcTransFutureBranchIDType BrokerBranchID;  
    ///交易日期  
    TUstpFtdcTransDateType TradeDate;  
    ///成交时间  
    TUstpFtdcTransTimeType TradeTime;  
    ///发起方流水号  
    TUstpFtdcTransSendSourceSerialType SendSourceSerial;  
    ///发起方关联流水号  
    TUstpFtdcTransSendSourceReletedSerialType SendSourceReletedSerial;  
    ///币种  
    TUstpFtdcTransCurrencyType CurrencyID;  
    ///客户姓名  
    TUstpFtdcTransIndividualNameType CustomerName;
```

```
///证件类型
TUstpFtdcTransIdCardTypeType    IdCardType;
///证件号码
TUstpFtdcTransIdentifiedCardNoType  IdentifiedCardNo;
///客户类型
TUstpFtdcTransCustTypeType    CustType;
///机构客户号
TUstpFtdcOrganCustNoType    OrganCustNo;
///老银行帐号
TUstpFtdcTransBankAccountType    OldBankAccount;
///老银行帐号类型
TUstpFtdcTransAccountTypeType    OldBankAccountType;
///老开户分行号
TUstpFtdcTransBankBranchIDType    OldBankBranchID;
///老银行账户密码类型 1
TUstpFtdcTransPwdTypeType    OldBankAccountPwdType1;
///老银行账户加密方式 1
TUstpFtdcTransPwdEncType    OldBankAccountPwdEnc1;
///老银行账户密码 1
TUstpFtdcTransPwdType    OldBankAccountPwd1;
///老银行账户密码类型 2
TUstpFtdcTransPwdTypeType    OldBankAccountPwdType2;
///老银行账户加密方式 2
TUstpFtdcTransPwdEncType    OldBankAccountPwdEnc2;
///老银行账户密码 2
TUstpFtdcTransPwdType    OldBankAccountPwd2;
///老银行账户户主名称
TUstpFtdcTransNameType    OldBankAccountName;
///老银行账户状态
TUstpFtdcTransStatusType    OldBankAccountStatus;
///老银行账户开户日期
TUstpFtdcTransRegisterDateType    OldBankAccountRegisterDate;
///老银行账户生效日期
TUstpFtdcTransValidDateType    OldBankAccountValidDate;
///新银行帐号
```

```
TUstpFtdcTransBankAccountType    NewBankAccount;
///新银行帐号类型

TUstpFtdcTransAccountTypeType    NewBankAccountType;
///新开户分行号

TUstpFtdcTransBankBranchIDType    NewBankBranchID;
///新银行账户密码类型 1

TUstpFtdcTransPwdTypeType    NewBankAccountPwdType1;
///新银行账户加密方式 1

TUstpFtdcTransPwdEncType    NewBankAccountPwdEnc1;
///新银行账户密码 1

TUstpFtdcTransPwdType    NewBankAccountPwd1;
///新银行账户密码类型 2

TUstpFtdcTransPwdTypeType    NewBankAccountPwdType2;
///新银行账户加密方式 2

TUstpFtdcTransPwdEncType    NewBankAccountPwdEnc2;
///新银行账户密码 2

TUstpFtdcTransPwdType    NewBankAccountPwd2;
///新银行账户户主名称

TUstpFtdcTransNameType    NewBankAccountName;
///新银行账户状态

TUstpFtdcTransStatusType    NewBankAccountStatus;
///新银行账户开户日期

TUstpFtdcTransRegisterDateType    NewBankAccountRegisterDate;
///新银行账户生效日期

TUstpFtdcTransValidDateType    NewBankAccountValidDate;
///资金帐号

TUstpFtdcTransAccountIDType    AccountID;
///资金帐号类型

TUstpFtdcTransAccountTypeType    AccountType;
///资金账户密码类型 1

TUstpFtdcTransPwdTypeType    AccountPwdType1;
///资金账户加密方式 1

TUstpFtdcTransPwdEncType    AccountPwdEnc1;
///资金账户密码 1

TUstpFtdcTransPwdType    AccountPwd1;
```

```
///资金账户密码类型 2
TUstpFtdcTransPwdTypeType    AccountPwdType2;
///资金账户加密方式 2
TUstpFtdcTransPwdEncType      AccountPwdEnc2;
///资金账户密码 2
TUstpFtdcTransPwdType         AccountPwd2;
///交易日
TUstpFtdcTransDateType        TradingDay;
///用户编号
TUstpFtdcUserIDType           UserID;
///投资者编号
TUstpFtdcInvestorIDType       InvestorID;
///性别
TUstpFtdcTransGenderType      Gender;
///国家代码
TUstpFtdcTransCountryCodeType CountryCode;
///地址
TUstpFtdcTransAddressType      Address;
///邮政编码类型
TUstpFtdcTransZipCodeType      ZipCode;
///电话号码
TUstpFtdcTransTelephoneType    Telephone;
///手机
TUstpFtdcTransMobilePhoneType  MobilePhone;
///传真
TUstpFtdcTransFaxType          Fax;
///电子邮件
TUstpFtdcTransEMailType        EMail;
///传呼
TUstpFtdcTransBpType           Bp;
///资金账户户主名称
TUstpFtdcTransNameType         AccountName;
///资金账户状态
TUstpFtdcTransStatusType       AccountStatus;
///资金账户开户日期
```



```
TUstpFtdcTransDateType  AccountRegisterDate;
///资金账户生效日期
TUstpFtdcTransDateType  AccountValidDate;
///汇钞标志
TUstpFtdcTransCashExCodeType  CashExCode;
};
```

6.3.45 ReqOpenSimTradeAccount 方法

仿真交易帐户申请。

函数原形:

```
int ReqOpenSimTradeAccount(CUstpFtdcSimTradeAccountInfoField *pSimTradeAccountInfo,
    int nRequestID)
```

参数: pSimTradeAccountInfo 指向仿真交易帐户结构的地址。

///仿真交易帐户信息

```
struct CUstpFtdcSimTradeAccountInfoField
{
    ///客户姓名
    TUstpFtdcIndividualNameType CustomerName;
    ///证件号码
    TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///手机
    TUstpFtdcMobilePhoneType MobilePhone;
    ///电子邮件
    TUstpFtdcEMailType EMail;
    ///开户密码
    TUstpFtdcPasswordType PassWord;
    ///用户代码
    TUstpFtdcUserIDType UserID;
};
```

6.3.46 ReqCheckOpenSimTradeAccount 方法

仿真交易帐户查询。

函数原形:

```
int ReqCheckOpenSimTradeAccount(
    CUstpFtdcSimTradeAccountInfoField *pSimTradeAccountInfo,
```

