

SUNGARD 金仕达

期 货 交 易 API 编程手册

文档标识

文档名称	金仕达期货交易 API 编程手册
版本号	<V1.12>
状况	<input type="radio"/> 草案 <input type="radio"/> 评审过的 <input type="radio"/> 更新过的 <input checked="" type="radio"/> 定为基线的

文档修订历史

版本	日期	描述	修订者
V1.0	<2011-08-15>	1. 兼容 CTP api 接口的相关字段 2. 头文件请使用命名空间 namespace KingstarAPI 3. 增加查投资者开盘前持仓、组合持仓明细查询接口	沈明明
V1.1	<2012-05-08>	1. 标出“保留字段”(紫色字体) 2. 组合单暂不支持、预埋单暂不支持、UDP 行情暂不支持、银行推送暂不支持	沈明明
V1.2	<2012-05-18>	1. 文档名称标准化	沈明明
V1.3	<2012-09-06>	1. 支持 UDP 行情 2. 增加指定本地文件路径接口	沈明明
V1.4	<2012-11-26>	1. 支持黄金夜市、仓单折抵 2. API 支持 linux 环境下开发	胡旭东
V1.5	<2013-01-07>	1. 支持 RegisterNameServer 接口	胡旭东
V1.6	<2013-03-06>	1. 支持批量撤单接口 2. 支持条件单接口	胡旭东
V1.7	<2014-01-17>	1. 支持上海单边优惠接口	沈明明
V1.8	<2014-04-04>	1. 支持期货期权	刘艳云
V1.9	<2014-08-04>	1. 增加 Voc 接口及做市商接口	刘艳云
V1.10	<2014-12-05>	1. 增加交叉汇率历史报单等接口	刘艳云
V1.11	<2015-02-12>	1. 接口更新 2. 增加 CKSVocMdApi 接口	刘艳云
V1.12	<2017-09-26>	1. 接口更新	胡闻涛
V1.13	<2022-03-02>	1. 接口更新	蔡永恒
V1.14	<2022-07-07>	1. 接口更新	蔡永恒

V1.15	<2023-03-07>	1. 增加出入金查询接口 2. 取消大额出金预约申请接口 3. 取消大额出金预约查询接口	王兵
V1.16	<2023-10-12>	1. 接口更新（郑州 SPBM 相关接口）	王兵
V1.17	<2024-02-23>	1. 接口更新（上海 SPMM 相关接口） 2. 接口更新（大连 RULE 相关接口）	王兵

正式核准

姓名	签字	日期

分发控制

副本	接受人	机构

目 录

1. 第一章 介绍	21
1.1. 概述	21
1.2. API 文件	21
2. 第二章 体系结构	23
2.1. 通讯模式	23
2.2. 数据流	23
2.3. 通讯流程	24
2.3.1. 交易流程	24
2.3.2. 行情流程	25
2.4. 报单指令（特别说明）	26
2.4.1. 指令正常	26
2.4.2. 指令异常	26
3. 第三章 接口模式	28
3.1. 对话模式的编程接口	28
3.2. 私有模式的编程接口	29
3.3. 广播模式的编程接口	29
4. 第四章 开发接口	29
4.1. 工作线程	29
4.2. 通用规则	29
4.3. CThostFtdcTraderSpi 接口	31
4.3.1. OnFrontConnected 方法	31
4.3.2. OnFrontDisconnected 方法	31

4.3.3. OnHeartBeatWarning 方法[暂不支持]	32
4.3.4. OnRspUserLogin 方法	32
4.3.5. OnRspUserLogout 方法	33
4.3.6. OnRspUserPasswordUpdate 方法	34
4.3.7. OnRspTradingAccountPasswordUpdate 方法	35
4.3.8. OnRspError 方法	35
4.3.9. OnRspOrderInsert 方法	36
4.3.10. OnRspOrderAction 方法	38
4.3.11. OnRspQueryMaxOrderVolume 方法	39
4.3.12. OnRspExecOrderInsert 方法	40
4.3.13. OnRspExecOrderAction 方法	42
4.3.14. OnRspSettlementInfoConfirm 方法	43
4.3.15. OnRspFromBankToFutureByFuture 方法	44
4.3.16. OnRspFromFutureToBankByFuture 方法	47
4.3.17. OnRspQueryBankAccountMoneyByFuture 方法	50
4.3.18. OnRspQryTransferSerial 方法	53
4.3.19. OnRspQryAccountregister 方法	55
4.3.20. OnRspQryOrder 方法	56
4.3.21. OnRspQryTrade 方法	60
4.3.22. OnRspQryInvestor 方法	63
4.3.23. OnRspQryInvestorPosition 方法	64
4.3.24. OnRspQryTradingAccount 方法	67
4.3.25. OnRspQryTradingCode 方法	70
4.3.26. OnRspQryExchange 方法	71

4.3.27. OnRspQryInstrument 方法	72
4.3.28. OnRspQryDepthMarketData 方法	74
4.3.29. OnRspQryInstrumentMarginRate 方法	78
4.3.30. OnRspQryInstrumentCommissionRate 方法	79
4.3.31. OnRspQryCFMMCTradingAccountKey 方法	80
4.3.32. OnRspQryEWarrantOffset 方法	81
4.3.33. OnRspQryInvestorProductGroupMargin 方法[暂不支持]	82
4.3.34. OnRspQryExchangeMarginRate 方法[暂不支持]	84
4.3.35. OnRspQryExchangeMarginRateAdjust 方法[暂不支持]	85
4.3.36. OnRspQrySettlementInfo 方法	86
4.3.37. OnRspQryTransferBank 方法	87
4.3.38. OnRspQryInvestorPositionDetail 方法	87
4.3.39. OnRspQryNotice 方法	90
4.3.40. OnRtnTrade 方法	90
4.3.41. OnRtnOrder 方法	92
4.3.42. OnErrRtnOrderInsert 方法[暂不支持]	96
4.3.43. OnErrRtnOrderAction 方法[暂不支持]	98
4.3.44. OnRtnInstrumentStatus 方法	100
4.3.45. OnRtnTradingNotice 方法	101
4.3.46. OnRtnErrorConditionalOrder 方法	102
4.3.47. OnRtnExecOrder 方法	105
4.3.48. OnErrRtnExecOrderInsert 方法	107
4.3.49. OnErrRtnExecOrderAction 方法	108
4.3.50. OnRspQrySettlementInfoConfirm 方法	110

4.3.51. OnRspQryContractBank 方法	111
4.3.52. OnRspQryParkedOrder 方法[暂不支持]	111
4.3.53. OnRspQryParkedOrderAction 方法[暂不支持]	114
4.3.54. OnRspQryTradingNotice 方法	115
4.3.55. OnRspQryBrokerTradingParams 方法	116
4.3.56. OnRspQryInvestorPositionCombineDetail 方法	117
4.3.57. OnRspParkedOrderInsert 方法[暂不支持]	118
4.3.58. OnRspParkedOrderAction 方法[暂不支持]	121
4.3.59. OnRspRemoveParkedOrder 方法[暂不支持]	122
4.3.60. OnRspRemoveParkedOrderAction 方法[暂不支持]	123
4.3.61. OnRspQryBrokerTradingAlgos 方法	124
4.3.62. OnRspForQuoteInsert 方法	124
4.3.63. OnRspQuoteInsert 方法	125
4.3.64. OnRspQuoteAction 方法	127
4.3.65. OnRspQryProduct 方法	128
4.3.66. OnRspQryExchangeRate 方法[暂不支持]	129
4.3.67. OnRspQrySecAgentACIDMap 方法[暂不支持]	130
4.3.68. OnRspQryForQuote 方法	131
4.3.69. OnRspQryQuote 方法	133
4.3.70. OnErrRtnForQuoteInsert 方法[暂不支持]	136
4.3.71. OnRtnQuote 方法	137
4.3.72. OnRtnForQuoteRsp 方法	140
4.3.73. OnErrRtnQuoteInsert 方法[暂不支持]	140
4.3.74. OnErrRtnQuoteAction 方法[暂不支持]	142

4.3.75. OnRspQueryCFMMCTradingAccountToken 方法[暂不支持]	144
4.3.76. OnRtnFromBankToFutureByBank 方法	144
4.3.77. OnRtnFromFutureToBankByBank 方法	147
4.3.78. OnRtnRepealFromBankToFutureByBank 方法	149
4.3.79. OnRtnRepealFromFutureToBankByBank 方法	152
4.3.80. OnRtnFromBankToFutureByFuture 方法	155
4.3.81. OnRtnFromFutureToBankByFuture 方法	158
4.3.82. OnRtnRepealFromBankToFutureByFutureManual 方法	160
4.3.83. OnRtnRepealFromFutureToBankByFutureManual 方法	163
4.3.84. OnRtnQueryBankBalanceByFuture 方法	166
4.3.85. OnErrRtnBankToFutureByFuture 方法	168
4.3.86. OnErrRtnFutureToBankByFuture 方法	171
4.3.87. OnErrRtnRepealBankToFutureByFutureManual 方法	173
4.3.88. OnErrRtnRepealFutureToBankByFutureManual 方法	176
4.3.89. OnErrRtnQueryBankBalanceByFuture 方法	179
4.3.90. OnRtnRepealFromBankToFutureByFuture 方法	182
4.3.91. OnRtnRepealFromFutureToBankByFuture 方法	185
4.3.92. OnRtnOpenAccountByBank 方法	187
4.3.93. OnRtnCancelAccountByBank 方法	190
4.3.94. OnRtnChangeAccountByBank 方法	193
4.3.95. OnRtnCFMMCTradingAccountToken 方法[暂不支持]	195
4.3.96. OnRspLockInsert 方法	196
4.3.97. OnRspCombActionInsert 方法	197
4.3.98. OnRspQryProductExchRate 方法	198

4.3.99. OnRspQryProductGroup 方法	199
4.3.100. OnRspQryInstrumentOrderCommRate 方法	199
4.3.101. OnRspQryOptionInstrTradeCost 方法	200
4.3.102. OnRspQryOptionInstrCommRate 方法	201
4.3.103. OnRspQryExecOrder 方法	202
4.3.104. OnRspQryLock 方法	205
4.3.105. OnRspQryLockPosition 方法	207
4.3.106. OnRspQryCombInstrumentGuard 方法	208
4.3.107. OnRspQryCombAction 方法	209
4.3.108. OnRtnLock 方法	211
4.3.109. OnErrRtnLockInsert 方法	213
4.3.110. OnRtnCombAction 方法	214
4.3.111. OnErrRtnCombActionInsert 方法	216
4.3.112. OnRspQryRiskSettleInvstPosition 方法	217
4.3.113. OnRspQryRiskSettleProductStatus 方法	221
4.3.114. OnRspOptionSelfCloseAction 方法	221
4.3.115. OnRspQryCustomerSuitableEvaResults 方法	223
4.3.116. OnRspQrySPBMFutureParameter 方法	224
4.3.117. OnRspQrySPBMOptionParameter 方法	225
4.3.118. OnRspQrySPBMIntraParameter 方法	226
4.3.119. OnRspQrySPBMInterParameter 方法	227
4.3.120. OnRspQrySPBMPortfDefinition 方法	227
4.3.121. OnRspQrySPBMInvestorPortfDef 方法	228
4.3.122. OnRspQryInvestorPortfMarginRatio 方法	229

4.3.123. OnRspQryInvestorProdSPBMDetail 方法	230
4.3.124. OnRspQryPortfTradeParamSetting 方法	232
4.3.125. OnRspQrySPBMAAddOnInterParameter 方法	233
4.3.126. OnRspQrySPMMInstParam 方法	233
4.3.127. OnRspQrySPMMProductParam 方法	234
4.3.128. OnRspQryInvestorCommoditySPMMMMargin 方法	235
4.3.129. OnRspQryInvestorCommodityGroupSPMMMMargin 方法	237
4.3.130. OnRspQryRULEInstrParameter 方法	239
4.3.131. OnRspQryRULEIntraParameter 方法	240
4.3.132. OnRspQryRULEInterParameter 方法	241
4.3.133. OnRspQryInvestorProdRULEMargin 方法	242
4.4. CthostFtdcTraderApi 接口	244
4.4.1. CreateFtdcTraderApi 方法	244
4.4.2. Release 方法	244
4.4.3. init 方法	245
4.4.4. join 方法	245
4.4.5. GetTradingDay 方法	245
4.4.6. GetApiVersion 方法	245
4.4.7. RegisterSpi 方法	245
4.4.8. RegisterFront 方法	245
4.4.9. SubscribePrivateTopic 方法	246
4.4.10. SubscribePublicTopic 方法	246
4.4.11. ReqUserLoginSM 方法	246
4.4.12. ReqAuthenticate 方法	247

4.4.13. ReqUserLogin 方法	248
4.4.14. ReqUserLogout 方法	249
4.4.15. ReqUserPasswordUpdate 方法	249
4.4.16. ReqTradingAccountPasswordUpdate 方法	250
4.4.17. ReqOrderInsert 方法	251
4.4.18. ReqOrderAction 方法	253
4.4.19. ReqQueryMaxOrderVolume 方法	254
4.4.20. ReqSettlementInfoConfirm 方法	255
4.4.21. ReqFromBankToFutureByFuture 方法	255
4.4.22. ReqFromFutureToBankByFuture 方法	258
4.4.23. ReqQueryBankAccountMoneyByFuture 方法	261
4.4.24. ReqQryTransferSerial 方法	264
4.4.25. ReqQryAccountregister 方法	265
4.4.26. ReqQryOrder 方法	265
4.4.27. ReqQryTrade 方法	266
4.4.28. ReqQryInvestor 方法	267
4.4.29. ReqQryInvestorPosition 方法	267
4.4.30. ReqQryTradingAccount 方法	268
4.4.31. ReqQryTradingCode 方法	269
4.4.32. ReqQryExchange 方法	269
4.4.33. ReqQryInstrument 方法	270
4.4.34. ReqQryDepthMarketData 方法	270
4.4.35. ReqQryInstrumentMarginRate 方法	271
4.4.36. ReqQryInstrumentCommissionRate 方法	272

4.4.37. ReqQryCFMMCTradingAccountKey 方法	272
4.4.38. ReqQryEWarrantOffset 方法	273
4.4.39. ReqQryInvestorProductGroupMargin 方法	273
4.4.40. ReqQryExchangeMarginRate 方法	274
4.4.41. ReqQryExchangeMarginRateAdjust 方法	275
4.4.42. ReqQrySettlementInfo 方法	275
4.4.43. ReqQryTransferBank 方法	276
4.4.44. ReqQryInvestorPositionDetail 方法	276
4.4.45. ReqQryNotice 方法	277
4.4.46. ReqQrySettlementInfoConfirm 方法	277
4.4.47. ReqQryContractBank 方法	278
4.4.48. ReqQryParkedOrder 方法	278
4.4.49. ReqQryParkedOrderAction 方法	279
4.4.50. ReqQryTradingNotice 方法	280
4.4.51. ReqQryBrokerTradingParams 方法	280
4.4.52. ReqQryInvestorPositionCombineDetail 方法	281
4.4.53. ReqParkedOrderInsert 方法	281
4.4.54. ReqParkedOrderAction 方法	283
4.4.55. ReqRemoveParkedOrder 方法	285
4.4.56. ReqRemoveParkedOrderAction 方法	285
4.4.57. ReqExecOrderInsert 方法	286
4.4.58. ReqExecOrderAction 方法	287
4.4.59. ReqQryBrokerTradingAlgos 方法	288
4.4.60. RegisterNameServer 方法	289

4.4.61. RegisterFensUserInfo 方法	289
4.4.62. LoadExtApi 方法	290
4.4.63. ReqForQuoteInsert 方法	290
4.4.64. ReqQuoteInsert 方法	291
4.4.65. ReqQuoteAction 方法	292
4.4.66. ReqQryProduct 方法	293
4.4.67. ReqQryExchangeRate 方法	294
4.4.68. ReqQrySecAgentACIDMap 方法	294
4.4.69. ReqQryForQuote 方法	295
4.4.70. ReqQryQuote 方法	296
4.4.71. ReqQueryCFMMCTradingAccountToken 方法	297
4.4.72. ReqLockInsert 方法	297
4.4.73. ReqCombActionInsert 方法	298
4.4.74. ReqQryProductExchRate 方法	299
4.4.75. ReqQryProductGroup 方法	300
4.4.76. ReqQryInstrumentOrderCommRate 方法	300
4.4.77. ReqQryOptionInstrTradeCost 方法	301
4.4.78. ReqQryOptionInstrCommRate 方法	301
4.4.79. ReqQryExecOrder 方法	302
4.4.80. ReqQryLock 方法	303
4.4.81. ReqQryLockPosition 方法	303
4.4.82. ReqQryCombInstrumentGuard 方法	304
4.4.83. ReqQryCombAction 方法	304
4.4.84. ReqQryRiskSettleInvstPosition 方法	305

4.4.85. ReqQryRiskSettleProductStatus 方法	306
4.4.86. ReqOptionSelfCloseAction 方法	306
4.4.87. ReqQryCustomerSuitableEvaResults 方法	307
4.4.88. ReqQrySPBMFutureParameter 方法	308
4.4.89. ReqQrySPBMOptionParameter 方法	309
4.4.90. ReqQrySPBMIntraParameter 方法	309
4.4.91. ReqQrySPBMInterParameter 方法	310
4.4.92. ReqQrySPBMPortfDefinition 方法	310
4.4.93. ReqQrySPBMInvestorPortfDef 方法	311
4.4.94. ReqQryInvestorPortfMarginRatio 方法	311
4.4.95. ReqQryInvestorProdSPBMDetail 方法	312
4.4.96. ReqQryPortfTradeParamSetting 方法	313
4.4.97. ReqQrySPBMAAddOnInterParameter 方法	313
4.4.98. ReqQrySPMMInstParam 方法	314
4.4.99. ReqQrySPMMProductParam 方法	314
4.4.100. ReqQryInvestorCommoditySPMMMMargin 方法	315
4.4.101. ReqQryInvestorCommodityGroupSPMMMMargin 方法	315
4.4.102. ReqQryRULEInstrParameter 方法	316
4.4.103. ReqQryRULEIntraParameter 方法	317
4.4.104. ReqQryRULEInterParameter 方法	317
4.4.105. ReqQryInvestorProdRULEMargin 方法	318
4.5. CthostFtdcMdSpi 接口	319
1.1.1 OnFrontConnected 方法	319
4.5.1. OnFrontDisconnected 方法	319

4.5.2. OnHeartBeatWarning 方法	319
4.5.3. OnRspUserLogin 方法	319
4.5.4. OnRspUserLogout 方法	321
4.5.5. OnRspError 方法	322
4.5.6. OnRspSubMarketData 方法	322
4.5.7. OnRspUnSubMarketData 方法	323
4.5.8. OnRspSubForQuoteRsp 方法	323
4.5.9. OnRspUnSubForQuoteRsp 方法	324
4.5.10. OnRtnDepthMarketData 方法	324
4.6. CthostFtdcMdApi 接口	327
4.6.1. CreateFtdcMdApi 方法	327
4.6.2. GetApiVersion 方法	328
4.6.3. Release 方法	328
4.6.4. Init 方法	328
4.6.5. Join 方法	328
4.6.6. GetTradingDay 方法	328
4.6.7. RegisterFront 方法	328
4.6.8. RegisterSpi 方法	329
4.6.9. SubscribeMarketData 方法	329
4.6.10. UnSubscribeMarketData 方法	329
4.6.11. SubscribeForQuoteRsp 方法	329
4.6.12. UnSubscribeForQuoteRsp 方法	330
4.6.13. ReqUserLogin 方法	330
4.6.14. ReqUserLogout 方法	331

4.6.15. RegisterFensUserInfo 方法	331
4.6.16. RegisterNameServer 方法	332
4.6.17. LoadExtMdApi 方法	332
4.7. CKSCosSpi 接口	333
4.7.1. OnRspInitInsertConditionalOrder 方法	333
4.7.2. OnRspQueryConditionalOrder 方法	335
4.7.3. OnRspModifyConditionalOrder 方法	337
4.7.4. OnRspPauseConditionalOrder 方法	339
4.7.5. OnRspRemoveConditionalOrder 方法	341
4.7.6. OnRspSelectConditionalOrder 方法	341
4.7.7. OnRspInsertProfitAndLossOrder 方法	342
4.7.8. OnRspModifyProfitAndLossOrder 方法	344
4.7.9. OnRspRemoveProfitAndLossOrder 方法	346
4.7.10. OnRspQueryProfitAndLossOrder 方法	347
4.7.11. OnRtnCOSAskSelect 方法	349
4.7.12. OnRtnCOSStatus 方法	350
4.7.13. OnRtnPLStatus 方法	352
4.8. CKSCosApi 接口	354
4.8.1. ReqInitInsertConditionalOrder 方法	354
4.8.2. ReqQueryConditionalOrder 方法	356
4.8.3. ReqModifyConditionalOrder 方法	356
4.8.4. ReqRemoveConditionalOrder 方法	358
4.8.5. ReqStateAlterConditionalOrder 方法	359
4.8.6. ReqSelectConditionalOrder 方法	359

4.8.7. ReqInsertProfitAndLossOrder 方法	360
4.8.8. ReqModifyProfitAndLossOrder 方法	361
4.8.9. ReqRemoveProfitAndLossOrder 方法	362
4.8.10. ReqQueryProfitAndLossOrder 方法	363
4.9. CKSOptionSpi 接口	364
4.9.1. OnRspQryIndexPrice 方法	364
4.9.2. OnRspQryOptionInstrGuard 方法	365
4.9.3. OnRspQryExecOrderVolume 方法	365
4.9.4. OnRspQryStockOptionAssignment 方法	366
4.9.5. OnRspQryInvestorTradeLevel 方法	368
4.9.6. OnRspQryPurchaseLimitAmt 方法	368
4.9.7. OnRspQryPositionLimitVol 方法	369
4.9.8. OnRspQryPurchaseLimitAmtS 方法	370
4.9.9. OnRspQryPositionLimitVolS 方法	371
4.9.10. OnRspPurchaseLimitAmtUpdate 方法	372
4.9.11. OnRspPositionLimitVolUpdate 方法	372
4.9.12. OnRspQryHistoryOrder 方法	373
4.9.13. OnRspQryHistoryTrade 方法	375
4.9.14. OnRspQryStockOptionHistoryAssignment 方法	377
4.9.15. OnRspQryStockOptionDelivDetail 方法	378
4.9.16. OnRspAutoExecOrderAction 方法	379
4.9.17. OnRspCombActionInsert 方法	380
4.9.18. OnRspQryInvestorCombinePosition 方法	381
4.9.19. OnRspQryCombActionVolume 方法	383

4.9.20. OnRspQuoteUpdate 方法	383
4.9.21. OnRtnCombAction 方法	384
4.10. CKSOptionApi 接口	387
4.10.1. ReqQryIndexPrice 方法	387
4.10.2. ReqQryOptionInstrGuard 方法	387
4.10.3. ReqQryExecOrderVolume 方法	388
4.10.4. ReqQryStockOptionAssignment 方法	389
4.10.5. ReqQryInvestorTradeLevel 方法	389
4.10.6. ReqQryPurchaseLimitAmtT 方法	390
4.10.7. ReqQryPositionLimitVolT 方法	390
4.10.8. ReqQryPurchaseLimitAmtS 方法	391
4.10.9. ReqQryPositionLimitVolS 方法	391
4.10.10. ReqPurchaseLimitAmtUpdate 方法	392
4.10.11. ReqPositionLimitVolUpdate 方法	392
4.10.12. ReqQryHistoryOrder 方法	393
4.10.13. ReqQryHistoryTrade 方法	394
4.10.14. ReqQryStockOptionHistoryAssignment 方法	395
4.10.15. ReqQryStockOptionDelivDetail 方法	395
4.10.16. ReqAutoExecOrderAction 方法	396
4.10.17. ReqCombActionInsert 方法	397
4.10.18. ReqQryInvestorCombinePosition 方法	398
4.10.19. ReqQryCombActionVolume 方法	399
4.10.20. ReqQuoteUpdate 方法	400
4.11. CKSVocSpi 接口	401

4.11.1. OnRspQryInvestorOpenPosition 方法	401
4.11.2. OnRspQryInvestorOpenCombinePosition 方法	403
4.11.3. OnRspBulkCancelOrder 方法	405
4.11.4. OnRspQryCloseStrategy 方法	406
4.11.5. OnRspQryCombStrategy 方法	406
4.11.6. OnRspQryOptionCombStrategy 方法	409
4.11.7. OnRspQryTransferInfo 方法	411
4.11.8. OnRspQryKSTradingNotice 方法	412
4.11.9. OnRtnKSTradingNotice 方法	413
4.11.10. OnRspQryUserProductUrl 方法	414
4.11.11. OnRspQryMaxCombActionVolume 方法	415
4.11.12. OnRspQryKSInstrCommRate 方法	415
4.11.13. OnRspQryKSETFOptionInstrCommRate 方法	416
4.11.14. OnRspQryKSIInstrumentMarginRate 方法	418
4.11.15. OnRspWithdrawCreditApply 方法	419
4.11.16. OnRspQryWithdrawCredit 方法	419
4.11.17. OnRspQryWithdrawCreditApply 方法	420
4.11.18. OnRspQryKSOOptionMargin 方法	421
4.11.19. OnRspQryKSTradingParams 方法	422
4.11.20. OnRspQryKSTradingAccount 方法	422
4.11.21. OnRspQryWithdrawReservedAccount 方法	425
4.11.22. OnRspWithdrawReservedApply 方法	426
4.11.23. OnRspWithdrawReservedCancel 方法	427
4.11.24. OnRspQryWithdrawReservedApply 方法	428

4.11.25. OnRspOptionSelfCloseAction 方法	429
4.11.26. OnRspQryDepositAndWithdraw 方法	430
4.11.27. OnRspQryInvestorPortfMarginRatio 方法	431
4.11.28. OnRspQryInvestorSPMMCombDiscountParam 方法	432
4.11.29. OnRspQryInvestorProdRULEDetail 方法	433
4.12. CKSVocApi 接口	435
4.12.1. ReqQueryInvestorOpenPosition 方法	435
4.12.2. ReqQueryInvestorOpenCombinePosition 方法	436
4.12.3. ReqBulkCancelOrder 方法	436
4.12.4. ReqQryCloseStrategy 方法	437
4.12.5. ReqQryCombStrategy 方法	437
4.12.6. ReqQryOptionCombStrategy 方法	438
4.12.7. ReqQryTransferInfo 方法	439
4.12.8. ReqQryKSTradingNotice 方法	439
4.12.9. ReqQryUserProductUrl 方法	440
4.12.10. ReqQryMaxCombActionVolume 方法	441
4.12.11. ReqQryKSInstrCommRate 方法	441
4.12.12. ReqQryKSETFOptionInstrCommRate 方法	442
4.12.13. ReqQryKSIInstrumentMarginRate 方法	443
4.12.14. ReqWithdrawCreditApply 方法	443
4.12.15. ReqQryWithdrawCredit 方法	444
4.12.16. ReqQryWithdrawCreditApply 方法	444
4.12.17. ReqKSEMailLogin 方法	445
4.12.18. ReqQryKSOOptionMargin 方法	445

4.12.19. ReqQryKSTradingParams 方法	446
4.12.20. ReqQryKSTradingAccount 方法	446
4.12.21. ReqQryWithdrawReservedAccount 方法	447
4.12.22. ReqWithdrawReservedApply 方法	448
4.12.23. ReqWithdrawReservedCancel 方法	448
4.12.24. ReqQryWithdrawReservedApply 方法	449
4.12.25. ReqOptionSelfCloseAction 方法	450
4.12.26. ReqQryDepositAndWithdraw 方法	451
4.12.27. ReqQryInvestorPortfMarginRatioAddClientID 方法	452
4.12.28. ReqQryInvestorSPMMCombDiscountParam 方法	452
4.12.29. ReqQryInvestorProdRULEDetail 方法	453
4.13. CKSVocMdSpi 接口	453
4.13.1. OnRspSubKSMarketData 方法	453
4.13.2. OnRspUnSubKSMarketData 方法	454
4.14. CKSVocMdApi 接口	455
4.14.1. SubscribeKSMarketData 方法	455
4.14.2. UnSubscribeKSMarketData 方法	455
4.14.3. ReqKSEMailLogin 方法	456
5. 第五章 示例代码	457
6. 第六章 Q&A 及问题反馈方式	457
6.1. 多实例的编程要求	457

1. 第一章 介绍

1.1. 概述

本手册为 Kingstar API 与 Kingstar 服务器之间的通讯数据包接口规范。通过 API，投资者可以接收来自上交所，大商所、郑商所和中金所的行情数据，并发送交易指令，接收相应的反馈和交易状态等信息。Kingstar API 将实现兼容上期技术综合交易平台 CTP 的 API。

1.2. API 文件

Kingstar 平台上使用的 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实现客户端和 Kingstar 服务器之间的数据传输。通过 API，客户端可以发出或撤销普通单、查询委托或交易状态、查询账户实时信息和交易头寸。API 程序库根据不同的操作系统平台所包含文件有所不同：

Windows 平台 API 包含文件如下：

库文件名	库文件描述
KSTraderApiEx.h	交易接口头文件
KSTradeAPI.h	
KSMdApiEx.h	行情接口头文件
KSMarketDataAPI.h	
KSUserApiDataTypeEx.h	定义了 API 所需的一系列数据类型的头文件
KSUserApiStructEx.h	定义了一系列业务相关的数据结构的头文件
KSCosApi.h	条件单接口头文件
KSOptionApi.h	期权单接口头文件
KSVocApi.h	客户需求接口头文件(Voice Of Customer)
KSVocApiDataType.h	定义了 Cos、Option、Voc API 所需的一系列数据类型的头文件
KSVocApiStruct.h	定义了一系列 Cos、Option、Voc 业务相关的数据结构的头文件
KSTradeAPI.dll	交易 API 动态链接库
KSMarketDataAPI.lib	行情 API 导入库文件
KSMarketDataAPI.dll	行情 API 动态链接库
lkcdll.dll	交易行情 API 运行需附加的动态链接库

ksPortalAPI.dll	门户集成功能接口库
SSPXEncode.dll	Sspx 协议编码接口库
KSIInterB2C.lkc	客户端授权文件

Linux 平台 API 包含文件如下：

库文件名	库文件描述
KSTraderApiEx.h	交易接口头文件
KSTradeAPI.h	
KSMdApiEx.h	行情接口头文件
KSMarketDataAPI.h	
KSUserApiDataTypeEx.h	定义了 API 所需的一系列数据类型的头文件
KSUserApiStructEx.h	定义了一系列业务相关的数据结构的头文件
KSCosApi.h	条件单接口头文件
KSOptionApi.h	期权单接口头文件
KSVocApi.h	客户需求接口头文件(Voice Of Customer)
KSVocApiDataType.h	定义了 Cos、Option、Voc API 所需的一系列数据类型的头文件
KSVocApiStruct.h	定义了一系列 Cos、Option、Voc 业务相关的数据结构的头文件
libkstradeapi.so	交易 API 动态链接库
libksmarketdataapi.so	行情 API 动态链接库
libkslkc64r.so 、 libkslkc32r.so	授权文件验证动态链接库
KSIInterB2C.lkc	客户端授权文件

注：编程请使用头文件命名空间 **namespace KingstarAPI**。

条件单头文件前缀“**KSCos**”术语解释为：**kingstar condition order system**

2. 第二章 体系结构

2.1. 通讯模式

金仕达的 API 和 Kingstar 服务器之间使用的通讯协议是期货交易数据协议（futures TradingData Exchange protocol，FTD），它基于 TCP 协议。

在 FTD 协议中，通讯模式包括以下三种模式：

对话模式，客户端给 Kingstar 服务器发送请求，该请求被交易所端接收和处理，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。

私有模式，Kingstar 服务器把特定的私人信息发送给对应的客户端，包括持仓信息、交易确认信息等。

广播模式，Kingstar 服务器将把公告等信息发送给所有的注册用户。

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。

2.2. 数据流

Kingstar 服务器提供对话、私有、广播等三种通讯模式。

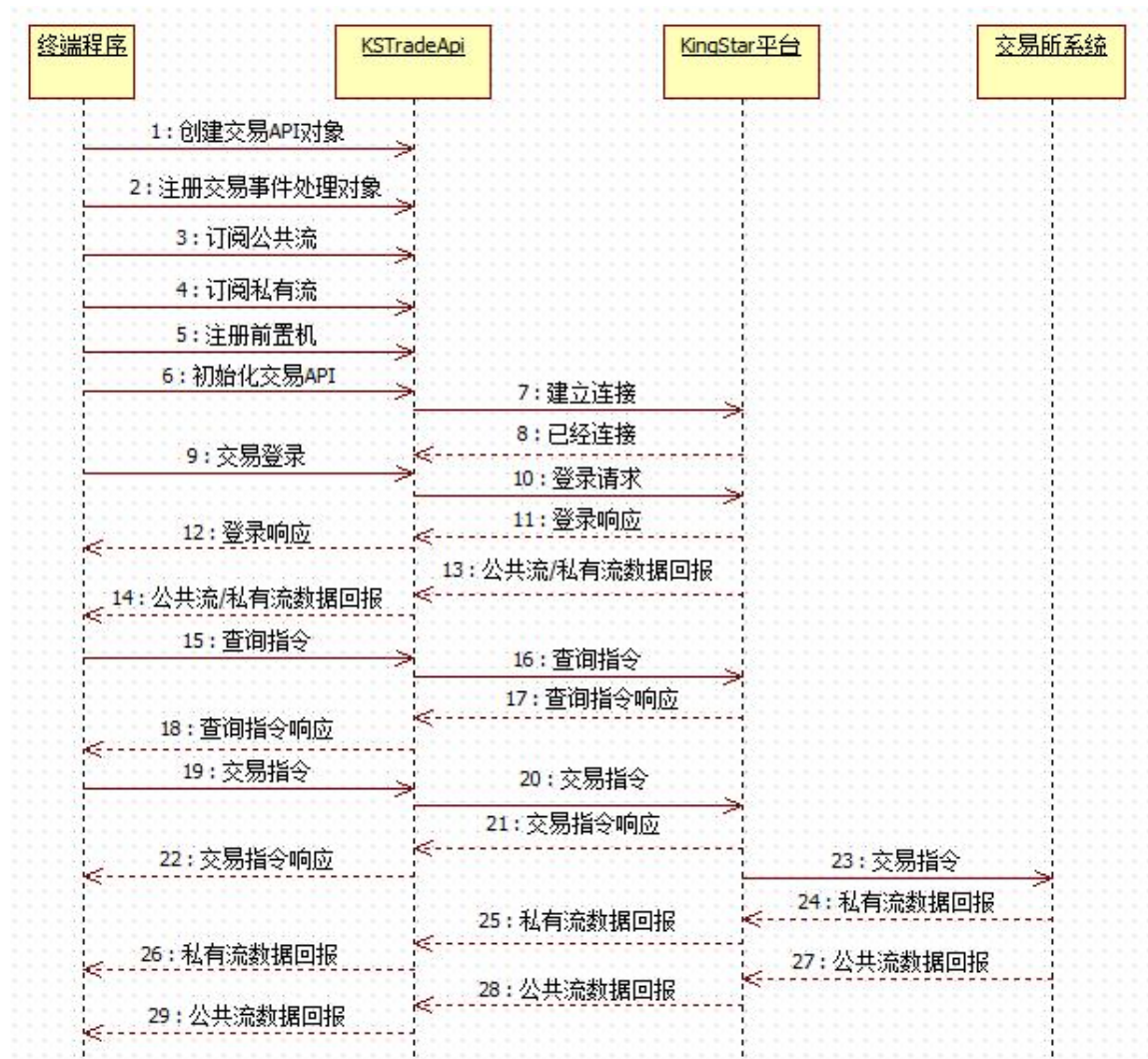
在对话模式中，传输的是对话数据流和查询数据流。对话和查询数据流是双向数据流，客户端发出请求，Kingstar 服务器返回结果。Kingstar 服务器并不保存对话和查询数据流。当故障发生时，比如连接中断后又重新连接，对话和查询数据流将会重置，通讯途中的数据可能会丢失。

在私人通讯模式中，传输的是私人数据流。私人数据流是单向数据流，Kingstar 服务器就是利用它来把相应的私人信息发给提出申请的客户端。私人信息包括，风险提示、指令状态、指令确认、交易确认等。私人数据流是可靠的。当客户端和 Kingstar 服务器失去连接后，在同一交易日的任何时间，客户端都可重新连上 Kingstar 服务器，获取一系列指定的私人信息而不用担心这些数据会丢失。

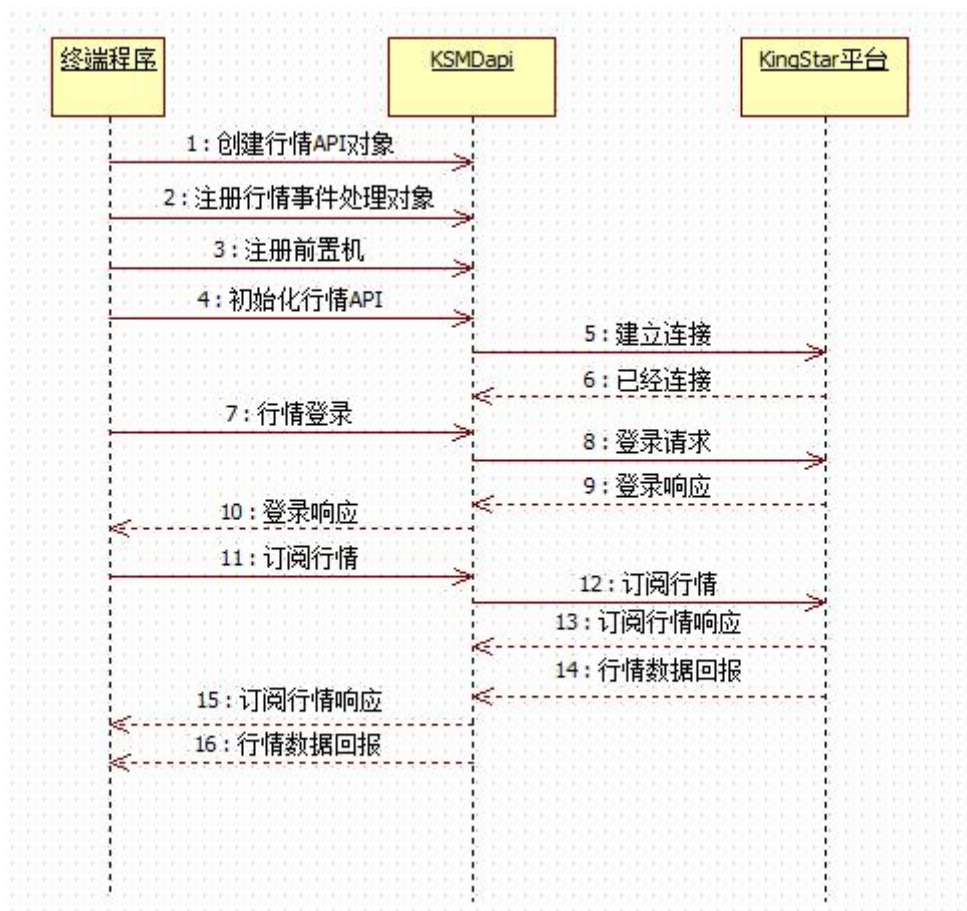
在广播通讯模式中，传输的是公共数据流。和私人数据流一样，它是单向数据流，而且可靠。区别在于，广播通讯数据会发送到所有连接的客户端上。它的主要用途就是发布公共设备的状态信息或重要的公共信息。