```
int nRequestID)
```

参数： pSimTradeAccountInfo 指向仿真交易帐户结构的地址。

///仿真交易帐户信息

```
struct CUstpFtdcSimTradeAccountInfoField
{
    ///客户姓名
    TUstpFtdcIndividualNameType CustomerName;
    ///证件号码
    TUstpFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///手机
    TUstpFtdcMobilePhoneType MobilePhone;
    ///电子邮件
    TUstpFtdcEmailType Email;
    ///开户密码
    TUstpFtdcPasswordType PassWord;
    ///用户代码
    TUstpFtdcUserIDType UserID;
};
```

6.3.47 ReqCFMMCTradingAccountKey 方法

查询保证金监管系统经纪公司资金账户密钥接口。

函数原形：

```
int ReqQryCFMMCTradingAccountKey(
    CUstpFtdcCFMMCTradingAccountKeyReqField *pCFMMCTradingAccountKeyReq,
    int nRequestID)
```

参数：

pCFMMCTradingAccountKeyReq: 指向系统时间查询结构的地址。

///保证金监管系统经纪公司资金账户密钥请求

```
struct CUstpFtdcCFMMCTradingAccountKeyReqField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
};
```

6.3.48 ReqQryEnableRtnMoneyPositoinChange 方法

允许持仓变化通知请求。

函数原形：

```
int ReqQryEnableRtnMoneyPositoinChange (
    CUstpFtdcEnableRtnMoneyPositoinChangeField
    *pEnableRtnMoneyPositoinChange,
    int nRequestID)
```

参数： pEnableRtnMoneyPositoinChange 指向允许持仓变化通知结构的地址。

///允许持仓变化通知

```
struct CUstpFtdcEnableRtnMoneyPositoinChangeField
{
    ///交易触发资金推送开关
    TUstpFtdcBoolType    TradeMoneyChangeFlag;
    ///行情触发资金推送开关
    TUstpFtdcBoolType    MDMoneyChangeFlag;
    ///交易触发持仓推送开关
    TUstpFtdcBoolType    TradePositiioinChangeFlag;
    ///行情触发持仓推送开关
    TUstpFtdcBoolType    MDPositiioinChangeFlag;
};
```

6.3.49 ReqQueryHisOrder 方法

历史订单查询请求。

函数原形：

```
int ReqQueryHisOrder(CUstpFtdcQryHisOrderField *pQryHisOrder,
    int nRequestID)
```

参数： pQryHisOrder 指向历史订单结构的地址。

///历史订单查询

```
struct CUstpFtdcQryHisOrderField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType    BrokerID;
    ///用户代码
```

```

TUstpFtdcUserIDType UserID;
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID; //必填
///报单编号
TUstpFtdcOrderSysIDType OrderSysID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///报单状态
TUstpFtdcOrderStatusType OrderStatus;
///委托类型
TUstpFtdcOrderTypeType OrderType;
///客户号
TUstpFtdcClientIDType ClientID;
///起始日期
TUstpFtdcDateType BeginDate;
///结束日期
TUstpFtdcDateType EndDate;
};

```

6.3.50 ReqQueryHisTrade 方法

历史成交单查询请求。

函数原形：

```
int ReqQueryHisTrade(CUstpFtdcQryHisTradeField *pQryHisTrade,
    int nRequestID)
```

参数： pQryHisTrade 指向历史成交结构的地址。

///历史成交查询

```

struct CUstpFtdcQryHisTradeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;

```

```
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID; //必填
///成交编号
TUstpFtdcTradeIDType TradeID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///客户号
TUstpFtdcClientIDType ClientID;
///起始日期
TUstpFtdcDateType BeginDate;
///结束日期
TUstpFtdcDateType EndDate;
};
```

6.3.51ReqCurrencyPledge 方法

货币质押解质请求。

函数原形:

```
int ReqCurrencyPledge(CUstpFtdcCurrencyPledgeField *pCurrencyPledge,
int nRequestID);
```

参数: pCurrencyPledge 指向货币质押结构的地址。

```
///货币质押
struct CUstpFtdcCurrencyPledgeField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///资金帐号
    TUstpFtdcAccountIDType AccountID;
    ///资金流水号
    TUstpFtdcAccountSeqNoType AccountSeqNo;
    ///币种
    TUstpFtdcCurrencyIDType SrcCurrency;
    ///金额
```

```

    TUstpFtdcMoneyType  SrcAmount;
    ///币种
    TUstpFtdcCurrencyIDType  DestCurrency;
    ///金额
    TUstpFtdcMoneyType  DestAmount;
    ///质押动作类型
    TUstpFtdcPledgeActionTypeType  Action;
    ///交易日
    TUstpFtdcTradingDayType  TradingDay;
    ///交易时间
    TUstpFtdcTimeType  TradingTime;
};

```

6.3.52 ReqSettlementInfoConfirm 方法

请求对账单确认。

函数原形：

```

int ReqSettlementInfoConfirm(
    CUstpFtdcInputSettlementInfoConfirmField *pInputSettlementInfoConfirm,
    int nRequestID)

```

参数：略

6.3.53 ReqQueryMaxOrderVolume 方法

用户发出查询最大可开/可平手数请求。

函数原形：

```

int ReqQueryMaxOrderVolume(
    CUstpFtdcReqQueryMaxOrderVolumeField *pReqQueryMaxOrderVolume,
    int nRequestID) ;

```

参数：

pReqQueryMaxOrderVolume ：指向用户查询最大可开可平手数请求结构的地址。

查询最大可开可平手数请求结构：

///查询最大报单数量请求

```

struct CUstpFtdcReqQueryMaxOrderVolumeField
{
    ///经纪公司编号

```

```
TUstpFtdcBrokerIDType  BrokerID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///客户代码
TUstpFtdcClientIDType   ClientID;
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///买卖方向
TUstpFtdcDirectionType  Direction;
///投机套保标志
TUstpFtdcHedgeFlagType   HedgeFlag;
///开平标志
TUstpFtdcOffsetFlagType  OffsetFlag;
///价格
TUstpFtdcPriceType       Price;
};
nRequestID: 用户登录请求的 ID, 该 ID 由用户指定, 管理。
返回值:
0, 代表成功。
-1, 表示网络连接失败;
-2, 表示未处理请求超过许可数;
-3, 表示每秒发送请求数超过许可数。
```

6.3.54 ReqQueryContractBank 方法

注意：查询签约银行请求接口

函数原形：

```
int ReqQueryContractBank(
    CUstpFtdcQryContractBankReqField *pQryContractBankReq,
    int nRequestID)
```

参数：

```
    //查询签约银行请求
    struct CUstpFtdcQryContractBankReqField
    {
```

```
    ///经纪公司编号
    TUstpFtdcTransBrokerIDType BrokerID;
    ///银行代码
    TUstpFtdcTransBankIDType BankID;
};
```

6.3.55 ReqQryInvestorPositionDetail 方法

投资者持仓明细查询，包含单腿持仓和组合持仓。

函数原形：