2.3. 通讯流程

2.3.1. 交易流程

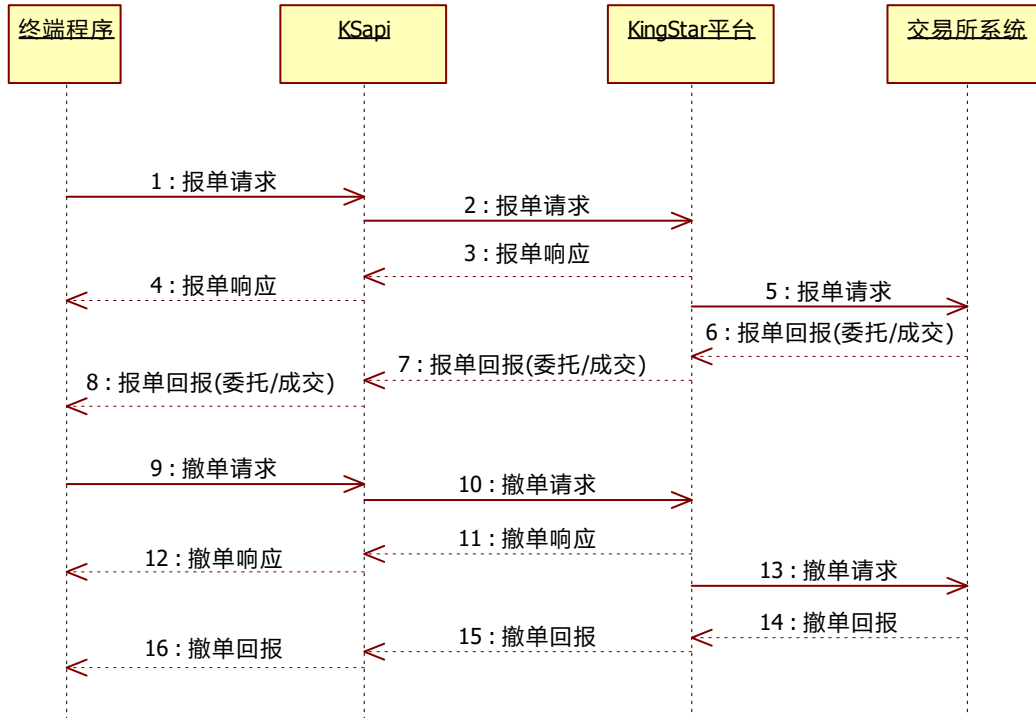


2.3.2. 行情流程



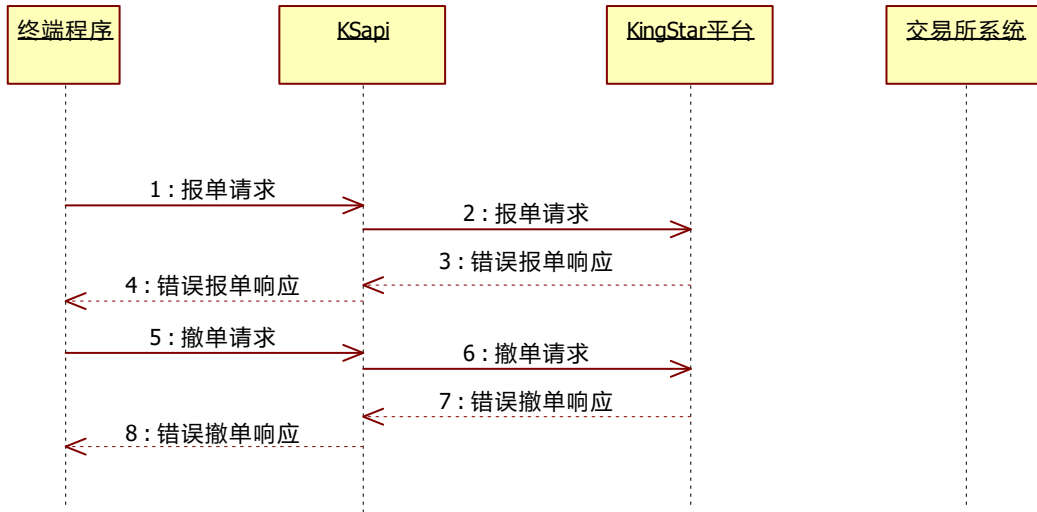
2.4. 报单指令（特别说明）

2.4.1. 指令正常

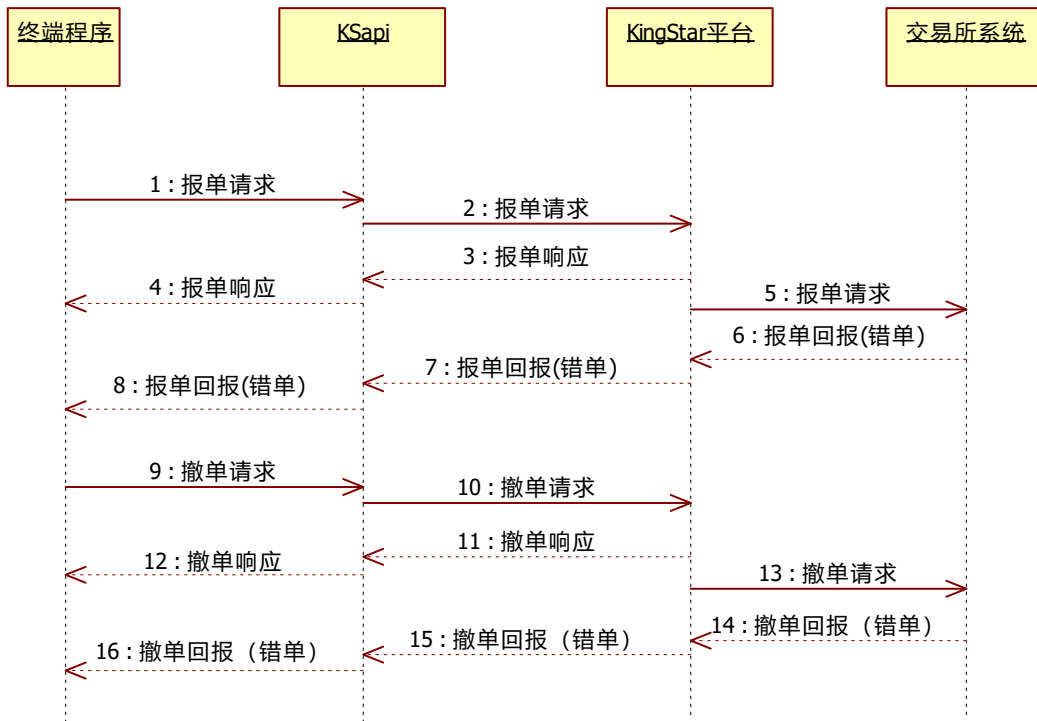


2.4.2. 指令异常

当终端程序发出报单、撤单指令错误时，分两种情况：图（一）显示指令填写错误，Kingstar 平台会在送交交易所之前进行检测给出错误提示；图（二）显示交易所系统对所发出的指令进行处理，当不符合交易条件时，交易所给出错误提示。



(一)



(二)

3. 第三章 接口模式

Kingstar 的交易 API 提供了两种编程接口：CThostFtdcTraderApi 与 CThostFtdcTraderSpi。行情 API 提供的是 CThostFtdcMdApi 与 CThostFtdcMdSpi 等两种编程接口。这四个接口对 FTD 协议进行了封装，方便客户端应用程序的开发。客户端可使用 CThostFtdcXXXApi 来发出操作请求，并通过继承 CThostFtdcXXXSpi 并重载回调函数来处理 Kingstar 服务器的响应。

3.1. 对话模式的编程接口

对话模式的通讯函数常常这样定义：

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

        int CThostFtdcMdApi::ReqXXX(
        CThostFtdcXXXField *pReqXXX,
        int nRequestID)

响应:  void CThostFtdcTraderSpi::OnRspXXX(
        CThostFtdcXXXField *pRspXXX,
        CThostFtdcRspInfoField *pRspInfo,
        int nRequestID,
        bool bIsLast)

        void CThostFtdcMdSpi::OnRspXXX(
        CThostFtdcXXXField *pRspXXX,
        CThostFtdcRspInfoField *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不要重复。在接收 Kingstar 服务器的响应时，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。

当客户端收到 Kingstar 服务器发出的反馈时，回调函数 CThostFtdcXXXSpi 被激活。如果反馈多于一条，回调函数 CThostFtdcXXXSpi 会反复被激活，直到数据全部接收完毕。

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。

最后一个参数是结束标识，其值为“true”时表示进行的是该请求的最后一个反馈。

3.2. 私有模式的编程接口

下面的例子为私有模式的常用接口：

```
void CThostFtdcTraderSpi::OnRtnXXX(CThostFtdcXXXField *pXXX)
void CThostFtdcTraderSpi::OnErrRtnXXX(CThostFtdcXXXField *pXXX,
                                         CThostFtdcRspInfoField *pRspInfo)
```

在私有模式里并没有连接 API 与 Kingstar 服务器的行情函数。当 Kingstar 服务器发出私有数据流时，CThostFtdcTradeSpi 的回调函数将被激活。所有回调函数的第一个参数都是 Kingstar 服务器返回的具体内容。OnErrRtnCThostFtdcTradeSpi 函数的第二个参数是报错时详细的错误信息。

3.3. 广播模式的编程接口

使用广播模式时，客户端可以用以下两种方式与 Kingstar 服务器进行通讯：

```
void CThostFtdcTraderSpi::OnRtnInstrumentStatus(
    CThostFtdcInstrumentStatusField *pInstrumentStatus)
void CThostFtdcMdSpi::OnRtnDepthMarketData(
    CThostFtdcDepthMarketDataField *pDepthMarketData)
```

回调函数 OnRtnInstrumentStatus 用于通知客户端合约状态的变化。

回调函数 OnRtnDepthMarketData 用于公布最新的交易所行情数据。

4. 第四章 开发接口

4.1. 工作线程

客户端进程需要两种线程，一是应用程序主线程，另一种是交易 API 工作线程。如果客户端想要接收行情数据，那么也需要行情 API 工作线程。API 工作线程连接了客户端和 Kingstar 服务器。

交易 API 和行情 API 是安全线程，客户端的应用程序可以同时使用两种或多种的工作线程，而无须担心线程冲突。客户端的应用程序要能尽快的处理回调信息，这样才能避免未处理的回调信息堵塞工作线程。要避免通讯堵塞，客户端的应用程序需要使用缓冲层来储存从 Kingstar 服务器接收的数据，当然缓冲层也可以用来保护客户端自有数据，以便使之与 API 的数据区分开。

4.2. 通用规则

客户端应用程序需要经过两步才能连接到 Kingstar 服务器：初始化与功能启用。

使用交易 API，客户端交易应用程序需要编写：

(1) 创建一个“CThostFtdcTraderApi”实例。

(2) 创建一个处理来自“CThostFtdcTraderSpi”接口的事件处理器，然后使用“CThostFtdcTraderApi”的“RegisterSpi”函数记录下这些事件。

(3) 使用“CThostFtdcTraderApi”的“SubscribePrivateTopic”函数处理私有数据流。

(4) 使用“CThostFtdcTraderApi”的“SubscribePublicTopic”函数处理公共数据流。

(5) 使用“CThostFtdcTraderApi”的“RegisterFront”函数记录 Kingstar 服务器的前端地址。客户端多运行几次这种函数，以便与服务器建立更可靠的联系。强烈建议。

(6) 使用“CThostFtdcTraderApi”的“Init”函数来连接 Kingstar 服务器。

(7) Kingstar 服务器连上之后，“CThostFtdcTraderSpi”接口的回调函数“OnFrontConnected”将被激活。函数运行过程中，对于非国密客户端，应用程序需要使用“CThostFtdcTraderApi”的“ReqAuthenticate”函数来提交认证请求，在返回的 OnRspAuthenticate 中继续调用“ReqUserLogin”函数来提交 login 登陆认证请求，对于国密用户，直接调用“ReqUserLoginSM”函数提交国密认证请求即可，里面已经包含了相关的认证登陆请求。

(8) 当 Kingstar 服务器确认登陆成功后，“CThostFtdcTraderSpi”接口回调函数“OnRspUserLogin”将被激活。

(9) 这样，客户端与 Kingstar 服务器的通讯就建立起来了。客户端交易应用程序可以使用其他金仕达 API 业务来与 Kingstar 服务器进行通讯。

使用行情 API，客户端行情应用程序需要编写：

(1) 创建一个“CThostFtdcMdApi”实例。

(2) 创建一个处理来自“CThostFtdcMdSpi”接口的事件处理器，然后使用“CThostFtdcMdApi”的“RegisterSpi”函数记录下这些事件，如果订阅行情合约的个数大于等于 2000 的时候，需要再开启新的行情 API 实例（需要多实例订阅）。

(3) 使用“CThostFtdcTraderApi”的“Init”函数来连接 Kingstar 服务器。

(4) Kingstar 服务器连上之后，“CThostFtdcTraderSpi”接口的回调函数“OnFrontConnected”将被激活。函数运行过程中，客户端的应用程序需要使用“CThostFtdcMdApi”的“ReqUserLogin”函数来提交“login”请求。

(5) 当 Kingstar 服务器确认登陆成功后，“CThostFtdcMdSpi”接口回调函数“OnRspUserLogin”将被激活。

(6) 这样，客户端与 Kingstar 服务器的通讯就建立起来了，客户端行情应用程序就可以进行行情的订阅与退订了。

使用条件单/期权单/Voc API，客户端交易应用程序需要编写：

(1) 通过继承条件单/期权单/Voc API 头文件中的 CKSCosSpi/CKSOptionSpi/CKSVocSpi 接口类，实现条件单/期权单/Voc API 回调实例类。

- (2) 条件单/期权单/Voc API 的注册，需要交易 API 的创建。故首先要创建一个“CThostFtdcTraderApi”实例
- (3) 通过条件单/期权单/Voc API 回调实例类，定义应答回调实例。
- (4) 通过交易 API 的 LoadExtApi 请求接口注册条件单/期权单/Voc API（ExtApiName 分别对应为"KSCosApi"/ "KSOOptionApi"/ "KSVocApi"）。此函数返回指向条件单/期权单/客户需求 API（Voc API）实例的指针。
- (5) 调用条件单/期权单/Voc API 接口中的函数。（注意，指令的调用需要在交易登陆之后进行，否则将会出现客户尚未登陆的报错信息）。

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，其他表示返回错误。

4.3. CThostFtdcTraderSpi 接口

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

4.3.1. OnFrontConnected 方法

当客户端与 Kingstar 服务器建立起通信连接时（还未登录前），该方法被调用。

函数原型：

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。

4.3.2. OnFrontDisconnected 方法

当客户端与 Kingstar 服务器通信连接断开时，该方法被调用。

函数原型：

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

0x2004 服务器主动断开

4.3.3. OnHeartBeatWarning 方法[暂不支持]

心跳超时警告。当长时间未收到报文时，该方法被调用。

函数原型：

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

参数：

nTimeLapse: 距离上次接收报文的时间

4.3.4. OnRspUserLogin 方法

当客户端发出登录请求之后，Kingstar 服务器返回响应时，该方法会被调用，通知客户端登录是否成功。

函数原型：

```
void OnRspUserLogin(
    CThostFtdcRspUserLoginField *pRspUserLogin,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pRspUserLogin: 返回用户登录信息的地址。

用户登录信息结构：

```
struct CThostFtdcRspUserLoginField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///登录成功时间
    TThostFtdcTimeType    LoginTime;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///交易系统名称
    TThostFtdcSystemNameType    SystemName;
    ///前置编号
    TThostFtdcFrontIDType    FrontID;
```

```

    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///最大报单引用
    TThostFtdcOrderRefType    MaxOrderRef;
    ///上期所时间
    TThostFtdcTimeType    SHFETime;
    ///大商所时间
    TThostFtdcTimeType    DCETime;
    ///郑商所时间
    TThostFtdcTimeType    CZCETime;
    ///中金所时间
    TThostFtdcTimeType    FFEXTime;
};

```

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

响应信息结构:

```

struct CThostFtdcRspInfoField
{
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    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。

4.3.5. OnRspUserLogout 方法

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

函数原型:

```

void OnRspUserLogout(
    CThostFtdcUserLogoutField *pUserLogout,
    CThostFtdcRspInfoField *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 CThostFtdcUserLogoutField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///用户代码
    TThostFtdcUserIDType   UserID;
};
```

4.3.6. 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，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
void OnRspUserPasswordUpdate(
    CThostFtdcUserPasswordUpdateField *pUserPasswordUpdate,
    CThostFtdcRspInfoField *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 CThostFtdcUserPasswordUpdateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///用户代码
    TThostFtdcUserIDType   UserID;
    ///原来的口令
    TThostFtdcPasswordType   OldPassword;
```

```

        ///新的口令
        TThostFtdcPasswordType    NewPassword;
    };

```

4.3.7. OnRspTradingAccountPasswordUpdate 方法

资金账户口令更新应答。当客户端发出资金账户口令更新指令后，Kingstar返回响应时，该方法会被调用。

函数原型：

```

void OnRspTradingAccountPasswordUpdate(
    CThostFtdcTradingAccountPasswordUpdateField *pTradingAccountPasswordUpdate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pTradingAccountPasswordUpdate: 指向资金账户口令变更域结构的地址，包含了用户密码修改请求的输入数据。

资金账户口令变更域结构：

```

struct CThostFtdcTradingAccountPasswordUpdateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者帐号
    TThostFtdcAccountIDType    AccountID;
    ///原来的口令
    TThostFtdcPasswordType    OldPassword;
    ///新的口令
    TThostFtdcPasswordType    NewPassword;
};

```

4.3.8. OnRspError 方法

针对用户请求的出错通知。

函数原型：

```

void OnRspError(
    CThostFtdcRspInfoField *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 CThostFtdcRspInfoField
```

```
{
```

```
    ///错误代码
```

```
    TThostFtdcErrorIDType ErrorID;
```

```
    ///错误信息
```

```
    TThostFtdcErrorMsgType    ErrorMsg;
```

```
};
```

4.3.9. OnRspOrderInsert 方法

报单录入应答。当客户端发出过报单录入指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
void OnRspOrderInsert(
    CThostFtdcInputOrderField *pInputOrder,
    CThostFtdcRspInfoField *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 CThostFtdcInputOrderField
```

```
{
```

```
    ///经纪公司代码
```

```
    TThostFtdcBrokerIDType    BrokerID;
```

```
    ///投资者代码
```

```
    TThostFtdcInvestorIDType    InvestorID;
```

```
    ///合约代码
```

```

TThostFtdcInstrumentIDType InstrumentID;
///报单引用

TThostFtdcOrderRefType    OrderRef;
///用户代码

TThostFtdcUserIDType  UserID;
///报单价格条件

TThostFtdcOrderPriceTypeType  OrderPriceType;
///买卖方向

TThostFtdcDirectionType    Direction;
///组合开平标志

TThostFtdcCombOffsetFlagType  CombOffsetFlag;
///组合投机套保标志

TThostFtdcCombHedgeFlagType  CombHedgeFlag;
///价格

TThostFtdcPriceType    LimitPrice;
///数量

TThostFtdcVolumeType  VolumeTotalOriginal;
///有效期类型

TThostFtdcTimeConditionType    TimeCondition;
///GTD 日期 （保留字段）

TThostFtdcDateType    GTDDate;
///成交量类型

TThostFtdcVolumeConditionType  VolumeCondition;
///最小成交量

TThostFtdcVolumeType  MinVolume;
///触发条件

TThostFtdcContingentConditionType  ContingentCondition;
///止损价

TThostFtdcPriceType    StopPrice;
///强平原因

TThostFtdcForceCloseReasonType    ForceCloseReason;
///自动挂起标志

```

```

    TThostFtdcBoolType    IsAutoSuspend;
    ///业务单元

    TThostFtdcBusinessUnitType BusinessUnit;
    ///请求编号

    TThostFtdcRequestIDType RequestID;
    ///用户强评标志

    TThostFtdcBoolType    UserForceClose;
    ///互换单标志

    TThostFtdcBoolType    IsSwapOrder;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.10. OnRspOrderAction 方法

报单操作应答。当客户端发出过报单操作指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspOrderAction(
    CThostFtdcOrderActionField *pOrderAction,
    CThostFtdcRspInfoField *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 CThostFtdcOrderActionField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType InvestorID;
    ///报单操作引用

```

```

    TThostFtdcOrderActionRefType OrderActionRef;
    ///报单引用

    TThostFtdcOrderRefType OrderRef;
    ///请求编号

    TThostFtdcRequestIDType RequestID;
    ///前置编号

    TThostFtdcFrontIDType FrontID;
    ///会话编号

    TThostFtdcSessionIDType SessionID;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
    ///报单编号

    TThostFtdcOrderSysIDType OrderSysID;
    ///操作标志

    TThostFtdcActionFlagType ActionFlag;
    ///价格

    TThostFtdcPriceType LimitPrice;
    ///数量变化

    TThostFtdcVolumeType VolumeChange;
    ///用户代码

    TThostFtdcUserIDType UserID;
    ///合约代码

    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.3.11. OnRspQueryMaxOrderVolume 方法

查询最大报单数量应答。当客户端发出查询最大报单数量指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQueryMaxOrderVolume(
    CThostFtdcQueryMaxOrderVolumeField *pQueryMaxOrderVolume,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,

```



```
bool bIsLast);
```

参数:

pQueryMaxOrderVolume: 指向查询最大报单数量结构的地址, 包含了最大允许报单数量。

最大报单数量结构:

```
struct CThostFtdcQueryMaxOrderVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///买卖方向
    TThostFtdcDirectionType   Direction;
    ///开平标志
    TThostFtdcOffsetFlagType   OffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///最大允许报单数量
    TThostFtdcVolumeType       MaxVolume;
    ///交易所代码
    TThostFtdcExchangeIDType   ExchangeID;
};
```

4.3.12. OnRspExecOrderInsert 方法

执行宣告录入应答。当客户端发出执行宣告录入指令后, Kingstar 服务器返回响应时, 该方法会被调用。

函数原型:

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

参数:

pInputExecOrder: 指向输入的执行宣告结构体的地址, 包含执行宣告引用和执行类型。

输入的执行宣告结构:

```
struct CThostFtdcInputExecOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///执行宣告引用
    TThostFtdcOrderRefType    ExecOrderRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///数量
    TThostFtdcVolumeType    Volume;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///开平标志
    TThostFtdcOffsetFlagType    OffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///执行类型
    TThostFtdcActionTypeType    ActionType;
    ///保留头寸申请的持仓方向
    TThostFtdcPosiDirectionType    PosiDirection;
    ///期权行权后是否保留期货头寸的标记
    TThostFtdcExecOrderPositionFlagType    ReservePositionFlag;
    ///期权行权后生成的头寸是否自动平仓
```

```

    TThostFtdcExecOrderCloseFlagType CloseFlag;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.13. OnRspExecOrderAction 方法

执行宣告操作应答。当客户端发出执行宣告操作指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspExecOrderAction(
    CThostFtdcInputExecOrderActionField *pInputExecOrderAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInputExecOrderAction： 指向输入的执行宣告操作结构体的地址，包含执行宣告操作引用和执行宣告引用。

输入执行宣告操作结构：

```

struct CThostFtdcInputExecOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///执行宣告操作引用
    TThostFtdcOrderActionRefType ExecOrderActionRef;
    ///执行宣告引用
    TThostFtdcOrderRefType ExecOrderRef;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///执行宣告操作编号
    TThostFtdcExecOrderSysIDType ExecOrderSysID;
}

```

```

    ///操作标志
    TThostFtdcActionFlagType  ActionFlag;
    ///用户代码
    TThostFtdcUserIDType  UserID;
    ///合约代码
    TThostFtdcInstrumentIDType  InstrumentID;
};

```

4.3.14. OnRspSettlementInfoConfirm 方法

投资者结算结果确认应答。当客户端发出投资者结算结果确认指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspSettlementInfoConfirm(
    CThostFtdcSettlementInfoConfirmField *pSettlementInfoConfirm,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pSettlementInfoConfirm: 指向投资者结算结果确认信息结构的地址，包含了最大允许报单数量。

投资者结算结果确认信息结构：

```

struct CThostFtdcSettlementInfoConfirmField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType  BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///确认日期
    TThostFtdcDateType  ConfirmDate;
    ///确认时间
    TThostFtdcTimeType  ConfirmTime;
};

```

4.3.15. OnRspFromBankToFutureByFuture 方法

请求银行资金转期货响应。当客户端发出请求银行资金转期货指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspFromBankToFutureByFuture (
    CThostFtdcReqTransferField *pReqTransfer,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pReqTransfer: 指向银行资金转期货请求响应结构的地址。

银行资金转期货请求响应结构：

```
struct CThostFtdcReqTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;
```

```
///银期平台消息流水号
TThostFtdcSerialType PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType LastFragment;
///会话号
TThostFtdcSessionIDType SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
```

```
///转帐金额
TThostFtdcTradeAmountType    TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType    FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType    FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType    CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType    BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType    Message;
///摘要
TThostFtdcDigestType    Digest;
///银行帐号类型
TThostFtdcBankAccTypeType    BankAccType;
///渠道标志
TThostFtdcDeviceIDType    DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType    BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType    BrokerIDByBank;
///期货单位帐号
TThostFtdcBankAccountType    BankSecuAcc;
///银行密码标志
TThostFtdcPwdFlagType    BankPwdFlag;
///期货资金密码核对标志
TThostFtdcPwdFlagType    SecuPwdFlag;
///交易柜员
TThostFtdcOperNoType    OperNo;
///请求编号
TThostFtdcRequestIDType    RequestID;
```

```

    ///交易 ID
    TThostFtdcTIDType    TID;

    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;

};

```

4.3.16. OnRspFromFutureToBankByFuture 方法

请求期货资金转银行响应。当客户端发出请求期货资金转银行指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspFromFutureToBankByFuture (
    CThostFtdcReqTransferField *pReqTransfer,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pReqTransfer： 指向期货资金转银行请求响应结构的地址。

期货资金转银行请求响应结构：

```

struct CThostFtdcReqTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;

    ///银行代码
    TThostFtdcBankIDType    BankID;

    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;

    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;

    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;

    ///交易日期
    TThostFtdcTradeDateType    TradeDate;

    ///交易时间

```



```
TThostFtdcTradeTimeType TradeTime;
///银行流水号

TThostFtdcBankSerialType BankSerial;
///交易系统日期

TThostFtdcTradeDateType TradingDay;
///银期平台消息流水号

TThostFtdcSerialType PlateSerial;
///最后分片标志

TThostFtdcLastFragmentType LastFragment;
///会话号

TThostFtdcSessionIDType SessionID;
///客户姓名

TThostFtdcIndividualNameType CustomerName;
///证件类型

TThostFtdcIdCardTypeType IdCardType;
///证件号码

TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型

TThostFtdcCustTypeType CustType;
///银行帐号

TThostFtdcBankAccountType BankAccount;
///银行密码

TThostFtdcPasswordType BankPassWord;
///投资者帐号

TThostFtdcAccountIDType AccountID;
///期货密码

TThostFtdcPasswordType Password;
///安装编号

TThostFtdcInstallIDType InstallID;
///期货公司流水号

TThostFtdcFutureSerialType FutureSerial;
///用户标识
```

```
TThostFtdcUserIDType UserID;
///验证客户证件号码标志

TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码

TThostFtdcCurrencyIDType CurrencyID;
///转帐金额

TThostFtdcTradeAmountType TradeAmount;
///期货可取金额

TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志

TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用

TThostFtdcCustFeeType CustFee;
///应收期货公司费用

TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息

TThostFtdcAddInfoType Message;
///摘要

TThostFtdcDigestType Digest;
///银行帐号类型

TThostFtdcBankAccTypeType BankAccType;
///渠道标志

TThostFtdcDeviceIDType DeviceID;
///期货单位帐号类型

TThostFtdcBankAccTypeType BankSecuAccType;
///期货公司银行编码

TThostFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号

TThostFtdcBankAccountType BankSecuAcc;
///银行密码标志

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

```

    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员

    TThostFtdcOperNoType    OperNo;
    ///请求编号

    TThostFtdcRequestIDType    RequestID;
    ///交易ID

    TThostFtdcTIDType    TID;
    ///转账交易状态

    TThostFtdcTransferStatusType    TransferStatus;
};

```

4.3.17. OnRspQueryBankAccountMoneyByFuture 方法

请求查询银行资金响应。当客户端发出请求查询银行资金指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQueryBankAccountMoneyByFuture (
    CThostFtdcReqQueryAccountField *pReqQueryAccount,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pReqQueryAccount: 指向查询银行资金请求响应结构的地址。

查询银行资金请求响应结构：

```

struct CThostFtdcReqQueryAccountField
{
    ///业务功能码

    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码

    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码

    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码

    TThostFtdcBrokerIDType    BrokerID;

```

```
///期商分支机构代码
TThostFtdcFutureBranchIDType BrokerBranchID;

///交易日期
TThostFtdcTradeDateType TradeDate;

///交易时间
TThostFtdcTradeTimeType TradeTime;

///银行流水号
TThostFtdcBankSerialType BankSerial;

///交易系统日期
TThostFtdcTradeDateType TradingDay;

///银期平台消息流水号
TThostFtdcSerialType PlateSerial;

///最后分片标志
TThostFtdcLastFragmentType LastFragment;

///会话号
TThostFtdcSessionIDType SessionID;

///客户姓名
TThostFtdcIndividualNameType CustomerName;

///证件类型
TThostFtdcIdCardTypeType IdCardType;

///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;

///客户类型
TThostFtdcCustTypeType CustType;

///银行帐号
TThostFtdcBankAccountType BankAccount;

///银行密码
TThostFtdcPasswordType BankPassWord;

///投资者帐号
TThostFtdcAccountIDType AccountID;

///期货密码
TThostFtdcPasswordType Password;
```

///期货公司流水号

TThostFtdcFutureSerialType *FutureSerial;*

///安装编号

TThostFtdcInstallIDType *InstallID;*

///用户标识

TThostFtdcUserIDType *UserID;*

///验证客户证件号码标志

TThostFtdcYesNoIndicatorType *VerifyCertNoFlag;*

///币种代码

TThostFtdcCurrencyIDType *CurrencyID;*

///摘要

TThostFtdcDigestType *Digest;*

///银行帐号类型

TThostFtdcBankAccTypeType *BankAccType;*

///渠道标志

TThostFtdcDeviceIDType *DeviceID;*

///期货单位帐号类型

TThostFtdcBankAccTypeType *BankSecuAccType;*

///期货公司银行编码

TThostFtdcBankCodingForFutureType *BrokerIDByBank;*

///期货单位帐号

TThostFtdcBankAccountType *BankSecuAcc;*

///银行密码标志

TThostFtdcPwdFlagType *BankPwdFlag;*

///期货资金密码核对标志

TThostFtdcPwdFlagType *SecuPwdFlag;*

///交易柜员

TThostFtdcOperNoType *OperNo;*

///请求编号

TThostFtdcRequestIDType *RequestID;*

///交易ID

TThostFtdcTIDType *TID;*

```
};
```

4.3.18. OnRspQryTransferSerial 方法

请求查询转账流水响应。当客户端发出请求查询转账流水指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryTransferSerial(
    CThostFtdcTransferSerialField *pTransferSerial,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pTransferSerial: 指向查询转账流水请求响应结构的地址。

查询转账流水请求响应结构：

```
struct CThostFtdcTransferSerialField
{
    ///平台流水号
    TThostFtdcPlateSerialType PlateSerial;
    ///交易发起方日期
    TThostFtdcTradeDateType TradeDate;
    ///交易日期
    TThostFtdcDateType TradingDay;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///交易代码
    TThostFtdcTradeCodeType TradeCode;
    ///会话编号
    TThostFtdcSessionIDType SessionID;
    ///银行编码
    TThostFtdcBankIDType BankID;
    ///银行分支机构编码
    TThostFtdcBankBrchIDType BankBranchID;
    ///银行帐号类型
```

```
TThostFtdcBankAccTypeType    BankAccType;
///银行帐号

TThostFtdcBankAccountType BankAccount;
///银行流水号

TThostFtdcBankSerialType    BankSerial;
///期货公司编码

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

TThostFtdcFutureBranchIDType BrokerBranchID;
///期货公司帐号类型

TThostFtdcFutureAccTypeType FutureAccType;
///投资者帐号

TThostFtdcAccountIDType    AccountID;
///投资者代码

TThostFtdcInvestorIDType    InvestorID;
///期货公司流水号

TThostFtdcFutureSerialType FutureSerial;
///证件类型

TThostFtdcIdCardTypeType    IdCardType;
///证件号码

TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///币种代码

TThostFtdcCurrencyIDType    CurrencyID;
///交易金额

TThostFtdcTradeAmountType    TradeAmount;
///应收客户费用

TThostFtdcCustFeeType CustFee;
///应收期货公司费用

TThostFtdcFutureFeeType    BrokerFee;
///有效标志

TThostFtdcAvailabilityFlagType AvailabilityFlag;
///操作员
```

```

    TThostFtdcOperatorCodeType    OperatorCode;
    ///新银行帐号

    TThostFtdcBankAccountType BankNewAccount;
    ///错误代码

    TThostFtdcErrorIDType ErrorID;
    ///错误信息

    TThostFtdcErrorMsgType    ErrorMsg;

};

```

4.3.19. OnRspQryAccountregister 方法

请求查询银期签约关系响应。当客户端发出请求查询银期签约关系指令之后，Kingstar 服务器发回响应时，该方法会被调用。

函数原型：

```

void OnRspQryAccountregister(
    CThostFtdcAccountregisterField *pAccountregister,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pAccountregister: 指向客户开销户信息表查询结构的地址。

客户开销户信息表查询结构：

```

struct CThostFtdcAccountregisterField
{
    ///交易日期
    TThostFtdcTradeDateType    TradeDay;
    ///银行编码
    TThostFtdcBankIDType BankID;
    ///银行分支机构编码
    TThostFtdcBankBrchIDType BankBranchID;
    ///银行帐号
    TThostFtdcBankAccountType BankAccount;
    ///期货公司编码
    TThostFtdcBrokerIDType    BrokerID;
    ///期货公司分支机构编码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///投资者帐号
    TThostFtdcAccountIDType    AccountID;
}

```



```

    ///证件类型
    TThostFtdcIdCardTypeType IdCardType;
    ///证件号码
    TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
    ///开销户类别
    TThostFtdcOpenOrDestroyType OpenOrDestroy;
    ///签约日期
    TThostFtdcTradeDateType RegDate;
    ///解约日期
    TThostFtdcTradeDateType OutDate;
    ///交易ID
    TThostFtdcTIDType TID;
    ///客户类型
    TThostFtdcCustTypeType CustType;
    ///银行帐号类型
    TThostFtdcBankAccTypeType BankAccType;
};

```

4.3.20. OnRspQryOrder 方法

报单查询请求。当客户端发出报单查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pOrder： 指向报单信息结构的地址。

报单信息结构：

```

struct CThostFtdcOrderField
{
    ///经纪公司代码

```

```
TThostFtdcBrokerIDType    BrokerID;
///投资者代码

TThostFtdcInvestorIDType   InvestorID;
///合约代码

TThostFtdcInstrumentIDType InstrumentID;
///报单引用

TThostFtdcOrderRefType     OrderRef;
///用户代码

TThostFtdcUserIDType       UserID;
///报单价格条件

TThostFtdcOrderPriceTypeType OrderPriceType;
///买卖方向

TThostFtdcDirectionType    Direction;
///组合开平标志

TThostFtdcCombOffsetFlagType CombOffsetFlag;
///组合投机套保标志

TThostFtdcCombHedgeFlagType CombHedgeFlag;
///价格

TThostFtdcPriceType         LimitPrice;
///数量

TThostFtdcVolumeType         VolumeTotalOriginal;
///有效期类型

TThostFtdcTimeConditionType  TimeCondition;
///GTD 日期

TThostFtdcDateType           GTDDate;
///成交量类型

TThostFtdcVolumeConditionType VolumeCondition;
///最小成交量

TThostFtdcVolumeType         MinVolume;
///触发条件

TThostFtdcContingentConditionType ContingentCondition;
///止损价
```

```
TThostFtdcPriceType    StopPrice;
///强平原因

TThostFtdcForceCloseReasonType    ForceCloseReason;
///自动挂起标志

TThostFtdcBoolType    IsAutoSuspend;
///业务单元

TThostFtdcBusinessUnitType    BusinessUnit;
///请求编号

TThostFtdcRequestIDType    RequestID;
///本地报单编号

TThostFtdcOrderLocalIDType    OrderLocalID;
///交易所代码

TThostFtdcExchangeIDType    ExchangeID;
///会员代码

TThostFtdcParticipantIDType    ParticipantID;
///客户代码

TThostFtdcClientIDType    ClientID;
///合约在交易所的代码

TThostFtdcExchangeInstIDType    ExchangeInstID;
///交易所交易员代码

TThostFtdcTraderIDType    TraderID;
///安装编号

TThostFtdcInstallIDType    InstallID;
///报单提交状态

TThostFtdcOrderSubmitStatusType    OrderSubmitStatus;
///报单提示序号

TThostFtdcSequenceNoType    NotifySequence;
///交易日

TThostFtdcDateType    TradingDay;
///结算编号

TThostFtdcSettlementIDType    SettlementID;
///报单编号
```

```
TThostFtdcOrderSysIDType OrderSysID;  
///报单来源  
TThostFtdcOrderSourceType OrderSource;  
///报单状态  
TThostFtdcOrderStatusType OrderStatus;  
///报单类型  
TThostFtdcOrderTypeType OrderType;  
///今成交数量  
TThostFtdcVolumeType VolumeTraded;  
///剩余数量  
TThostFtdcVolumeType VolumeTotal;  
///报单日期  
TThostFtdcDateType InsertDate;  
///委托时间  
TThostFtdcTimeType InsertTime;  
///激活时间  
TThostFtdcTimeType ActiveTime;  
///挂起时间  
TThostFtdcTimeType SuspendTime;  
///最后修改时间  
TThostFtdcTimeType UpdateTime;  
///撤销时间  
TThostFtdcTimeType CancelTime;  
///最后修改交易所交易员代码  
TThostFtdcTraderIDType ActiveTraderID;  
///结算会员编号  
TThostFtdcParticipantIDType ClearingPartID;  
///序号  
TThostFtdcSequenceNoType SequenceNo;  
///前置编号  
TThostFtdcFrontIDType FrontID;  
///会话编号
```

```

    TThostFtdcSessionIDType    SessionID;
    ///用户端产品信息
    TThostFtdcProductInfoType  UserProductInfo;
    ///状态信息
    TThostFtdcErrorMsgType     StatusMsg;
    ///用户强评标志
    TThostFtdcBoolType         UserForceClose;
    ///操作用户代码
    TThostFtdcUserIDType       ActiveUserID;
    ///经纪公司报单编号
    TThostFtdcSequenceNoType   BrokerOrderSeq;
    ///相关报单
    TThostFtdcOrderSysIDType   RelativeOrderSysID;
    ///郑商所成交数量
    TThostFtdcVolumeType       ZCETotalTradedVolume;
    ///互换单标志
    TThostFtdcBoolType         IsSwapOrder;
    ///营业部编号
    TThostFtdcBranchIDType     BranchID;
};

```

4.3.21. OnRspQryTrade 方法

成交单查询应答。当客户端发出成交单查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryTrade(
    CThostFtdcTradeField *pTrade,
    CThostFtdcRspInfoField *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 CThostFtdcTradeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///用户代码
    TThostFtdcUserIDType      UserID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///成交编号
    TThostFtdcTradeIDType      TradeID;
    ///买卖方向
    TThostFtdcDirectionType    Direction;
    ///报单编号
    TThostFtdcOrderSysIDType   OrderSysID;
    ///会员代码
    TThostFtdcParticipantIDType ParticipantID;
    ///客户代码
    TThostFtdcClientIDType      ClientID;
    ///交易角色
    TThostFtdcTradingRoleType   TradingRole;
    ///合约在交易所的代码
    TThostFtdcExchangeInstIDType ExchangeInstID;
    ///开平标志
    TThostFtdcOffsetFlagType    OffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType     HedgeFlag;
```

```
///价格
TThostFtdcPriceType Price;
///数量
TThostFtdcVolumeType Volume;
///成交时期
TThostFtdcDateType TradeDate;
///成交时间
TThostFtdcTimeType TradeTime;
///成交类型
TThostFtdcTradeTypeType TradeType;
///成交价来源
TThostFtdcPriceSourceType PriceSource;
///交易所交易员代码
TThostFtdcTraderIDType TraderID;
///本地报单编号
TThostFtdcOrderLocalIDType OrderLocalID;
///结算会员编号
TThostFtdcParticipantIDType ClearingPartID;
///业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///序号
TThostFtdcSequenceNoType SequenceNo;
///交易日
TThostFtdcDateType TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;
///经纪公司报单编号
TThostFtdcSequenceNoType BrokerOrderSeq;
///成交来源
TThostFtdcTradeSourceType TradeSource;
};
```

4.3.22. OnRspQryInvestor 方法

会员客户查询应答。当客户端发出会员客户查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryInvestor (
    CThostFtdcInvestorField *pInvestor,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pInvestor： 指向投资者信息结构的地址。

投资者信息结构：

```
struct CThostFtdcInvestorField
{
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///经纪公司代码
    TThostFtdcBrokerIDType     BrokerID;
    ///投资者分组代码
    TThostFtdcInvestorIDType   InvestorGroupID;
    ///投资者名称
    TThostFtdcPartyNameType    InvestorName;
    ///证件类型
    TThostFtdcIdCardTypeType   IdentifiedCardType;
    ///证件号码
    TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///是否活跃
    TThostFtdcBoolType         IsActive;
    ///联系电话
    TThostFtdcTelephoneType    Telephone;
    ///通讯地址
    TThostFtdcAddressType       Address;
```



```

    ///开户日期
    TThostFtdcDateType   OpenDate;

    ///手机
    TThostFtdcMobileType  Mobile;

    ///手续费率模板代码
    TThostFtdcInvestorIDType  CommModelID;

    ///保证金率模板代码
    TThostFtdcInvestorIDType  MarginModelID;

};

```

4.3.23. 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，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorPosition(
    CThostFtdcInvestorPositionField *pInvestorPosition,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInvestorPosition ：指向投资者持仓应答结构的地址。

投资者持仓应答结构：

```

struct CThostFtdcInvestorPositionField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;

    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;

    ///持仓多空方向
    TThostFtdcPosiDirectionType  PosiDirection;

    ///投机套保标志

```

TThostFtdcHedgeFlagType HedgeFlag;
///持仓日期

TThostFtdcPositionDateType PositionDate;
///上日持仓

TThostFtdcVolumeType YdPosition;
///今日持仓

TThostFtdcVolumeType Position;
///多头冻结

TThostFtdcVolumeType LongFrozen;
///空头冻结

TThostFtdcVolumeType ShortFrozen;
///多头开仓冻结金额

TThostFtdcMoneyType LongFrozenAmount;
///空头开仓冻结金额

TThostFtdcMoneyType ShortFrozenAmount;
///开仓量

TThostFtdcVolumeType OpenVolume;
///平仓量

TThostFtdcVolumeType CloseVolume;
///开仓金额

TThostFtdcMoneyType OpenAmount;
///平仓金额

TThostFtdcMoneyType CloseAmount;
///持仓成本

TThostFtdcMoneyType PositionCost;
///上次占用的保证金

TThostFtdcMoneyType PreMargin;
///占用的保证金

TThostFtdcMoneyType UseMargin;
///冻结的保证金

TThostFtdcMoneyType FrozenMargin;
///冻结的资金

```
TThostFtdcMoneyType FrozenCash;  
///冻结的手续费  
TThostFtdcMoneyType FrozenCommission;  
///资金差额  
TThostFtdcMoneyType CashIn;  
///手续费  
TThostFtdcMoneyType Commission;  
///平仓盈亏  
TThostFtdcMoneyType CloseProfit;  
///持仓盈亏  
TThostFtdcMoneyType PositionProfit;  
///上次结算价  
TThostFtdcPriceType PreSettlementPrice;  
///本次结算价  
TThostFtdcPriceType SettlementPrice;  
///交易日  
TThostFtdcDateType TradingDay;  
///结算编号  
TThostFtdcSettlementIDType SettlementID;  
///开仓成本  
TThostFtdcMoneyType OpenCost;  
///交易所保证金  
TThostFtdcMoneyType ExchangeMargin;  
///组合成交形成的持仓  
TThostFtdcVolumeType CombPosition;  
///组合多头冻结  
TThostFtdcVolumeType CombLongFrozen;  
///组合空头冻结  
TThostFtdcVolumeType CombShortFrozen;  
///逐日盯市平仓盈亏  
TThostFtdcMoneyType CloseProfitByDate;  
///逐笔对冲平仓盈亏
```

```

    TThostFtdcMoneyType CloseProfitByTrade;
    ///今日持仓

    TThostFtdcVolumeType TodayPosition;
    ///保证金率

    TThostFtdcRatioType MarginRateByMoney;
    ///保证金率(按手数)

    TThostFtdcRatioType MarginRateByVolume;
    ///执行冻结

    TThostFtdcVolumeType StrikeFrozen;
    ///执行冻结金额

    TThostFtdcMoneyType StrikeFrozenAmount;
    ///放弃执行冻结

    TThostFtdcVolumeType AbandonFrozen;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.24. OnRspQryTradingAccount 方法

请求查询资金账户响应。当客户端发出请求查询资金账户指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryTradingAccount(
    CThostFtdcTradingAccountField *pTradingAccount,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pTradingAccount : 指向资金账户结构的地址。

资金账户结构：

```

struct CThostFtdcTradingAccountField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType BrokerID;
    ///投资者帐号

    TThostFtdcAccountIDType AccountID;

```

```
///上次质押金额
TThostFtdcMoneyType PreMortgage;
///上次信用额度
TThostFtdcMoneyType PreCredit;
///上次存款额
TThostFtdcMoneyType PreDeposit;
///上次结算准备金
TThostFtdcMoneyType PreBalance;
///上次占用的保证金
TThostFtdcMoneyType PreMargin;
///利息基数
TThostFtdcMoneyType InterestBase;
///利息收入
TThostFtdcMoneyType Interest;
///入金金额
TThostFtdcMoneyType Deposit;
///出金金额
TThostFtdcMoneyType Withdraw;
///冻结的保证金
TThostFtdcMoneyType FrozenMargin;
///冻结的资金
TThostFtdcMoneyType FrozenCash;
///冻结的手续费
TThostFtdcMoneyType FrozenCommission;
///当前保证金总额
TThostFtdcMoneyType CurrMargin;
///资金差额
TThostFtdcMoneyType CashIn;
///手续费
TThostFtdcMoneyType Commission;
///平仓盈亏
TThostFtdcMoneyType CloseProfit;
```

```
///持仓盈亏
TThostFtdcMoneyType  PositionProfit;
///期货结算准备金
TThostFtdcMoneyType  Balance;
///可用资金
TThostFtdcMoneyType  Available;
///可取资金
TThostFtdcMoneyType  WithdrawQuota;
///基本准备金
TThostFtdcMoneyType  Reserve;
///交易日
TThostFtdcDateType   TradingDay;
///结算编号
TThostFtdcSettlementIDType  SettlementID;
///信用额度
TThostFtdcMoneyType  Credit;
///质押金额
TThostFtdcMoneyType  Mortgage;
///交易所保证金
TThostFtdcMoneyType  ExchangeMargin;
///投资者交割保证金
TThostFtdcMoneyType  DeliveryMargin;
///交易所交割保证金
TThostFtdcMoneyType  ExchangeDeliveryMargin;
///保底期货结算准备金
TThostFtdcMoneyType  ReserveBalance;
///币种代码
TThostFtdcCurrencyIDType  CurrencyID;
///上次货币质入金额
TThostFtdcMoneyType  PreFundMortgageIn;
///上次货币质出金额
TThostFtdcMoneyType  PreFundMortgageOut;
///货币质入金额
TThostFtdcMoneyType  FundMortgageIn;
```

```

    ///货币质出金额
    TThostFtdcMoneyType FundMortgageOut;
    ///货币质押余额
    TThostFtdcMoneyType FundMortgageAvailable;
    ///可质押货币金额
    TThostFtdcMoneyType MortgageableFund;
    ///特殊产品占用保证金
    TThostFtdcMoneyType SpecProductMargin;
    ///特殊产品冻结保证金
    TThostFtdcMoneyType SpecProductFrozenMargin;
    ///特殊产品手续费
    TThostFtdcMoneyType SpecProductCommission;
    ///特殊产品冻结手续费
    TThostFtdcMoneyType SpecProductFrozenCommission;
    ///特殊产品持仓盈亏
    TThostFtdcMoneyType SpecProductPositionProfit;
    ///特殊产品平仓盈亏
    TThostFtdcMoneyType SpecProductCloseProfit;
    ///根据持仓盈亏算法计算的特殊产品持仓盈亏
    TThostFtdcMoneyType SpecProductPositionProfitByAlg;
    ///特殊产品交易所保证金
    TThostFtdcMoneyType SpecProductExchangeMargin;
    ///业务类型
    TThostFtdcBizType BizType;
};

```

4.3.25. OnRspQryTradingCode 方法

请求查询交易编码响应。当客户端发出请求查询交易编码指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryTradingCode(
    CThostFtdcTradingCodeField *pTradingCode,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pTradingCode：指向交易编码结构的地址。

交易编码结构：

```

struct CThostFtdcTradingCodeField
{
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///经纪公司代码
    TThostFtdcBrokerIDType     BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType   ExchangeID;
    ///客户代码
    TThostFtdcClientIDType      ClientID;
    ///是否活跃
    TThostFtdcBoolType          IsActive;
    ///交易编码类型
    TThostFtdcClientIDTypeType   ClientIDType;
    ///营业部编号
    TThostFtdcBranchIDType       BranchID;
    ///业务类型
    TThostFtdcBizTypeType        BizType;
};

```

4.3.26. OnRspQryExchange 方法

请求查询交易所响应。当客户端发出请求查询交易所指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pExchange： 指向交易所结构的地址。

交易所结构：

```

struct CThostFtdcExchangeField

```



```

{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易所名称
    TThostFtdcExchangeNameType ExchangeName;
    ///交易所属性
    TThostFtdcExchangePropertyType ExchangeProperty;
};

```

4.3.27. OnRspQryInstrument 方法

请求查询合约响应。当客户端发出请求查询合约指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pInstrument： 指向合约结构的地址。

合约结构：

```

struct CThostFtdcInstrumentField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约名称
    TThostFtdcInstrumentNameType InstrumentName;
    ///合约在交易所的代码
    TThostFtdcExchangeInstIDType ExchangeInstID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;

```

```
///产品类型
TThostFtdcProductClassType    ProductClass;

///交割年份
TThostFtdcYearType    DeliveryYear;

///交割月
TThostFtdcMonthType    DeliveryMonth;

///市价单最大下单量
TThostFtdcVolumeType    MaxMarketOrderVolume;

///市价单最小下单量
TThostFtdcVolumeType    MinMarketOrderVolume;

///限价单最大下单量
TThostFtdcVolumeType    MaxLimitOrderVolume;

///限价单最小下单量
TThostFtdcVolumeType    MinLimitOrderVolume;

///合约数量乘数
TThostFtdcVolumeMultipleType    VolumeMultiple;

///最小变动价位
TThostFtdcPriceType    PriceTick;

///创建日
TThostFtdcDateType    CreateDate;

///上市日
TThostFtdcDateType    OpenDate;

///到期日
TThostFtdcDateType    ExpireDate;

///开始交割日
TThostFtdcDateType    StartDelivDate;

///结束交割日
TThostFtdcDateType    EndDelivDate;

///合约生命周期状态
TThostFtdcInstLifePhaseType    InstLifePhase;

///当前是否交易
TThostFtdcBoolType    IsTrading;
```

```

    ///持仓类型
    TThostFtdcPositionTypeType PositionType;
    ///持仓日期类型
    TThostFtdcPositionDateTypeType PositionDateType;
    ///多头保证金率
    TThostFtdcRatioType LongMarginRatio;
    ///空头保证金率
    TThostFtdcRatioType ShortMarginRatio;
    ///是否使用大额单边保证金算法
    TThostFtdcMaxMarginSideAlgorithmType MaxMarginSideAlgorithm;
    ///基础商品代码
    TThostFtdcInstrumentIDType UnderlyingInstrID;
    ///执行价
    TThostFtdcPriceType StrikePrice;
    ///期权类型
    TThostFtdcOptionsTypeType OptionsType;
    ///合约基础商品乘数
    TThostFtdcUnderlyingMultipleType UnderlyingMultiple;
    ///组合类型
    TThostFtdcCombinationTypeType CombinationType;
    ///最小买下单单位
    TThostFtdcVolumeType MinBuyVolume;
    ///最小卖下单单位
    TThostFtdcVolumeType MinSellVolume;
    ///合约标识码
    TThostFtdcInstrumentCodeType InstrumentCode;
};

```

4.3.28. OnRspQryDepthMarketData 方法

请求查询行情响应。当客户端发出请求查询行情指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryDepthMarketData(
    CThostFtdcDepthMarketDataField *pDepthMarketData,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,

```

```
bool bIsLast);
```

参数:

pDepthMarketData : 指向深度行情结构的地址。

深度行情结构:

```
struct CThostFtdcDepthMarketDataField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约在交易所的代码
    TThostFtdcExchangeInstIDType ExchangeInstID;
    ///最新价
    TThostFtdcPriceType    LastPrice;
    ///上次结算价
    TThostFtdcPriceType    PreSettlementPrice;
    ///昨收盘
    TThostFtdcPriceType    PreClosePrice;
    ///昨持仓量
    TThostFtdcLargeVolumeTypePreOpenInterest;
    ///今开盘
    TThostFtdcPriceType    OpenPrice;
    ///最高价
    TThostFtdcPriceType    HighestPrice;
    ///最低价
    TThostFtdcPriceType    LowestPrice;
    ///数量
    TThostFtdcVolumeType    Volume;
    ///成交金额
    TThostFtdcMoneyType    Turnover;
```

```
///持仓量
TThostFtdcLargeVolumeTypeOpenInterest;
///今收盘
TThostFtdcPriceType   ClosePrice;
///本次结算价
TThostFtdcPriceType   SettlementPrice;
///涨停板价
TThostFtdcPriceType   UpperLimitPrice;
///跌停板价
TThostFtdcPriceType   LowerLimitPrice;
///昨虚实度
TThostFtdcRatioType   PreDelta;
///今虚实度
TThostFtdcRatioType   CurrDelta;
///最后修改时间
TThostFtdcTimeType    UpdateTime;
///最后修改毫秒
TThostFtdcMillisecType UpdateMillisec;
///申买价一
TThostFtdcPriceType   BidPrice1;
///申买量一
TThostFtdcVolumeType BidVolume1;
///申卖价一
TThostFtdcPriceType   AskPrice1;
///申卖量一
TThostFtdcVolumeType AskVolume1;
///申买价二
TThostFtdcPriceType   BidPrice2;
///申买量二
TThostFtdcVolumeType BidVolume2;
///申卖价二
TThostFtdcPriceType   AskPrice2;
```

```
/// 申卖量二
TThostFtdcVolumeType AskVolume2;
/// 申买价三
TThostFtdcPriceType BidPrice3;
/// 申买量三
TThostFtdcVolumeType BidVolume3;
/// 申卖价三
TThostFtdcPriceType AskPrice3;
/// 申卖量三
TThostFtdcVolumeType AskVolume3;
/// 申买价四
TThostFtdcPriceType BidPrice4;
/// 申买量四
TThostFtdcVolumeType BidVolume4;
/// 申卖价四
TThostFtdcPriceType AskPrice4;
/// 申卖量四
TThostFtdcVolumeType AskVolume4;
/// 申买价五
TThostFtdcPriceType BidPrice5;
/// 申买量五
TThostFtdcVolumeType BidVolume5;
/// 申卖价五
TThostFtdcPriceType AskPrice5;
/// 申卖量五
TThostFtdcVolumeType AskVolume5;
/// 当日均价
TThostFtdcPriceType AveragePrice;
/// 业务日期
TThostFtdcDateType ActionDay;
};
```

4.3.29. OnRspQryInstrumentMarginRate 方法

请求查询合约保证金率响应。当客户端发出请求查询合约保证金率指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryInstrumentMarginRate(
    CThostFtdcInstrumentMarginRateField *pInstrumentMarginRate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pInstrumentMarginRate： 指向合约保证金率结构的地址。

合约保证金率结构：

```
struct CThostFtdcInstrumentMarginRateField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投资者范围
    TThostFtdcInvestorRangeType InvestorRange;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///多头保证金率
    TThostFtdcRatioType LongMarginRatioByMoney;
    ///多头保证金费
    TThostFtdcMoneyType LongMarginRatioByVolume;
    ///空头保证金率
    TThostFtdcRatioType ShortMarginRatioByMoney;
    ///空头保证金费
    TThostFtdcMoneyType ShortMarginRatioByVolume;
```

```

    ///是否相对交易所收取
    TThostFtdcBoolType    IsRelative;
};

```

4.3.30. OnRspQryInstrumentCommissionRate 方法

请求查询合约手续费率响应。当客户端发出请求查询合约手续费率指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pInstrumentCommissionRate： 指向合约手续费率结构的地址。

合约手续费率结构：

```

struct CThostFtdcInstrumentCommissionRateField
{
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///投资者范围
    TThostFtdcInvestorRangeType    InvestorRange;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///开仓手续费率
    TThostFtdcRatioType    OpenRatioByMoney;
    ///开仓手续费
    TThostFtdcRatioType    OpenRatioByVolume;
    ///平仓手续费率
    TThostFtdcRatioType    CloseRatioByMoney;
    ///平仓手续费

```



```

    TThostFtdcRatioType    CloseRatioByVolume;
    ///平今手续费率

    TThostFtdcRatioType    CloseTodayRatioByMoney;
    ///平今手续费

    TThostFtdcRatioType    CloseTodayRatioByVolume;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
    ///业务类型

    TThostFtdcBizTypeType  BizType;
};

```

4.3.31. OnRspQryCFMMCTradingAccountKey 方法

请求查询保证金监控中心密钥响应。当客户端发出请求查询保证金监控中心密钥指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryCFMMCTradingAccountKey(
    CThostFtdcCFMMCTradingAccountKeyField *pCFMMCTradingAccountKey,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pCFMMCTradingAccountKey: 指向保证金监控中心密钥结构的地址。

保证金监控中心密钥结构：

```

struct CThostFtdcCFMMCTradingAccountKeyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///经纪公司统一编码

    TThostFtdcParticipantIDType    ParticipantID;
    ///投资者帐号

    TThostFtdcAccountIDType    AccountID;
    ///密钥编号

    TThostFtdcSequenceNoType    KeyID;

```

```

    ///动态密钥
    TThostFtdcCFMMCKeyType CurrentKey;
};

```

4.3.32. OnRspQryEWarrantOffset 方法

请求查询仓单折抵信息响应。当客户端发出请求查询仓单折抵信息指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryEWarrantOffset(
    CThostFtdcEWarrantOffsetField *pEWarrantOffset,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pEWarrantOffset: 指向仓单折抵信息结构的地址。

仓单折抵信息结构：

```

struct CThostFtdcEWarrantOffsetField
{
    ///交易日期
    TThostFtdcTradeDateType TradingDay;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///买卖方向
    TThostFtdcDirectionType Direction;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///数量
    TThostFtdcVolumeType Volume;
}

```

```
};
```

4.3.33. OnRspQryInvestorProductGroupMargin 方法[暂不支持]

请求查询投资者品种/跨品种保证金响应。当客户端发出请求查询投资者品种/跨品种保证金指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryInvestorProductGroupMargin(
    CThostFtdcInvestorProductGroupMarginField *pInvestorProductGroupMargin,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pInvestorProductGroupMargin： 指向投资者品种/跨品种保证金结构的地址。

投资者品种/跨品种保证金结构：

```
struct CThostFtdcInvestorProductGroupMarginField
```

```
{
```

```
    ///品种/跨品种标示
```

```
    TThostFtdcInstrumentIDType ProductGroupID;
```

```
    ///经纪公司代码
```

```
    TThostFtdcBrokerIDType    BrokerID;
```

```
    ///投资者代码
```

```
    TThostFtdcInvestorIDType  InvestorID;
```

```
    ///交易日
```

```
    TThostFtdcDateType    TradingDay;
```

```
    ///结算编号
```

```
    TThostFtdcSettlementIDType SettlementID;
```

```
    ///冻结的保证金
```

```
    TThostFtdcMoneyType  FrozenMargin;
```

```
    ///多头冻结的保证金
```

```
    TThostFtdcMoneyType  LongFrozenMargin;
```

```
    ///空头冻结的保证金
```

```
    TThostFtdcMoneyType  ShortFrozenMargin;
```

```
    ///占用的保证金
```

```
    TThostFtdcMoneyType  UseMargin;
```

```
///多头保证金
TThostFtdcMoneyType LongUseMargin;
///空头保证金
TThostFtdcMoneyType ShortUseMargin;
///交易所保证金
TThostFtdcMoneyType ExchMargin;
///交易所多头保证金
TThostFtdcMoneyType LongExchMargin;
///交易所空头保证金
TThostFtdcMoneyType ShortExchMargin;
///平仓盈亏
TThostFtdcMoneyType CloseProfit;
///冻结的手续费
TThostFtdcMoneyType FrozenCommission;
///手续费
TThostFtdcMoneyType Commission;
///冻结的资金
TThostFtdcMoneyType FrozenCash;
///资金差额
TThostFtdcMoneyType CashIn;
///持仓盈亏
TThostFtdcMoneyType PositionProfit;
///折抵总金额
TThostFtdcMoneyType OffsetAmount;
///多头折抵总金额
TThostFtdcMoneyType LongOffsetAmount;
///空头折抵总金额
TThostFtdcMoneyType ShortOffsetAmount;
///交易所折抵总金额
TThostFtdcMoneyType ExchOffsetAmount;
///交易所多头折抵总金额
TThostFtdcMoneyType LongExchOffsetAmount;
///交易所空头折抵总金额
TThostFtdcMoneyType ShortExchOffsetAmount;
```

```

    ///投机套保标志
    TThostFtdcHedgeFlagType  HedgeFlag;
};

```

4.3.34. OnRspQryExchangeMarginRate 方法[暂不支持]

请求查询交易所保证金率响应。当客户端发出请求查询交易所保证金率指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pExchangeMarginRate： 指向交易所保证金率结构的地址。

交易所保证金率结构：

```

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

```

4.3.35. OnRspQryExchangeMarginRateAdjust 方法[暂不支持]

请求查询交易所调整保证金率响应。当客户端发出请求查询交易所调整保证金率指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

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

参数：

pExchangeMarginRateAdjust： 指向交易所调整保证金率结构的地址。

交易所调整保证金率结构：

```
struct CThostFtdcExchangeMarginRateAdjustField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///跟随交易所投资者多头保证金率
    TThostFtdcRatioType LongMarginRatioByMoney;
    ///跟随交易所投资者多头保证金费
    TThostFtdcMoneyType LongMarginRatioByVolume;
    ///跟随交易所投资者空头保证金率
    TThostFtdcRatioType ShortMarginRatioByMoney;
    ///跟随交易所投资者空头保证金费
    TThostFtdcMoneyType ShortMarginRatioByVolume;
    ///交易所多头保证金率
    TThostFtdcRatioType ExchLongMarginRatioByMoney;
    ///交易所多头保证金费
    TThostFtdcMoneyType ExchLongMarginRatioByVolume;
    ///交易所空头保证金率
```

```

TThostFtdcRatioType   ExchShortMarginRatioByMoney;
///交易所空头保证金费

TThostFtdcMoneyType   ExchShortMarginRatioByVolume;
///不跟随交易所投资者多头保证金率

TThostFtdcRatioType   NoLongMarginRatioByMoney;
///不跟随交易所投资者多头保证金费

TThostFtdcMoneyType   NoLongMarginRatioByVolume;
///不跟随交易所投资者空头保证金率

TThostFtdcRatioType   NoShortMarginRatioByMoney;
///不跟随交易所投资者空头保证金费

TThostFtdcMoneyType   NoShortMarginRatioByVolume;

};

```

4.3.36. OnRspQrySettlementInfo 方法

请求查询投资者结算结果响应。当客户端发出请求查询投资者结算结果指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQrySettlementInfo(
    CThostFtdcSettlementInfoField *pSettlementInfo,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pSettlementInfo ：指向投资者结算结果结构的地址。

投资者结算结果结构：

```

struct CThostFtdcSettlementInfoField
{
    ///交易日
    TThostFtdcDateType   TradingDay;
    ///结算编号
    TThostFtdcSettlementIDType   SettlementID;
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码

```

```

    TThostFtdcInvestorIDType InvestorID;
    ///序号
    TThostFtdcSequenceNoType SequenceNo;
    ///消息正文
    TThostFtdcContentType Content;
};

```

4.3.37. OnRspQryTransferBank 方法

请求查询转帐银行响应。当客户端发出请求查询转帐银行指令后，Kingstar 返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryTransferBank(
    CThostFtdcTransferBankField *pTransferBank,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pTransferBank： 指向转帐银行结构的地址。

转帐银行结构：

```

struct CThostFtdcTransferBankField
{
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分中心代码
    TThostFtdcBankBrchIDType BankBrchID;
    ///银行名称
    TThostFtdcBankNameType BankName;
    ///是否活跃
    TThostFtdcBoolType IsActive;
};

```

4.3.38. OnRspQryInvestorPositionDetail 方法

请求查询投资者持仓明细响应。当客户端发出请求查询投资者持仓明细指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryInvestorPositionDetail(
    CThostFtdcInvestorPositionDetailField *pInvestorPositionDetail,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast) ;
```

参数：

pInvestorPositionDetail ：指向投资者持仓明细结构的地址。

投资者持仓明细结构：

```
struct CThostFtdcInvestorPositionDetailField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
    /// 投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    /// 买卖
    TThostFtdcDirectionType Direction;
    /// 开仓日期
    TThostFtdcDateType OpenDate;
    /// 成交编号
    TThostFtdcTradeIDType TradeID;
    /// 数量
    TThostFtdcVolumeType Volume;
    /// 开仓价
    TThostFtdcPriceType OpenPrice;
    /// 交易日
    TThostFtdcDateType TradingDay;
    /// 结算编号
```

```

    TThostFtdcSettlementIDType SettlementID;
    ///成交类型

    TThostFtdcTradeTypeType TradeType;
    ///组合合约代码

    TThostFtdcInstrumentIDType CombInstrumentID;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
    ///逐日盯市平仓盈亏

    TThostFtdcMoneyType CloseProfitByDate;
    ///逐笔对冲平仓盈亏

    TThostFtdcMoneyType CloseProfitByTrade;
    ///逐日盯市持仓盈亏

    TThostFtdcMoneyType PositionProfitByDate;
    ///逐笔对冲持仓盈亏

    TThostFtdcMoneyType PositionProfitByTrade;
    ///投资者保证金

    TThostFtdcMoneyType Margin;
    ///交易所保证金

    TThostFtdcMoneyType ExchMargin;
    ///保证金率

    TThostFtdcRatioType MarginRateByMoney;
    ///保证金率(按手数)

    TThostFtdcRatioType MarginRateByVolume;
    ///昨结算价

    TThostFtdcPriceType LastSettlementPrice;
    ///结算价

    TThostFtdcPriceType SettlementPrice;
    ///平仓量

    TThostFtdcVolumeType CloseVolume;
    ///平仓金额

    TThostFtdcMoneyType CloseAmount;
};

```

4.3.39. OnRspQryNotice 方法

请求查询客户通知响应。当客户端发出请求查询客户通知指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryNotice(
    CThostFtdcNoticeField *pNotice,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pNotice: 指向客户通知结构的地址。

客户通知结构：

```
struct CThostFtdcNoticeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///消息正文
    TThostFtdcContentType    Content;
    ///经纪公司通知内容序列号
    TThostFtdcSequenceLabelType    SequenceLabel;
};
```

4.3.40. OnRtnTrade 方法

成交回报。当发生成交时 Kingstar 服务器会通知客户端，该方法会被调用。

函数原型：

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

参数：

pTrade: 指向成交信息结构的地址。

成交信息结构：

```
struct CThostFtdcTradeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
```

```
///投资者代码
TThostFtdcInvestorIDType InvestorID;
///合约代码
TThostFtdcInstrumentIDType InstrumentID;
///报单引用
TThostFtdcOrderRefType OrderRef;
///用户代码
TThostFtdcUserIDType UserID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///成交编号
TThostFtdcTradeIDType TradeID;
///买卖方向
TThostFtdcDirectionType Direction;
///报单编号
TThostFtdcOrderSysIDType OrderSysID;
///会员代码
TThostFtdcParticipantIDType ParticipantID;
///客户代码
TThostFtdcClientIDType ClientID;
///交易角色
TThostFtdcTradingRoleType TradingRole;
///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///开平标志
TThostFtdcOffsetFlagType OffsetFlag;
///投机套保标志
TThostFtdcHedgeFlagType HedgeFlag;
///价格
TThostFtdcPriceType Price;
///数量
TThostFtdcVolumeType Volume;
```

```

    ///成交时期
    TThostFtdcDateType    TradeDate;
    ///成交时间
    TThostFtdcTimeType    TradeTime;
    ///成交类型
    TThostFtdcTradeTypeType    TradeType;
    ///成交价来源
    TThostFtdcPriceSourceType    PriceSource;
    ///交易所交易员代码
    TThostFtdcTraderIDType    TraderID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType    OrderLocalID;
    ///结算会员编号
    TThostFtdcParticipantIDType    ClearingPartID;
    ///业务单元
    TThostFtdcBusinessUnitType    BusinessUnit;
    ///序号
    TThostFtdcSequenceNoType    SequenceNo;
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///结算编号
    TThostFtdcSettlementIDType    SettlementID;
    ///经纪公司报单编号
    TThostFtdcSequenceNoType    BrokerOrderSeq;
    ///成交来源
    TThostFtdcTradeSourceType    TradeSource;
};

```

4.3.41. 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（部分成交）导致报单状态发生变化时，Kingstar 服务器会主动通知客户端，该方法会被调用。

函数原型：

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

参数:

pOrder: 指向报单信息结构的地址。

报单信息结构:

```
struct CThostFtdcOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///用户代码
    TThostFtdcUserIDType      UserID;
    ///报单价格条件
    TThostFtdcOrderPriceTypeType OrderPriceType;
    ///买卖方向
    TThostFtdcDirectionType    Direction;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType CombOffsetFlag;
    ///组合投机套保标志
    TThostFtdcCombHedgeFlagType CombHedgeFlag;
    ///价格
    TThostFtdcPriceType        LimitPrice;
    ///数量
    TThostFtdcVolumeType       VolumeTotalOriginal;
    ///有效期类型
    TThostFtdcTimeConditionType TimeCondition;
    ///GTD 日期
    TThostFtdcDateType         GTDDate;
    ///成交量类型
```

```
TThostFtdcVolumeConditionType VolumeCondition;
///最小成交量

TThostFtdcVolumeType MinVolume;
///触发条件

TThostFtdcContingentConditionType ContingentCondition;
///止损价

TThostFtdcPriceType StopPrice;
///强平原因

TThostFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志

TThostFtdcBoolType IsAutoSuspend;
///业务单元

TThostFtdcBusinessUnitType BusinessUnit;
///请求编号

TThostFtdcRequestIDType RequestID;
///本地报单编号

TThostFtdcOrderLocalIDType OrderLocalID;
///交易所代码

TThostFtdcExchangeIDType ExchangeID;
///会员代码

TThostFtdcParticipantIDType ParticipantID;
///客户代码

TThostFtdcClientIDType ClientID;
///合约在交易所的代码

TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所交易员代码

TThostFtdcTraderIDType TraderID;
///安装编号

TThostFtdcInstallIDType InstallID;
///报单提交状态

TThostFtdcOrderSubmitStatusType OrderSubmitStatus;
///报单提示序号
```

TThostFtdcSequenceNoType *NotifySequence;*

///交易日

TThostFtdcDateType *TradingDay;*

///结算编号

TThostFtdcSettlementIDType *SettlementID;*

///报单编号

TThostFtdcOrderSysIDType *OrderSysID;*

///报单来源

TThostFtdcOrderSourceType *OrderSource;*

///报单状态

TThostFtdcOrderStatusType *OrderStatus;*

///报单类型

TThostFtdcOrderTypeType *OrderType;*

///今成交数量

TThostFtdcVolumeType *VolumeTraded;*

///剩余数量

TThostFtdcVolumeType *VolumeTotal;*

///报单日期

TThostFtdcDateType *InsertDate;*

///委托时间

TThostFtdcTimeType *InsertTime;*

///激活时间

TThostFtdcTimeType *ActiveTime;*

///挂起时间

TThostFtdcTimeType *SuspendTime;*

///最后修改时间

TThostFtdcTimeType *UpdateTime;*

///撤销时间

TThostFtdcTimeType *CancelTime;*

///最后修改交易所交易员代码

TThostFtdcTraderIDType *ActiveTraderID;*

///结算会员编号


```

    TThostFtdcParticipantIDType    ClearingPartID;
    ///序号

    TThostFtdcSequenceNoType    SequenceNo;
    ///前置编号

    TThostFtdcFrontIDType    FrontID;
    ///会话编号

    TThostFtdcSessionIDType    SessionID;
    ///用户端产品信息

    TThostFtdcProductInfoType    UserProductInfo;
    ///状态信息

    TThostFtdcErrorMsgType    StatusMsg;
    ///用户强评标志

    TThostFtdcBoolType    UserForceClose;
    ///操作用户代码

    TThostFtdcUserIDType    ActiveUserID;
    ///经纪公司报单编号

    TThostFtdcSequenceNoType    BrokerOrderSeq;
    ///相关报单

    TThostFtdcOrderSysIDType    RelativeOrderSysID;
    ///郑商所成交数量

    TThostFtdcVolumeType    ZCETotalTradedVolume;
    ///互换单标志

    TThostFtdcBoolType    IsSwapOrder;
    ///营业部编号

    TThostFtdcBranchIDType    BranchID;
};

```

4.3.42. OnErrRtnOrderInsert 方法[暂不支持]

报单录入错误回报。由 Kingstar 服务器主动通知客户端，该方法会被调用。

函数原型：

```

void OnErrRtnOrderInsert(
    CThostFtdcInputOrderField *pInputOrder,
    CThostFtdcRspInfoField *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 CThostFtdcInputOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///报单价格条件
    TThostFtdcOrderPriceTypeType  OrderPriceType;
    ///买卖方向
    TThostFtdcDirectionType    Direction;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType  CombOffsetFlag;
    ///组合投机套保标志
    TThostFtdcCombHedgeFlagType  CombHedgeFlag;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量
    TThostFtdcVolumeType  VolumeTotalOriginal;
    ///有效期类型
    TThostFtdcTimeConditionType    TimeCondition;
    ///GTD 日期
    TThostFtdcDateType    GTDDate;
```

```

    ///成交量类型
    TThostFtdcVolumeConditionType VolumeCondition;
    ///最小成交量
    TThostFtdcVolumeType MinVolume;
    ///触发条件
    TThostFtdcContingentConditionType ContingentCondition;
    ///止损价
    TThostFtdcPriceType StopPrice;
    ///强平原因
    TThostFtdcForceCloseReasonType ForceCloseReason;
    ///自动挂起标志
    TThostFtdcBoolType IsAutoSuspend;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///用户强评标志
    TThostFtdcBoolType UserForceClose;
    ///互换单标志
    TThostFtdcBoolType IsSwapOrder;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.43. OnErrRtnOrderAction 方法[暂不支持]

报单操作错误回报。由 Kingstar 服务器主动通知客户端，该方法会被调用。

函数原型：

```

void OnErrRtnOrderAction (
    CThostFtdcOrderActionField *pOrderAction,
    CThostFtdcRspInfoField *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，和后台返回的报价编号。