```
int ReqQryInvestorPositionDetail(
    CUstpFtdcReqQryInvestorPositionDetailField *pReqQryInvestorPositionDetail,
    int nRequestID)
```

参数：

pReqQryInvestorPositionDetail 指向投资者持仓明细查询结构的地址。

投资者持仓明细查询结构

```
struct CUstpFtdcReqQryInvestorPositionDetailField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///单腿合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///交易编码
    TUstpFtdcClientIDType ClientID;
};
```

6.3.56 ReqQryInvestorPositionCombDetail 方法

投资者查询组合持仓明细。

函数原形：

```
int ReqQryMarginPrefParam(
    CUstpFtdcReqQryInvestorPositionCombDetailField
```



```
*pReqQryInvestorPositionCombDetail,
int nRequestID)
```

参数:

pReqQryInvestorPositionCombDetail 指向组合持仓明细结构的地址。

///请求查询持仓明细

```
struct CUstpFtdcReqQryInvestorPositionCombDetailField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType ExchangeID;
```

///用户代码

```
TUstpFtdcUserIDType UserID;
```

///投资者编号

```
TUstpFtdcInvestorIDType InvestorID;
```

///组合合约代码

```
TUstpFtdcInstrumentIDType CombInstrumentID;
```

///交易编码

```
TUstpFtdcClientIDType ClientID;
```

```
};
```

6.3.57 ReqQryMarginPrefParam 方法

投资者查询保证金优惠参数请求，获取组合合约在盘中交易期间是否优惠、优先级等信息。

函数原形:

```
int ReqQryMarginPrefParam(
    CUstpFtdcReqQryMarginPrefParamField *pReqQryMarginPrefParam,
    int nRequestID)
```

参数:

pReqQryMarginPrefParam 指向保证金优惠参数结构的地址。

```
struct CUstpFtdcReqQryMarginPrefParamField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///交易所代码

```

TUstpFtdcExchangeIDType ExchangeID;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///组合合约代码
TUstpFtdcCombInstrumentIDType CombInstrumentID;
///组合合约名称
TUstpFtdcCombInstrumentIDType CombInstrumentName;
};

```

6.3.58 ReqDSUserCertification 方法

投资者穿透监管客户认证请求。

函数原形:

```
int ReqDSUserCertification(CUstpFtdcDSUserInfoField *pDSUserInfo, int nRequestID)
```

参数:

pDSUserInfo 指向穿透式监管客户认证结构的地址。

///穿透监管客户信息

```

struct CUstpFtdcDSUserInfoField
{
    ///用户 AppID
    TUstpFtdcDSAppIDType AppID;
    ///用户授权码
    TUstpFtdcDSAAuthorizationCodeType AuthCode;
    ///密钥加密类型
    TUstpFtdcDSKeyEncryptType EncryptType;
};

```

6.3.59 ReqDSProxySubmitInfo 方法

穿透监管信息采集中继上传信息请求。

函数原形:

```

int ReqDSProxySubmitInfo(
    CUstpFtdcDSProxySubmitDataField *pDSProxySubmitData,
    int nRequestID)

```

参数:

pDSProxySubmitData 指向穿透式监管中继上传信息结构的地址。

///穿透监管中继提交信息

```
struct CUstpFtdcDSProxySubmitDataField
```

```

{
    ///AppID
    TUstpFtdcDSAppIDType    AppID;
    ///客户终端公网 IP
    TUstpFtdcDSPubNetIPType TerminalPubNetIP;
    ///客户终端公网端口号
    TUstpFtdcDSPubNetPortType TerminalPubNetPort;
    ///客户终端登入时间
    TUstpFtdcDSLoginTimeType TerminalLoginTime;
    ///异常标识
    TUstpFtdcDSExceptionFlagType ExceptionFlag;
    ///RelayID
    TUstpFtdcDSAppIDType    RelayID;
    ///终端采集信息
    TUstpFtdcDSLocalSystemDataType TerminalSystemData;
};

```

6.3.60 ReqQrySettlementInfoConfirm 方法

请求查询对账单确认。

函数原形:

```

int ReqQrySettlementInfoConfirm (
    CUstpFtdcQrySettlementInfoConfirmField * pQrySettlementInfoConfirm,
    int nRequestID)

```

参数:

///结算单确认请求

```

struct CUstpFtdcInputSettlementInfoConfirmField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType    BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
};

```

6.3.61 ReqQryExchangeRate 方法（未启用）

交叉汇率查询。

注意：外汇期货合约专用接口，目前尚未启用。

函数原形：

```
int ReqQryExchangeRate(  
    CUstpFtdcQryExchangeRateField *pQryExchangeRate,  
    int nRequestID)
```

参数：

pQryExchangeRate 指向交叉汇率查询结构的地址。

交叉汇率查询结构

```
struct CUSTPFTDCQRYEXCHANGERATEFIELD  
{  
    ///交易所代码  
    TUSTPFTDCExchangeIDType ExchangeID;  
    ///品种代码  
    TUSTPFTDCProductIDType ProductID;  
};
```

6.3.62 ReqQrySystemTime 方法（未启用）

注意：查询飞马系统时间信息。

函数原形：

```
int ReqQrySystemTime (  
    CUSTPFTDCReqQrySystemTimeField *pReqQrySystemTime,  
    int nRequestID)
```

参数： pReqQrySystemTime 指向系统时间查询结构的地址。

系统时间查询结构

```
struct CUSTPFTDCReqQrySystemTimeField  
{  
    ///交易所代码  
    TUSTPFTDCExchangeIDType ExchangeID;  
};
```

6.3.63 ReqQrySettlementInfo 方法（未启用）

注意：查询投资者结算结果接口，目前尚未启用

函数原形：

```
int ReqQrySettlementInfo(  
    CUstpFtdcSettlementQryReqField *pSettlementQryReq,  
    int nRequestID)
```

参数：略

6.3.64 ReqQryEWarrantOffset 方法（未启用）

注意：请求查询仓单折抵信息接口，目前尚未启用

函数原形：

```
int ReqQryEWarrantOffset(  
    CUstpFtdcEWarrantOffsetFieldReqField *pEWarrantOffsetFieldReq,  
    int nRequestID)
```