报价操作结构:

```
struct CThostFtdcOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///报单操作引用
    TThostFtdcOrderActionRefType  OrderActionRef;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报单编号
    TThostFtdcOrderSysIDType  OrderSysID;
    ///操作标志
    TThostFtdcActionFlagType  ActionFlag;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量变化
    TThostFtdcVolumeType  VolumeChange;
    ///操作日期
    TThostFtdcDateType    ActionDate;
    ///操作时间
    TThostFtdcTimeType    ActionTime;
    ///交易所交易员代码
```

```

    TThostFtdcTraderIDType    TraderID;
    ///安装编号

    TThostFtdcInstallIDType    InstallID;
    ///本地报单编号

    TThostFtdcOrderLocalIDType    OrderLocalID;
    ///操作本地编号

    TThostFtdcOrderLocalIDType    ActionLocalID;
    ///会员代码

    TThostFtdcParticipantIDType    ParticipantID;
    ///客户代码

    TThostFtdcClientIDType    ClientID;
    ///业务单元

    TThostFtdcBusinessUnitType    BusinessUnit;
    ///报单操作状态

    TThostFtdcOrderActionStatusType    OrderActionStatus;
    ///用户代码

    TThostFtdcUserIDType    UserID;
    ///状态信息

    TThostFtdcErrorMsgType    StatusMsg;
    ///合约代码

    TThostFtdcInstrumentIDType    InstrumentID;
    ///营业部编号

    TThostFtdcBranchIDType    BranchID;
};

```

4.3.44. OnRtnInstrumentStatus 方法

合约交易状态通知。由 Kingstar 主动通知客户合约交易状态改变时，该方法会被调用。

函数原型：

```

void OnRtnInstrumentStatus(
    CThostFtdcInstrumentStatusField *pInstrumentStatus);

```

参数：

pInstrumentStatus： 指向合约状态结构的地址。

合约状态结构：

```

struct CThostFtdcInstrumentStatusField

```

```

{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约在交易所的代码
    TThostFtdcExchangeInstIDType ExchangeInstID;
    ///结算组代码
    TThostFtdcSettlementGroupIDType SettlementGroupID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///合约交易状态
    TThostFtdcInstrumentStatusType InstrumentStatus;
    ///交易阶段编号
    TThostFtdcTradingSegmentSNTYPE TradingSegmentSN;
    ///进入本状态时间
    TThostFtdcTimeType EnterTime;
    ///进入本状态原因
    TThostFtdcInstStatusEnterReasonType EnterReason;
};

```

4.3.45. OnRtnTradingNotice 方法

交易通知。当交易发生时 Kingstar 服务器会通知客户端，该方法会被调用。

函数原型：

```

void OnRtnTradingNotice(
    CThostFtdcTradingNoticeInfoField *pTradingNoticeInfo);

```

参数：

pTradingNoticeInfo：指向用户事件通知信息结构的地址。

用户事件通知信息结构：

```

struct CThostFtdcTradingNoticeInfoField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///发送时间
    TThostFtdcTimeType SendTime;
    ///消息正文
    TThostFtdcContentType FieldContent;
    ///序列系列号

```

```

    TThostFtdcSequenceSeriesType    SequenceSeries;
    ///序列号
    TThostFtdcSequenceNoType    SequenceNo;
};

```

4.3.46. OnRtnErrorConditionalOrder 方法

提示条件单校验错误，当条件单校验错误时 Kingstar 服务器通知客户端，该方法会被调用。

函数原型：

```

void OnRtnErrorConditionalOrder(
    CThostFtdcErrorConditionalOrderField *pErrorConditionalOrder);

```

参数：

pErrorConditionalOrder： 指向查询错误报单操作结构的地址。

查询错误报单操作结构：

```

struct CThostFtdcErrorConditionalOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///报单价格条件
    TThostFtdcOrderPriceTypeType    OrderPriceType;
    ///买卖方向
    TThostFtdcDirectionType    Direction;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType    CombOffsetFlag;
    ///组合投机套保标志
    TThostFtdcCombHedgeFlagType    CombHedgeFlag;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量
    TThostFtdcVolumeType    VolumeTotalOriginal;
    ///有效期类型
    TThostFtdcTimeConditionType    TimeCondition;
}

```

```

///GTD日期
TThostFtdcDateType    GTDDate;
///成交量类型
TThostFtdcVolumeConditionType VolumeCondition;
///最小成交量
TThostFtdcVolumeType MinVolume;
///触发条件
TThostFtdcContingentConditionType ContingentCondition;
///止损价
TThostFtdcPriceType    StopPrice;
///强平原因
TThostFtdcForceCloseReasonType    ForceCloseReason;
///自动挂起标志
TThostFtdcBoolType    IsAutoSuspend;
///业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///请求编号
TThostFtdcRequestIDType    RequestID;
///本地报单编号
TThostFtdcOrderLocalIDType    OrderLocalID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///会员代码
TThostFtdcParticipantIDType    ParticipantID;
///客户代码
TThostFtdcClientIDType    ClientID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所交易员代码
TThostFtdcTraderIDType    TraderID;
///安装编号
TThostFtdcInstallIDType    InstallID;
///报单提交状态
TThostFtdcOrderSubmitStatusType    OrderSubmitStatus;
///报单提示序号
TThostFtdcSequenceNoType NotifySequence;
///交易日
TThostFtdcDateType    TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;

```



```

///报单编号
TThostFtdcOrderSysIDType OrderSysID;
///报单来源
TThostFtdcOrderSourceType OrderSource;
///报单状态
TThostFtdcOrderStatusType OrderStatus;
///报单类型
TThostFtdcOrderTypeType OrderType;
///今成交数量
TThostFtdcVolumeType VolumeTraded;
///剩余数量
TThostFtdcVolumeType VolumeTotal;
///报单日期
TThostFtdcDateType InsertDate;
///委托时间
TThostFtdcTimeType InsertTime;
///激活时间
TThostFtdcTimeType ActiveTime;
///挂起时间
TThostFtdcTimeType SuspendTime;
///最后修改时间
TThostFtdcTimeType UpdateTime;
///撤销时间
TThostFtdcTimeType CancelTime;
///最后修改交易所交易员代码
TThostFtdcTraderIDType ActiveTraderID;
///结算会员编号
TThostFtdcParticipantIDType ClearingPartID;
///序号
TThostFtdcSequenceNoType SequenceNo;
///前置编号
TThostFtdcFrontIDType FrontID;
///会话编号
TThostFtdcSessionIDType SessionID;
///用户端产品信息
TThostFtdcProductInfoType UserProductInfo;
///状态信息
TThostFtdcErrorMsgType StatusMsg;
///用户强评标志
TThostFtdcBoolType UserForceClose;

```

```

    ///操作用户代码
    TThostFtdcUserIDType ActiveUserID;
    ///经纪公司报单编号
    TThostFtdcSequenceNoType BrokerOrderSeq;
    ///相关报单
    TThostFtdcOrderSysIDType RelativeOrderSysID;
    ///郑商所成交数量
    TThostFtdcVolumeType ZCETotalTradedVolume;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
    ///互换单标志
    TThostFtdcBoolType IsSwapOrder;
    ///营业部编号
    TThostFtdcBranchIDType BranchID;
};

```

4.3.47. OnRtnExecOrder 方法

执行宣告通知。当执行宣告时 Kingstar 服务器通知客户端，该方法会被调用。

函数原型：

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

参数：

pExecOrder： 指向执行宣告结构的地址。

执行宣告结构：

```

struct CThostFtdcExecOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///执行宣告引用
    TThostFtdcOrderRefType ExecOrderRef;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///数量
    TThostFtdcVolumeType Volume;
    ///请求编号

```

```

TThostFtdcRequestIDType RequestID;
///业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///开平标志
TThostFtdcOffsetFlagType OffsetFlag;
///投机套保标志
TThostFtdcHedgeFlagType HedgeFlag;
///执行类型
TThostFtdcActionTypeType ActionType;
///保留头寸申请的持仓方向
TThostFtdcPosiDirectionType PosiDirection;
///期权行权后是否保留期货头寸的标记
TThostFtdcExecOrderPositionFlagType ReservePositionFlag;
///期权行权后生成的头寸是否自动平仓
TThostFtdcExecOrderCloseFlagType CloseFlag;
///本地执行宣告编号
TThostFtdcOrderLocalIDType ExecOrderLocalID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///会员代码
TThostFtdcParticipantIDType ParticipantID;
///客户代码
TThostFtdcClientIDType ClientID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所交易员代码
TThostFtdcTraderIDType TraderID;
///安装编号
TThostFtdcInstallIDType InstallID;
///执行宣告提交状态
TThostFtdcOrderSubmitStatusType OrderSubmitStatus;
///报单提示序号
TThostFtdcSequenceNoType NotifySequence;
///交易日
TThostFtdcDateType TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;
///执行宣告编号
TThostFtdcExecOrderSysIDType ExecOrderSysID;
///报单日期

```

```

    TThostFtdcDateType    InsertDate;
    ///插入时间
    TThostFtdcTimeType    InsertTime;
    ///撤销时间
    TThostFtdcTimeType    CancelTime;
    ///执行结果
    TThostFtdcExecResultType ExecResult;
    ///结算会员编号
    TThostFtdcParticipantIDType    ClearingPartID;
    ///序号
    TThostFtdcSequenceNoType SequenceNo;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///用户端产品信息
    TThostFtdcProductInfoType UserProductInfo;
    ///状态信息
    TThostFtdcErrorMsgType    StatusMsg;
    ///操作用户代码
    TThostFtdcUserIDType ActiveUserID;
    ///经纪公司报单编号
    TThostFtdcSequenceNoType BrokerExecOrderSeq;
    ///营业部编号
    TThostFtdcBranchIDType    BranchID;
};

```

4.3.48. OnErrRtnExecOrderInsert 方法

执行宣告录入错误回报。当执行宣告录入发生错误时 Kingstar 服务器通知客户端，该方法会被调用。

函数原型：

```

void OnErrRtnExecOrderInsert(
    CThostFtdcInputExecOrderField *pInputExecOrder,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pInputExecOrder： 指向输入的执行宣告结构的地址。

输入的执行宣告结构：

```

struct CThostFtdcInputExecOrderField
{

```

```

    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///执行宣告引用
    TThostFtdcOrderRefType ExecOrderRef;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///数量
    TThostFtdcVolumeType Volume;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///开平标志
    TThostFtdcOffsetFlagType OffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///执行类型
    TThostFtdcActionTypeType ActionType;
    ///保留头寸申请的持仓方向
    TThostFtdcPosiDirectionType PosiDirection;
    ///期权行权后是否保留期货头寸的标记
    TThostFtdcExecOrderPositionFlagType ReservePositionFlag;
    ///期权行权后生成的头寸是否自动平仓
    TThostFtdcExecOrderCloseFlagType CloseFlag;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.49. OnErrRtnExecOrderAction 方法

执行宣告操作错误回报。当执行宣告操作发生错误时 Kingstar 服务器通知客户端，该方法会被调用。

函数原型：

```

void OnErrRtnExecOrderAction(
    CThostFtdcExecOrderActionField *pExecOrderAction,
    CThostFtdcRspInfoField *pRspInfo);

```

参数:

pExecOrderAction: 指向执行宣告操作结构的地址。

执行宣告操作结构:

```
struct CThostFtdcExecOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///执行宣告操作引用
    TThostFtdcOrderActionRefType  ExecOrderActionRef;
    ///执行宣告引用
    TThostFtdcOrderRefType    ExecOrderRef;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///前置编号
    TThostFtdcFrontIDType    FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///执行宣告操作编号
    TThostFtdcExecOrderSysIDType  ExecOrderSysID;
    ///操作标志
    TThostFtdcActionFlagType    ActionFlag;
    ///操作日期
    TThostFtdcDateType    ActionDate;
    ///操作时间
    TThostFtdcTimeType    ActionTime;
    ///交易所交易员代码
    TThostFtdcTraderIDType    TraderID;
    ///安装编号
    TThostFtdcInstallIDType    InstallID;
    ///本地执行宣告编号
    TThostFtdcOrderLocalIDType    ExecOrderLocalID;
    ///操作本地编号
    TThostFtdcOrderLocalIDType    ActionLocalID;
    ///会员代码
    TThostFtdcParticipantIDType    ParticipantID;
    ///客户代码
```

```

    TThostFtdcClientIDType    ClientID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///报单操作状态
    TThostFtdcOrderActionStatusType    OrderActionStatus;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///执行类型
    TThostFtdcActionTypeType    ActionType;
    ///状态信息
    TThostFtdcErrorMsgType    StatusMsg;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///营业部编号
    TThostFtdcBranchIDType    BranchID;
};

```

4.3.50. OnRspQrySettlementInfoConfirm 方法

查询结算确认响应。由 Kingstar 主动通知客户端，该方法会被调用。

函数原型：

```

void OnRspQrySettlementInfoConfirm(
    CThostFtdcSettlementInfoConfirmField *pSettlementInfoConfirm,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pSettlementInfoConfirm : 指向返回的结算确认信息结构。

结算确认结构：

```

struct CThostFtdcSettlementInfoConfirmField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///确认日期
    TThostFtdcDateType    ConfirmDate;

```

```

        ///确认时间
        TThostFtdcTimeType    ConfirmTime;
    };

```

4.3.51. OnRspQryContractBank 方法

请求查询签约银行响应。

函数原型：

```

void OnRspQryContractBank(
    CThostFtdcContractBankField *pContractBank,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pContractBank： 指向查询签约银行响应结构。

查询签约银行响应结构：

```

struct CThostFtdcContractBankField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分中心代码
    TThostFtdcBankBrchIDType BankBrchID;
    ///银行名称
    TThostFtdcBankNameType    BankName;
};

```

4.3.52. OnRspQryParkedOrder 方法[暂不支持]

请求查询预埋单响应。

函数原型：

```

void OnRspQryParkedOrder(
    CThostFtdcParkedOrderField *pParkedOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,

```


bool bIsLast);

参数:

pParkedOrder: 指向请求查询预埋单响应结构。

请求查询预埋单响应结构:

```
struct CThostFtdcParkedOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///报单价格条件
    TThostFtdcOrderPriceTypeType    OrderPriceType;
    ///买卖方向
    TThostFtdcDirectionType    Direction;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType    CombOffsetFlag;
    ///组合投机套保标志
    TThostFtdcCombHedgeFlagType    CombHedgeFlag;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量
    TThostFtdcVolumeType    VolumeTotalOriginal;
    ///有效期类型
    TThostFtdcTimeConditionType    TimeCondition;
    ///GTD 日期
    TThostFtdcDateType    GTDDate;
```

```

///成交量类型
TThostFtdcVolumeConditionType VolumeCondition;
///最小成交量
TThostFtdcVolumeType MinVolume;
///触发条件
TThostFtdcContingentConditionType ContingentCondition;
///止损价
TThostFtdcPriceType StopPrice;
///强平原因
TThostFtdcForceCloseReasonType ForceCloseReason;
///自动挂起标志
TThostFtdcBoolType IsAutoSuspend;
///业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///请求编号
TThostFtdcRequestIDType RequestID;
///用户强评标志
TThostFtdcBoolType UserForceClose;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///预埋报单编号
TThostFtdcParkedOrderIDType ParkedOrderID;
///用户类型
TThostFtdcUserTypeType UserType;
///预埋单状态
TThostFtdcParkedOrderStatusType Status;
///错误代码
TThostFtdcErrorIDType ErrorID;
///错误信息
TThostFtdcErrorMsgType ErrorMsg;
///互换单标志
TThostFtdcBoolType IsSwapOrder;

```

```
};
```

4.3.53. OnRspQryParkedOrderAction 方法[暂不支持]

请求查询预埋撤单响应。

函数原型：

```
void OnRspQryParkedOrderAction(
    CThostFtdcParkedOrderActionField *pParkedOrderAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pParkedOrderAction ：指向请求查询预埋撤单响应结构。

请求查询预埋撤单响应结构：

```
struct CThostFtdcParkedOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///报单操作引用
    TThostFtdcOrderActionRefType OrderActionRef;
    ///报单引用
    TThostFtdcOrderRefType   OrderRef;
    ///请求编号
    TThostFtdcRequestIDType  RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报单编号
    TThostFtdcOrderSysIDType OrderSysID;
```

```

    ///操作标志
    TThostFtdcActionFlagType ActionFlag;

    ///价格
    TThostFtdcPriceType LimitPrice;

    ///数量变化
    TThostFtdcVolumeType VolumeChange;

    ///用户代码
    TThostFtdcUserIDType UserID;

    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

    ///预埋撤单单编号
    TThostFtdcParkedOrderActionIDType ParkedOrderActionID;

    ///用户类型
    TThostFtdcUserTypeType UserType;

    ///预埋撤单状态
    TThostFtdcParkedOrderStatusType Status;

    ///错误代码
    TThostFtdcErrorIDType ErrorID;

    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;

};

```

4.3.54. OnRspQryTradingNotice 方法

请求查询交易通知响应。

函数原型：

```

void OnRspQryTradingNotice(
    CThostFtdcTradingNoticeField *pTradingNotice,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pTradingNotice： 指向查询交易通知结构。

查询交易通知结构：

```

struct CThostFtdcTradingNoticeField

```

```

{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者范围
    TThostFtdcInvestorRangeType    InvestorRange;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///序列系列号
    TThostFtdcSequenceSeriesType    SequenceSeries;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///发送时间
    TThostFtdcTimeType    SendTime;
    ///序列号
    TThostFtdcSequenceNoType    SequenceNo;
    ///消息正文
    TThostFtdcContentType    FieldContent;
};

```

4.3.55. OnRspQryBrokerTradingParams 方法

请求查询经济公司交易参数响应。

函数原型：

```

void OnRspQryBrokerTradingParams(
    CThostFtdcBrokerTradingParamsField *pBrokerTradingParams,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pBrokerTradingParams: 指向查询经济公司交易参数结构的地址。

查询经济公司交易参数结构：

```

struct CThostFtdcBrokerTradingParamsField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///保证金价格类型
    TThostFtdcMarginPriceTypeType    MarginPriceType;
    ///盈亏算法

```

```

TThostFtdcAlgorithmType   Algorithm;
///可用是否包含平仓盈利
TThostFtdcIncludeCloseProfitType   AvailIncludeCloseProfit;
///币种代码
TThostFtdcCurrencyIDType   CurrencyID;
///期权权利金价格类型
TThostFtdcOptionRoyaltyPriceTypeType   OptionRoyaltyPriceType;
};

```

4.3.56. OnRspQryInvestorPositionCombineDetail 方法

请求查询投资者组合持仓明细响应。

函数原型：

```

void OnRspQryInvestorPositionCombineDetail(
    CThostFtdcInvestorPositionCombineDetailField *pInvestorPositionCombineDetail,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInvestorPositionCombineDetail：指向请求查询投资者组合持仓明细响应结构。

请求查询投资者组合持仓明细响应结构

```

struct CThostFtdcInvestorPositionCombineDetailField

```

```

{
    ///交易日
    TThostFtdcDateType   TradingDay;
    ///开仓日期
    TThostFtdcDateType   OpenDate;
    ///交易所代码
    TThostFtdcExchangeIDType   ExchangeID;
    ///结算编号
    TThostFtdcSettlementIDType   SettlementID;
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///组合编号

```

```

    TThostFtdcTradeIDType    ComTradeID;
    ///撮合编号

    TThostFtdcTradeIDType    TradeID;
    ///合约代码

    TThostFtdcInstrumentIDType InstrumentID;
    ///投机套保标志

    TThostFtdcHedgeFlagType  HedgeFlag;
    ///买卖

    TThostFtdcDirectionType  Direction;
    ///持仓量

    TThostFtdcVolumeType  TotalAmt;
    ///投资者保证金

    TThostFtdcMoneyType  Margin;
    ///交易所保证金

    TThostFtdcMoneyType  ExchMargin;
    ///保证金率

    TThostFtdcRatioType  MarginRateByMoney;
    ///保证金率(按手数)

    TThostFtdcRatioType  MarginRateByVolume;
    ///单腿编号

    TThostFtdcLegIDType  LegID;
    ///单腿乘数

    TThostFtdcLegMultipleType  LegMultiple;
    ///组合持仓合约编码

    TThostFtdcInstrumentIDType  CombInstrumentID;
    ///成交组合

    TThostFtdcTradeGroupIDType  TradeGroupID;
};

```

4.3.57. OnRspParkedOrderInsert 方法[暂不支持]

预埋单录入请求响应。

函数原型：

```
void OnRspParkedOrderInsert(
```

```

CThostFtdcParkedOrderField *pParkedOrder,
CThostFtdcRspInfoField *pRspInfo,
int nRequestID,
bool bIsLast);

```

参数：

pParkedOrder ：指向预埋单录入请求响应结构。

预埋单录入请求响应结构：

```

struct CThostFtdcParkedOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///报单价格条件
    TThostFtdcOrderPriceTypeType  OrderPriceType;
    ///买卖方向
    TThostFtdcDirectionType    Direction;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType  CombOffsetFlag;
    ///组合投机套保标志
    TThostFtdcCombHedgeFlagType  CombHedgeFlag;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量
    TThostFtdcVolumeType  VolumeTotalOriginal;
    ///有效期类型

```



```
TThostFtdcTimeConditionType    TimeCondition;
///GTD 日期

TThostFtdcDateType    GTDDate;
///成交量类型

TThostFtdcVolumeConditionType VolumeCondition;
///最小成交量

TThostFtdcVolumeType    MinVolume;
///触发条件

TThostFtdcContingentConditionType    ContingentCondition;
///止损价

TThostFtdcPriceType    StopPrice;
///强平原因

TThostFtdcForceCloseReasonType    ForceCloseReason;
///自动挂起标志

TThostFtdcBoolType    IsAutoSuspend;
///业务单元

TThostFtdcBusinessUnitType    BusinessUnit;
///请求编号

TThostFtdcRequestIDType    RequestID;
///用户强评标志

TThostFtdcBoolType    UserForceClose;
///交易所代码

TThostFtdcExchangeIDType    ExchangeID;
///预埋报单编号

TThostFtdcParkedOrderIDType    ParkedOrderID;
///用户类型

TThostFtdcUserTypeType    UserType;
///预埋单状态

TThostFtdcParkedOrderStatusType    Status;
///错误代码

TThostFtdcErrorIDType    ErrorID;
///错误信息
```

```

    TThostFtdcErrorMsgType    ErrorMsg;

    ///呼换单标志

    TThostFtdcBoolType        IsSwapOrder;

};

```

4.3.58. OnRspParkedOrderAction 方法[暂不支持]

预埋撤单录入请求响应。

函数原型：

```

void OnRspParkedOrderAction(
    CThostFtdcParkedOrderActionField *pParkedOrderAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pParkedOrderAction ：指向预埋撤单录入请求响应结构。

预埋撤单录入请求响应结构：

```

struct CThostFtdcParkedOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;

    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;

    ///报单操作引用
    TThostFtdcOrderActionRefType  OrderActionRef;

    ///报单引用
    TThostFtdcOrderRefType    OrderRef;

    ///请求编号
    TThostFtdcRequestIDType    RequestID;

    ///前置编号
    TThostFtdcFrontIDType      FrontID;

    ///会话编号
    TThostFtdcSessionIDType    SessionID;

    ///交易所代码

```

```

    TThostFtdcExchangeIDType ExchangeID;
    ///报单编号

    TThostFtdcOrderSysIDType OrderSysID;
    ///操作标志

    TThostFtdcActionFlagType ActionFlag;
    ///价格

    TThostFtdcPriceType LimitPrice;
    ///数量变化

    TThostFtdcVolumeType VolumeChange;
    ///用户代码

    TThostFtdcUserIDType UserID;
    ///合约代码

    TThostFtdcInstrumentIDType InstrumentID;
    ///预埋撤单单编号

    TThostFtdcParkedOrderActionIDType ParkedOrderActionID;
    ///用户类型

    TThostFtdcUserTypeType UserType;
    ///预埋撤单状态

    TThostFtdcParkedOrderStatusType Status;
    ///错误代码

    TThostFtdcErrorIDType ErrorID;
    ///错误信息

    TThostFtdcErrorMsgType ErrorMsg;

};

```

4.3.59. OnRspRemoveParkedOrder 方法[暂不支持]

删除预埋单响应。

函数原型：

```

void OnRspRemoveParkedOrder(
    CThostFtdcRemoveParkedOrderField *pRemoveParkedOrder
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pRemoveParkedOrder : 指向删除预埋单响应结构。

删除预埋单响应结构:

```
struct CThostFtdcRemoveParkedOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///预埋报单编号
    TThostFtdcParkedOrderIDType  ParkedOrderID;
};
```

4.3.60. OnRspRemoveParkedOrderAction 方法[暂不支持]

删除预埋撤单响应。

函数原型:

```
void OnRspRemoveParkedOrderAction(
    CThostFtdcRemoveParkedOrderActionField *pRemoveParkedOrderAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pRemoveParkedOrderAction: 指向删除预埋撤单响应结构。

删除预埋撤单响应结构:

```
struct CThostFtdcRemoveParkedOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///预埋撤单编号
    TThostFtdcParkedOrderActionIDType  ParkedOrderActionID;
};
```

4.3.61. OnRspQryBrokerTradingAlgos 方法

请求查询经纪公司交易算法响应。

函数原型：

```
void OnRspQryBrokerTradingAlgos(
    CThostFtdcBrokerTradingAlgosField *pBrokerTradingAlgos,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pBrokerTradingAlgos： 指向请求经纪公司交易算法响应结构。

请求查询经纪公司交易算法响应结构

```
struct CThostFtdcBrokerTradingAlgosField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///持仓处理算法编号
    TThostFtdcHandlePositionAlgoIDType HandlePositionAlgoID;
    ///寻找保证金率算法编号
    TThostFtdcFindMarginRateAlgoIDType FindMarginRateAlgoID;
    ///资金处理算法编号
    TThostFtdcHandleTradingAccountAlgoIDType HandleTradingAccountAlgoID;
};
```

4.3.62. OnRspForQuoteInsert 方法

请求询价录入响应。

函数原型：

```
void OnRspForQuoteInsert (
    CThostFtdcInputForQuoteField *pInputForQuote,
    CThostFtdcRspInfoField *pRspInfo,
```

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

参数:

pInputForQuote: 指向查询询价录入响应结构。

查询询价录入响应结构如下:

```
struct CThostFtdcInputForQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///询价引用
    TThostFtdcOrderRefType    ForQuoteRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
};
```

4.3.63. OnRspQuoteInsert 方法

请求报价录入响应。

函数原型:

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

参数:

pInputQuote: 指向查询报价录入响应结构。

查询报价录入响应结构如下:

```
struct CThostFtdcInputQuoteField
{
```

```

///经纪公司代码
TThostFtdcBrokerIDType BrokerID;
///投资者代码
TThostFtdcInvestorIDType InvestorID;
///合约代码
TThostFtdcInstrumentIDType InstrumentID;
///报价引用
TThostFtdcOrderRefType QuoteRef;
///用户代码
TThostFtdcUserIDType UserID;
///卖价格
TThostFtdcPriceType AskPrice;
///买价格
TThostFtdcPriceType BidPrice;
///卖数量
TThostFtdcVolumeType AskVolume;
///买数量
TThostFtdcVolumeType BidVolume;
///请求编号
TThostFtdcRequestIDType RequestID;
///业务单元
TThostFtdcBusinessUnitType BusinessUnit;
///卖开平标志
TThostFtdcOffsetFlagType AskOffsetFlag;
///买开平标志
TThostFtdcOffsetFlagType BidOffsetFlag;
///卖投机套保标志
TThostFtdcHedgeFlagType AskHedgeFlag;
///买投机套保标志
TThostFtdcHedgeFlagType BidHedgeFlag;
///衍生卖报单引用
TThostFtdcOrderRefType AskOrderRef;
///衍生买报单引用

```

```

    TThostFtdcOrderRefType    BidOrderRef;
    ///应价编号
    TThostFtdcOrderSysIDType  ForQuoteSysID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.64. OnRspQuoteAction 方法

请求报价操作响应。

函数原型：

```

void OnRspQuoteAction (
    CThostFtdcInputQuoteActionField *pInputQuoteAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInputQuoteAction： 指向查询报价操作响应结构。

查询报价操作响应结构如下：

```

struct CThostFtdcInputQuoteActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///报价操作引用
    TThostFtdcOrderActionRefType  QuoteActionRef;
    ///报价引用
    TThostFtdcOrderRefType    QuoteRef;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
}

```



```

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报价操作编号
    TThostFtdcOrderSysIDType QuoteSysID;
    ///操作标志
    TThostFtdcActionFlagType ActionFlag;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.3.65. OnRspQryProduct 方法

请求查询产品响应。

函数原型：

```

void OnRspQryProduct (
    CThostFtdcProductField *pProduct,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pProduct： 指向查询产品响应结构。

查询产品响应结构如下：

```

struct CThostFtdcProductField
{
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///产品名称
    TThostFtdcProductNameType ProductName;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品类型
    TThostFtdcProductClassType ProductClass;

```

```

    /// 合约数量乘数
    TThostFtdcVolumeMultipleType  VolumeMultiple;
    /// 最小变动价位
    TThostFtdcPriceType  PriceTick;
    /// 市价单最大下单量
    TThostFtdcVolumeType  MaxMarketOrderVolume;
    /// 市价单最小下单量
    TThostFtdcVolumeType  MinMarketOrderVolume;
    /// 限价单最大下单量
    TThostFtdcVolumeType  MaxLimitOrderVolume;
    /// 限价单最小下单量
    TThostFtdcVolumeType  MinLimitOrderVolume;
    /// 持仓类型
    TThostFtdcPositionTypeType  PositionType;
    /// 持仓日期类型
    TThostFtdcPositionDateTypeType  PositionDateType;
    /// 平仓处理类型
    TThostFtdcCloseDealTypeType  CloseDealType;
    /// 交易币种类型
    TThostFtdcCurrencyIDType  TradeCurrencyID;
    /// 质押资金可用范围
    TThostFtdcMortgageFundUseRangeType  MortgageFundUseRange;
    /// 交易所产品代码
    TThostFtdcInstrumentIDType  ExchangeProductID;
    /// 合约基础商品乘数
    TThostFtdcUnderlyingMultipleType  UnderlyingMultiple;
};

```

4.3.66. OnRspQryExchangeRate 方法[暂不支持]

请求查询汇率响应。

函数原型：

```

void OnRspQryExchangeRate (
    CThostFtdcExchangeRateField *pExchangeRate,

```

```

        CThostFtdcRspInfoField *pRspInfo,
        int nRequestID,
        bool bIsLast);

```

参数:

pExchangeRate: 指向查询汇率响应结构。

查询汇率响应结构如下:

```

struct CThostFtdcExchangeRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///源币种
    TThostFtdcCurrencyIDType  FromCurrencyID;
    ///源币种单位数量
    TThostFtdcCurrencyUnitType    FromCurrencyUnit;
    ///目标币种
    TThostFtdcCurrencyIDType  ToCurrencyID;
    ///汇率
    TThostFtdcExchangeRateType    ExchangeRate;
};

```

4.3.67. OnRspQrySecAgentACIDMap 方法[暂不支持]

请求查询二级代理操作员银期权限响应。

函数原型:

```

void OnRspQrySecAgentACIDMap (
    CThostFtdcSecAgentACIDMapField *pSecAgentACIDMap,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pSecAgentACIDMap: 指向查询二级代理操作员银期权限响应结构。

查询二级代理操作员银期权限响应结构如下:

```

struct CThostFtdcSecAgentACIDMapField
{

```

```

    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///用户代码
    TThostFtdcUserIDType   UserID;
    ///资金账户
    TThostFtdcAccountIDType   AccountID;
    ///币种
    TThostFtdcCurrencyIDType   CurrencyID;
    ///境外中介机构资金帐号
    TThostFtdcAccountIDType   BrokerSecAgentID;
};

```

4.3.68. OnRspQryForQuote 方法

请求查询询价响应。

函数原型：

```

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

```

参数：

pForQuote：指向查询询价响应结构。

查询询价响应结构如下：

```

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

```

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

```
};
```

4.3.69. OnRspQryQuote 方法

请求查询报价响应。

函数原型：

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

参数：

pQuote： 指向查询报价响应结构。

查询报价响应结构如下：

```
struct CThostFtdcQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报价引用
    TThostFtdcOrderRefType   QuoteRef;
    ///用户代码
    TThostFtdcUserIDType   UserID;
    ///卖价格
    TThostFtdcPriceType   AskPrice;
    ///买价格
    TThostFtdcPriceType   BidPrice;
    ///卖数量
    TThostFtdcVolumeType AskVolume;
    ///买数量
    TThostFtdcVolumeType BidVolume;
```

```
///请求编号
TThostFtdcRequestIDType RequestID;

///业务单元
TThostFtdcBusinessUnitType BusinessUnit;

///卖开平标志
TThostFtdcOffsetFlagType AskOffsetFlag;

///买开平标志
TThostFtdcOffsetFlagType BidOffsetFlag;

///卖投机套保标志
TThostFtdcHedgeFlagType AskHedgeFlag;

///买投机套保标志
TThostFtdcHedgeFlagType BidHedgeFlag;

///本地报价编号
TThostFtdcOrderLocalIDType QuoteLocalID;

///交易所代码
TThostFtdcExchangeIDType ExchangeID;

///会员代码
TThostFtdcParticipantIDType ParticipantID;

///客户代码
TThostFtdcClientIDType ClientID;

///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;

///交易所交易员代码
TThostFtdcTraderIDType TraderID;

///安装编号
TThostFtdcInstallIDType InstallID;

///报价提示序号
TThostFtdcSequenceNoType NotifySequence;

///报价提交状态
TThostFtdcOrderSubmitStatusType OrderSubmitStatus;

///交易日
TThostFtdcDateType TradingDay;
```

```
///结算编号
TThostFtdcSettlementIDType SettlementID;
///报价编号
TThostFtdcOrderSysIDType QuoteSysID;
///报单日期
TThostFtdcDateType InsertDate;
///插入时间
TThostFtdcTimeType InsertTime;
///撤销时间
TThostFtdcTimeType CancelTime;
///报价状态
TThostFtdcOrderStatusType QuoteStatus;
///结算会员编号
TThostFtdcParticipantIDType ClearingPartID;
///序号
TThostFtdcSequenceNoType SequenceNo;
///卖方报单编号
TThostFtdcOrderSysIDType AskOrderSysID;
///买方报单编号
TThostFtdcOrderSysIDType BidOrderSysID;
///前置编号
TThostFtdcFrontIDType FrontID;
///会话编号
TThostFtdcSessionIDType SessionID;
///用户端产品信息
TThostFtdcProductInfoType UserProductInfo;
///状态信息
TThostFtdcErrorMsgType StatusMsg;
///操作用户代码
TThostFtdcUserIDType ActiveUserID;
///经纪公司报价编号
TThostFtdcSequenceNoType BrokerQuoteSeq;
```



```

    ///衍生卖报单引用
    TThostFtdcOrderRefType    AskOrderRef;
    ///衍生买报单引用
    TThostFtdcOrderRefType    BidOrderRef;
    ///应价编号
    TThostFtdcOrderSysIDType  ForQuoteSysID;
    ///营业部编号
    TThostFtdcBranchIDType    BranchID;
};

```

4.3.70. OnErrRtnForQuoteInsert 方法[暂不支持]

询价录入错误响应。

函数原型：

```

void OnErrRtnForQuoteInsert (
    CThostFtdcInputForQuoteField *pInputForQuote,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pInputForQuote： 指向询价录入错误响应结构。

询价录入错误响应结构如下：

```

struct CThostFtdcInputForQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///询价引用
    TThostFtdcOrderRefType    ForQuoteRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;

```

```
};
```

4.3.71. OnRtnQuote 方法

报价通知。

函数原型：

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

参数：

pQuote： 指向报价通知结构。

报价通知结构如下：

```
struct CThostFtdcQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报价引用
    TThostFtdcOrderRefType    QuoteRef;
    ///用户代码
    TThostFtdcUserIDType     UserID;
    ///卖价格
    TThostFtdcPriceType      AskPrice;
    ///买价格
    TThostFtdcPriceType      BidPrice;
    ///卖数量
    TThostFtdcVolumeType     AskVolume;
    ///买数量
    TThostFtdcVolumeType     BidVolume;
    ///请求编号
    TThostFtdcRequestIDType   RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
```

```

///卖开平标志
TThostFtdcOffsetFlagType  AskOffsetFlag;
///买开平标志
TThostFtdcOffsetFlagType  BidOffsetFlag;
///卖投机套保标志
TThostFtdcHedgeFlagType  AskHedgeFlag;
///买投机套保标志
TThostFtdcHedgeFlagType  BidHedgeFlag;
///本地报价编号
TThostFtdcOrderLocalIDType  QuoteLocalID;
///交易所代码
TThostFtdcExchangeIDType  ExchangeID;
///会员代码
TThostFtdcParticipantIDType  ParticipantID;
///客户代码
TThostFtdcClientIDType  ClientID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType  ExchangeInstID;
///交易所交易员代码
TThostFtdcTraderIDType  TraderID;
///安装编号
TThostFtdcInstallIDType  InstallID;
///报价提示序号
TThostFtdcSequenceNoType  NotifySequence;
///交易日
TThostFtdcDateType  TradingDay;
///结算编号
TThostFtdcSettlementIDType  SettlementID;
///报价编号
TThostFtdcOrderSysIDType  QuoteSysID;
///报单日期
TThostFtdcDateType  InsertDate;

```

```

///插入时间
TThostFtdcTimeType    InsertTime;

///撤销时间
TThostFtdcTimeType    CancelTime;

///报价状态
TThostFtdcOrderStatusType QuoteStatus;

///结算会员编号
TThostFtdcParticipantIDType    ClearingPartID;

///序号
TThostFtdcSequenceNoType SequenceNo;

///卖方报单编号
TThostFtdcOrderSysIDType AskOrderSysID;

///买方报单编号
TThostFtdcOrderSysIDType BidOrderSysID;

///前置编号
TThostFtdcFrontIDType FrontID;

///会话编号
TThostFtdcSessionIDType SessionID;

///用户端产品信息
TThostFtdcProductInfoType UserProductInfo;

///状态信息
TThostFtdcErrorMsgType StatusMsg;

///操作用户代码
TThostFtdcUserIDType ActiveUserID;

///经纪公司报价编号
TThostFtdcSequenceNoType BrokerQuoteSeq;

///衍生卖报单引用
TThostFtdcOrderRefType AskOrderRef;

///衍生买报单引用
TThostFtdcOrderRefType BidOrderRef;

///应价编号
TThostFtdcOrderSysIDType ForQuoteSysID;

```

```

        ///营业部编号
        TThostFtdcBranchIDType   BranchID;
    };

```

4.3.72. OnRtnForQuoteRsp 方法

询价通知。

函数原型：

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

参数：

pForQuoteRsp：指向询价通知结构。

询价通知结构如下：

```

struct CThostFtdcForQuoteRspField
{
    ///交易日
    TThostFtdcDateType   TradingDay;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///询价编号
    TThostFtdcOrderSysIDType ForQuoteSysID;
    ///询价时间
    TThostFtdcTimeType   ForQuoteTime;
    ///业务日期
    TThostFtdcDateType   ActionDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.73. OnErrRtnQuoteInsert 方法[暂不支持]

报价录入错误响应。

函数原型：

```

void OnErrRtnQuoteInsert (
    CThostFtdcInputQuoteField *pInputQuote,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pInputQuote：指向报价录入错误响应结构。

报价录入错误响应结构如下:

```
struct CThostFtdcInputQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报价引用
    TThostFtdcOrderRefType    QuoteRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///卖价格
    TThostFtdcPriceType    AskPrice;
    ///买价格
    TThostFtdcPriceType    BidPrice;
    ///卖数量
    TThostFtdcVolumeType    AskVolume;
    ///买数量
    TThostFtdcVolumeType    BidVolume;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///卖开平标志
    TThostFtdcOffsetFlagType    AskOffsetFlag;
    ///买开平标志
    TThostFtdcOffsetFlagType    BidOffsetFlag;
    ///卖投机套保标志
    TThostFtdcHedgeFlagType    AskHedgeFlag;
    ///买投机套保标志
```

```

    TThostFtdcHedgeFlagType BidHedgeFlag;
    ///衍生卖报单引用

    TThostFtdcOrderRefType AskOrderRef;
    ///衍生买报单引用

    TThostFtdcOrderRefType BidOrderRef;
    ///应价编号

    TThostFtdcOrderSysIDType ForQuoteSysID;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.74. OnErrRtnQuoteAction 方法[暂不支持]

报价操作错误响应。

函数原型：

```

void OnErrRtnQuoteAction (
    CThostFtdcQuoteActionField *pQuoteAction,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pQuoteAction：指向报价操作错误响应结构。

报价操作错误响应结构如下：

```

struct CThostFtdcQuoteActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///报价操作引用
    TThostFtdcOrderActionRefType QuoteActionRef;
    ///报价引用
    TThostFtdcOrderRefType QuoteRef;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///前置编号

```

```

TThostFtdcFrontIDType FrontID;
///会话编号

TThostFtdcSessionIDType SessionID;
///交易所代码

TThostFtdcExchangeIDType ExchangeID;
///报价操作编号

TThostFtdcOrderSysIDType QuoteSysID;
///操作标志

TThostFtdcActionFlagType ActionFlag;
///操作日期

TThostFtdcDateType ActionDate;
///操作时间

TThostFtdcTimeType ActionTime;
///交易所交易员代码

TThostFtdcTraderIDType TraderID;
///安装编号

TThostFtdcInstallIDType InstallID;
///本地报价编号

TThostFtdcOrderLocalIDType QuoteLocalID;
///操作本地编号

TThostFtdcOrderLocalIDType ActionLocalID;
///会员代码

TThostFtdcParticipantIDType ParticipantID;
///客户代码

TThostFtdcClientIDType ClientID;
///业务单元

TThostFtdcBusinessUnitType BusinessUnit;
///报单操作状态

TThostFtdcOrderActionStatusType OrderActionStatus;
///用户代码

TThostFtdcUserIDType UserID;
///状态信息

```



```

    TThostFtdcErrorMsgType    StatusMsg;
    ///合约代码

    TThostFtdcInstrumentIDType    InstrumentID;
    ///营业部编号

    TThostFtdcBranchIDType    BranchID;
};

```

4.3.75. OnRspQueryCFMMCTradingAccountToken 方法[暂不支持]

请求查询保证金监控中心用户令牌响应。当客户端发出请求查询保证金监控中心用户令牌指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQueryCFMMCTradingAccountToken(
    CThostFtdcQueryCFMMCTradingAccountTokenField *
    pQueryCFMMCTradingAccountToken, CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pQueryCFMMCTradingAccountToken： 指向保证金监控中心用户令牌结构的地址。

保证金监控中心用户令牌结构：

```

struct CThostFtdcQueryCFMMCTradingAccountTokenField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType    InvestorID;
};

```

4.3.76. OnRtnFromBankToFutureByBank 方法

银行发起银行资金转期货通知。

函数原型：

```

OnRtnFromBankToFutureByBank(CThostFtdcRspTransferField *pRspTransfer);

```

参数：

pRspTransfer： 指向银行发起银行资金转期货通知结构。

银行发起银行资金转期货通知结构：

```

struct CThostFtdcRspTransferField

```

```

{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType LastFragment;
    ///会话号
    TThostFtdcSessionIDType SessionID;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///证件类型
    TThostFtdcIdCardTypeType IdCardType;
    ///证件号码
    TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///客户类型
    TThostFtdcCustTypeType CustType;
    ///银行帐号
    TThostFtdcBankAccountType BankAccount;
    ///银行密码
    TThostFtdcPasswordType BankPassWord;
    ///投资者帐号
    TThostFtdcAccountIDType AccountID;
    ///期货密码

```

```

TThostFtdcPasswordType Password;
/// 安装编号
TThostFtdcInstallIDType InstallID;
/// 期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
/// 用户标识
TThostFtdcUserIDType UserID;
/// 验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
/// 币种代码
TThostFtdcCurrencyIDType CurrencyID;
/// 转帐金额
TThostFtdcTradeAmountType TradeAmount;
/// 期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
/// 费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
/// 应收客户费用
TThostFtdcCustFeeType CustFee;
/// 应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
/// 发送方给接收方的消息
TThostFtdcAddInfoType Message;
/// 摘要
TThostFtdcDigestType Digest;
/// 银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
/// 渠道标志
TThostFtdcDeviceIDType DeviceID;
/// 期货单位帐号类型
TThostFtdcBankAccTypeType BankSecuAccType;
/// 期货公司银行编码
TThostFtdcBankCodingForFutureType BrokerIDByBank;
/// 期货单位帐号
TThostFtdcBankAccountType BankSecuAcc;
/// 银行密码标志
TThostFtdcPwdFlagType BankPwdFlag;
/// 期货资金密码核对标志
TThostFtdcPwdFlagType SecuPwdFlag;
/// 交易柜员

```

```

TThostFtdcOperNoType OperNo;
///请求编号
TThostFtdcRequestIDType RequestID;
///交易ID
TThostFtdcTIDType TID;
///转账交易状态
TThostFtdcTransferStatusType TransferStatus;
///错误代码
TThostFtdcErrorIDType ErrorID;
///错误信息
TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.77. OnRtnFromFutureToBankByBank 方法

银行发起期货资金转银行通知。

函数原型：

```
void OnRtnFromFutureToBankByBank(CThostFtdcRspTransferField *pRspTransfer);
```

参数：

pRspTransfer： 指向银行发起期货资金转银行通知结构。

银行发起期货资金转银行通知结构：

```

struct CThostFtdcRspTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;

```

```

///银期平台消息流水号
TThostFtdcSerialType PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType LastFragment;
///会话号
TThostFtdcSessionIDType SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;

```

```

    ///应收期货公司费用
    TThostFtdcFutureFeeType BrokerFee;
    ///发送方给接收方的消息
    TThostFtdcAddInfoType Message;
    ///摘要
    TThostFtdcDigestType Digest;
    ///银行帐号类型
    TThostFtdcBankAccTypeType BankAccType;
    ///渠道标志
    TThostFtdcDeviceIDType DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///交易ID
    TThostFtdcTIDType TID;
    ///转账交易状态
    TThostFtdcTransferStatusType TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.78. OnRtnRepealFromBankToFutureByBank 方法

银行发起冲正银行转期货通知。

函数原型：

```
void OnRtnRepealFromBankToFutureByBank(CThostFtdcRspRepealField *pRspRepeal);
```

参数：

pRspRepeal：指向银行发起冲正银行转期货通知结构。

银行发起冲正银行转期货通知结构:

```
struct CThostFtdcRspRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TThostFtdcRepealedTimesType    RepealedTimes;
    ///银行冲正标志
    TThostFtdcBankRepealFlagType    BankRepealFlag;
    ///期商冲正标志
    TThostFtdcBrokerRepealFlagType    BrokerRepealFlag;
    ///被冲正平台流水号
    TThostFtdcPlateSerialType    PlateRepealSerial;
    ///被冲正银行流水号
    TThostFtdcBankSerialType    BankRepealSerial;
    ///被冲正期货流水号
    TThostFtdcFutureSerialType    FutureRepealSerial;
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType    TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType    BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType    TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType    PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType    LastFragment;
    ///会话号
```

```
TThostFtdcSessionIDType SessionID;  
///客户姓名  
TThostFtdcIndividualNameType CustomerName;  
///证件类型  
TThostFtdcIdCardTypeType IdCardType;  
///证件号码  
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;  
///客户类型  
TThostFtdcCustTypeType CustType;  
///银行帐号  
TThostFtdcBankAccountType BankAccount;  
///银行密码  
TThostFtdcPasswordType BankPassWord;  
///投资者帐号  
TThostFtdcAccountIDType AccountID;  
///期货密码  
TThostFtdcPasswordType Password;  
///安装编号  
TThostFtdcInstallIDType InstallID;  
///期货公司流水号  
TThostFtdcFutureSerialType FutureSerial;  
///用户标识  
TThostFtdcUserIDType UserID;  
///验证客户证件号码标志  
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;  
///币种代码  
TThostFtdcCurrencyIDType CurrencyID;  
///转帐金额  
TThostFtdcTradeAmountType TradeAmount;  
///期货可取金额  
TThostFtdcTradeAmountType FutureFetchAmount;  
///费用支付标志  
TThostFtdcFeePayFlagType FeePayFlag;  
///应收客户费用  
TThostFtdcCustFeeType CustFee;  
///应收期货公司费用  
TThostFtdcFutureFeeType BrokerFee;  
///发送方给接收方的消息  
TThostFtdcAddInfoType Message;  
///摘要
```



```

    TThostFtdcDigestType  Digest;
    ///银行帐号类型
    TThostFtdcBankAccTypeType    BankAccType;
    ///渠道标志
    TThostFtdcDeviceIDType    DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
};

```

4.3.79. OnRtnRepealFromFutureToBankByBank 方法

银行发起冲正期货转银行通知。

函数原型：

```
void OnRtnRepealFromFutureToBankByBank(CThostFtdcRspRepealField *pRspRepeal);
```

参数：

pRspRepeal: 指向银行发起冲正期货转银行通知结构。

银行发起冲正期货转银行通知结构：

```

struct CThostFtdcRspRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType    RepealTimeInterval;

```

```

///已经冲正次数
TThostFtdcRepealedTimesType   RepealedTimes;
///银行冲正标志
TThostFtdcBankRepealFlagType   BankRepealFlag;
///期商冲正标志
TThostFtdcBrokerRepealFlagType BrokerRepealFlag;
///被冲正平台流水号
TThostFtdcPlateSerialType   PlateRepealSerial;
///被冲正银行流水号
TThostFtdcBankSerialType   BankRepealSerial;
///被冲正期货流水号
TThostFtdcFutureSerialType   FutureRepealSerial;
///业务功能码
TThostFtdcTradeCodeType   TradeCode;
///银行代码
TThostFtdcBankIDType   BankID;
///银行分支机构代码
TThostFtdcBankBrchIDType   BankBranchID;
///期商代码
TThostFtdcBrokerIDType   BrokerID;
///期商分支机构代码
TThostFtdcFutureBranchIDType   BrokerBranchID;
///交易日期
TThostFtdcTradeDateType   TradeDate;
///交易时间
TThostFtdcTradeTimeType   TradeTime;
///银行流水号
TThostFtdcBankSerialType   BankSerial;
///交易系统日期
TThostFtdcTradeDateType   TradingDay;
///银期平台消息流水号
TThostFtdcSerialType   PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType   SessionID;
///客户姓名
TThostFtdcIndividualNameType   CustomerName;
///证件类型
TThostFtdcIdCardTypeType   IdCardType;

```

```

///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType Message;
///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType DeviceID;

```

```

    ///期货单位帐号类型
    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
};

```

4.3.80. OnRtnFromBankToFutureByFuture 方法

期货发起银行资金转期货通知。

函数原型：

```
void OnRtnFromBankToFutureByFuture(CThostFtdcRspTransferField *pRspTransfer);
```

参数：

pRspTransfer：指向期货发起银行资金转期货通知结构。

期货发起银行资金转期货通知结构：

```

struct CThostFtdcRspTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码

```

```

TThostFtdcBrokerIDType    BrokerID;
///期商分支机构代码
TThostFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TThostFtdcTradeDateType    TradeDate;
///交易时间
TThostFtdcTradeTimeType    TradeTime;
///银行流水号
TThostFtdcBankSerialType    BankSerial;
///交易系统日期
TThostFtdcTradeDateType    TradingDay;
///银期平台消息流水号
TThostFtdcSerialType    PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType    LastFragment;
///会话号
TThostFtdcSessionIDType    SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType    IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType    CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType    BankPassWord;
///投资者帐号
TThostFtdcAccountIDType    AccountID;
///期货密码
TThostFtdcPasswordType    Password;
///安装编号
TThostFtdcInstallIDType    InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType    UserID;
///验证客户证件号码标志

```

```

TThostFtdcYesNoIndicatorType  VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType  CurrencyID;
///转帐金额
TThostFtdcTradeAmountType  TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType  FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType  FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType  CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType  BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType  Message;
///摘要
TThostFtdcDigestType  Digest;
///银行帐号类型
TThostFtdcBankAccTypeType  BankAccType;
///渠道标志
TThostFtdcDeviceIDType  DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType  BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType  BrokerIDByBank;
///期货单位帐号
TThostFtdcBankAccountType  BankSecuAcc;
///银行密码标志
TThostFtdcPwdFlagType  BankPwdFlag;
///期货资金密码核对标志
TThostFtdcPwdFlagType  SecuPwdFlag;
///交易柜员
TThostFtdcOperNoType  OperNo;
///请求编号
TThostFtdcRequestIDType  RequestID;
///交易ID
TThostFtdcTIDType  TID;
///转账交易状态
TThostFtdcTransferStatusType  TransferStatus;
///错误代码

```

```

    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;

};

```

4.3.81. OnRtnFromFutureToBankByFuture 方法

期货发起期货资金转银行通知。

函数原型：

```
void OnRtnFromFutureToBankByFuture(CThostFtdcRspTransferField *pRspTransfer);
```

参数：

pRspTransfer：指向期货发起期货资金转银行通知结构。

期货发起期货资金转银行通知结构：

```

struct CThostFtdcRspTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType    TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType    BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType    TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType    PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType    LastFragment;
    ///会话号
    TThostFtdcSessionIDType    SessionID;
    ///客户姓名
    TThostFtdcIndividualNameType    CustomerName;

```

```

///证件类型
TThostFtdcIdCardTypeType IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType Message;
///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;

```



```

    ///渠道标志
    TThostFtdcDeviceIDType DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///交易ID
    TThostFtdcTIDType TID;
    ///转账交易状态
    TThostFtdcTransferStatusType TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.82. OnRtnRepealFromBankToFutureByFutureManual 方法

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。

函数原型：

```

virtual void OnRtnRepealFromBankToFutureByFutureManual(
    CThostFtdcRspRepealField *pRspRepeal);

```

参数：

pRspRepeal： 指向冲正响应结构的地址。

冲正响应结构：

```

struct CThostFtdcRspRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType RepealTimeInterval;
    ///已经冲正次数

```

```

TThostFtdcRepealedTimesType   RepealedTimes;
///银行冲正标志
TThostFtdcBankRepealFlagType   BankRepealFlag;
///期商冲正标志
TThostFtdcBrokerRepealFlagType   BrokerRepealFlag;
///被冲正平台流水号
TThostFtdcPlateSerialType   PlateRepealSerial;
///被冲正银行流水号
TThostFtdcBankSerialType   BankRepealSerial;
///被冲正期货流水号
TThostFtdcFutureSerialType   FutureRepealSerial;
///业务功能码
TThostFtdcTradeCodeType   TradeCode;
///银行代码
TThostFtdcBankIDType   BankID;
///银行分支机构代码
TThostFtdcBankBrchIDType   BankBranchID;
///期商代码
TThostFtdcBrokerIDType   BrokerID;
///期商分支机构代码
TThostFtdcFutureBranchIDType   BrokerBranchID;
///交易日期
TThostFtdcTradeDateType   TradeDate;
///交易时间
TThostFtdcTradeTimeType   TradeTime;
///银行流水号
TThostFtdcBankSerialType   BankSerial;
///交易系统日期
TThostFtdcTradeDateType   TradingDay;
///银期平台消息流水号
TThostFtdcSerialType   PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType   SessionID;
///客户姓名
TThostFtdcIndividualNameType   CustomerName;
///证件类型
TThostFtdcIdCardTypeType   IdCardType;
///证件号码

```

```

TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType Message;
///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType DeviceID;
///期货单位帐号类型

```

```

    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType    BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType    BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType    OperNo;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType    ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
};

```

4.3.83. OnRtnRepealFromFutureToBankByFutureManual 方法

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。

函数原型：

```

void OnRtnRepealFromFutureToBankByFutureManual(
    CThostFtdcRspRepealField *pRspRepeal);

```

参数：

pRspRepeal：指向冲正响应结构的地址。

冲正响应结构：

```

struct CThostFtdcRspRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TThostFtdcRepealedTimesType    RepealedTimes;
    ///银行冲正标志
    TThostFtdcBankRepealFlagType    BankRepealFlag;

```

```

///期商冲正标志
TThostFtdcBrokerRepealFlagType BrokerRepealFlag;
///被冲正平台流水号
TThostFtdcPlateSerialType PlateRepealSerial;
///被冲正银行流水号
TThostFtdcBankSerialType BankRepealSerial;
///被冲正期货流水号
TThostFtdcFutureSerialType FutureRepealSerial;
///业务功能码
TThostFtdcTradeCodeType TradeCode;
///银行代码
TThostFtdcBankIDType BankID;
///银行分支机构代码
TThostFtdcBankBrchIDType BankBranchID;
///期商代码
TThostFtdcBrokerIDType BrokerID;
///期商分支机构代码
TThostFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TThostFtdcTradeDateType TradeDate;
///交易时间
TThostFtdcTradeTimeType TradeTime;
///银行流水号
TThostFtdcBankSerialType BankSerial;
///交易系统日期
TThostFtdcTradeDateType TradingDay;
///银期平台消息流水号
TThostFtdcSerialType PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType LastFragment;
///会话号
TThostFtdcSessionIDType SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;

```

```

///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType    BankPassWord;
///投资者帐号
TThostFtdcAccountIDType   AccountID;
///期货密码
TThostFtdcPasswordType    Password;
///安装编号
TThostFtdcInstallIDType    InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType      UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType  CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType  FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType     CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType   BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType     Message;
///摘要
TThostFtdcDigestType      Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType     DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType BrokerIDByBank;

```

```

    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.84. OnRtnQueryBankBalanceByFuture 方法

期货发起查询银行余额通知。

函数原型:

```

void OnRtnQueryBankBalanceByFuture(
    CThostFtdcNotifyQueryAccountField *pNotifyQueryAccount);

```

参数;

pNotifyQueryAccount: 指向查询银行余额通知结构。

查询银行余额通知结构:

```

struct CThostFtdcNotifyQueryAccountField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码

```

```

TThostFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TThostFtdcTradeDateType TradeDate;
///交易时间
TThostFtdcTradeTimeType TradeTime;
///银行流水号
TThostFtdcBankSerialType BankSerial;
///交易系统日期
TThostFtdcTradeDateType TradingDay;
///银期平台消息流水号
TThostFtdcSerialType PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType LastFragment;
///会话号
TThostFtdcSessionIDType SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///安装编号
TThostFtdcInstallIDType InstallID;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码

```



```

    TThostFtdcCurrencyIDType CurrencyID;
    ///摘要
    TThostFtdcDigestType Digest;
    ///银行帐号类型
    TThostFtdcBankAccTypeType BankAccType;
    ///渠道标志
    TThostFtdcDeviceIDType DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///交易ID
    TThostFtdcTIDType TID;
    ///银行可用金额
    TThostFtdcTradeAmountType BankUseAmount;
    ///银行可取金额
    TThostFtdcTradeAmountType BankFetchAmount;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.85. OnErrRtnBankToFutureByFuture 方法

期货发起银行资金转期货错误回报。

函数原型：

```

void OnErrRtnBankToFutureByFuture(
    CThostFtdcReqTransferField *pReqTransfer,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pReqTransfer: 指向银行资金转期货请求结构。

银行资金转期货请求结构:

```
struct CThostFtdcReqTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType LastFragment;
    ///会话号
    TThostFtdcSessionIDType SessionID;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///证件类型
    TThostFtdcIdCardTypeType IdCardType;
    ///证件号码
    TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///客户类型
    TThostFtdcCustTypeType CustType;
    ///银行帐号
    TThostFtdcBankAccountType BankAccount;
    ///银行密码
    TThostFtdcPasswordType BankPassWord;
```

```

///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType Message;
///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TThostFtdcBankAccountType BankSecuAcc;
///银行密码标志
TThostFtdcPwdFlagType BankPwdFlag;

```

```

    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;
};

```

4.3.86. OnErrRtnFutureToBankByFuture 方法

期货发起期货资金转银行错误回报。

函数原型：

```

void OnErrRtnFutureToBankByFuture(
    CThostFtdcReqTransferField *pReqTransfer,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pReqTransfer： 指向期货资金转银行请求结构。

期货资金转银行请求结构：

```

struct CThostFtdcReqTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType    TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType    BankSerial;

```

```

///交易系统日期
TThostFtdcTradeDateType   TradingDay;
///银期平台消息流水号
TThostFtdcSerialType   PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType   SessionID;
///客户姓名
TThostFtdcIndividualNameType   CustomerName;
///证件类型
TThostFtdcIdCardTypeType   IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType   IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType   CustType;
///银行帐号
TThostFtdcBankAccountType   BankAccount;
///银行密码
TThostFtdcPasswordType   BankPassWord;
///投资者帐号
TThostFtdcAccountIDType   AccountID;
///期货密码
TThostFtdcPasswordType   Password;
///安装编号
TThostFtdcInstallIDType   InstallID;
///期货公司流水号
TThostFtdcFutureSerialType   FutureSerial;
///用户标识
TThostFtdcUserIDType   UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType   VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType   CurrencyID;
///转帐金额
TThostFtdcTradeAmountType   TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType   FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType   FeePayFlag;

```

```

    ///应收客户费用
    TThostFtdcCustFeeType CustFee;
    ///应收期货公司费用
    TThostFtdcFutureFeeType BrokerFee;
    ///发送方给接收方的消息
    TThostFtdcAddInfoType Message;
    ///摘要
    TThostFtdcDigestType Digest;
    ///银行帐号类型
    TThostFtdcBankAccTypeType BankAccType;
    ///渠道标志
    TThostFtdcDeviceIDType DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///交易ID
    TThostFtdcTIDType TID;
    ///转账交易状态
    TThostFtdcTransferStatusType TransferStatus;
};

```

4.3.87. OnErrRtnRepealBankToFutureByFutureManual 方法

系统运行时期货端手工发起冲正银行转期货错误回报。

函数原型：

```

void OnErrRtnRepealBankToFutureByFutureManual(
    CThostFtdcReqRepealField *pReqRepeal,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pReqRepeal: 指向冲正银行转期货请求结构。

冲正银行转期货请求:

```
struct CThostFtdcReqRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TThostFtdcRepealedTimesType    RepealedTimes;
    ///银行冲正标志
    TThostFtdcBankRepealFlagType    BankRepealFlag;
    ///期商冲正标志
    TThostFtdcBrokerRepealFlagType    BrokerRepealFlag;
    ///被冲正平台流水号
    TThostFtdcPlateSerialType    PlateRepealSerial;
    ///被冲正银行流水号
    TThostFtdcBankSerialType    BankRepealSerial;
    ///被冲正期货流水号
    TThostFtdcFutureSerialType    FutureRepealSerial;
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType    TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType    BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType    TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType    PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType    LastFragment;
```

```

///会话号
TThostFtdcSessionIDType SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType Message;

```



```

///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TThostFtdcBankAccountType BankSecuAcc;
///银行密码标志
TThostFtdcPwdFlagType BankPwdFlag;
///期货资金密码核对标志
TThostFtdcPwdFlagType SecuPwdFlag;
///交易柜员
TThostFtdcOperNoType OperNo;
///请求编号
TThostFtdcRequestIDType RequestID;
///交易ID
TThostFtdcTIDType TID;
///转账交易状态
TThostFtdcTransferStatusType TransferStatus;
};

```

4.3.88. OnErrRtnRepealFutureToBankByFutureManual 方法

系统运行时期货端发起冲正期货转银行错误回报。

函数原型:

```

void OnErrRtnRepealFutureToBankByFutureManual(
    CThostFtdcReqRepealField *pReqRepeal,
    CThostFtdcRspInfoField *pRspInfo);

```

参数:

pReqRepeal: 指向冲正期货转银行请求结构。

冲正期货转银行请求结构:

```

struct CThostFtdcReqRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType RepealTimeInterval;

```

```

///已经冲正次数
TThostFtdcRepealedTimesType   RepealedTimes;
///银行冲正标志
TThostFtdcBankRepealFlagType   BankRepealFlag;
///期商冲正标志
TThostFtdcBrokerRepealFlagType BrokerRepealFlag;
///被冲正平台流水号
TThostFtdcPlateSerialType      PlateRepealSerial;
///被冲正银行流水号
TThostFtdcBankSerialType        BankRepealSerial;
///被冲正期货流水号
TThostFtdcFutureSerialType      FutureRepealSerial;
///业务功能码
TThostFtdcTradeCodeType         TradeCode;
///银行代码
TThostFtdcBankIDType            BankID;
///银行分支机构代码
TThostFtdcBankBrchIDType        BankBranchID;
///期商代码
TThostFtdcBrokerIDType          BrokerID;
///期商分支机构代码
TThostFtdcFutureBranchIDType    BrokerBranchID;
///交易日期
TThostFtdcTradeDateType         TradeDate;
///交易时间
TThostFtdcTradeTimeType         TradeTime;
///银行流水号
TThostFtdcBankSerialType        BankSerial;
///交易系统日期
TThostFtdcTradeDateType         TradingDay;
///银期平台消息流水号
TThostFtdcSerialType            PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType       LastFragment;
///会话号
TThostFtdcSessionIDType         SessionID;
///客户姓名
TThostFtdcIndividualNameType    CustomerName;
///证件类型
TThostFtdcIdCardTypeType        IdCardType;

```

```

///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///安装编号
TThostFtdcInstallIDType InstallID;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///转帐金额
TThostFtdcTradeAmountType TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType FeePayFlag;
///应收客户费用
TThostFtdcCustFeeType CustFee;
///应收期货公司费用
TThostFtdcFutureFeeType BrokerFee;
///发送方给接收方的消息
TThostFtdcAddInfoType Message;
///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType DeviceID;

```

```

    ///期货单位帐号类型
    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;
};

```

4.3.89. OnErrRtnQueryBankBalanceByFuture 方法

银行发起查询银行余额错误回报。

函数原型：

```

void OnErrRtnQueryBankBalanceByFuture(
    CThostFtdcReqQueryAccountField *pReqQueryAccount,
    CThostFtdcRspInfoField *pRspInfo);

```

参数：

pReqQueryAccount：指向查询银行余额信息结构。

查询银行余额信息结构：

```

struct CThostFtdcReqQueryAccountField
{
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;

```

```

///期商代码
TThostFtdcBrokerIDType   BrokerID;
///期商分支机构代码
TThostFtdcFutureBranchIDType BrokerBranchID;
///交易日期
TThostFtdcTradeDateType   TradeDate;
///交易时间
TThostFtdcTradeTimeType   TradeTime;
///银行流水号
TThostFtdcBankSerialType   BankSerial;
///交易系统日期
TThostFtdcTradeDateType   TradingDay;
///银期平台消息流水号
TThostFtdcSerialType   PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType   SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType   IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType   CustType;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType   BankPassWord;
///投资者帐号
TThostFtdcAccountIDType   AccountID;

```

```
///期货密码
TThostFtdcPasswordType Password;
///期货公司流水号
TThostFtdcFutureSerialType FutureSerial;
///安装编号
TThostFtdcInstallIDType InstallID;
///用户标识
TThostFtdcUserIDType UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///摘要
TThostFtdcDigestType Digest;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///渠道标志
TThostFtdcDeviceIDType DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType BrokerIDByBank;
///期货单位帐号
TThostFtdcBankAccountType BankSecuAcc;
///银行密码标志
TThostFtdcPwdFlagType BankPwdFlag;
///期货资金密码核对标志
TThostFtdcPwdFlagType SecuPwdFlag;
///交易柜员
TThostFtdcOperNoType OperNo;
///请求编号
TThostFtdcRequestIDType RequestID;
```

```

        ///交易ID
        TThostFtdcTIDType    TID;
    };

```

4.3.90. OnRtnRepealFromBankToFutureByFuture 方法

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。

函数原型：

```
void OnRtnRepealFromBankToFutureByFuture(CThostFtdcRspRepealField *pRspRepeal);
```

参数：

pRspRepeal：指向冲正银行转期货请求响应结构。

冲正银行转期货请求响应结构：

```

struct CThostFtdcRspRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TThostFtdcRepealedTimesType    RepealedTimes;
    ///银行冲正标志
    TThostFtdcBankRepealFlagType    BankRepealFlag;
    ///期商冲正标志
    TThostFtdcBrokerRepealFlagType    BrokerRepealFlag;
    ///被冲正平台流水号
    TThostFtdcPlateSerialType    PlateRepealSerial;
    ///被冲正银行流水号
    TThostFtdcBankSerialType    BankRepealSerial;
    ///被冲正期货流水号
    TThostFtdcFutureSerialType    FutureRepealSerial;
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;

```

```

///交易时间
TThostFtdcTradeTimeType   TradeTime;
///银行流水号
TThostFtdcBankSerialType   BankSerial;
///交易系统日期
TThostFtdcTradeDateType    TradingDay;
///银期平台消息流水号
TThostFtdcSerialType       PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType     SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType    IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType      CustType;
///银行帐号
TThostFtdcBankAccountType   BankAccount;
///银行密码
TThostFtdcPasswordType      BankPassWord;
///投资者帐号
TThostFtdcAccountIDType     AccountID;
///期货密码
TThostFtdcPasswordType      Password;
///安装编号
TThostFtdcInstallIDType     InstallID;
///期货公司流水号
TThostFtdcFutureSerialType   FutureSerial;
///用户标识
TThostFtdcUserIDType        UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType     CurrencyID;
///转帐金额
TThostFtdcTradeAmountType    TradeAmount;

```



```

    ///期货可取金额
    TThostFtdcTradeAmountType    FutureFetchAmount;
    ///费用支付标志
    TThostFtdcFeePayFlagType    FeePayFlag;
    ///应收客户费用
    TThostFtdcCustFeeType    CustFee;
    ///应收期货公司费用
    TThostFtdcFutureFeeType    BrokerFee;
    ///发送方给接收方的消息
    TThostFtdcAddInfoType    Message;
    ///摘要
    TThostFtdcDigestType    Digest;
    ///银行帐号类型
    TThostFtdcBankAccTypeType    BankAccType;
    ///渠道标志
    TThostFtdcDeviceIDType    DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType    BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType    BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType    OperNo;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///转账交易状态
    TThostFtdcTransferStatusType    TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType    ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
};

```

4.3.91. OnRtnRepealFromFutureToBankByFuture 方法

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。

函数原型：

```
void OnRtnRepealFromFutureToBankByFuture(CThostFtdcRspRepealField *pRspRepeal);
```

参数：

pRspRepeal：指向冲正期货转银行请求响应结构。

冲正期货转银行请求响应结构：

```
struct CThostFtdcRspRepealField
{
    ///冲正时间间隔
    TThostFtdcRepealTimeIntervalType    RepealTimeInterval;
    ///已经冲正次数
    TThostFtdcRepealedTimesType    RepealedTimes;
    ///银行冲正标志
    TThostFtdcBankRepealFlagType    BankRepealFlag;
    ///期商冲正标志
    TThostFtdcBrokerRepealFlagType    BrokerRepealFlag;
    ///被冲正平台流水号
    TThostFtdcPlateSerialType    PlateRepealSerial;
    ///被冲正银行流水号
    TThostFtdcBankSerialType    BankRepealSerial;
    ///被冲正期货流水号
    TThostFtdcFutureSerialType    FutureRepealSerial;
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType    TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType    BankSerial;
```

```

///交易系统日期
TThostFtdcTradeDateType   TradingDay;
///银期平台消息流水号
TThostFtdcSerialType   PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType   SessionID;
///客户姓名
TThostFtdcIndividualNameType   CustomerName;
///证件类型
TThostFtdcIdCardTypeType   IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType   IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType   CustType;
///银行帐号
TThostFtdcBankAccountType   BankAccount;
///银行密码
TThostFtdcPasswordType   BankPassWord;
///投资者帐号
TThostFtdcAccountIDType   AccountID;
///期货密码
TThostFtdcPasswordType   Password;
///安装编号
TThostFtdcInstallIDType   InstallID;
///期货公司流水号
TThostFtdcFutureSerialType   FutureSerial;
///用户标识
TThostFtdcUserIDType   UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType   VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType   CurrencyID;
///转帐金额
TThostFtdcTradeAmountType   TradeAmount;
///期货可取金额
TThostFtdcTradeAmountType   FutureFetchAmount;
///费用支付标志
TThostFtdcFeePayFlagType   FeePayFlag;

```

```

    ///应收客户费用
    TThostFtdcCustFeeType CustFee;
    ///应收期货公司费用
    TThostFtdcFutureFeeType BrokerFee;
    ///发送方给接收方的消息
    TThostFtdcAddInfoType Message;
    ///摘要
    TThostFtdcDigestType Digest;
    ///银行帐号类型
    TThostFtdcBankAccTypeType BankAccType;
    ///渠道标志
    TThostFtdcDeviceIDType DeviceID;
    ///期货单位帐号类型
    TThostFtdcBankAccTypeType BankSecuAccType;
    ///期货公司银行编码
    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///交易ID
    TThostFtdcTIDType TID;
    ///转账交易状态
    TThostFtdcTransferStatusType TransferStatus;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.92. OnRtnOpenAccountByBank 方法

银行发起银期开户通知。

函数原型：

```
void OnRtnOpenAccountByBank(CThostFtdcOpenAccountField *pOpenAccount);
```

参数:

pOpenAccount: 指向银期开户信息结构。

银期开户信息结构:

```
struct CThostFtdcOpenAccountField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType LastFragment;
    ///会话号
    TThostFtdcSessionIDType SessionID;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///证件类型
    TThostFtdcIdCardTypeType IdCardType;
    ///证件号码
    TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
    ///性别
    TThostFtdcGenderType Gender;
    ///国家代码
    TThostFtdcCountryCodeType CountryCode;
    ///客户类型
```

```

TThostFtdcCustTypeType    CustType;
///地址
TThostFtdcAddressType Address;
///邮编
TThostFtdcZipCodeType     ZipCode;
///电话号码
TThostFtdcTelephoneType   Telephone;
///手机
TThostFtdcMobilePhoneType MobilePhone;
///传真
TThostFtdcFaxType         Fax;
///电子邮件
TThostFtdcEMailType       EMail;
///资金账户状态
TThostFtdcMoneyAccountStatusType MoneyAccountStatus;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType    BankPassWord;
///投资者帐号
TThostFtdcAccountIDType   AccountID;
///期货密码
TThostFtdcPasswordType    Password;
///安装编号
TThostFtdcInstallIDType   InstallID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType  CurrencyID;
///汇钞标志
TThostFtdcCashExchangeCodeType CashExchangeCode;
///摘要
TThostFtdcDigestType      Digest;
///银行帐号类型
TThostFtdcBankAccTypeType  BankAccType;
///渠道标志
TThostFtdcDeviceIDType     DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType  BankSecuAccType;
///期货公司银行编码

```

```

    TThostFtdcBankCodingForFutureType BrokerIDByBank;
    ///期货单位帐号
    TThostFtdcBankAccountType BankSecuAcc;
    ///银行密码标志
    TThostFtdcPwdFlagType    BankPwdFlag;
    ///期货资金密码核对标志
    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员
    TThostFtdcOperNoType OperNo;
    ///交易ID
    TThostFtdcTIDType    TID;
    ///用户标识
    TThostFtdcUserIDType UserID;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
};

```

4.3.93. OnRtnCancelAccountByBank 方法

银行发起银期销户通知。

函数原型：

```
void OnRtnCancelAccountByBank(CThostFtdcCancelAccountField *pCancelAccount);
```

参数：

pCancelAccount: 指向银期销户信息结构。

银期销户信息结构：

```

struct CThostFtdcCancelAccountField
{
    ///业务功能码
    TThostFtdcTradeCodeType    TradeCode;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType    BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType    BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType    BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType    TradeDate;
};

```

```

///交易时间
TThostFtdcTradeTimeType   TradeTime;
///银行流水号
TThostFtdcBankSerialType   BankSerial;
///交易系统日期
TThostFtdcTradeDateType    TradingDay;
///银期平台消息流水号
TThostFtdcSerialType       PlateSerial;
///最后分片标志
TThostFtdcLastFragmentType   LastFragment;
///会话号
TThostFtdcSessionIDType     SessionID;
///客户姓名
TThostFtdcIndividualNameType CustomerName;
///证件类型
TThostFtdcIdCardTypeType    IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///性别
TThostFtdcGenderType        Gender;
///国家代码
TThostFtdcCountryCodeType    CountryCode;
///客户类型
TThostFtdcCustTypeType       CustType;
///地址
TThostFtdcAddressType        Address;
///邮编
TThostFtdcZipCodeType        ZipCode;
///电话号码
TThostFtdcTelephoneType      Telephone;
///手机
TThostFtdcMobilePhoneType     MobilePhone;
///传真
TThostFtdcFaxType            Fax;
///电子邮件
TThostFtdcEMailType          EMail;
///资金账户状态
TThostFtdcMoneyAccountStatusType MoneyAccountStatus;
///银行帐号
TThostFtdcBankAccountType     BankAccount;

```