参数：略

6.3.65 SubscribeUserTopic 方法（未启用）

订阅交易员流。该方法要在 Init 方法前调用。若不调用则不会收到交易员的数据。

函数原形：

```
void SubscribeUserTopic(USTP_TE_RESUME_TYPE nResumeType);
```

参数：

nResumeType: 公共流重传方式

USTP_TERT_RESTART: 从本交易日开始重传

USTP_TERT_RESUME: 从上次收到的续传

USTP_TERT_QUICK: 只传送登录后公共流的内容

6.3.66 ReqQryComplianceParam 方法（不启用）

6.3.67 ReqProxyUserLogin 方法

穿透式监管启用后，中继代理进行认证后，直接登录柜台的方式，同时将采集到的客户端信

息提交给柜台。

函数原形:

///中继登录请求

```
int ReqProxyUserLogin(CUstpFtdcReqProxyUserLoginField *pReqProxyUserLogin,  
                      int nRequestID)
```

参数:

pReqProxyUserLogin: 指向中继用户结构的地址。

中继用户结构:

///系统用户登录请求

```
struct CUstpFtdcReqProxyUserLoginField
```

```
{
```

///交易日

```
TUstpFtdcDateType    TradingDay;
```

///交易用户代码

```
TUstpFtdcUserIDType  UserID;
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///密码

```
TUstpFtdcPasswordType Password;
```

///用户端产品信息

```
TUstpFtdcProductInfoType  UserProductInfo;
```

///接口端产品信息

```
TUstpFtdcProductInfoType  InterfaceProductInfo;
```

///协议信息

```
TUstpFtdcProtocolInfoType  ProtocolInfo;
```

///IP 地址

```
TUstpFtdcIPAddressType  IPAddress;
```

///Mac 地址

```
TUstpFtdcMacAddressType  MacAddress;
```

///数据中心代码

```
TUstpFtdcDataCenterIDType  DataCenterID;
```

///客户端运行文件大小

```
TUstpFtdcFileSizeType  UserProductFileSize;
```

```

    ///客户认证类型
    TUSTpFtdcAuthenticate2TypeType Authenticate2Type;

    ///客户认证密码
    TUSTpFtdcAuthenticate2PasswordType Authenticate2Password;

    ///开发厂商终端编码
    TUSTpFtdcTerminalCodeType TerminalCode;

    ///AppID
    TUSTpFtdcDSAppIDType AppID;

    ///客户终端公网 IP
    TUSTpFtdcDSPubNetIPType TerminalPubNetIP;

    ///客户终端公网端口号
    TUSTpFtdcDSPubNetPortType TerminalPubNetPort;

    ///客户终端登入时间
    TUSTpFtdcDSLoginTimeType TerminalLoginTime;

    ///异常标识
    TUSTpFtdcDSExceptionFlagType ExceptionFlag;

    ///RelayID
    TUSTpFtdcDSAppIDType RelayID;

    ///终端采集信息
    TUSTpFtdcDSLSystemDataType TerminalSystemData;

};

```

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

返回值:

0, 代表成功。

-1, 表示网络连接失败;

-2, 表示未处理请求超过许可数;

-3, 表示每秒发送请求数超过许可数。

注意:

此接口跟普通用户登录的调用方式相同, 仅为提供中继代理使用。

穿透式监管启用后, 中继代理连接飞马进行认证后, 直接登录飞马柜台的方式, 同时将采集到的客户端信息提交给柜台。

特别注意, 此接口对应的 rsp 接口与普通登录的 rsp 接口相同, 均为 OnRspUserLogin()。

根据穿透式监管要求, 需采集客户端的 IP 地址和 Mac 地址, 而飞马再中继模式下, 采集的是中继服务器地址, 所以中继模式下调用该接口需要将 IP、mac 两字段赋值为客户终端的真实 IP、mac 地

址。

6.3.68 ReqQrySystemTime 方法（主柜台不支持）

客户端发出查询某个交易所时间请求，以获取交易所时间（并不是严格意义上的交易所时间）。

函数原形：

///用户请求出入金

```
int ReqQrySystemTime(CUstpFtdcReqQrySystemTimeField *pReqQrySystemTime,
                    int nRequestID) ;
```

参数：

pReqQrySystemTime：指向查询交易所时间结构的地址。

交易所时间结构结构：

```
struct CUstpFtdcReqQrySystemTimeField
{
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
};
```

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

返回值：

0，代表成功。

-1，表示网络连接失败；

-2，表示未处理请求超过许可数；

-3，表示每秒发送请求数超过许可数。

6.3.69 ReqQrySpotPrevClosePrice 方法

查询现货标的昨日收盘价请求。

函数原形：

```
int ReqQrySpotPrevClosePrice(
    CUstpFtdcQrySpotPrevClosePriceField *pQrySpotPrevClosePrice,
    int nRequestID) ;
```

参数：

///查询现货标的昨日收盘价请求

```
struct CUstpFtdcQrySpotPrevClosePriceField
{
    ///经纪公司编号
```



```
TUstpFtdcBrokerIDType   BrokerID;
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///品种代码
TUstpFtdcProductIDType   ProductID;
///产品类型
TUstpFtdcProductClassType ProductClass;
};
```

6.3.70 ReqQryOptionTradeCost 方法

查询期权交易成本请求。

函数原形:

```
int ReqQryOptionTradeCost(
    CUstpFtdcQryOptionTradeCostField *pQryOptionTradeCost,
    int nRequestID)
```

参数:

```
///期权成交成本查询请求
struct CUstpFtdcQryOptionTradeCostField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///用户代码
    TUstpFtdcUserIDType   UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TUstpFtdcHedgeFlagType   HedgeFlag;
    ///期权合约报价
    TUstpFtdcPriceType   InputPrice;
    ///标的价格
    TUstpFtdcPriceType   UnderlyingPrice;
```

```
};;
```

6.3.71 ReqQryProduct 方法

产品信息查询请求。

函数原形:

```
int ReqQryProduct(CUstpFtdcQryProductField *pQryProduct, int nRequestID);
```

参数:

///产品信息查询请求

```
struct CUstpFtdcQryProductField
{
    ///经纪公司编号
    TustpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TustpFtdcExchangeIDType ExchangeID;
    ///品种代码
    TustpFtdcProductIDType  ProductID;
    ///产品类型
    TustpFtdcProductClassType ProductClass;
    ///特殊品种标志
    TustpFtdcProductTypeType ProductType;
};;
```

6.3.72 ReqQryInputCombAction 方法

组合拆分查询请求。

函数原形:

```
int ReqQryInputCombAction(
    CUstpFtdcQryInputCombActionField *pQryInputCombAction,
    int nRequestID);
```

参数:

///组合拆分申请查询请求

```
struct CUstpFtdcQryInputCombActionField
{
    ///经纪公司编号
    TustpFtdcBrokerIDType   BrokerID;
```

```
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///用户代码
TUstpFtdcUserIDType UserID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///组合合约代码
TUstpFtdcCombInstrumentIDType CombInstrumentID;
///组合动作方向
TUstpFtdcDirectionType CombDirection;
///交易编码
TUstpFtdcClientIDType ClientID;
};
```

6.3.73 ReqQryMarginCombCancel 方法

组合申请撤销查询请求。

函数原形:

```
int ReqQryMarginCombCancel(
    CUstpFtdcQryMarginCombCancelField *pQryMarginCombCancel,
    int nRequestID);
```

参数:

```
///查询组合申请撤销
struct CUstpFtdcQryMarginCombCancelField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
```

```
TUstpFtdcUserIDType  UserID;
///用户本地编号
TUstpFtdcUserOrderLocalIDType  CancelUserActionLocalID;
///客户代码
TUstpFtdcClientIDType  ClientID;
///组合合约代码
TUstpFtdcCombInstrumentIDType  CombInstrumentID;
};;
```

6.3.74 ReqMarginCombCancel 方法

组合申请撤销请求。

函数原形:

```
Int ReqMarginCombCancel(
    CUstpFtdcstpMarginCombCancelField *pstpMarginCombCancel,
    int nRequestID)
```

参数:

```
///组合申请撤销
struct CUstpFtdcstpMarginCombCancelField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType  BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType  ExchangeID;
    ///系统报单编号
    TUstpFtdcOrderSysIDType  OrderSysID;
    ///投资者编号
    TUstpFtdcInvestorIDType  InvestorID;
    ///用户代码
    TUstpFtdcUserIDType  UserID;
    ///用户本地编号
    TUstpFtdcUserOrderLocalIDType  CancelUserActionLocalID;
    ///客户代码
    TUstpFtdcClientIDType  ClientID;
    ///本地编号
    TUstpFtdcOrderLocalIDType  ActionLocalID;
```

```

///投机套保标志
TustpFtdcHedgeFlagType   HedgeFlag;
///组合合约代码
TustpFtdcCombInstrumentIDType   CombInstrumentID;
///组合方向
TustpFtdcDirectionType   Direction;
///组合动作方向
TustpFtdcCombDirectionType   CombDirection;
///组合数量
TustpFtdcVolumeType   CombVolume;
};

```

6.3.75 ReqQryExecSelfClose 方法

期权对冲设置查询请求

函数原形:

```

int ReqQryExecSelfClose(
    CUSTpFtdcQryExecSelfCloseField *pQryExecSelfClose, int nRequestID)

```

参数:

```

    ///期权对冲设置查询接口
struct CUSTpFtdcQryExecSelfCloseField
{
    ///经纪公司编号
    TustpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TustpFtdcExchangeIDType   ExchangeID;
    ///用户代码
    TustpFtdcUserIDType   UserID;
    ///投资者编号
    TustpFtdcInvestorIDType   InvestorID;
    ///合约代码
    TustpFtdcInstrumentIDType   InstrumentID;
    ///是否有效
    TustpFtdcIsActiveType   ActiveFlag;
    ///履约头寸自对冲标志
    TustpFtdcSelfCloseFlagType   SelfCloseFlag;

```

```
///开始时间
TUstpFtdcTimeType InsertTimeStart;
///结束时间
TUstpFtdcTimeType InsertTimeEnd;
};;
```

6.3.76 ReqExecSelfCloseInsert 方法

期权对冲设置录入请求。

函数原形：

```
int ReqExecSelfCloseInsert(
    CUstpFtdcInputExecSelfCloseField *pInputExecSelfClose,
    int nRequestID)
```

参数：

```
///期权对冲设置录入请求
struct CUstpFtdcInputExecSelfCloseField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TUstpFtdcExchangeIDType ExchangeID;
    ///用户代码
    TUstpFtdcUserIDType    UserID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///合约代码
    TUstpFtdcInstrumentIDType InstrumentID;
    ///本地期权自对冲编号
    TUstpFtdcUserSelfCloseLocalIDType UserSelfCloseLocalID;
    ///履约头寸自对冲标志
    TUstpFtdcSelfCloseFlagType SelfCloseFlag;
    ///投机套保标志
    TUstpFtdcHedgeFlagType   HedgeFlag;
    ///数量
    TUstpFtdcVolumeType      Volume;
    ///业务单元
```

```

TustpFtdcBusinessUnitType BusinessUnit;
///币种
TustpFtdcCurrencyIDType Currency;
///交易编码
TustpFtdcClientIDType ClientID;
};

```

字段	说明
ExchangeID	支持大商所、上期所、能源中心
SelfCloseFlag	///对冲(期货)#define USTP_FTDC_DF_Delivery '1' (废除) ///对冲(期权)#define USTP_FTDC_DF_OptDelivery '2' (上期、能源、大商支持) ///保留持仓#define USTP_FTDC_DF_OptHold '3' (上期和能源的做市商支持) ///履约后对冲#define USTP_FTDC_DF_SellDelivery '4' (上期、能源、大商支持) ///履约后不对冲#define USTP_FTDC_DF_SellNoDelivery '5' (大商支持)
InstrumentID	大商所的履约后对冲、履约后不对冲是对所有合约生效的，故该字段无效，可填“ALL”。 注：大商所的履约后对冲、履约后不对冲设置，两者只能有一个存在，切换时，需要撤销后再设置；且履约类是一次设置，一直有效的，不需要每天设置。 其他对冲设置都是当日有效。
Volume	大商所不带量，只能填 0；上期所、能源所需要带量。

6.3.77 ReqExecSelfCloseAction 方法

期权对冲设置撤单录入请求

函数原形：

```

int ReqExecSelfCloseAction(
    CustpFtdcInputExecSelfCloseActionField *pInputExecSelfCloseAction,
    int nRequestID)

```

参数：

///期权对冲设置录入撤销请求

```

struct CUSTPFTDCINPUTEXECSELF_CLOSE_ACTION_FIELD
{

```

```
///经纪公司编号
TUstpFtdcBrokerIDType BrokerID;
///交易所代码
TUstpFtdcExchangeIDType ExchangeID;
///用户代码
TUstpFtdcUserIDType UserID;
///投资者编号
TUstpFtdcInvestorIDType InvestorID;
///合约代码
TUstpFtdcInstrumentIDType InstrumentID;
///投机套保标志
TUstpFtdcHedgeFlagType HedgeFlag;
///本地期权自对冲撤单编号
TUstpFtdcUserSelfCloseActionLocalIDType UserSelfCloseActionLocalID;
///自对冲编号
TUstpFtdcSelfCloseSysIDType SelfCloseSysID;
///会话号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
///本地期权自对冲编号
TUstpFtdcUserSelfCloseLocalIDType UserSelfCloseLocalID;
///履约头寸自对冲标志
TUstpFtdcSelfCloseFlagType SelfCloseFlag;
///交易编码
TUstpFtdcClientIDType ClientID;
};;
```

6.3.78 ReqChangeCenter 方法

切换交易中心请求。

函数原形:

```
int ReqChangeCenter(
CUstpFtdcReqChangeCenterField *pReqChangeCenter,
int nRequestID)
```

参数:

///切换交易中心请求

```
struct CUstpFtdcReqChangeCenterField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
    ///新所属交易中心
    TUstpFtdcDataCenterIDType NewCenterID;
    ///原所属交易中心
    TUstpFtdcDataCenterIDType OldCenterID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
};
```

6.3.79 ReqQryTradeCenter 方法

交易中心查询请求。

函数原形:

```
int ReqQryTradeCenter(
CUstpFtdcQryTradeCenterField *pQryTradeCenter,
int nRequestID)
```

参数:

```
///查询交易中心请求
struct CUstpFtdcQryTradeCenterField
{
    ///经纪公司编号
    TUstpFtdcBrokerIDType BrokerID;
    ///投资者编号
    TUstpFtdcInvestorIDType InvestorID;
    ///用户代码
    TUstpFtdcUserIDType UserID;
};
```

6.3.80 ReqParkedOrderInsert 方法

预埋单报单录入请求。

函数原形：

```
int ReqParkedOrderInsert(  
CUstpFtdcParkedOrderInsertField *pParkedOrderInsert,  
int nRequestID)
```

参数：

///预埋单报单

```
struct CUstpFtdcParkedOrderInsertField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType ExchangeID;
```

///投资者编号

```
TUstpFtdcInvestorIDType InvestorID;
```

///用户代码

```
TUstpFtdcUserIDType UserID;
```

///指定通过此席位序号下单

```
TUstpFtdcSeatNoType SeatNo;
```

///合约代码/套利组合合约号

```
TUstpFtdcInstrumentIDType InstrumentID;
```

///预埋单报单编号

```
TUstpFtdcUserOrderLocalIDType UserOrderLocalID;
```

///报单类型

```
TUstpFtdcOrderPriceTypeType OrderPriceType;
```

///买卖方向

```
TUstpFtdcDirectionType Direction;
```

///开平标志

```
TUstpFtdcOffsetFlagType OffsetFlag;
```

///投机套保标志

```
TUstpFtdcHedgeFlagType HedgeFlag;
```

///价格

```
TUstpFtdcPriceType LimitPrice;
```

```
///数量
TUstpFtdcVolumeType Volume;
///有效期类型
TUstpFtdcTimeConditionType TimeCondition;
///GTD 日期
TUstpFtdcDateType GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType VolumeCondition;
///最小成交量
TUstpFtdcVolumeType MinVolume;
///止损价止赢价
TUstpFtdcPriceType StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志
TUstpFtdcBoolType IsAutoSuspend;
///业务单元
TUstpFtdcBusinessUnitType BusinessUnit;
///用户自定义域
TUstpFtdcCustomType UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType BusinessLocalID;
///业务发生日期
TUstpFtdcDateType ActionDay;
///策略类别
TUstpFtdcArbiTypeType ArbiType;
///客户号
TUstpFtdcClientIDType ClientID;
///会话编号
TUstpFtdcSessionIDType SessionID;
///前置编号
TUstpFtdcFrontIDType FrontID;
///预埋单状态
TUstpFtdcParkedOrderStatusType ParkedOrderStatus;
};
```

6.3.81 ReqParkedOrderAction 方法

预埋单撤单请求。

函数原形：

```
int ReqParkedOrderAction(  
CUstpFtdcParkedOrderActionField *pParkedOrderAction,  
int nRequestID)
```

参数：

///预埋单撤单

```
struct CUstpFtdcParkedOrderActionField
```

```
{
```

///交易所代码

TUstpFtdcExchangeIDType

ExchangeID;

///经纪公司编号

TUstpFtdcBrokerIDType

BrokerID;

///投资者编号

TUstpFtdcInvestorIDType

InvestorID;

///用户代码

TUstpFtdcUserIDType

UserID;

///本次撤单操作的本地编号

TUstpFtdcUserOrderLocalIDType

UserOrderActionLocalID;

///被撤预埋单报单编号

TUstpFtdcUserOrderLocalIDType

UserOrderLocalID;

///本地业务标识

TUstpFtdcBusinessLocalIDType

BusinessLocalID;

///客户代码

TUstpFtdcClientIDType

ClientID;

///会话编号

TUstpFtdcSessionIDType

SessionID;

///前置编号

TUstpFtdcFrontIDType

FrontID;

///合约代码

TUstpFtdcInstrumentIDType

InstrumentID;

```
};
```

6.3.82 ReqQryParkedOrder 方法

预埋单查询请求。

函数原形：

```
int ReqQryParkedOrder(  
CUstpFtdcQryParkedOrderField *pQryParkedOrder,  
int nRequestID)
```

参数：

///预埋单查询请求

```
struct CUstpFtdcQryParkedOrderField
```

```
{
```

///经纪公司编号

TUstpFtdcBrokerIDType

BrokerID;

///用户代码

TUstpFtdcUserIDType

UserID;

///交易所代码

TUstpFtdcExchangeIDType

ExchangeID;

///投资者编号

TUstpFtdcInvestorIDType

InvestorID;

///预埋单报单编号

TUstpFtdcUserOrderLocalIDType

UserOrderLocalID;

///合约代码

TUstpFtdcInstrumentIDType

InstrumentID;

///预埋单状态

TUstpFtdcParkedOrderStatusType

ParkedOrderStatus;

///买卖方向

TUstpFtdcDirectionType

Direction;

///客户号

TUstpFtdcClientIDType

ClientID;

///会话编号

TUstpFtdcSessionIDType

SessionID;

///前置编号

TUstpFtdcFrontIDType

FrontID;

```
};
```

6.3.83 ReqConditionalOrderInsert 方法

条件单报单录入请求。

函数原形：

```
int ReqConditionalOrderInsert(  
CUstpFtdcConditionalOrderInsertField *pConditionalOrderInsert,  
int nRequestID)
```

参数：

///条件单报单

```
struct CUstpFtdcConditionalOrderInsertField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType          BrokerID;
```

///交易所代码