```

///银行密码
TThostFtdcPasswordType   BankPassWord;
///投资者帐号
TThostFtdcAccountIDType   AccountID;
///期货密码
TThostFtdcPasswordType   Password;
///安装编号
TThostFtdcInstallIDType   InstallID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType   VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType   CurrencyID;
///汇钞标志
TThostFtdcCashExchangeCodeType   CashExchangeCode;
///摘要
TThostFtdcDigestType   Digest;
///银行帐号类型
TThostFtdcBankAccTypeType   BankAccType;
///渠道标志
TThostFtdcDeviceIDType   DeviceID;
///期货单位帐号类型
TThostFtdcBankAccTypeType   BankSecuAccType;
///期货公司银行编码
TThostFtdcBankCodingForFutureType   BrokerIDByBank;
///期货单位帐号
TThostFtdcBankAccountType   BankSecuAcc;
///银行密码标志
TThostFtdcPwdFlagType   BankPwdFlag;
///期货资金密码核对标志
TThostFtdcPwdFlagType   SecuPwdFlag;
///交易柜员
TThostFtdcOperNoType   OperNo;
///交易ID
TThostFtdcTIDType   TID;
///用户标识
TThostFtdcUserIDType   UserID;
///错误代码
TThostFtdcErrorIDType   ErrorID;
///错误信息
TThostFtdcErrorMsgType   ErrorMsg;

```

```
};
```

4.3.94. OnRtnChangeAccountByBank 方法

银行发起变更银行账号通知。

函数原型：

```
void OnRtnChangeAccountByBank(CThostFtdcChangeAccountField *pChangeAccount);
```

参数：

pChangeAccount：指向银行账号变更通知结构。

银行帐号变更通知结构：

```
struct CThostFtdcChangeAccountField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType LastFragment;
    ///会话号
    TThostFtdcSessionIDType SessionID;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///证件类型
    TThostFtdcIdCardTypeType IdCardType;
    ///证件号码
```

```

TThostFtdcIdentifiedCardNoType IdentifiedCardNo;
///性别
TThostFtdcGenderType Gender;
///国家代码
TThostFtdcCountryCodeType CountryCode;
///客户类型
TThostFtdcCustTypeType CustType;
///地址
TThostFtdcAddressType Address;
///邮编
TThostFtdcZipCodeType ZipCode;
///电话号码
TThostFtdcTelephoneType Telephone;
///手机
TThostFtdcMobilePhoneType MobilePhone;
///传真
TThostFtdcFaxType Fax;
///电子邮件
TThostFtdcEMailType EMail;
///资金账户状态
TThostFtdcMoneyAccountStatusType MoneyAccountStatus;
///银行帐号
TThostFtdcBankAccountType BankAccount;
///银行密码
TThostFtdcPasswordType BankPassWord;
///新银行帐号
TThostFtdcBankAccountType NewBankAccount;
///新银行密码
TThostFtdcPasswordType NewBankPassWord;
///投资者帐号
TThostFtdcAccountIDType AccountID;
///期货密码
TThostFtdcPasswordType Password;
///银行帐号类型
TThostFtdcBankAccTypeType BankAccType;
///安装编号
TThostFtdcInstallIDType InstallID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;
///币种代码

```

```

TThostFtdcCurrencyIDType CurrencyID;
///期货公司银行编码
TThostFtdcBankCodingForFutureType BrokerIDByBank;
///银行密码标志
TThostFtdcPwdFlagType BankPwdFlag;
///期货资金密码核对标志
TThostFtdcPwdFlagType SecuPwdFlag;
///交易ID
TThostFtdcTIDType TID;
///摘要
TThostFtdcDigestType Digest;
///错误代码
TThostFtdcErrorIDType ErrorID;
///错误信息
TThostFtdcErrorMsgType ErrorMsg;
};

```

4.3.95. OnRtnCFMMCTradingAccountToken 方法[暂不支持]

保证金监控中心用户令牌。

函数原型：

```

void OnRtnCFMMCTradingAccountToken(
    CThostFtdcCFMMCTradingAccountTokenField *
    pCFMMCTradingAccountToken);

```

参数：

pCFMMCTradingAccountToken： 指向监控中心用户令牌结构的地址。

监控中心用户令牌结构：

```

struct CThostFtdcCFMMCTradingAccountTokenField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///经纪公司统一编码
    TThostFtdcParticipantIDType ParticipantID;
    ///投资者帐号
    TThostFtdcAccountIDType AccountID;
    ///密钥编号
    TThostFtdcSequenceNoType KeyID;

```

```

    ///动态令牌
    TThostFtdcCFMMCTokenType    Token;
};

```

4.3.96. OnRspLockInsert 方法

锁定应答。当客户端发出过锁定录入指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspLockInsert (
    CThostFtdcInputLockField *pInputLock,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInputLock: 指向锁定录入结构的地址。

锁定录入结构：

```

struct CThostFtdcInputLockField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///锁定引用
    TThostFtdcOrderRefType    LockRef;
    ///用户代码
    TThostFtdcUserIDType  UserID;
    ///数量
    TThostFtdcVolumeType  Volume;
    ///请求编号
    TThostFtdcRequestIDType  RequestID;
    ///业务单元

```

```

    TThostFtdcBusinessUnitType BusinessUnit;
    ///锁定类型

    TThostFtdcLockTypeType    LockType;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;

};

```

4.3.97. OnRspCombActionInsert 方法

申请组合录入请求应答。当客户端发出过申请组合录入请求指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pInputCombAction： 指向申请组合录入请求结构的地址。

申请组合录入请求结构：

```

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

```

```

        ///数量
        TThostFtdcVolumeType Volume;
        ///组合指令方向
        TThostFtdcCombDirectionType CombDirection;
        ///投机套保标志
        TThostFtdcHedgeFlagType HedgeFlag;
        ///交易所代码
        TThostFtdcExchangeIDType ExchangeID;
    };

```

4.3.98. OnRspQryProductExchRate 方法

请求查询产品报价汇率响应。当客户端发出请求查询产品报价汇率指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryProductExchRate (
    CThostFtdcProductExchRateField *pProductExchRate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pProductExchRate: 指向产品报价汇率查询结构的地址。

产品报价汇率查询结构：

```

struct CThostFtdcProductExchRateField
{
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///报价币种类型
    TThostFtdcCurrencyIDType QuoteCurrencyID;
    ///汇率
    TThostFtdcExchangeRateType ExchangeRate;
};

```

4.3.99. OnRspQryProductGroup 方法

请求查询产品组响应。当客户端发出请求查询产品组指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryProductGroup(
    CThostFtdcProductGroupField *pProductGroup,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pProductGroup：指向投资者品种/跨品种保证金产品组结构的地址。

投资者品种/跨品种保证金产品组结构：

```
struct CThostFtdcProductGroupField
{
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品组代码
    TThostFtdcInstrumentIDType ProductGroupID;
};
```

4.3.100. OnRspQryInstrumentOrderCommRate 方法

请求查询报单手续费响应。当客户端发出请求查询报单手续费指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryInstrumentOrderCommRate(
    CThostFtdcInstrumentOrderCommRateField *pInstrumentOrderCommRate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pInstrumentOrderCommRate：指向当前报单手续费的详细内容结构地址。

当前报单手续费的详细内容结构

```
struct CThostFtdcInstrumentOrderCommRateField
{
```



```

    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投资者范围
    TThostFtdcInvestorRangeType InvestorRange;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///报单手续费
    TThostFtdcRatioType OrderCommByVolume;
    ///撤单手续费
    TThostFtdcRatioType OrderActionCommByVolume;
};

```

4.3.101. OnRspQryOptionInstrTradeCost 方法

请求查询期权交易成本响应。当客户端发出请求查询期权交易成本指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pOptionInstrTradeCost： 指向期权交易成本查询结构的地址。

期权交易成本查询结构：

```

struct CThostFtdcOptionInstrTradeCostField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

```

```

    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///期权合约保证金不变部分
    TThostFtdcMoneyType FixedMargin;
    ///期权合约最小保证金
    TThostFtdcMoneyType MiniMargin;
    ///期权合约权利金
    TThostFtdcMoneyType Royalty;
    ///交易所期权合约保证金不变部分
    TThostFtdcMoneyType ExchFixedMargin;
    ///交易所期权合约最小保证金
    TThostFtdcMoneyType ExchMiniMargin;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.102. OnRspQryOptionInstrCommRate 方法

请求查询期权合约手续费响应。当客户端发出请求查询期权合约手续费指令后，Kingstar 服务器返回响应是，该方法会被调用。

函数原型：

```

void OnRspQryOptionInstrCommRate(
    CThostFtdcOptionInstrCommRateField *pOptionInstrCommRate,

    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pOptionInstrCommRate: 指向当前期权合约手续费的详细内容结构的地址

当前期权合约手续费的详细内容结构：

```

struct CThostFtdcOptionInstrCommRateField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投资者范围
    TThostFtdcInvestorRangeType InvestorRange;

```

```

    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///开仓手续费率
    TThostFtdcRatioType    OpenRatioByMoney;
    ///开仓手续费
    TThostFtdcRatioType    OpenRatioByVolume;
    ///平仓手续费率
    TThostFtdcRatioType    CloseRatioByMoney;
    ///平仓手续费
    TThostFtdcRatioType    CloseRatioByVolume;
    ///平今手续费率
    TThostFtdcRatioType    CloseTodayRatioByMoney;
    ///平今手续费
    TThostFtdcRatioType    CloseTodayRatioByVolume;
    ///执行手续费率
    TThostFtdcRatioType    StrikeRatioByMoney;
    ///执行手续费
    TThostFtdcRatioType    StrikeRatioByVolume;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.103. OnRspQryExecOrder 方法

请求查询执行宣告响应。当客户端发出请求查询执行宣告指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryExecOrder(
    CThostFtdcExecOrderField *pExecOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

PExecOrder: 指向执行宣告结构地址。

执行宣告结构：

```

struct CThostFtdcExecOrderField
{
    ///经纪公司代码

```

```

TThostFtdcBrokerIDType   BrokerID;
///投资者代码

TThostFtdcInvestorIDType  InvestorID;
///合约代码

TThostFtdcInstrumentIDType InstrumentID;
///执行宣告引用

TThostFtdcOrderRefType    ExecOrderRef;
///用户代码

TThostFtdcUserIDType      UserID;
///数量

TThostFtdcVolumeType      Volume;
///请求编号

TThostFtdcRequestIDType   RequestID;
///业务单元

TThostFtdcBusinessUnitType BusinessUnit;
///开平标志

TThostFtdcOffsetFlagType  OffsetFlag;
///投机套保标志

TThostFtdcHedgeFlagType   HedgeFlag;
///执行类型

TThostFtdcActionTypeType  ActionType;
///保留头寸申请的持仓方向

TThostFtdcPosiDirectionType PosiDirection;
///期权行权后是否保留期货头寸的标记

TThostFtdcExecOrderPositionFlagType ReservePositionFlag;
///期权行权后生成的头寸是否自动平仓

TThostFtdcExecOrderCloseFlagType CloseFlag;
///本地执行宣告编号

TThostFtdcOrderLocalIDType ExecOrderLocalID;
///交易所代码

TThostFtdcExchangeIDType  ExchangeID;
///会员代码

TThostFtdcParticipantIDType ParticipantID;
///客户代码

TThostFtdcClientIDType     ClientID;
///合约在交易所的代码

TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所交易员代码

TThostFtdcTraderIDType     TraderID;
///安装编号

```

```

    TThostFtdcInstallIDType    InstallID;
    ///执行宣告提交状态
    TThostFtdcOrderSubmitStatusType    OrderSubmitStatus;
    ///报单提示序号
    TThostFtdcSequenceNoType    NotifySequence;
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///结算编号
    TThostFtdcSettlementIDType    SettlementID;
    ///执行宣告编号
    TThostFtdcExecOrderSysIDType    ExecOrderSysID;
    ///报单日期
    TThostFtdcDateType    InsertDate;
    ///插入时间
    TThostFtdcTimeType    InsertTime;
    ///撤销时间
    TThostFtdcTimeType    CancelTime;
    ///执行结果
    TThostFtdcExecResultType    ExecResult;
    ///结算会员编号
    TThostFtdcParticipantIDType    ClearingPartID;
    ///序号
    TThostFtdcSequenceNoType    SequenceNo;
    ///前置编号
    TThostFtdcFrontIDType    FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///用户端产品信息
    TThostFtdcProductInfoType    UserProductInfo;
    ///状态信息
    TThostFtdcErrorMsgType    StatusMsg;
    ///操作用户代码
    TThostFtdcUserIDType    ActiveUserID;
    ///经纪公司报单编号
    TThostFtdcSequenceNoType    BrokerExecOrderSeq;
    ///营业部编号
    TThostFtdcBranchIDType    BranchID;
};

```

4.3.104. OnRspQryLock 方法

请求查询锁定响应。当客户端发出请求查询锁定指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryLock (
    CThostFtdcLockField *pLock,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pLock： 指向锁定查询结构的地址。

锁定查询结构：

```
struct CThostFtdcLockField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///锁定引用
    TThostFtdcOrderRefType   LockRef;
    ///用户代码
    TThostFtdcUserIDType   UserID;
    ///数量
    TThostFtdcVolumeType   Volume;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///锁定类型
    TThostFtdcLockTypeType   LockType;
```

```
///本地锁定编号
TThostFtdcOrderLocalIDType    LockLocalID;
///交易所代码
TThostFtdcExchangeIDType    ExchangeID;
///会员代码
TThostFtdcParticipantIDType    ParticipantID;
///客户代码
TThostFtdcClientIDType    ClientID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType    ExchangeInstID;
///交易所交易员代码
TThostFtdcTraderIDType    TraderID;
///安装编号
TThostFtdcInstallIDType    InstallID;
///执行宣告提交状态
TThostFtdcOrderSubmitStatusType    OrderSubmitStatus;
///报单提示序号
TThostFtdcSequenceNoType    NotifySequence;
///交易日
TThostFtdcDateType    TradingDay;
///结算编号
TThostFtdcSettlementIDType    SettlementID;
///锁定编号
TThostFtdcOrderSysIDType    LockSysID;
///报单日期
TThostFtdcDateType    InsertDate;
///插入时间
TThostFtdcTimeType    InsertTime;
///撤销时间
TThostFtdcTimeType    CancelTime;
///锁定状态
TThostFtdcOrderActionStatusType    LockStatus;
```

```

    ///结算会员编号
    TThostFtdcParticipantIDType    ClearingPartID;
    ///序号
    TThostFtdcSequenceNoType    SequenceNo;
    ///前置编号
    TThostFtdcFrontIDType    FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///用户端产品信息
    TThostFtdcProductInfoType    UserProductInfo;
    ///状态信息
    TThostFtdcErrorMsgType    StatusMsg;
    ///操作用户代码
    TThostFtdcUserIDType    ActiveUserID;
    ///经纪公司报单编号
    TThostFtdcSequenceNoType    BrokerLockSeq;
    ///营业部编号
    TThostFtdcBranchIDType    BranchID;
};

```

4.3.105. OnRspQryLockPosition 方法

请求查询锁定证券仓位响应。当客户端发出请求查询锁定证券仓位指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryLockPosition (
    CThostFtdcLockPositionField *pLockPosition,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pLockPosition： 指向锁定证券仓位查询结构的地址。

锁定证券仓位查询结构：

```

struct CThostFtdcLockPositionField

```



```

{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///数量
    TThostFtdcVolumeType Volume;
    ///冻结数量
    TThostFtdcVolumeType FrozenVolume;
};

```

4.3.106. OnRspQryCombInstrumentGuard 方法

请求查询组合合约安全系数响应。当客户端发出请求查询组合合约安全系数指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryCombInstrumentGuard (
    CThostFtdcCombInstrumentGuardField *pCombInstrumentGuard,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pCombInstrumentGuard： 指向组合合约安全系数查询结构的地址。

组合合约安全系数查询结构：

```

struct CThostFtdcCombInstrumentGuardField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

```

```

    ///安全系数
    TThostFtdcRatioType    GuarantRatio;

};

```

4.3.107. OnRspQryCombAction 方法

请求查询申请组合响应。当客户端发出请求查询申请组合指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

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

```

参数：

pCombAction： 指向申请组合查询结构的地址。

申请组合查询结构：

```

struct CThostFtdcCombActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;

    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;

    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

    ///组合引用
    TThostFtdcOrderRefType    CombActionRef;

    ///用户代码
    TThostFtdcUserIDType    UserID;

    ///买卖方向
    TThostFtdcDirectionType    Direction;

    ///数量
    TThostFtdcVolumeType    Volume;

    ///组合指令方向

```

```
TThostFtdcCombDirectionType CombDirection;  
///投机套保标志  
TThostFtdcHedgeFlagType HedgeFlag;  
///本地申请组合编号  
TThostFtdcOrderLocalIDType ActionLocalID;  
///交易所代码  
TThostFtdcExchangeIDType ExchangeID;  
///会员代码  
TThostFtdcParticipantIDType ParticipantID;  
///客户代码  
TThostFtdcClientIDType ClientID;  
///合约在交易所的代码  
TThostFtdcExchangeInstIDType ExchangeInstID;  
///交易所交易员代码  
TThostFtdcTraderIDType TraderID;  
///安装编号  
TThostFtdcInstallIDType InstallID;  
///组合状态  
TThostFtdcOrderActionStatusType ActionStatus;  
///报单提示序号  
TThostFtdcSequenceNoType NotifySequence;  
///交易日  
TThostFtdcDateType TradingDay;  
///结算编号  
TThostFtdcSettlementIDType SettlementID;  
///序号  
TThostFtdcSequenceNoType SequenceNo;  
///前置编号  
TThostFtdcFrontIDType FrontID;  
///会话编号  
TThostFtdcSessionIDType SessionID;  
///用户端产品信息
```

```

    TThostFtdcProductInfoType UserProductInfo;
    ///状态信息
    TThostFtdcErrorMsgType StatusMsg;
};

```

4.3.108. OnRtnLock 方法

锁定通知。

函数原型：

```
void OnRtnLock (CThostFtdcLockField *pLock);
```

参数：

pLock: 指向锁定通知结构。

锁定通知结构如下：

```

struct CThostFtdcLockField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///锁定引用
    TThostFtdcOrderRefType LockRef;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///数量
    TThostFtdcVolumeType Volume;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///锁定类型
    TThostFtdcLockTypeType LockType;
    ///本地锁定编号

```

```
TThostFtdcOrderLocalIDType    LockLocalID;  
///交易所代码  
  
TThostFtdcExchangeIDType ExchangeID;  
///会员代码  
  
TThostFtdcParticipantIDType    ParticipantID;  
///客户代码  
  
TThostFtdcClientIDType    ClientID;  
///合约在交易所的代码  
  
TThostFtdcExchangeInstIDType ExchangeInstID;  
///交易所交易员代码  
  
TThostFtdcTraderIDType    TraderID;  
///安装编号  
  
TThostFtdcInstallIDType    InstallID;  
///执行宣告提交状态  
  
TThostFtdcOrderSubmitStatusType    OrderSubmitStatus;  
///报单提示序号  
  
TThostFtdcSequenceNoType NotifySequence;  
///交易日  
  
TThostFtdcDateType    TradingDay;  
///结算编号  
  
TThostFtdcSettlementIDType SettlementID;  
///锁定编号  
  
TThostFtdcOrderSysIDType LockSysID;  
///报单日期  
  
TThostFtdcDateType    InsertDate;  
///插入时间  
  
TThostFtdcTimeType    InsertTime;  
///撤销时间  
  
TThostFtdcTimeType    CancelTime;  
///锁定状态  
  
TThostFtdcOrderActionStatusType    LockStatus;  
///结算会员编号
```

```

    TThostFtdcParticipantIDType    ClearingPartID;
    ///序号

    TThostFtdcSequenceNoType    SequenceNo;
    ///前置编号

    TThostFtdcFrontIDType    FrontID;
    ///会话编号

    TThostFtdcSessionIDType    SessionID;
    ///用户端产品信息

    TThostFtdcProductInfoType    UserProductInfo;
    ///状态信息

    TThostFtdcErrorMsgType    StatusMsg;
    ///操作用户代码

    TThostFtdcUserIDType    ActiveUserID;
    ///经纪公司报单编号

    TThostFtdcSequenceNoType    BrokerLockSeq;
    ///营业部编号

    TThostFtdcBranchIDType    BranchID;
};

```

4.3.109. OnErrRtnLockInsert 方法

锁定错误通知。当客户端发出锁定错误指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnErrRtnLockInsert (
    CThostFtdcInputLockField *pInputLock,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInputLock：指向申请锁定错误结构的地址。

锁定错误结构：

```

struct CThostFtdcInputLockField
{

```

```

    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///锁定引用
    TThostFtdcOrderRefType LockRef;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///数量
    TThostFtdcVolumeType Volume;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///锁定类型
    TThostFtdcLockTypeType LockType;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.3.110. OnRtnCombAction 方法

组合通知。

函数原型：

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

参数：

pCombAction： 指向组合通知结构。

组合通知结构如下：

```

struct CThostFtdcCombActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;

```

```
///投资者代码
TThostFtdcInvestorIDType InvestorID;

///合约代码
TThostFtdcInstrumentIDType InstrumentID;

///组合引用
TThostFtdcOrderRefType CombActionRef;

///用户代码
TThostFtdcUserIDType UserID;

///买卖方向
TThostFtdcDirectionType Direction;

///数量
TThostFtdcVolumeType Volume;

///组合指令方向
TThostFtdcCombDirectionType CombDirection;

///投机套保标志
TThostFtdcHedgeFlagType HedgeFlag;

///本地申请组合编号
TThostFtdcOrderLocalIDType ActionLocalID;

///交易所代码
TThostFtdcExchangeIDType ExchangeID;

///会员代码
TThostFtdcParticipantIDType ParticipantID;

///客户代码
TThostFtdcClientIDType ClientID;

///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;

///交易所交易员代码
TThostFtdcTraderIDType TraderID;

///安装编号
TThostFtdcInstallIDType InstallID;

///组合状态
TThostFtdcOrderActionStatusType ActionStatus;
```



```

    ///报单提示序号
    TThostFtdcSequenceNoType NotifySequence;

    ///交易日
    TThostFtdcDateType TradingDay;

    ///结算编号
    TThostFtdcSettlementIDType SettlementID;

    ///序号
    TThostFtdcSequenceNoType SequenceNo;

    ///前置编号
    TThostFtdcFrontIDType FrontID;

    ///会话编号
    TThostFtdcSessionIDType SessionID;

    ///用户端产品信息
    TThostFtdcProductInfoType UserProductInfo;

    ///状态信息
    TThostFtdcErrorMsgType StatusMsg;

};

```

4.3.111. OnErrRtnCombActionInsert 方法

申请组合录入错误通知。当客户端发出申请组合录入错误指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnErrRtnCombActionInsert (
    CThostFtdcInputCombActionField *pInputCombAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInputCombAction： 指向申请组合录入错误结构的地址。

申请组合录入错误结构：

```

struct CThostFtdcInputCombActionField
{
    ///经纪公司代码

```

```

    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码

    TThostFtdcInstrumentIDTypeInstrumentID;
    ///组合引用

    TThostFtdcOrderRefType    CombActionRef;
    ///用户代码

    TThostFtdcUserIDType     UserID;
    ///买卖方向

    TThostFtdcDirectionType  Direction;
    ///数量

    TThostFtdcVolumeType     Volume;
    ///组合指令方向

    TThostFtdcCombDirectionType  CombDirection;
    ///投机套保标志

    TThostFtdcHedgeFlagType  HedgeFlag;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;

};

```

4.3.112. OnRspQryRiskSettleInvstPosition 方法

请求查询投资者风险结算持仓响应。当客户端发出请求投资者风险结算持仓指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryRiskSettleInvstPosition(
    CThostFtdcRiskSettleInvstPositionField *pRiskSettleInvstPosition,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pRiskSettleInvstPosition： 指向投资者风险结算持仓查询结构的地址。

投资者风险结算持仓查询结构：

```

struct CThostFtdcRiskSettleInvstPositionField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
    /// 持仓多空方向
    TThostFtdcPosiDirectionType PosiDirection;
    /// 投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    /// 持仓日期
    TThostFtdcPositionDateType PositionDate;
    /// 上日持仓
    TThostFtdcVolumeType YdPosition;
    /// 今日持仓
    TThostFtdcVolumeType Position;
    /// 多头冻结
    TThostFtdcVolumeType LongFrozen;
    /// 空头冻结
    TThostFtdcVolumeType ShortFrozen;
    /// 开仓冻结金额
    TThostFtdcMoneyType LongFrozenAmount;
    /// 开仓冻结金额
    TThostFtdcMoneyType ShortFrozenAmount;
    /// 开仓量
    TThostFtdcVolumeType OpenVolume;
    /// 平仓量
    TThostFtdcVolumeType CloseVolume;
    /// 开仓金额
    TThostFtdcMoneyType OpenAmount;

```

```
///平仓金额
TThostFtdcMoneyType CloseAmount;
///持仓成本
TThostFtdcMoneyType PositionCost;
///上次占用的保证金
TThostFtdcMoneyType PreMargin;
///占用的保证金
TThostFtdcMoneyType UseMargin;
///冻结的保证金
TThostFtdcMoneyType FrozenMargin;
///冻结的资金
TThostFtdcMoneyType FrozenCash;
///冻结的手续费
TThostFtdcMoneyType FrozenCommission;
///资金差额
TThostFtdcMoneyType CashIn;
///手续费
TThostFtdcMoneyType Commission;
///平仓盈亏
TThostFtdcMoneyType CloseProfit;
///持仓盈亏
TThostFtdcMoneyType PositionProfit;
///上次结算价
TThostFtdcPriceType PreSettlementPrice;
///本次结算价
TThostFtdcPriceType SettlementPrice;
///交易日
TThostFtdcDateType TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;
///开仓成本
TThostFtdcMoneyType OpenCost;
```

```

///交易所保证金
TThostFtdcMoneyType ExchangeMargin;
///组合成交形成的持仓
TThostFtdcVolumeType CombPosition;
///组合多头冻结
TThostFtdcVolumeType CombLongFrozen;
///组合空头冻结
TThostFtdcVolumeType CombShortFrozen;
///逐日盯市平仓盈亏
TThostFtdcMoneyType CloseProfitByDate;
///逐笔对冲平仓盈亏
TThostFtdcMoneyType CloseProfitByTrade;
///今日持仓
TThostFtdcVolumeType TodayPosition;
///保证金率
TThostFtdcRatioType MarginRateByMoney;
///保证金率(按手数)
TThostFtdcRatioType MarginRateByVolume;
///执行冻结
TThostFtdcVolumeType StrikeFrozen;
///执行冻结金额
TThostFtdcMoneyType StrikeFrozenAmount;
///放弃执行冻结
TThostFtdcVolumeType AbandonFrozen;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///执行冻结的昨仓
TThostFtdcVolumeType YdStrikeFrozen;
///投资单元代码
TThostFtdcInvestUnitIDType InvestUnitID;
///大商所持仓成本差值，只有大商所使用
TThostFtdcMoneyType PositionCostOffset;

```

```

    ///tas 持仓手数
    TThostFtdcVolumeType TasPosition;

    ///tas 持仓成本
    TThostFtdcMoneyType TasPositionCost;

};

```

4.3.113. OnRspQryRiskSettleProductStatus 方法

请求查询风险结算产品响应。当客户端发出请求风险结算产品指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryRiskSettleProductStatus(
    CThostFtdcRiskSettleProductStatusField *pRiskSettleProductStatus,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pRiskSettleProductStatus： 指向风险结算产品查询结构的地址。

风险结算产品查询结构：

```

struct CThostFtdcRiskSettleProductStatusField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;

    ///产品编号
    TThostFtdcInstrumentIDType ProductID;

    ///产品结算状态
    TThostFtdcProductStatusType ProductStatus;

    ///是否追平完成
    TThostFtdcSyncDeltaStatusType SyncDeltaStatus;

};

```

4.3.114. OnRspOptionSelfCloseAction 方法

期权自对冲操作请求应答。当客户端发出过期权自对冲操作请求指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
void OnRspOptionSelfCloseAction(
    CThostFtdcInputOptionSelfCloseActionField *pInputOptionSelfCloseAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pInputOptionSelfCloseAction: 指向期权自对冲操作请求请求结构的地址。

期权自对冲操作请求结构:

```
struct CThostFtdcInputOptionSelfCloseActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///期权自对冲操作引用
    TThostFtdcOrderActionRefType  OptionSelfCloseActionRef;
    ///期权自对冲引用
    TThostFtdcOrderRefType    OptionSelfCloseRef;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///期权自对冲操作编号
    TThostFtdcOrderSysIDType  OptionSelfCloseSysID;
    ///操作标志
    TThostFtdcActionFlagType  ActionFlag;
    ///用户代码
```

```

    TThostFtdcUserIDType UserID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投资单元代码
    TThostFtdcInvestUnitIDType InvestUnitID;
    ///IP 地址
    TThostFtdcIPAddressType IPAddress;
    ///Mac 地址
    TThostFtdcMacAddressType MacAddress;
};

```

4.3.115. OnRspQryCustomerSuitableEvaResults 方法

请求查询客户适当性评测结果响应。当客户端发出请求查询客户适当性评测结果指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryCustomerSuitableEvaResults(
    CThostFtdcCustomerSuitableEvaResultsField *pCustomerSuitableEvaResults,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pCustomerSuitableEvaResults：指向客户适当性评测结果查询结构的地址。

客户适当性评测结果查询结构：

```

struct CThostFtdcCustomerSuitableEvaResultsField
{
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///有效期
    TThostFtdcDateType ExpiryDate;
};

```


4.3.116. OnRspQrySPBMFutureParameter 方法

SPBM 期货合约参数查询响应。当客户端发出 SPBM 期货合约参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQrySPBMFutureParameter(CThostFtdcSPBMFutureParameterField
    *pSPBMFutureParameter, CThostFtdcRspInfoField *pRspInfo,
    int nRequestID, bool bIsLast);
```

参数：

pSPBMFutureParameter：指向 SPBM 期货合约保证金参数地址。

期货合约保证金参数结构：

```
struct CThostFtdcSPBMFutureParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///期货合约因子
    TThostFtdcVolumeMultipleType    Cvf;
    ///阶段标识
    TThostFtdcTimeRangeType    TimeRange;
    ///品种保证金标准
    TThostFtdcRatioType    MarginRate;
    ///期货合约内部对锁仓费率折扣比例
    TThostFtdcRatioType    LockRateX;
    ///提高保证金标准
    TThostFtdcRatioType    AddOnRate;
    ///昨结算价
    TThostFtdcPriceType    PreSettlementPrice;
```

```

        ///期货合约内部附加费率折扣比例 X2
        TThostFtdcRatioType AddOnLockRateX2;
    };

```

4.3.117. OnRspQrySPBMOptionParameter 方法

SPBM 期权合约参数查询响应。当客户端发出 SPBM 期权合约参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQrySPBMOptionParameter(CThostFtdcSPBMOptionParameterField
    *pSPBMOptionParameter, CThostFtdcRspInfoField *pRspInfo, int nRequestID,
    bool bIsLast);

```

参数：

pSPBMFutureParameter：指向 SPBM 期货合约保证金参数地址。

期权合约保证金参数结构：

```

struct CThostFtdcSPBMOptionParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///期权合约因子
    TThostFtdcVolumeMultipleType    Cvf;
    ///期权冲抵价格
    TThostFtdcPriceType    DownPrice;
    ///Delta 值
    TThostFtdcDeltaType    Delta;
    ///卖方期权风险转换最低值

```

```

    TThostFtdcDeltaType    SlimiDelta;
    ///昨结算价

    TThostFtdcPriceType    PreSettlementPrice;

};

```

4.3.118. OnRspQrySPBMIntraParameter 方法

SPBM 品种内对锁仓折扣参数查询响应。当客户端发出 SPBM 品种内对锁仓折扣参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQrySPBMIntraParameter(CThostFtdcSPBMIntraParameterField
    *pSPBMIntraParameter, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);

```

参数：

pSPBMIntraParameter: SPBM 品种内对锁仓折扣参数地址。

SPBM 品种内对锁仓折扣参数结构：

```

struct CThostFtdcSPBMIntraParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///品种代码
    TThostFtdcInstrumentIDType    ProdFamilyCode;
    ///品种内合约间对锁仓费率折扣比例
    TThostFtdcRatioType    IntraRateY;
    ///品种内合约间对锁仓费率附加折扣比例
    TThostFtdcRatioType    AddOnIntraRateY2;
};

```

4.3.119. OnRspQrySPBMInterParameter 方法

SPBM 跨品种抵扣参数查询响应。当客户端发出 SPBM 跨品种抵扣参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQrySPBMInterParameter(CThostFtdcSPBMInterParameterField
    *pSPBMInterParameter, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);
```

参数：

pSPBMInterParameter：SPBM 跨品种抵扣参数地址。

SPBM 跨品种抵扣参数结构：

```
struct CThostFtdcSPBMInterParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///优先级
    TThostFtdcSpreadIDType    SpreadID;
    ///品种间对锁仓费率折扣比例
    TThostFtdcRatioType    InterRateZ;
    ///第一腿构成品种
    TThostFtdcInstrumentIDType Leg1ProdFamilyCode;
    ///第二腿构成品种
    TThostFtdcInstrumentIDType Leg2ProdFamilyCode;
};
```

4.3.120. OnRspQrySPBMPortfDefinition 方法

SPBM 组合保证金套餐查询响应。当客户端发出 SPBM 组合保证金套餐查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
OnRspQrySPBMPortfDefinition(CThostFtdcSPBMPortfDefinitionField
    *pSPBMPortfDefinition, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);
```

参数:

pSPBMPortfDefinition: SPBM 组合保证金参数地址。

SPBM 组合保证金参数结构:

```
struct CThostFtdcSPBMPortfDefinitionField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///组合保证金套餐代码
    TThostFtdcPortfolioDefIDType PortfolioDefID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///是否启用 SPBM
    TThostFtdcBoolType IsSPBM;
};
```

4.3.121. OnRspQrySPBMInvestorPortfDef 方法

投资者 SPBM 套餐选择查询响应。当客户端发出投资者 SPBM 套餐选择查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
void OnRspQrySPBMInvestorPortfDef(CThostFtdcSPBMInvestorPortfDefField
    *pSPBMInvestorPortfDef, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);
```

参数:

pSPBMInvestorPortfDef: 投资者 SPBM 套餐选择参数地址。

投资者 SPBM 套餐选择参数结构:

```
struct CThostFtdcSPBMInvestorPortfDefField
{
    ///交易所代码
```

```

    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码

    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType InvestorID;
    ///组合保证金套餐代码

    TThostFtdcPortfolioDefIDType PortfolioDefID;

};

```

4.3.122. OnRspQryInvestorPortfMarginRatio 方法

投资者新型组合保证金系数查询响应。当客户端发出投资者新型组合保证金系数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorPortfMarginRatio(CThostFtdcInvestorPortfMarginRatioField
    *pInvestorPortfMarginRatio, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);

```

参数：

pInvestorPortfMarginRatio：投资者新型组合保证金系数地址。

投资者新型组合保证金参数结构：

```

struct CThostFtdcInvestorPortfMarginRatioField
{
    ///投资者范围

    TThostFtdcInvestorRangeType InvestorRange;
    ///经纪公司代码

    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
    ///会员对投资者收取的保证金和交易所对投资者收取的保证金的比例

    TThostFtdcRatioType MarginRatio;
}

```

```

        ///交易编码
        TThostFtdcClientIDType ClientID;
    };

```

4.3.123. OnRspQryInvestorProdSPBMDetail 方法

投资者产品 SPBM 明细查询响应。当客户端发出投资者产品 SPBM 明细查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorProdSPBMDetail(CThostFtdcInvestorProdSPBMDetailField
    *pInvestorProdSPBMDetail, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);

```

参数：

pInvestorProdSPBMDetail：投资者产品 SPBM 明细地址。

投资者产品 SPBM 明细地址参数结构：

```

struct CThostFtdcInvestorProdSPBMDetailField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///合约内对锁保证金
    TThostFtdcMoneyType IntraInstrMargin;
    ///买归集保证金
    TThostFtdcMoneyType BCollectingMargin;
    ///卖归集保证金
    TThostFtdcMoneyType SCollectingMargin;
    ///品种内合约间对锁保证金

```

```

    TThostFtdcMoneyType IntraProdMargin;
    ///净保证金

    TThostFtdcMoneyType NetMargin;
    ///产品间对锁保证金

    TThostFtdcMoneyType InterProdMargin;
    ///裸保证金

    TThostFtdcMoneyType SingleMargin;
    ///附加保证金

    TThostFtdcMoneyType AddOnMargin;
    ///交割月保证金

    TThostFtdcMoneyType DeliveryMargin;
    ///看涨期权最低风险

    TThostFtdcMoneyType CallOptionMinRisk;
    ///看跌期权最低风险

    TThostFtdcMoneyType PutOptionMinRisk;
    ///卖方期权最低风险

    TThostFtdcMoneyType OptionMinRisk;
    ///买方期权冲抵价值

    TThostFtdcMoneyType OptionValueOffset;
    ///卖方期权权利金

    TThostFtdcMoneyType OptionRoyalty;
    ///价值冲抵

    TThostFtdcMoneyType RealOptionValueOffset;
    ///保证金

    TThostFtdcMoneyType Margin;
    ///交易所保证金

    TThostFtdcMoneyType ExchMargin;
    ///交易编码

    TThostFtdcClientIDType ClientID;
};

```


4.3.124. OnRspQryPortfTradeParamSetting 方法

组保交易参数设置查询响应。当客户端发出组保交易参数设置查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryPortfTradeParamSetting(
    CThostFtdcPortfTradeParamSettingField *pPortfTradeParamSetting,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数：

pPortfTradeParamSetting：组保交易参数设置查询地址。

组保交易参数设置查询结构：

```
struct CThostFtdcPortfTradeParamSettingField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///新型组保算法
    TThostFtdcPortfolioType Portfolio;
    ///撤单是否验资
    TThostFtdcBoolType IsActionVerify;
    ///平仓是否验资
    TThostFtdcBoolType IsCloseVerify;
    ///客户名称
    TThostFtdcCSRCOpenInvestorNameType InvestorName;
    ///交易编码
    TThostFtdcClientIDType ClientID;
};
```