```
TUstpFtdcExchangeIDType        ExchangeID;
```

///投资者编号

```
TUstpFtdcInvestorIDType        InvestorID;
```

///用户代码

```
TUstpFtdcUserIDType            UserID;
```

///指定通过此席位序号下单

```
TUstpFtdcSeatNoType            SeatNo;
```

///合约代码/套利组合合约号

```
TUstpFtdcInstrumentIDType      InstrumentID;
```

///条件单报单编号

```
TUstpFtdcUserOrderLocalIDType  UserOrderLocalID;
```

///报单类型

```
TUstpFtdcOrderPriceType        OrderPriceType;
```

///买卖方向

```
TUstpFtdcDirectionType         Direction;
```

///开平标志

```
TUstpFtdcOffsetFlagType        OffsetFlag;
```

///投机套保标志

```
TUstpFtdcHedgeFlagType         HedgeFlag;
```

///价格

```
TUstpFtdcPriceType             LimitPrice;
```

```
///数量
TUstpFtdcVolumeType          Volume;
///有效期类型
TUstpFtdcTimeConditionType    TimeCondition;
///GTD 日期
TUstpFtdcDateType             GTDDate;
///成交量类型
TUstpFtdcVolumeConditionType  VolumeCondition;
///最小成交量
TUstpFtdcVolumeType           MinVolume;
///触发价格
TUstpFtdcPriceType            StopPrice;
///强平原因
TUstpFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志
TUstpFtdcBoolType             IsAutoSuspend;
///业务单元
TUstpFtdcBusinessUnitType     BusinessUnit;
///用户自定义域
TUstpFtdcCustomType           UserCustom;
///本地业务标识
TUstpFtdcBusinessLocalIDType  BusinessLocalID;
///业务发生日期
TUstpFtdcDateType             ActionDay;
///策略类别
TUstpFtdcArbiTypeType         ArbiType;
///客户号
TUstpFtdcClientIDType         ClientID;
///会话编号
TUstpFtdcSessionIDType        SessionID;
///前置编号
TUstpFtdcFrontIDType          FrontID;
///条件单状态
TUstpFtdcConditionalOrderStatusType ConditionalOrderStatus;
};
```

6.3.84 ReqConditionalOrderAction 方法

条件单撤单请求。

函数原形：

```
int ReqConditionalOrderAction(  
CUstpFtdcConditionalOrderActionField *pConditionalOrderAction,  
int nRequestID)
```

参数：

///条件单撤单

```
struct CUstpFtdcConditionalOrderActionField
```

```
{
```

///交易所代码

```
TUstpFtdcExchangeIDType          ExchangeID;
```

///经纪公司编号

```
TUstpFtdcBrokerIDType            BrokerID;
```

///投资者编号

```
TUstpFtdcInvestorIDType          InvestorID;
```

///用户代码

```
TUstpFtdcUserIDType              UserID;
```

///本次撤单操作的本地编号

```
TUstpFtdcUserOrderLocalIDType    UserOrderActionLocalID;
```

///被撤条件单报单编号

```
TUstpFtdcUserOrderLocalIDType    UserOrderLocalID;
```

///本地业务标识

```
TUstpFtdcBusinessLocalIDType     BusinessLocalID;
```

///客户代码

```
TUstpFtdcClientIDType            ClientID;
```

///会话编号

```
TUstpFtdcSessionIDType           SessionID;
```

///前置编号

```
TUstpFtdcFrontIDType             FrontID;
```

///合约代码

```
TUstpFtdcInstrumentIDType        InstrumentID;
```

```
};
```


6.3.85 ReqQryConditionalOrder 方法

条件单查询请求。

函数原形：

```
int ReqQryConditionalOrder(  
CUstpFtdcQryConditionalOrderField *pQryConditionalOrder,  
int nRequestID)
```

参数：

///条件单查询请求

```
struct CUstpFtdcQryConditionalOrderField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType          BrokerID;
```

///用户代码

```
TUstpFtdcUserIDType            UserID;
```

///交易所代码

```
TUstpFtdcExchangeIDType        ExchangeID;
```

///投资者编号

```
TUstpFtdcInvestorIDType         InvestorID;
```

///条件单报单编号

```
TUstpFtdcUserOrderLocalIDType   UserOrderLocalID;
```

///合约代码

```
TUstpFtdcInstrumentIDType        InstrumentID;
```

///条件单状态

```
TUstpFtdcConditionalOrderStatusType ConditionalOrderStatus;
```

///买卖方向

```
TUstpFtdcDirectionType          Direction;
```

///客户号

```
TUstpFtdcClientIDType            ClientID;
```

///会话编号

```
TUstpFtdcSessionIDType           SessionID;
```

///前置编号

```
TUstpFtdcFrontIDType              FrontID;
```

```
};
```

6.3.86 ReqQryQuote 方法

报价查询请求。

函数原形：

```
int ReqQryQuote(  
CUstpFtdcQryQuoteField *pQryQuote,  
int nRequestID)
```

参数：

///报价查询请求

```
struct CUstpFtdcQryQuoteField
```

```
{
```

///经纪公司编号

TUstpFtdcBrokerIDType

BrokerID;

///用户代码

TUstpFtdcUserIDType

UserID;

///投资者编号

TUstpFtdcInvestorIDType

InvestorID;

///交易所代码

TUstpFtdcExchangeIDType

ExchangeID;

///合约代码

TUstpFtdcInstrumentIDType

InstrumentID;

///报价状态

TUstpFtdcQuoteStatusType

QuoteStatus;

///买卖方向

TUstpFtdcDirectionType

Direction;

///系统报价编号

TUstpFtdcQuoteSysIDType

QuoteSysID;

///客户号

TUstpFtdcClientIDType

ClientID;

```
};
```

6.3.87 ReqQryExecOrder 方法

行权查询请求。

函数原形：

```
int ReqQryExecOrder(  

```

```
CUstpFtdcQryExecOrderField *pQryExecOrder,  
int nRequestID)
```

参数:

///行权查询请求

```
struct CUstpFtdcQryExecOrderField
```

```
{
```

///经纪公司编号

```
TUstpFtdcBrokerIDType          BrokerID;
```

///用户代码

```
TUstpFtdcUserIDType            UserID;
```

///交易所代码

```
TUstpFtdcExchangeIDType        ExchangeID;
```

///投资者编号

```
TUstpFtdcInvestorIDType        InvestorID;
```

///系统报单编号

```
TUstpFtdcOrderSysIDType        OrderSysID;
```

///合约代码

```
TUstpFtdcInstrumentIDType      InstrumentID;
```

///报单状态

```
TUstpFtdcOrderStatusType       OrderStatus;
```

///委托类型

```
TUstpFtdcOrderTypeType         OrderType;
```

///客户号

```
TUstpFtdcClientIDType          ClientID;
```

```
};
```

6.3.88 ReqQueryHisTransferSerial 方法

查询历史转帐流水请求。

函数原形:

```
int ReqQueryHisTransferSerial(
CUSTPFtdcHisTransferSerialFieldReqField *pHisTransferSerialFieldReq,
int nRequestID)
```

参数:

///查询历史转帐流水请求

```
struct CUSTPFtdcHisTransferSerialFieldReqField
```

```
{
    ///经纪公司编号
    TUSTPFtdcBrokerIDType          BrokerID;
    ///用户编号
    TUSTPFtdcUserIDType            UserID;
    ///资金帐号
    TUSTPFtdcAccountIDType         AccountID;
    ///银行代码
    TUSTPFtdcBankIDType            BankID;
    ///开始日期
    TUSTPFtdcDateType              StartDay;
    ///结束日期
    TUSTPFtdcDateType              EndDay;
};
```

7. 开发实例

```
// testtradeapi.cpp :
// 一个简单的例子，介绍 CUSTPFtdcTraderApi 和 CUSTPFtdcTraderSpi 接口的使用。
// 本例将演示一个报单录入操作的过程
#include <stdio.h>
#include <windows.h>
#include "USTPFtdcTraderApi.h"
// 报单录入操作是否完成的标志
// Create a manual reset event with no signal
HANDLE g_hEvent = CreateEvent(NULL, true, false, NULL);
// 经纪公司代码
TUSTPFtdcBrokerIDType g_ch BrokerID;
// 交易用户代码
TUSTPFtdcUserIDType g_chUserID;
//用户本地最大报单号
```

```

int g_UserOrderLocalID;
class CSimpleHandler : public CUstpFtdcTraderSpi
{
public:
// 构造函数，需要一个有效的指向 CUstpFtdcMduserApi 实例的指针
CSimpleHandler(CUstpFtdcTraderApi *pUserApi) :
m_pUserApi (pUserApi) {}
~CSimpleHandler () {}
// 当客户端与飞马平台建立起通信连接，客户端需要进行登录
virtual void OnFrontConnected()
{
    CUstpFtdcReqUserLoginField reqUserLogin;
    // get BrokerID
    printf("BrokerID:");
    scanf("%s", &g_chBrokerID);
    strcpy(reqUserLogin. BrokerID, g_chBrokerID);
    // get userid
    printf("userid:");
    scanf("%s", &g_chUserID);
    strcpy(reqUserLogin.UserID, g_chUserID);
    // get password
    printf("password:");
    scanf("%s", &reqUserLogin.Password);
    // 发出登陆请求
    m_pUserApi->ReqUserLogin(&reqUserLogin, 0);
}
// 当客户端与飞马平台通信连接断开时，该方法被调用
virtual void OnFrontDisconnected(int nReason)
{
    // 当发生这个情况后，API 会自动重新连接，客户端可不做处理
    printf("OnFrontDisconnected. \n");
}
// 当客户端发出登录请求之后，该方法会被调用，通知客户端登录是否成功
virtual void OnRspUserLogin(
    CUstpFtdcRspUserLoginField *pRspUserLogin,
    CUstpFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
{
    printf("OnRspUserLogin:\n");
    printf("ErrorCode=[%d], ErrorMsg=[%s]\n",
        pRspInfo->ErrorID, pRspInfo->ErrorMsg);
    printf("RequestID=[%d], Chain=[%d]\n", nRequestID, bIsLast);
    if (pRspInfo->ErrorID != 0)

```

```
{
    // 端登失败，客户端需进行错误处理
    printf("Failed to login, errorcode=%d errormsg=%s
           requestid=%d chain=%d",
           pRspInfo->ErrorID, pRspInfo->ErrorMsg,
           nRequestID, bIsLast);
    exit(-1);
}

//用户最大本地报单号
g_UserOrderLocalID=atoi(pRspUserLogin->MaxOrderLocalID)+1;
// 端登成功, 发出报单录入请求
CUstpFtdcInputOrderField ord;
memset(&ord, 0, sizeof(ord));
//经纪公司代码
strcpy(ord.BrokerID, g_chBrokerID);
// 合约代码
strcpy(ord.InstrumentID, "IF1306");
//投资者代码
strcpy(ord.InvestorID, "000101");
// 用户代码
strcpy(ord.UserID, g_chUserID);
// 买卖方向
strcpy(ord.Direction, "0");
// 开平标志
strcpy(ord.OffsetFlag, "0");
// 投机套保标志
strcpy(ord.HedgeFlag, "1");
// 价格
ord.LimitPrice = 5000;
// 数量
ord.VolumeTotalOriginal = 10;
// 有效期类型
strcpy(ord.TimeCondition, "1");
// 自动挂起标志
ord.IsAutoSuspend = 0;
//交易所
strcpy(ord.ExchangeID, "CFFEX");
//本地报单号
sprintf(ord.UserOrderLocalID, "%012d", g_UserOrderLocalID++);
m_pUserApi->ReqOrderInsert(&ord, 1);
}

// 报单录入应答
virtual void OnRspOrderInsert(
    CUstpFtdcInputOrderField *pInputOrder,
```

```

        CUSTpFtdcRspInfoField *pRspInfo,
        int nRequestID,
        bool bIsLast)
    {
        // 输出报单录入结果
        printf("ErrorCode=[%d], ErrorMsg=[%s]\n",
            pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        // 通知报单录入完成
        SetEvent(g_hEvent);
    };
    ///报单回报
    virtual void OnRtnOrder(CUSTpFtdcOrderField *pOrder)
    {
        printf("OnRtnOrder:\n");
        printf("OrderSysID=[%s]\n", pOrder->OrderSysID);
    }
    // 针对用户请求的出错通知
    virtual void OnRspError(
        CUSTpFtdcRspInfoField *pRspInfo,
        int nRequestID,
        bool bIsLast)
    {
        printf("OnRspError:\n");
        printf("ErrorCode=[%d], ErrorMsg=[%s]\n",
            pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        printf("RequestID=[%d], Chain=[%d]\n", nRequestID, bIsLast);
        // 客户端需进行错误处理
        {
            //客户端的错误处理
        }
    }
private:
    // 指向 CUSTpFtdcMduserApi 实例的指针
    CUSTpFtdcTraderApi *m_pUserApi;
};

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

```

```
// 订阅私有流
// USTP_TERT_RESTART:从本交易日开始重传
// USTP_TERT_RESUME:从上次收到的续传
// USTP_TERT_QUICK:只传送登录后私有流的内容
    pUserApi->SubscribePrivateTopic(USTP_TERT_RESUME);
// 订阅公共流
// USTP_TERT_RESTART:从本交易日开始重传
// USTP_TERT_RESUME:从上次收到的续传
// USTP_TERT_QUICK:只传送登录后公共流的内容
    pUserApi->SubscribePublicTopic(USTP_TERT_RESUME);
// 设置飞马平台服务的地址，可以注册多个地址备用
    pUserApi->RegisterFront("tcp://172.28.21.133:15555");
// 使客户端开始与后台服务建立连接
    pUserApi->Init();
// 客户端等待报单操作完成
    WaitForSingleObject(g_hEvent, INFINITE);
// 释放 API 实例
    pUserApi->Release();
return 0;
}
```