4.3.125. OnRspQrySPBMAAddOnInterParameter 方法

SPBM 跨品种抵扣参数查询响应。当客户端发出 SPBM 跨品种抵扣参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQrySPBMAAddOnInterParameter(CThostFtdcSPBMAAddOnInterParameterField
    *pSPBMAAddOnInterParameter, CThostFtdcRspInfoField *pRspInfo, int nRequestID,
    bool bIsLast);
```

参数：

pSPBMAAddOnInterParameter：SPBM 跨品种附加抵扣参数地址。

SPBM 跨品种附加抵扣参数结构：

```
struct CThostFtdcSPBMAAddOnInterParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///优先级
    TThostFtdcSpreadIDType    SpreadID;
    ///品种间对锁仓费率附加折扣比例
    TThostFtdcRatioType    AddOnInterRateZ2;
    ///第一腿构成品种
    TThostFtdcInstrumentIDType Leg1ProdFamilyCode;
    ///第二腿构成品种
    TThostFtdcInstrumentIDType Leg2ProdFamilyCode;
};
```

4.3.126. OnRspQrySPMMInstParam 方法

SPMM 合约参数查询响应。当客户端发出 SPMM 合约参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
OnRspQrySPMMInstParam(CThostFtdcSPMMInstParamField *pSPMMInstParam,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

参数:

pSPMMInstParam: SPMM 合约参数地址。

SPMM 合约参数结构:

```
struct CThostFtdcSPMMInstParamField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///SPMM 合约保证金算法
    TThostFtdcInstMarginCalIDType InstMarginCalID;
    ///商品组代码
    TThostFtdcSPMMProductIDType CommodityID;
    ///商品群代码
    TThostFtdcSPMMProductIDType CommodityGroupID;
};
```

4.3.127. OnRspQrySPMMProductParam 方法

SPMM 产品参数查询响应。当客户端发出 SPMM 产品参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
void OnRspQrySPMMProductParam(CThostFtdcSPMMProductParamField
    *pSPMMProductParam, CThostFtdcRspInfoField *pRspInfo,
    int nRequestID, bool bIsLast);
```

参数:

pSPMMProductParam: SPMM 产品参数地址。

SPMM 产品参数结构:

```
struct CThostFtdcSPMMProductParamField
{
```

```

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcSPMMProductIDType ProductID;
    ///商品组代码
    TThostFtdcSPMMProductIDType CommodityID;
    ///商品群代码
    TThostFtdcSPMMProductIDType CommodityGroupID;
};

```

4.3.128. OnRspQryInvestorCommoditySPMMMargin 方法

投资者商品组 SPMM 记录查询响应。当客户端发出投资者商品组 SPMM 记录查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorCommoditySPMMMargin(
    CThostFtdcInvestorCommoditySPMMMarginField *pInvestorCommoditySPMMMargin,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

```

参数：

pInvestorCommoditySPMMMargin：投资者商品组 SPMM 记录地址。

投资者商品组 SPMM 记录参数结构：

```

struct CThostFtdcInvestorCommoditySPMMMarginField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///商品组代码
    TThostFtdcSPMMProductIDType CommodityID;
    ///优惠仓位应收保证金

```

```

TThostFtdcMoneyType  MarginBeforeDiscount;
///不优惠仓位应收保证金

TThostFtdcMoneyType  MarginNoDiscount;
///多头实仓风险

TThostFtdcMoneyType  LongPosRisk;
///多头开仓冻结风险

TThostFtdcMoneyType  LongOpenFrozenRisk;
///多头被平冻结风险

TThostFtdcMoneyType  LongCloseFrozenRisk;
///空头实仓风险

TThostFtdcMoneyType  ShortPosRisk;
///空头开仓冻结风险

TThostFtdcMoneyType  ShortOpenFrozenRisk;
///空头被平冻结风险

TThostFtdcMoneyType  ShortCloseFrozenRisk;
///SPMM 品种内跨期优惠系数

TThostFtdcSPMMDiscountRatioType  IntraCommodityRate;
///SPMM 期权优惠系数

TThostFtdcSPMMDiscountRatioType  OptionDiscountRate;
///实仓对冲优惠金额

TThostFtdcMoneyType  PosDiscount;
///开仓报单对冲优惠金额

TThostFtdcMoneyType  OpenFrozenDiscount;
///品种风险净头

TThostFtdcMoneyType  NetRisk;
///平仓冻结保证金

TThostFtdcMoneyType  CloseFrozenMargin;
///冻结的手续费

TThostFtdcMoneyType  FrozenCommission;
///手续费

TThostFtdcMoneyType  Commission;
///冻结的资金

```

```

    TThostFtdcMoneyType FrozenCash;
    ///资金差额

    TThostFtdcMoneyType CashIn;
    ///行权冻结资金

    TThostFtdcMoneyType StrikeFrozenMargin;

};

```

4.3.129. OnRspQryInvestorCommodityGroupSPMMMargin 方法

投资者商品群 SPMM 记录查询响应。当客户端发出投资者商品群 SPMM 记录查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorCommodityGroupSPMMMargin(
    CThostFtdcInvestorCommodityGroupSPMMMarginField
    *pInvestorCommodityGroupSPMMMargin, CThostFtdcRspInfoField *pRspInfo,
    int nRequestID, bool bIsLast);

```

参数：

pInvestorCommodityGroupSPMMMargin：投资者商品群 SPMM 记录地址。

投资者商品群 SPMM 记录参数结构：

```

struct CThostFtdcInvestorCommodityGroupSPMMMarginField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///商品群代码
    TThostFtdcSPMMProductIDType CommodityGroupID;
    ///优惠仓位应收保证金
    TThostFtdcMoneyType MarginBeforeDiscount;
    ///不优惠仓位应收保证金

```

```

    TThostFtdcMoneyType  MarginNoDiscount;
    ///多头风险

    TThostFtdcMoneyType  LongRisk;
    ///空头风险

    TThostFtdcMoneyType  ShortRisk;
    ///商品群平仓冻结保证金

    TThostFtdcMoneyType  CloseFrozenMargin;
    ///SPMM 跨品种优惠系数

    TThostFtdcSPMMDiscountRatioType  InterCommodityRate;
    ///商品群最小保证金比例

    TThostFtdcSPMMDiscountRatioType  MiniMarginRatio;
    ///投资者保证金和交易所保证金的比例

    TThostFtdcRatioType  AdjustRatio;
    ///SPMM 品种内优惠汇总

    TThostFtdcMoneyType  IntraCommodityDiscount;
    ///SPMM 跨品种优惠

    TThostFtdcMoneyType  InterCommodityDiscount;
    ///交易所保证金

    TThostFtdcMoneyType  ExchMargin;
    ///投资者保证金

    TThostFtdcMoneyType  InvestorMargin;
    ///冻结的手续费

    TThostFtdcMoneyType  FrozenCommission;
    ///手续费

    TThostFtdcMoneyType  Commission;
    ///冻结的资金

    TThostFtdcMoneyType  FrozenCash;
    ///资金差额

    TThostFtdcMoneyType  CashIn;
    ///行权冻结资金

    TThostFtdcMoneyType  StrikeFrozenMargin;
};

```

4.3.130. OnRspQryRULEInstrParameter 方法

RULE 合约保证金参数查询响应。当客户端发出 RULE 合约保证金参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```
void OnRspQryRULEInstrParameter(
    CThostFtdcRULEInstrParameterField *pRULEInstrParameter,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

CThostFtdcRULEInstrParameterField : RULE 合约保证金参数地址。

RULE 合约保证金参数结构：

```
struct CThostFtdcRULEInstrParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///合约类型
    TThostFtdcInstrumentClassType InstrumentClass;
    ///标准合约
    TThostFtdcInstrumentIDType StdInstrumentID;
    ///投机买折算系数
    TThostFtdcRatioType    BSpecRatio;
    ///投机卖折算系数
    TThostFtdcRatioType    SSpecRatio;
    ///套保买折算系数
    TThostFtdcRatioType    BHedgeRatio;
    ///套保卖折算系数
    TThostFtdcRatioType    SHedgeRatio;
```



```

    ///买附加风险保证金
    TThostFtdcMoneyType BAddOnMargin;
    ///卖附加风险保证金
    TThostFtdcMoneyType SAddOnMargin;
    ///商品群号
    TThostFtdcCommodityGroupIDType CommodityGroupID;
};

```

4.3.131. OnRspQryRULEIntraParameter 方法

RULE 品种内对锁仓折扣参数查询响应。当客户端发出 RULE 品种内对锁仓折扣参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryRULEIntraParameter(
    CThostFtdcRULEIntraParameterField *pRULEIntraParameter,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

```

参数：

CThostFtdcRULEIntraParameterField : RULE 品种内对锁仓折扣参数地址。

RULE 品种内对锁仓折扣参数结构：

```

struct CThostFtdcRULEIntraParameterField
{
    ///交易日
    TThostFtdcDateType TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///标准合约
    TThostFtdcInstrumentIDType StdInstrumentID;
    ///标准合约保证金
    TThostFtdcMoneyType StdInstrMargin;
    ///一般月份合约组合保证金系数

```

```

    TThostFtdcRatioType    UsualIntraRate;
    ///临近交割合约组合保证金系数
    TThostFtdcRatioType    DeliveryIntraRate;
};

```

4.3.132. OnRspQryRULEInterParameter 方法

RULE 跨品种抵扣参数查询响应。当客户端发出 RULE 跨品种抵扣参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryRULEInterParameter(
    CThostFtdcRULEInterParameterField *pRULEInterParameter,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

```

参数：

CThostFtdcRULEInterParameterField : RULE 跨品种抵扣参数地址。

RULE 跨品种抵扣参数结构：

```

struct CThostFtdcRULEInterParameterField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///优先级
    TThostFtdcSpreadIDType    SpreadId;
    ///品种间对锁仓费率折扣比例
    TThostFtdcRatioType    InterRate;
    ///第一腿构成品种
    TThostFtdcInstrumentIDType    Leg1ProdFamilyCode;
    ///第二腿构成品种
    TThostFtdcInstrumentIDType    Leg2ProdFamilyCode;
    ///腿比例系数
    TThostFtdcCommonIntType    Leg1PropFactor;

```

```

    ///腿比例系数
    TThostFtdcCommonIntType Leg2PropFactor;
    ///商品群号
    TThostFtdcCommodityGroupIDType CommodityGroupID;
    ///商品群名称
    TThostFtdcInstrumentNameType CommodityGroupName;
};

```

4.3.133. OnRspQryInvestorProdRULEMargin 方法

投资者产品 RULE 保证金查询响应。当客户端发出投资者产品 RULE 保证金查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorProdRULEMargin(
    CThostFtdcInvestorProdRULEMarginField *pInvestorProdRULEMargin,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

```

参数：

CThostFtdcInvestorProdRULEMarginField : 投资者产品 RULE 保证金查询参数地址。

投资者产品 RULE 保证金参数结构：

```

struct CThostFtdcInvestorProdRULEMarginField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///合约类型
    TThostFtdcInstrumentClassType InstrumentClass;
    ///商品群号

```

TThostFtdcCommodityGroupIDType CommodityGroupID;

///买标准持仓

TThostFtdcStdPositionType BStdPosition;

///卖标准持仓

TThostFtdcStdPositionType SStdPosition;

///买标准开仓冻结

TThostFtdcStdPositionType BStdOpenFrozen;

///卖标准开仓冻结

TThostFtdcStdPositionType SStdOpenFrozen;

///买标准平仓冻结

TThostFtdcStdPositionType BStdCloseFrozen;

///卖标准平仓冻结

TThostFtdcStdPositionType SStdCloseFrozen;

///品种内对冲标准持仓

TThostFtdcStdPositionType IntraProdStdPosition;

///品种内单腿标准持仓

TThostFtdcStdPositionType NetStdPosition;

///品种间对冲标准持仓

TThostFtdcStdPositionType InterProdStdPosition;

///单腿标准持仓

TThostFtdcStdPositionType SingleStdPosition;

///品种内对锁保证金

TThostFtdcMoneyType IntraProdMargin;

///品种间对锁保证金

TThostFtdcMoneyType InterProdMargin;

///跨品种单腿保证金

TThostFtdcMoneyType SingleMargin;

///非组合合约保证金

TThostFtdcMoneyType NonCombMargin;

///附加保证金

TThostFtdcMoneyType AddOnMargin;

///交易所保证金

```

    TThostFtdcMoneyType ExchMargin;
    ///附加冻结保证金

    TThostFtdcMoneyType AddOnFrozenMargin;
    ///开仓冻结保证金

    TThostFtdcMoneyType OpenFrozenMargin;
    ///平仓冻结保证金

    TThostFtdcMoneyType CloseFrozenMargin;
    ///品种保证金

    TThostFtdcMoneyType Margin;
    ///冻结保证金

    TThostFtdcMoneyType FrozenMargin;

};

```

4.4. CthostFtdcTraderApi 接口

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

4.4.1. CreateFtdcTraderApi 方法

客户端应用程序使用此函数来创建一个“CThostFtdcTradeApi”实例。注意：创建实例时不要使用“new”。

函数原型：

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

返回值：

返回一个指向 CThostFtdcTradeApi 实例的指针。

4.4.2. Release 方法

释放一个 CThostFtdcTradeApi 实例。不能使用 delete 方法

函数原型：

```
void Release();
```

4.4.3. init 方法

使客户端开始与 Kingstar 服务器建立连接，连接成功后可以进行登陆。

函数原型：

```
void Init();
```

4.4.4. join 方法

客户端等待一个接口实例线程的结束。

函数原型：

```
void Join();
```

4.4.5. GetTradingDay 方法

获得当前交易日。只有当与 Kingstar 服务器连接建立后才会取到正确的值。

函数原型：

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

返回值：

返回一个指向日期信息字符串的常量指针。

4.4.6. GetApiVersion 方法

获得 API 版本号。只有当与 Kingstar 服务器连接建立后才会取到正确的值。

函数原型：

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

返回值：

返回一个指向日期信息字符串的常量指针。

4.4.7. RegisterSpi 方法

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

函数原型：

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

参数：

pSpi: 实现了 CThostFtdcTraderSpi 接口的实例指针。

4.4.8. RegisterFront 方法

设置交易托管系统的网络通讯地址，Kingstar 服务器拥有多个通信地址，但用户只需要选择一个通信地址。客户端 RegisterFront 多次时，使用第一个有效的通信地址。

函数原型：

```
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://127.0.0.1:17993"。 "tcp"代表传输协议, "127.0.0.1"代表服务器地址, "17993"代表服务器端口号。同时 API 也提供了对代理服务器的支持, 包括 socks4、socks5 及 http, 客户端开发时只需通过传递给 API 不同的连接字符串就可实现, 例如 "socks5://127.0.0.1:17993 /user:pass@127.0.0.1:18993" 。

4.4.9. 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。

函数原型:

```
void SubscribePrivateTopic(TE_RESUME_TYPE nResumeType);
```

参数:

nResumeType: 私有流重传方式;

TERT_RESTART: 从本交易日开始重传;

TERT_RESUME: 从上次收到的续传;

TERT_QUICK: 只传送登录后私有流的内容。

4.4.10. 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。

函数原型:

```
void SubscribePublicTopic(TE_RESUME_TYPE nResumeType);
```

参数:

nResumeType: 私有流重传方式;

TERT_RESTART: 从本交易日开始重传;

TERT_RESUME: 从上次收到的续传;

TERT_QUICK: 只传送登录后公有流的内容。

4.4.11. ReqUserLoginSM 方法

国密用户登陆请求。

函数原型:

```
int ReqUserLoginSM(  
    CThostFtdcReqUserLoginSMField *pReqUserLoginSM,  
    int nRequestID)
```

参数:

pReqUserLoginSM: 指向国密客户端认证请求结构

国密客户端认证请求结构:

```
struct CThostFtdcReqUserLoginSMField : public CThostFtdcReqUserLoginField
{
    ///认证码
    TThostFtdcAuthCodeType      AuthCode;
    ///App代码
    TThostFtdcAppIDType         AppID;
    ///PIN码
    TThostFtdcPasswordType      PIN;

    //厂商信安新增如下几个字段
    //MAuth 服务标识
    TThostFtdcAppIDType         MAuthServiceID;
    //MAuth AppID
    TThostFtdcAppIDType         MAuthAppID;
    //MAuth SecretKey
    TThostFtdcSecretKeyType      MAuthSecretKey;
};
```

备注: 其中继承子CThostFtdcReqUserLoginField相关字段具体内容可参考4.4.13

4.4.12. ReqAuthenticate 方法

客户端认证请求。

函数原型:

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

参数:

pReqAuthenticateField: 指向客户端认证请求结构

客户端认证请求结构:

```
struct CThostFtdcReqAuthenticateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType      BrokerID;
    ///用户代码
    TThostFtdcUserIDType        UserID;
    ///用户端产品信息
    TThostFtdcProductInfoType   UserProductInfo;
    ///认证码
    TThostFtdcAuthCodeType      AuthCode;
```



```
};
```

4.4.13. ReqUserLogin 方法

用户发出登陆请求。

函数原型：

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

参数：

pReqUserLoginField: 指向用户登录请求结构的地址。

用户登录请求结构：

```
struct CThostFtdcReqUserLoginField  
{  
    ///交易日  
    TThostFtdcDateType    TradingDay;  
    ///经纪公司代码  
    TThostFtdcBrokerIDType    BrokerID;  
    ///用户代码  
    TThostFtdcUserIDType    UserID;  
    ///密码  
    TThostFtdcPasswordType    Password;  
    ///用户端产品信息  
    TThostFtdcProductInfoType    UserProductInfo;  
    ///接口端产品信息  
    TThostFtdcProductInfoType    InterfaceProductInfo;  
    ///协议信息  
    TThostFtdcProtocolInfoType    ProtocolInfo;  
    ///Mac 地址  
    TThostFtdcMacAddressType    MacAddress;  
    ///动态密码  
    TThostFtdcPasswordType    OneTimePassword;  
    ///终端 IP 地址  
    TThostFtdcIPAddressType    ClientIPAddress;
```

```
};
```

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，软件开发商、版本号等，例如：*KSTraderV100*。*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。

4.4.14. ReqUserLogout 方法

用户发出登出请求。

函数原型：

```
int ReqUserLogout(
    CThostFtdcUserLogoutField *pUserLogout,
    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 CThostFtdcUserLogoutField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///用户代码
    TThostFtdcUserIDType UserID;
};
```

4.4.15. ReqUserPasswordUpdate 方法

用户密码修改请求。

函数原型：

```
int ReqUserPasswordUpdate(
    CThostFtdcUserPasswordUpdateField
    *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 CThostFtdcUserPasswordUpdateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///原来的口令
    TThostFtdcPasswordType    OldPassword;
    ///新的口令
    TThostFtdcPasswordType    NewPassword;
};
```

4.4.16. ReqTradingAccountPasswordUpdate 方法

资金账户口令更新请求。

函数原型:

```
int ReqTradingAccountPasswordUpdate(
    CThostFtdcTradingAccountPasswordUpdateField
    *pTradingAccountPasswordUpdate,
    int nRequestID) ;
```

参数:

pUserPasswordUpdate: 指向资金账户口令修改结构的地址。

资金账户口令修改结构:

```
struct CThostFtdcTradingAccountPasswordUpdateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者帐号
    TThostFtdcAccountIDType    AccountID;
    ///原来的口令
    TThostFtdcPasswordType    OldPassword;
```

```

    ///新的口令
    TThostFtdcPasswordType    NewPassword;
    ///币种代码
    TThostFtdcCurrencyIDType   CurrencyID;
};

```

4.4.17. ReqOrderInsert 方法

客户端发出报单录入请求。

函数原型：

```

int ReqOrderInsert(
    CThostFtdcInputOrderField *pInputOrder,
    int nRequestID);

```

参数：

pInputOrder: 指向输入报单结构的地址。

输入报单结构：

```

struct CThostFtdcInputOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///报单引用
    TThostFtdcOrderRefType     OrderRef;
    ///用户代码
    TThostFtdcUserIDType       UserID;
    ///报单价格条件
    TThostFtdcOrderPriceTypeType OrderPriceType;
    ///买卖方向
    TThostFtdcDirectionType     Direction;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType CombOffsetFlag;

```

```

    ///组合投机套保标志
    TThostFtdcCombHedgeFlagType CombHedgeFlag;
    ///价格
    TThostFtdcPriceType LimitPrice;
    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///有效期类型
    TThostFtdcTimeConditionType TimeCondition;
    ///GTD 日期
    TThostFtdcDateType GTDDate;
    ///成交量类型
    TThostFtdcVolumeConditionType VolumeCondition;
    ///最小成交量
    TThostFtdcVolumeType MinVolume;
    ///触发条件
    TThostFtdcContingentConditionType ContingentCondition;
    ///止损价
    TThostFtdcPriceType StopPrice;
    ///强平原因
    TThostFtdcForceCloseReasonType ForceCloseReason;
    ///自动挂起标志
    TThostFtdcBoolType IsAutoSuspend;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///用户强评标志
    TThostFtdcBoolType UserForceClose;
    ///互换单标志
    TThostFtdcBoolType IsSwapOrder;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

OrderRef : 报单引用, 只能单调递增。每次登成功后, 可以从 *OnRspUserLogin* 的输出参数 *CThostFtdcRspUserLoginField* 中获得上次登录用过的最大 *OrderRef:MaxOrderRef*。

4.4.18. 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、报单的激活、报单的修改。

函数原型:

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

参数:

pInputOrderAction: 指向报单操作结构的地址。

报单操作结构:

```
struct CThostFtdcInputOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///报单操作引用
    TThostFtdcOrderActionRefType OrderActionRef;
    ///报单引用
    TThostFtdcOrderRefType OrderRef;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报单编号
    TThostFtdcOrderSysIDType OrderSysID;
    ///操作标志
```

```

    TThostFtdcActionFlagType ActionFlag;
    ///价格
    TThostFtdcPriceType LimitPrice;
    ///数量变化
    TThostFtdcVolumeType VolumeChange;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.4.19. ReqQueryMaxOrderVolume 方法

查询最大报单数量请求。

函数原型：

```

int ReqQueryMaxOrderVolume(
    CThostFtdcQueryMaxOrderVolumeField *pQueryMaxOrderVolume,
    int nRequestID);

```

参数：

pQueryMaxOrderVolume： 指向查询最大报单数量结构的地址。

查询最大报单数量结构：

```

struct CThostFtdcQueryMaxOrderVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///买卖方向
    TThostFtdcDirectionType Direction;
    ///开平标志
    TThostFtdcOffsetFlagType OffsetFlag;
    ///投机套保标志

```

```

    TThostFtdcHedgeFlagType HedgeFlag;
    ///最大允许报单数量

    TThostFtdcVolumeType MaxVolume;
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
};

```

4.4.20. ReqSettlementInfoConfirm 方法

投资者结算结果确认。

函数原型：

```

int ReqSettlementInfoConfirm(
    CThostFtdcSettlementInfoConfirmField *pSettlementInfoConfirm,
    int nRequestID);

```

参数：

pSettlementInfoConfirm： 指向投资者结算结果确认结构的地址。

投资者结算结果确认结构：

```

struct CThostFtdcSettlementInfoConfirmField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType InvestorID;
    ///确认日期

    TThostFtdcDateType ConfirmDate;
    ///确认时间

    TThostFtdcTimeType ConfirmTime;
};

```

4.4.21. ReqFromBankToFutureByFuture 方法

请求银行资金转期货。

函数原型：

```

int ReqFromBankToFutureByFuture (
    CThostFtdcReqTransferField *pReqTransfer,
    int nRequestID);

```


参数:

pReqTransfer: 指向银行资金转期货请求结构的地址。

银行资金转期货请求结构:

```
struct CThostFtdcReqTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType   TradeCode;
    ///银行代码
    TThostFtdcBankIDType   BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType   BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType   BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType   BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType   TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType   TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType   BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType   TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType   PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType   LastFragment;
    ///会话号
    TThostFtdcSessionIDType   SessionID;
    ///客户姓名
    TThostFtdcIndividualNameType   CustomerName;
    ///证件类型
```

```
TThostFtdcIdCardTypeType IdCardType;  
///证件号码  
TThostFtdcIdentifiedCardNoType IdentifiedCardNo;  
///客户类型  
TThostFtdcCustTypeType CustType;  
///银行帐号  
TThostFtdcBankAccountType BankAccount;  
///银行密码  
TThostFtdcPasswordType BankPassWord;  
///投资者帐号  
TThostFtdcAccountIDType AccountID;  
///期货密码  
TThostFtdcPasswordType Password;  
///安装编号  
TThostFtdcInstallIDType InstallID;  
///期货公司流水号  
TThostFtdcFutureSerialType FutureSerial;  
///用户标识  
TThostFtdcUserIDType UserID;  
///验证客户证件号码标志  
TThostFtdcYesNoIndicatorType VerifyCertNoFlag;  
///币种代码  
TThostFtdcCurrencyIDType CurrencyID;  
///转帐金额  
TThostFtdcTradeAmountType TradeAmount;  
///期货可取金额  
TThostFtdcTradeAmountType FutureFetchAmount;  
///费用支付标志  
TThostFtdcFeePayFlagType FeePayFlag;  
///应收客户费用  
TThostFtdcCustFeeType CustFee;  
///应收期货公司费用
```

```

    TThostFtdcFutureFeeType    BrokerFee;
    ///发送方给接收方的消息

    TThostFtdcAddInfoType    Message;
    ///摘要

    TThostFtdcDigestType    Digest;
    ///银行帐号类型

    TThostFtdcBankAccTypeType    BankAccType;
    ///渠道标志

    TThostFtdcDeviceIDType    DeviceID;
    ///期货单位帐号类型

    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码

    TThostFtdcBankCodingForFutureType    BrokerIDByBank;
    ///期货单位帐号

    TThostFtdcBankAccountType    BankSecuAcc;
    ///银行密码标志

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

    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员

    TThostFtdcOperNoType    OperNo;
    ///请求编号

    TThostFtdcRequestIDType    RequestID;
    ///交易 ID

    TThostFtdcTIDType    TID;
    ///转账交易状态

    TThostFtdcTransferStatusType    TransferStatus;

};

```

4.4.22. ReqFromFutureToBankByFuture 方法

请求期货资金转银行。

函数原型：

```
int ReqFromFutureToBankByFuture (
```

```
CThostFtdcReqTransferField *pReqTransfer,
int nRequestID);
```

参数:

pReqTransfer: 指向期货资金转银行请求结构的地址。

期货资金转银行请求结构:

```
struct CThostFtdcReqTransferField
{
    ///业务功能码
    TThostFtdcTradeCodeType TradeCode;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType LastFragment;
    ///会话号
    TThostFtdcSessionIDType SessionID;
    ///客户姓名
```

```
TThostFtdcIndividualNameType  CustomerName;
///证件类型

TThostFtdcIdCardTypeType  IdCardType;
///证件号码

TThostFtdcIdentifiedCardNoType  IdentifiedCardNo;
///客户类型

TThostFtdcCustTypeType  CustType;
///银行帐号

TThostFtdcBankAccountType  BankAccount;
///银行密码

TThostFtdcPasswordType  BankPassWord;
///投资者帐号

TThostFtdcAccountIDType  AccountID;
///期货密码

TThostFtdcPasswordType  Password;
///安装编号

TThostFtdcInstallIDType  InstallID;
///期货公司流水号

TThostFtdcFutureSerialType  FutureSerial;
///用户标识

TThostFtdcUserIDType  UserID;
///验证客户证件号码标志

TThostFtdcYesNoIndicatorType  VerifyCertNoFlag;
///币种代码

TThostFtdcCurrencyIDType  CurrencyID;
///转帐金额

TThostFtdcTradeAmountType  TradeAmount;
///期货可取金额

TThostFtdcTradeAmountType  FutureFetchAmount;
///费用支付标志

TThostFtdcFeePayFlagType  FeePayFlag;
///应收客户费用
```

```

    TThostFtdcCustFeeType    CustFee;
    ///应收期货公司费用

    TThostFtdcFutureFeeType   BrokerFee;
    ///发送方给接收方的消息

    TThostFtdcAddInfoType     Message;
    ///摘要

    TThostFtdcDigestType      Digest;
    ///银行帐号类型

    TThostFtdcBankAccTypeType   BankAccType;
    ///渠道标志

    TThostFtdcDeviceIDType      DeviceID;
    ///期货单位帐号类型

    TThostFtdcBankAccTypeType   BankSecuAccType;
    ///期货公司银行编码

    TThostFtdcBankCodingForFutureType   BrokerIDByBank;
    ///期货单位帐号

    TThostFtdcBankAccountType   BankSecuAcc;
    ///银行密码标志

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

    TThostFtdcPwdFlagType       SecuPwdFlag;
    ///交易柜员

    TThostFtdcOperNoType        OperNo;
    ///请求编号

    TThostFtdcRequestIDType     RequestID;
    ///交易 ID

    TThostFtdcTIDType           TID;
    ///转账交易状态

    TThostFtdcTransferStatusType   TransferStatus;
};

```

4.4.23. ReqQueryBankAccountMoneyByFuture 方法

请求查询银行资金。

函数原型：

```
int ReqQueryBankAccountMoneyByFuture (
    CThostFtdcReqQueryAccountField *pReqQueryAccount,
    int nRequestID);
```

参数：

pReqQueryAccount： 指向查询银行资金请求结构的地址。

查询银行资金请求结构：

```
struct CThostFtdcReqQueryAccountField
{
    ///业务功能码
    TThostFtdcTradeCodeType  TradeCode;
    ///银行代码
    TThostFtdcBankIDType  BankID;
    ///银行分支机构代码
    TThostFtdcBankBrchIDType  BankBranchID;
    ///期商代码
    TThostFtdcBrokerIDType  BrokerID;
    ///期商分支机构代码
    TThostFtdcFutureBranchIDType  BrokerBranchID;
    ///交易日期
    TThostFtdcTradeDateType  TradeDate;
    ///交易时间
    TThostFtdcTradeTimeType  TradeTime;
    ///银行流水号
    TThostFtdcBankSerialType  BankSerial;
    ///交易系统日期
    TThostFtdcTradeDateType  TradingDay;
    ///银期平台消息流水号
    TThostFtdcSerialType  PlateSerial;
    ///最后分片标志
    TThostFtdcLastFragmentType  LastFragment;
    ///会话号
```

```

TThostFtdcSessionIDType  SessionID;
///客户姓名
TThostFtdcIndividualNameType  CustomerName;
///证件类型
TThostFtdcIdCardTypeType  IdCardType;
///证件号码
TThostFtdcIdentifiedCardNoType  IdentifiedCardNo;
///客户类型
TThostFtdcCustTypeType  CustType;
///银行帐号
TThostFtdcBankAccountType  BankAccount;
///银行密码
TThostFtdcPasswordType  BankPassWord;
///投资者帐号
TThostFtdcAccountIDType  AccountID;
///期货密码
TThostFtdcPasswordType  Password;
///期货公司流水号
TThostFtdcFutureSerialType  FutureSerial;
///安装编号
TThostFtdcInstallIDType  InstallID;
///用户标识
TThostFtdcUserIDType  UserID;
///验证客户证件号码标志
TThostFtdcYesNoIndicatorType  VerifyCertNoFlag;
///币种代码
TThostFtdcCurrencyIDType  CurrencyID;
///摘要
TThostFtdcDigestType  Digest;
///银行帐号类型
TThostFtdcBankAccTypeType  BankAccType;
///渠道标志

```



```

    TThostFtdcDeviceIDType    DeviceID;
    ///期货单位帐号类型

    TThostFtdcBankAccTypeType    BankSecuAccType;
    ///期货公司银行编码

    TThostFtdcBankCodingForFutureType    BrokerIDByBank;
    ///期货单位帐号

    TThostFtdcBankAccountType    BankSecuAcc;
    ///银行密码标志

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

    TThostFtdcPwdFlagType    SecuPwdFlag;
    ///交易柜员

    TThostFtdcOperNoType    OperNo;
    ///请求编号

    TThostFtdcRequestIDType    RequestID;
    ///交易ID

    TThostFtdcTIDType    TID;
};

```

4.4.24. ReqQryTransferSerial 方法

请求查询转帐流水。

函数原型：

```

int ReqQryTransferSerial(
    CThostFtdcQryTransferSerialField *pQryTransferSerial,
    int nRequestID);

```

参数：

pQryTransferSerial: 指向查询转账流水请求结构的地址。

查询转账流水请求结构：

```

struct CThostFtdcQryTransferSerialField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType    BrokerID;
    ///投资者帐号

```

```

    TThostFtdcAccountIDType AccountID;
    ///银行编码

    TThostFtdcBankIDType BankID;
    ///币种代码

    TThostFtdcCurrencyIDType CurrencyID;
};

```

4.4.25. ReqQryAccountregister 方法

客户端发出查询银期签约关系请求。

函数原型：

```

int ReqQryAccountregister(
    CThostFtdcQryAccountregisterField *pQryAccountregister,
    int nRequestID);

```

参数：

pQryAccountregister：指向查询银期签约关系请求结构。

查询银期签约关系请求结构：

```

struct CThostFtdcQryAccountregisterField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者帐号
    TThostFtdcAccountIDType AccountID;
    ///银行编码
    TThostFtdcBankIDType BankID;
    ///银行分支机构编码
    TThostFtdcBankBrchIDType BankBranchID;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
};

```

4.4.26. ReqQryOrder 方法

报单查询请求。

函数原型：

```

int ReqQryOrder(
    CThostFtdcQryOrderField *pQryOrder,
    int nRequestID);

```

参数：

pQryOrder: 指向报单查询结构的地址。

报单查询结构:

```
struct CThostFtdcQryOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///报单编号
    TThostFtdcOrderSysIDType   OrderSysID;
    ///开始时间
    TThostFtdcTimeType         InsertTimeStart;
    ///结束时间
    TThostFtdcTimeType         InsertTimeEnd;
};
```

4.4.27. ReqQryTrade 方法

成交单查询请求。

函数原型:

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

参数:

pQryTrade: 指向成交查询结构的地址。

成交查询结构:

```
struct CThostFtdcQryTradeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
```

```

    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///成交编号
    TThostFtdcTradeIDType TradeID;
    ///开始时间
    TThostFtdcTimeType TradeTimeStart;
    ///结束时间
    TThostFtdcTimeType TradeTimeEnd;
};

```

4.4.28. ReqQryInvestor 方法

会员客户查询请求。

函数原型：

```

int ReqQry Investor (
    CThostFtdcQryInvestorField *pQryInvestor,
    int nRequestID);

```

参数：

pQryInvestor: 指向客户查询结构的地址。

客户查询结构：

```

struct CThostFtdcQryInvestorField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
};

```

4.4.29. ReqQryInvestorPosition 方法

会员持仓查询请求。

函数原型：

```
int ReqQryInvestorPosition(
    CThostFtdcQryInvestorPositionField *pQryInvestorPosition,
    int nRequestID);
```

参数:

pQryInvestorPosition: 指向会员持仓查询结构的地址。

会员持仓查询结构:

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

4.4.30. ReqQryTradingAccount 方法

请求查询资金账户。

函数原型:

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

参数:

pQryTradingAccount: 指向查询资金账户结构的地址。

查询资金账户结构:

```
struct CThostFtdcQryTradingAccountField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
```

```

    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
    ///业务类型
    TThostFtdcBizTypeType BizType;
};

```

4.4.31. ReqQryTradingCode 方法

请求查询交易编码。

函数原型：

```

int ReqQryTradingCode(
    CThostFtdcQryTradingCodeField *pQryTradingCode,
    int nRequestID);

```

参数：

pQryTradingCode： 指向查询交易编码结构的地址。

查询交易编码结构：

```

struct CThostFtdcQryTradingCodeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///客户代码
    TThostFtdcClientIDType ClientID;
    ///交易编码类型
    TThostFtdcClientIDTypeType ClientIDType;
};

```

4.4.32. ReqQryExchange 方法

请求查询交易所。

函数原型：

```

int ReqQryExchange(
    CThostFtdcQryExchangeField *pQryExchange,

```

```
int nRequestID);
```

参数:

pQryExchange: 指向查询交易编码结构的地址。

查询交易所编码结构:

```
struct CThostFtdcQryExchangeField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.4.33. ReqQryInstrument 方法

请求查询合约。

函数原型:

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

参数:

pQryInstrument: 指向查询合约结构的地址。

查询合约结构:

```
struct CThostFtdcQryInstrumentField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约在交易所的代码
    TThostFtdcExchangeInstIDType ExchangeInstID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
};
```

4.4.34. ReqQryDepthMarketData 方法

请求查询行情。

函数原型:

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

参数:

pQryDepthMarketData: 指向查询行情结构的地址。

查询行情结构:

```
struct CThostFtdcQryDepthMarketDataField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.4.35. ReqQryInstrumentMarginRate 方法

请求查询合约保证金率。

函数原型:

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

参数:

pQryInstrumentMarginRate: 指向查询合约保证金率结构的地址。

查询合约保证金率结构:

```
struct CThostFtdcQryInstrumentMarginRateField
{
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
```



```
};
```

4.4.36. ReqQryInstrumentCommissionRate 方法

请求查询合约手续费率。

函数原型：

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

参数：

pQryInstrumentCommissionRate： 指向查询合约手续费率结构的地址。

查询合约手续费率结构：

```
struct CThostFtdcQryInstrumentCommissionRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
};
```

4.4.37. ReqQryCFMMCTradingAccountKey 方法

请求查询保证金监管系统经纪公司资金账户密钥。

函数原型：

```
int ReqQryCFMMCTradingAccountKey(
    CThostFtdcQryCFMMCTradingAccountKeyField *pQryCFMMCTradingAccountKey,
    int nRequestID);
```

参数：

pQryCFMMCTradingAccountKey： 指向查询监控中心密钥结构的地址。

查询监控中心密钥结构：

```
struct CThostFtdcQryCFMMCTradingAccountKeyField
{
```

```

    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
};

```

4.4.38. ReqQryEWarrantOffset 方法

请求查询仓单折抵信息

函数原型:

```

int ReqQryEWarrantOffset(
    CThostFtdcQryEWarrantOffsetField *pQryEWarrantOffset,
    int nRequestID);

```

参数:

pQryEWarrantOffset: 指向查询仓单折抵信息结构的地址。

查询仓单折抵信息结构:

```

struct CThostFtdcQryEWarrantOffsetField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType   ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.4.39. ReqQryInvestorProductGroupMargin 方法

请求查询投资者品种/跨品种保证金

函数原型:

```

int ReqQryInvestorProductGroupMargin(
    CThostFtdcQryInvestorProductGroupMarginField *pQryInvestorProductGroupMargin,
    int nRequestID);

```

参数:

pQryInvestorProductGroupMargin: 指向查询投资者品种/跨品种保证金结构的地址。

查询投资者品种/跨品种保证金结构:

```
struct CThostFtdcQryInvestorProductGroupMarginField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///品种/跨品种标示
    TThostFtdcInstrumentIDType ProductGroupID;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
};
```

4.4.40. ReqQryExchangeMarginRate 方法

请求查询交易所保证金率

函数原型:

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

参数:

pQryExchangeMarginRate: 指向查询交易所保证金率结构的地址。

查询交易所保证金率结构:

```
struct CThostFtdcExchangeMarginRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
};
```

4.4.41. ReqQryExchangeMarginRateAdjust 方法

请求查询交易所调整保证金率

函数原型:

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

参数:

pQryExchangeMarginRateAdjust: 指向查询交易所调整保证金率结构的地址。

查询交易所调整保证金率结构:

```
struct CThostFtdcExchangeMarginRateAdjustField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
};
```

4.4.42. ReqQrySettlementInfo 方法

请求查询投资者结算结果。

函数原型:

```
int ReqQrySettlementInfo(
    CThostFtdcQrySettlementInfoField *pQrySettlementInfo,
    int nRequestID);
```

参数:

pQrySettlementInfo: 指向查询投资者结算结果结构的地址。

查询投资者结算结果结构:

```
struct CThostFtdcQrySettlementInfoField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
```

```

        ///交易日
        TThostFtdcDateType    TradingDay;
};

```

4.4.43. ReqQryTransferBank 方法

请求查询转帐银行。

函数原型：

```

int ReqQryTransferBank(
    CThostFtdcQryTransferBankField *pQryTransferBank,
    int nRequestID);

```

参数：

pQryTransferBank： 指向查询转帐银行结构的地址。

查询转帐银行结构：

```

struct CThostFtdcQryTransferBankField
{
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///银行分中心代码
    TThostFtdcBankBrchIDType BankBrchID;
};

```

4.4.44. ReqQryInvestorPositionDetail 方法

请求查询投资者持仓明细。

函数原型：

```

int ReqQryInvestorPositionDetail(
    CThostFtdcQryInvestorPositionDetailField *pQryInvestorPositionDetail,
    int nRequestID);

```

参数：

pQryInvestorPositionDetail： 指向查询投资者持仓明细结构的地址。

查询投资者持仓明细结构：

```

struct CThostFtdcQryInvestorPositionDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;

```

```

    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;

    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;

};

```

4.4.45. ReqQryNotice 方法

请求查询客户通知。

函数原型：

```

int ReqQryNotice(
    CThostFtdcQryNoticeField *pQryNotice,
    int nRequestID);

```

参数：

pQryNotice： 指向查询客户通知结构的地址。

查询客户通知结构：

```

struct CThostFtdcQryNoticeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;

};

```

4.4.46. ReqQrySettlementInfoConfirm 方法

查询结算信息确认。

函数原型：

```

int ReqQrySettlementInfoConfirm(
    CThostFtdcQrySettlementInfoConfirmField
    *pQrySettlementInfoConfirm,
    int nRequestID);

```

参数：

pQrySettlementInfoConfirm： 指向查询结算信息确认结构的地址。

查询结算信息确认：

```

struct CThostFtdcQrySettlementInfoConfirmField

```

```
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
};
```

4.4.47. ReqQryContractBank 方法

请求查询签约银行。

函数原型：

```
int ReqQryContractBank(
    CThostFtdcQryContractBankField *pQryContractBank,
    int nRequestID);
```

参数：

pQryContractBank：指向查询签约银行请求结构的地址。其中 *BrokerID* 不能为空，*BankID* 和 *BankBrchID* 可以为空。

查询签约银行请求结构：

```
struct CThostFtdcQryContractBankField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///银行代码
    TThostFtdcBankIDType    BankID;
    ///银行分中心代码
    TThostFtdcBankBrchIDType BankBrchID;
};
```

4.4.48. ReqQryParkedOrder 方法

请求查询预埋单。

函数原型：

```
int ReqQryParkedOrder(
    CThostFtdcQryParkedOrderField *pQryParkedOrder,
    int nRequestID);
```

参数：

pQryParkedOrder: 指向查询预埋单请求结构的地址。

查询预埋单请求结构:

```
struct CThostFtdcQryParkedOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.4.49. ReqQryParkedOrderAction 方法

请求查询预埋撤单。

函数原型:

```
int ReqQryParkedOrderAction(
    CThostFtdcQryParkedOrderActionField *pQryParkedOrderAction,
    int nRequestID);
```

参数:

pQryParkedOrderAction: 指向请求查询预埋撤单结构的地址。

请求查询预埋撤单结构:

```
struct CThostFtdcQryParkedOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
```



```
};
```

4.4.50. ReqQryTradingNotice 方法

客户端发出查询交易通知请求。

函数原型：

```
int ReqQryTradingNotice(
    CThostFtdcQryTradingNoticeField *pQryTradingNotice,
    int nRequestID);
```

参数：

pQryTradingNotice： 指向查询交易通知请求结构。

查询交易通知请求结构：

```
struct CThostFtdcQryTradingNoticeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
};
```

4.4.51. ReqQryBrokerTradingParams 方法

客户端发起查询经济公司交易参数请求。

函数原型：

```
int ReqQryBrokerTradingParams(
    CThostFtdcQryBrokerTradingParamsField *pQryBrokerTradingParams,
    int nRequestID);
```

参数：

pQryBrokerTradingParams： 指向查询经济公司交易参数请求结构。

查询经济公司交易参数请求结构：

```
struct CThostFtdcQryBrokerTradingParamsField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
};
```

4.4.52. ReqQryInvestorPositionCombineDetail 方法

请求查询投资者组合持仓明细。

函数模型：

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

参数：

pQryInvestorPositionCombineDetail： 指向请求查询投资者组合持仓明细请求结构的地址。

查询组合持仓明细请求结构：

```
struct CThostFtdcQryInvestorPositionCombineDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///组合持仓合约编码
    TThostFtdcInstrumentIDType CombInstrumentID;
};
```

4.4.53. ReqParkedOrderInsert 方法

预埋单录入请求。

函数模型：

```
int ReqParkedOrderInsert (
    CThostFtdcParkedOrderField *pParkedOrder,
    int nRequestID);
```

参数：

pParkedOrder： 指向预埋单录入请求结构的地址。

预埋单结构：

```
struct CThostFtdcParkedOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
```

```
///投资者代码
TThostFtdcInvestorIDType InvestorID;

///合约代码
TThostFtdcInstrumentIDType InstrumentID;

///报单引用
TThostFtdcOrderRefType OrderRef;

///用户代码
TThostFtdcUserIDType UserID;

///报单价格条件
TThostFtdcOrderPriceTypeType OrderPriceType;

///买卖方向
TThostFtdcDirectionType Direction;

///组合开平标志
TThostFtdcCombOffsetFlagType CombOffsetFlag;

///组合投机套保标志
TThostFtdcCombHedgeFlagType CombHedgeFlag;

///价格
TThostFtdcPriceType LimitPrice;

///数量
TThostFtdcVolumeType VolumeTotalOriginal;

///有效期类型
TThostFtdcTimeConditionType TimeCondition;

///GTD 日期
TThostFtdcDateType GTDDate;

///成交量类型
TThostFtdcVolumeConditionType VolumeCondition;

///最小成交量
TThostFtdcVolumeType MinVolume;

///触发条件
TThostFtdcContingentConditionType ContingentCondition;

///止损价
TThostFtdcPriceType StopPrice;
```

```

    ///强平原因
    TThostFtdcForceCloseReasonType    ForceCloseReason;

    ///自动挂起标志
    TThostFtdcBoolType    IsAutoSuspend;

    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;

    ///请求编号
    TThostFtdcRequestIDType    RequestID;

    ///用户强评标志
    TThostFtdcBoolType    UserForceClose;

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;

    ///预埋报单编号
    TThostFtdcParkedOrderIDType    ParkedOrderID;

    ///用户类型
    TThostFtdcUserTypeType    UserType;

    ///预埋单状态
    TThostFtdcParkedOrderStatusType    Status;

    ///错误代码
    TThostFtdcErrorIDType ErrorID;

    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;

    ///互换单标志
    TThostFtdcBoolType    IsSwapOrder;

};

```

4.4.54. ReqParkedOrderAction 方法

预埋撤单录入请求。

函数原型：

```

int ReqParkedOrderAction(
    CThostFtdcParkedOrderActionField *pParkedOrderAction,
    int nRequestID);

```

参数：

pParkedOrderAction: 指向预埋撤单录入请求结构的地址。

预埋撤单录入请求结构:

```
struct CThostFtdcParkedOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///报单操作引用
    TThostFtdcOrderActionRefType  OrderActionRef;
    ///报单引用
    TThostFtdcOrderRefType    OrderRef;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报单编号
    TThostFtdcOrderSysIDType OrderSysID;
    ///操作标志
    TThostFtdcActionFlagType  ActionFlag;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量变化
    TThostFtdcVolumeType VolumeChange;
    ///用户代码
    TThostFtdcUserIDType  UserID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
```

```

    ///预埋撤单单编号
    TThostFtdcParkedOrderActionIDType ParkedOrderActionID;
    ///用户类型
    TThostFtdcUserTypeType    UserType;
    ///预埋撤单状态
    TThostFtdcParkedOrderStatusType    Status;
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
};

```

4.4.55. ReqRemoveParkedOrder 方法

请求删除预埋单。

函数原型：

```

int ReqRemoveParkedOrder(
    CThostFtdcRemoveParkedOrderField *pRemoveParkedOrder,
    int nRequestID);

```

参数：

pRemoveParkedOrder： 指向请求删除预埋单结构的地址。

请求删除预埋单结构：

```

struct CThostFtdcRemoveParkedOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///预埋报单编号
    TThostFtdcParkedOrderIDType    ParkedOrderID;
};

```

4.4.56. ReqRemoveParkedOrderAction 方法

请求删除预埋撤单。

函数原型：

```
int ReqRemoveParkedOrderAction(
    CThostFtdcRemoveParkedOrderActionField *pRemoveParkedOrderAction,
    int nRequestID);;
```

参数:

pRemoveParkedOrderAction: 指向请求删除预埋撤单结构的地址。

请求删除预埋撤单结构:

```
struct CThostFtdcRemoveParkedOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///预埋撤单编号
    TThostFtdcParkedOrderActionIDType ParkedOrderActionID;
};
```

4.4.57. ReqExecOrderInsert 方法

执行宣告录入请求。

函数原型:

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

参数:

pInputExecOrder: 指向宣告录入请求结构。

宣告录入请求结构:

```
struct CThostFtdcInputExecOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///执行宣告引用
    TThostFtdcOrderRefType    ExecOrderRef;
    ///用户代码
```

```

    TThostFtdcUserIDType UserID;
    ///数量
    TThostFtdcVolumeType Volume;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///开平标志
    TThostFtdcOffsetFlagType OffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///执行类型
    TThostFtdcActionTypeType ActionType;
    ///保留头寸申请的持仓方向
    TThostFtdcPosiDirectionType PosiDirection;
    ///期权行权后是否保留期货头寸的标记
    TThostFtdcExecOrderPositionFlagType ReservePositionFlag;
    ///期权行权后生成的头寸是否自动平仓
    TThostFtdcExecOrderCloseFlagType CloseFlag;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.4.58. ReqExecOrderAction 方法

执行宣告操作请求。

函数原型：

```

int ReqExecOrderAction(
    CThostFtdcInputExecOrderActionField *pInputExecOrderAction,
    int nRequestID);

```

参数：

pInputExecOrderAction： 指向执行宣告操作请求结构。

执行宣告操作请求结构：

```

struct CThostFtdcInputExecOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///执行宣告操作引用

```



```

TThostFtdcOrderActionRefType   ExecOrderActionRef;
///执行宣告引用
TThostFtdcOrderRefType         ExecOrderRef;
///请求编号
TThostFtdcRequestIDType        RequestID;
///前置编号
TThostFtdcFrontIDType          FrontID;
///会话编号
TThostFtdcSessionIDType        SessionID;
///交易所代码
TThostFtdcExchangeIDType       ExchangeID;
///执行宣告操作编号
TThostFtdcExecOrderSysIDType    ExecOrderSysID;
///操作标志
TThostFtdcActionFlagType        ActionFlag;
///用户代码
TThostFtdcUserIDType            UserID;
///合约代码
TThostFtdcInstrumentIDType      InstrumentID;
};

```

4.4.59. ReqQryBrokerTradingAlgos 方法

查询经纪公司交易算法

函数模型：

```

int ReqQryBrokerTradingAlgos(
    CThostFtdcQryBrokerTradingAlgosField *pQryBrokerTradingAlgos,
    int nRequestID);

```

参数：

pQryBrokerTradingAlgos: 指向请求查询经纪公司交易算法请求结构的地址。

查询经纪公司交易算法请求结构：

```

struct CThostFtdcQryBrokerTradingAlgosField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///合约代码

```

```

        TThostFtdcInstrumentIDType    InstrumentID;
    };

```

4.4.60. RegisterNameServer 方法

注册名字服务器（门户）

客户端可通过调用此方法注册名字服务器从而获取最优网关以实现快速登录。Kingstar API 支持多次注册名字服务器，也就是说，此函数可以被调用多次，每次调用都能实现从此名字服务器相关的最优网关列表中返回一个网关。需要注意的是，一旦多次调用此方法，后续返回的网关地址会替代上次返回的网关地址。

目前此接口只支持 API 的 windows 版本。

函数模型：

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

参数：

pszNsAddress：指向名字服务器地址（也叫门户服务器地址）的指针。名字服务器地址格式如下所示：“协议:ip 地址:端口/代理用户名:代理用户密码@代理服务器 ip 地址:代理服务器端口/网关标识/客户号”。其中代理相关字段是可选的。例如：“tcp://127.0.0.1:11000/A/80001”，“tcp”代表通讯协议，“127.0.0.1”标识名字服务器的 ip 地址，“11000”标识名字服务器的端口，“A”代表网关标识，“80001”标识客户号。

网关标识字段如下所示：

值	描述
‘A’	大智慧
‘B’	文华一键通
‘C’	彭博闪电手
‘D’	托瑞邦泽
‘E’	快期
‘F’	金字塔
‘G’	达钱

4.4.61. RegisterFensUserInfo 方法

注册名字服务器用户信息

函数模型：

```
void RegisterFensUserInfo(CThostFtdcFensUserInfoField *pFensUserInfo);
```

参数：

pFensUserInfo：指向名字服务器用户信息的指针。

4.4.62. LoadExtApi 方法

注册条件单实例

函数模型：

```
void * LoadExtApi(
    void * spi,
    const char *ExtApiName)
```

参数：

Spi: 指向 spi 实例的指针

ExtApiName: API 的名称

4.4.63. ReqForQuoteInsert 方法

询价录入请求

函数模型：

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

参数：

pInputForQuote: 指向询价录入结构的地址。

询价录入结构如下：

```
struct CThostFtdcInputForQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///询价引用
    TThostFtdcOrderRefType    ForQuoteRef;
    ///用户代码
    TThostFtdcUserIDType      UserID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
};
```

ForQuoteRef : 询价引用, 只能单调递增。每次登成功后, 可以从 *OnRspUserLogin* 的输出参数 *CThostFtdcRspUserLoginField* 中获得上次登录用过的最大 *OrderRef:MaxOrderRef*。

4.4.64. ReqQuoteInsert 方法

报价录入请求

函数模型:

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

参数:

pInputQuote: 指向报价录入结构的地址。

报价录入结构如下:

```
struct CThostFtdcInputQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///报价引用
    TThostFtdcOrderRefType    QuoteRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///卖价格
    TThostFtdcPriceType    AskPrice;
    ///买价格
    TThostFtdcPriceType    BidPrice;
    ///卖数量
    TThostFtdcVolumeType    AskVolume;
    ///买数量
    TThostFtdcVolumeType    BidVolume;
    ///请求编号
```

```

TThostFtdcRequestIDType RequestID;
///业务单元

TThostFtdcBusinessUnitType BusinessUnit;
///卖开平标志

TThostFtdcOffsetFlagType AskOffsetFlag;
///买开平标志

TThostFtdcOffsetFlagType BidOffsetFlag;
///卖投机套保标志

TThostFtdcHedgeFlagType AskHedgeFlag;
///买投机套保标志

TThostFtdcHedgeFlagType BidHedgeFlag;
///衍生卖报单引用

TThostFtdcOrderRefType AskOrderRef;
///衍生买报单引用

TThostFtdcOrderRefType BidOrderRef;
///应价编号

TThostFtdcOrderSysIDType ForQuoteSysID;
///交易所代码

TThostFtdcExchangeIDType ExchangeID;
};

```

QuoteRef : 报价引用, 只能单调递增。每次登成功后, 可以从 `OnRspUserLogin` 的输出参数 `CThostFtdcRspUserLoginField` 中获得上次登录用过的最大 `OrderRef:MaxOrderRef`。

4.4.65. ReqQuoteAction 方法

报价操作请求

函数模型:

```

int ReqQuoteAction (
    CThostFtdcInputQuoteActionField *pInputQuoteAction,
    int nRequestID)

```

参数:

pInputQuoteAction: 指向报价操作结构的地址。

报价操作结构如下:

```

struct CThostFtdcInputQuoteActionField

```

```

{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///报价操作引用
    TThostFtdcOrderActionRefType QuoteActionRef;
    ///报价引用
    TThostFtdcOrderRefType QuoteRef;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报价操作编号
    TThostFtdcOrderSysIDType QuoteSysID;
    ///操作标志
    TThostFtdcActionFlagType ActionFlag;
    ///用户代码
    TThostFtdcUserIDType UserID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.4.66. ReqQryProduct 方法

查询产品请求

函数模型:

```

int ReqQryProduct(
    CThostFtdcQryProductField *pQryProduct,
    int nRequestID)

```

参数:

pQryProduct: 指向查询产品结构的地址。

查询产品结构如下:

```
struct CThostFtdcQryProductField
{
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///产品类型
    TThostFtdcProductClassType    ProductClass;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
};
```

4.4.67. ReqQryExchangeRate 方法

查询汇率请求

函数模型:

```
int ReqQryExchangeRate(
    CThostFtdcQryExchangeRateField *pQryExchangeRate,
    int nRequestID)
```

参数:

pQryExchangeRate: 指向查询汇率结构的地址。

查询汇率结构如下:

```
struct CThostFtdcQryExchangeRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///源币种
    TThostFtdcCurrencyIDType  FromCurrencyID;
    ///目标币种
    TThostFtdcCurrencyIDType  ToCurrencyID;
};
```

4.4.68. ReqQrySecAgentACIDMap 方法

查询二级代理操作员银期权限请求

函数模型：

```
int ReqQrySecAgentACIDMap (
    CThostFtdcQrySecAgentACIDMapField *pQrySecAgentACIDMap,
    int nRequestID)
```

参数：

pQrySecAgentACIDMap： 指向查询二级代理操作员银期权限结构的地址。

查询二级代理操作员银期权限结构如下：

```
struct CThostFtdcQrySecAgentACIDMapField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///用户代码
    TThostFtdcUserIDType   UserID;
    ///资金账户
    TThostFtdcAccountIDType   AccountID;
    ///币种
    TThostFtdcCurrencyIDType   CurrencyID;
};
```

4.4.69. ReqQryForQuote 方法

查询询价请求

函数模型：

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

参数：

pQryForQuote： 指向查询询价结构的地址。

查询询价结构如下：

```
struct CThostFtdcQryForQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
```



```

    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///开始时间
    TThostFtdcTimeType InsertTimeStart;
    ///结束时间
    TThostFtdcTimeType InsertTimeEnd;
};

```

4.4.70. ReqQryQuote 方法

查询报价请求

函数模型：

```

int ReqQryQuote (
    CThostFtdcQryQuoteField *pQryQuote,
    int nRequestID)

```

参数：

pQryQuote：指向查询报价结构的地址。

查询报价结构如下：

```

struct CThostFtdcQryQuoteField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报价编号
    TThostFtdcOrderSysIDType QuoteSysID;
    ///开始时间

```

```

        TThostFtdcTimeType    InsertTimeStart;
        ///结束时间

        TThostFtdcTimeType    InsertTimeEnd;

    };

```

4.4.71. ReqQueryCFMMCTradingAccountToken 方法

请求查询监控中心用户令牌。

函数原型：

```

int ReqQueryCFMMCTradingAccountToken(
    CThostFtdcQueryCFMMCTradingAccountTokenField *pQueryCFMMCTradingAccountToken,
    int nRequestID);

```

参数：

pQueryCFMMCTradingAccountToken： 指向查询监控中心用户令牌结构的地址。

查询监控中心用户令牌结构：

```

struct CThostFtdcQueryCFMMCTradingAccountTokenField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType    BrokerID;

    ///投资者代码

    TThostFtdcInvestorIDType    InvestorID;

};

```

4.4.72. ReqLockInsert 方法

客户端发出锁定请求。

函数原型：

```

int ReqLockInsert(
    CThostFtdcInputLockField *pInputLock,
    int nRequestID);

```

参数：

pInputLock： 指向输入锁定结构的地址。

输入锁定结构：

```

struct CThostFtdcInputLockField
{
    ///经纪公司代码

```

```

    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType   InvestorID;
    ///合约代码

    TThostFtdcInstrumentIDType InstrumentID;
    ///锁定引用

    TThostFtdcOrderRefType     LockRef;
    ///用户代码

    TThostFtdcUserIDType       UserID;
    ///数量

    TThostFtdcVolumeType       Volume;
    ///请求编号

    TThostFtdcRequestIDType     RequestID;
    ///业务单元

    TThostFtdcBusinessUnitType BusinessUnit;
    ///锁定类型

    TThostFtdcLockTypeType      LockType;
    ///交易所代码

    TThostFtdcExchangeIDType    ExchangeID;
};

```

LockRef：锁定报单引用，只能单调递增。每次登成功后，可以从 *OnRspUserLogin* 的输出参数 *CThostFtdcRspUserLoginField* 中获得上次登录用过的最大 *OrderRef:MaxOrderRef*。

4.4.73. ReqCombActionInsert 方法

客户端发出组合录入请求。

函数原型：

```

int ReqCombActionInsert (
    CThostFtdcInputCombActionField *pInputCombAction,,
    int nRequestID);

```

参数：

pInputCombAction：指向组合录入结构的地址。

组合录入结构：

```

struct CThostFtdcInputCombActionField

```

```

{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///组合引用
    TThostFtdcOrderRefType    CombActionRef;
    ///用户代码
    TThostFtdcUserIDType      UserID;
    ///买卖方向
    TThostFtdcDirectionType   Direction;
    ///数量
    TThostFtdcVolumeType      Volume;
    ///组合指令方向
    TThostFtdcCombDirectionType CombDirection;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///交易所代码
    TThostFtdcExchangeIDType   ExchangeID;
};

```

CombActionRef: 组合报单引用，只能单调递增。每次登录后，可以从 OnRspUserLogin 的输出参数 CThostFtdcRspUserLoginField 中获得上次登录用过的最大 OrderRef:MaxOrderRef。

4.4.74. ReqQryProductExchRate 方法

客户端发出查询产品报价费率请求。

函数原型：

```

int ReqQryProductExchRate (
    CThostFtdcQryProductExchRateField *pQryProductExchRate,
    int nRequestID);

```

参数：

pQryProductExchRate: 指查询产品报价费率结构的地址。

查询产品报价费率结构:

```
struct CThostFtdcQryProductExchRateField
{
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
};
```

4.4.75. ReqQryProductGroup 方法

客户端发出查询产品组请求。

函数原型:

```
int ReqQryProductGroup(
    CThostFtdcQryProductGroupField *pQryProductGroup,
    int nRequestID);
```

参数:

pQryProductGroup: 指向查询产品组请求结构。

查询产品组请求结构:

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

4.4.76. ReqQryInstrumentOrderCommRate 方法

客户端发出查询报单手续费请求。

函数原型:

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

参数:

pQryInstrumentOrderCommRate: 指向查询报单手续费请求结构。

查询报单手续费请求结构:

```
struct CThostFtdcQryInstrumentOrderCommRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
```

```

    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.4.77. ReqQryOptionInstrTradeCost 方法

客户端发出查询期权交易成本请求。

函数原型：

```

int ReqQryOptionInstrTradeCost(
    CThostFtdcQryOptionInstrTradeCostField *pQryOptionInstrTradeCost,
    int nRequestID);

```

参数：

pQryOptionInstrTradeCost: 指向查询期权交易成本请求结构。

查询期权交易成本请求结构：

```

struct CThostFtdcQryOptionInstrTradeCostField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///期权合约报价
    TThostFtdcPriceType InputPrice;
    ///标的价格,填则用昨结算价
    TThostFtdcPriceType UnderlyingPrice;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.4.78. ReqQryOptionInstrCommRate 方法

客户端发出查询期权合约手续费请求。

函数原型：

```

int ReqQryOptionInstrCommRate(
    CThostFtdcQryOptionInstrCommRateField *pQryOptionInstrCommRate,
    int nRequestID);

```

参数:

pQryOptionInstrCommRate: 指向查询期权合约手续费请求结构。

查询期权合约手续费请求结构:

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

4.4.79. ReqQryExecOrder 方法

客户端发出查询执行宣告请求。

函数原型:

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

参数:

pQryExecOrder: 指向查询执行宣告请求结构。

查询执行宣告请求结构:

```
struct CThostFtdcQryExecOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///执行宣告编号
    TThostFtdcExecOrderSysIDType ExecOrderSysID;
    ///开始时间
    TThostFtdcTimeType       InsertTimeStart;
    ///结束时间
```

```

        TThostFtdcTimeType    InsertTimeEnd;
    };

```

4.4.80. ReqQryLock 方法

客户端发出查询锁定请求。

函数原型：

```

int ReqQryLock (
    CThostFtdcQryLockField *pQryLock,
    int nRequestID);

```

参数：

pQryLock： 指查询锁定结构的地址。

查询锁定结构：

```

struct CThostFtdcQryLockField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///锁定编号
    TThostFtdcOrderSysIDType   LockSysID;
    ///开始时间
    TThostFtdcTimeType         InsertTimeStart;
    ///结束时间
    TThostFtdcTimeType         InsertTimeEnd;
};

```

4.4.81. ReqQryLockPosition 方法

客户端发出查询锁定证券仓位请求。

函数原型：

```

int ReqQryLockPosition (
    CThostFtdcQryLockPositionField *pQryLockPosition,

```



```
int nRequestID);
```

参数:

pQryLockPosition: 指查询锁定证券仓位结构的地址。

查询锁定证券仓位结构:

```
struct CThostFtdcQryLockPositionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.4.82. ReqQryCombInstrumentGuard 方法

客户端发出查询组合合约安全系数请求。

函数原型:

```
int ReqQryCombInstrumentGuard (
    CThostFtdcQryCombInstrumentGuardField *pQryCombInstrumentGuard,
    int nRequestID);
```

参数:

pQryCombInstrumentGuard: 指查询组合合约安全系数结构的地址。

查询组合合约安全系数结构:

```
struct CThostFtdcCombInstrumentGuardField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
};
```

4.4.83. ReqQryCombAction 方法

客户端发出查询申请组合请求。

函数原型:

```
int ReqQryCombAction (
    CThostFtdcQryCombActionField *pQryCombAction,
    int nRequestID);
```

参数:

pQryCombAction: 指查询申请组合结构的地址。

查询申请组合结构:

```
struct CThostFtdcQryCombActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.4.84. ReqQryRiskSettleInvstPosition 方法

客户端发出查询投资者风险结算持仓请求。

函数原型:

```
int ReqQryRiskSettleInvstPosition(
    CThostFtdcQryRiskSettleInvstPositionField*pQryRiskSettleInvstPosition,
    int nRequestID);
```

参数:

pQryRiskSettleInvstPosition: 指查询投资者风险结算持仓结构的地址。

查询投资者风险结算持仓结构:

```
struct CThostFtdcQryRiskSettleInvstPositionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
```

```

        /// 合约代码
        TThostFtdcInstrumentIDType InstrumentID;
    };

```

4.4.85. ReqQryRiskSettleProductStatus 方法

客户端发出查询风险结算产品请求。

函数原型：

```

int ReqQryRiskSettleProductStatus(
    CThostFtdcQryRiskSettleProductStatusField* pQryRiskSettleProductStatus,
    int nRequestID);

```

参数：

pQryRiskSettleProductStatus： 指查询风险结算产品结构的地址。

查询风险结算产品结构：

```

struct CThostFtdcQryRiskSettleProductStatusField
{
    /// 产品代码
    TThostFtdcInstrumentIDType ProductID;
};

```

4.4.86. ReqOptionSelfCloseAction 方法

客户端发出期权自对冲操作请求。

函数原型：

```

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

```

参数：

pInputOptionSelfCloseAction： 指向期权自对冲操作请求的地址。

期权自对冲操作请求结构：

```

struct CThostFtdcInputOptionSelfCloseActionField
{
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
};

```

```

///期权自对冲操作引用
TThostFtdcOrderActionRefType  OptionSelfCloseActionRef;
///期权自对冲引用
TThostFtdcOrderRefType  OptionSelfCloseRef;
///请求编号
TThostFtdcRequestIDType  RequestID;
///前置编号
TThostFtdcFrontIDType FrontID;
///会话编号
TThostFtdcSessionIDType  SessionID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///期权自对冲操作编号
TThostFtdcOrderSysIDType  OptionSelfCloseSysID;
///操作标志
TThostFtdcActionFlagType  ActionFlag;
///用户代码
TThostFtdcUserIDType  UserID;
///合约代码
TThostFtdcInstrumentIDType InstrumentID;
///投资单元代码
TThostFtdcInvestUnitIDType InvestUnitID;
///IP 地址
TThostFtdcIPAddressType  IPAddress;
///Mac 地址
TThostFtdcMacAddressType  MacAddress;
};

```

说明：撤单请调用 CKSVocApi 的同名接口，并传入自对冲类型；大商所取消到期日自动行权同样调用 CKSVocApi 的同名接口，且将数量置为 0，对冲类型为'4'。

4.4.87. ReqQryCustomerSuitableEvaResults 方法

客户端发出查询客户适当性评测结果请求。

函数原型：

```
int ReqQryCustomerSuitableEvaResults(
    CThostFtdcQryCustomerSuitableEvaResultsField* pQryCustomerSuitableEvaResults,
    int nRequestID);
```

参数：

pQryCustomerSuitableEvaResults： 指查询客户适当性评测结果结构的地址。

查询客户适当性评测结果结构：

```
struct CThostFtdcQryCustomerSuitableEvaResultsField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
};
```

4.4.88. ReqQrySPBMFutureParameter 方法

SPBM 期货合约参数查询请求。

函数原型：

```
int ReqQrySPBMFutureParameter(CThostFtdcQrySPBMFutureParameterField
    *pQrySPBMFutureParameter, int nRequestID);
```

参数：

pQrySPBMFutureParameter： 指向 SPBM 期货合约参数查询请求地址。

SPBM 期货合约参数查询结构：

```
struct CThostFtdcQrySPBMFutureParameterField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
};
```

4.4.89. ReqQrySPBMOptionParameter 方法

SPBM 期权合约参数查询请求。

函数原型：

```
int ReqQrySPBMOptionParameter(CThostFtdcQrySPBMOptionParameterField
    *pQrySPBMOptionParameter, int nRequestID);
```

参数：

pQrySPBMOptionParameter：指向 SPBM 期权合约参数查询请求地址。

SPBM 期权合约参数查询结构：

```
struct CThostFtdcQrySPBMOptionParameterField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
};
```

4.4.90. ReqQrySPBMIntraParameter 方法

SPBM 品种内对锁仓折扣参数查询请求。

函数原型：

```
int ReqQrySPBMIntraParameter(CThostFtdcQrySPBMIntraParameterField
    *pQrySPBMIntraParameter, int nRequestID);
```

参数：

pQrySPBMIntraParameter：指向 SPBM 品种内对锁仓折扣参数查询地址。

SPBM 品种内对锁仓折扣参数查询结构：

```
struct CThostFtdcQrySPBMIntraParameterField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///品种代码
```

```

        TThostFtdcInstrumentIDTypeProdFamilyCode;
    };

```

4.4.91. ReqQrySPBMInterParameter 方法

SPBM 跨品种抵扣参数查询。

函数原型：

```

int ReqQrySPBMInterParameter(CThostFtdcQrySPBMInterParameterField
    *pQrySPBMInterParameter, int nRequestID);

```

参数：

pQrySPBMInterParameter：指向 SPBM 跨品种抵扣参数查询地址。

SPBM 跨品种抵扣参数查询结构：

```

struct CThostFtdcQrySPBMInterParameterField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///第一腿构成品种
    TThostFtdcInstrumentIDTypeLeg1ProdFamilyCode;
    ///第二腿构成品种
    TThostFtdcInstrumentIDTypeLeg2ProdFamilyCode;
};

```

4.4.92. ReqQrySPBMPortfDefinition 方法

SPBM 组合保证金套餐查询。

函数原型：

```

int ReqQrySPBMPortfDefinition(CThostFtdcQrySPBMPortfDefinitionField
    *pQrySPBMPortfDefinition, int nRequestID);

```

参数：

pQrySPBMPortfDefinition：指向 SPBM 组合保证金查询地址。

SPBM 组合保证金查询结构：

```

struct CThostFtdcQrySPBMPortfDefinitionField
{
    ///交易所代码

```

```

    TThostFtdcExchangeIDType ExchangeID;
    ///组合保证金套餐代码

    TThostFtdcPortfolioDefIDType PortfolioDefID;
    ///品种代码

    TThostFtdcInstrumentIDType ProdFamilyCode;

};

```

4.4.93. ReqQrySPBMInvestorPortfDef 方法

投资者 SPBM 套餐选择查询。

函数原型：

```

int ReqQrySPBMInvestorPortfDef(CThostFtdcQrySPBMInvestorPortfDefField
    *pQrySPBMInvestorPortfDef, int nRequestID);

```

参数：

pQrySPBMInvestorPortfDef：指向投资者 SPBM 套餐查询地址。

SPBM 套餐查询结构：

```

struct CThostFtdcQrySPBMInvestorPortfDefField
{
    ///交易所代码

    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码

    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType InvestorID;

};

```

4.4.94. ReqQryInvestorPortfMarginRatio 方法

投资者新型组合保证金系数查询。

函数原型：

```

int ReqQryInvestorPortfMarginRatio(CThostFtdcQryInvestorPortfMarginRatioField
    *pQryInvestorPortfMarginRatio, int nRequestID);

```

参数：

pQryInvestorPortfMarginRatio：投资者新型组合保证金系数查询地址。

投资者新型组合保证金系数查询结构:

```
struct CThostFtdcQryInvestorPortfMarginRatioField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
};
```

4.4.95. ReqQryInvestorProdSPBMDetail 方法

投资者产品 SPBM 明细查询。

函数原型:

```
int ReqQryInvestorProdSPBMDetail(CThostFtdcQryInvestorProdSPBMDetailField
    *pQryInvestorProdSPBMDetail, int nRequestID);
```

参数:

pQryInvestorProdSPBMDetail: 投资者产品 SPBM 明细查询地址。

投资者产品 SPBM 明细查询结构:

```
struct CThostFtdcQryInvestorProdSPBMDetailField
{
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///品种代码
    TThostFtdcInstrumentIDTypeProdFamilyCode;
};
```

4.4.96. ReqQryPortfTradeParamSetting 方法

组保交易参数设置查询。

函数原型：

```
int ReqQryPortfTradeParamSetting(CThostFtdcQryPortfTradeParamSettingField
    *pQryPortfTradeParamSetting, int nRequestID);
```

参数：

pQryPortfTradeParamSetting：组保交易参数设置查询地址。

组保交易参数设置查询结构：

```
struct CThostFtdcQryPortfTradeParamSettingField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易编码
    TThostFtdcClientIDType ClientID;
};
```

4.4.97. ReqQrySPBMAddOnInterParameter 方法

SPBM 跨品种附加抵扣参数查询。

函数原型：

```
int ReqQrySPBMAddOnInterParameter(CThostFtdcQrySPBMAddOnInterParameterField
    *pQrySPBMAddOnInterParameter, int nRequestID);
```

参数：

pQrySPBMInterParameter：指向 SPBM 跨品种抵扣参数查询地址。

SPBM 跨品种抵扣参数查询结构：

```
struct CThostFtdcQrySPBMAddOnInterParameterField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
```

```

        ///第一腿构成品种
        TThostFtdcInstrumentIDTypeLeg1ProdFamilyCode;
        ///第二腿构成品种
        TThostFtdcInstrumentIDTypeLeg2ProdFamilyCode;
    };

```

4.4.98. ReqQrySPMMInstParam 方法

SPMM 合约参数查询。

函数原型：

```

int ReqQrySPMMInstParam(CThostFtdcQrySPMMInstParamField *pQrySPMMInstParam,
    int nRequestID);

```

参数：

pQrySPMMInstParam：指向 SPMM 合约参数查询地址。

SPMM 合约参数查询结构：

```

struct CThostFtdcQrySPMMInstParamField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.4.99. ReqQrySPMMProductParam 方法

SPMM 产品参数查询。

函数原型：

```

ReqQrySPMMProductParam(CThostFtdcQrySPMMProductParamField
    *pQrySPMMProductParam, int nRequestID);

```

参数：

pQrySPMMProductParam：指向 SPMM 产品参数查询地址。

SPMM 产品参数查询结构：

```

struct CThostFtdcQrySPMMProductParamField
{

```

```

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcSPMMProductIDType ProductID;
};

```

4.4.100. ReqQryInvestorCommoditySPMMMargin 方法

投资者商品组 SPMM 记录查询。

函数原型：

```

int ReqQryInvestorCommoditySPMMMargin(
    CThostFtdcQryInvestorCommoditySPMMMarginField
    *pQryInvestorCommoditySPMMMargin, int nRequestID);

```

参数：

pQryInvestorCommoditySPMMMargin：指向投资者商品组 SPMM 记录查询地址。

投资者商品组 SPMM 记录查询结构：

```

struct CThostFtdcQryInvestorCommoditySPMMMarginField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///商品组代码
    TThostFtdcSPMMProductIDType CommodityID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易编码
    TThostFtdcClientIDType ClientID;
};

```

4.4.101. ReqQryInvestorCommodityGroupSPMMMargin 方法

投资者商品群 SPMM 记录查询。

函数原型：

```
ReqQryInvestorCommodityGroupSPMMMargin(
    CThostFtdcQryInvestorCommodityGroupSPMMMarginField
    *pQryInvestorCommodityGroupSPMMMargin, int nRequestID);
```

参数:

pQryInvestorCommodityGroupSPMMMargin: 指向投资者商品群 SPMM 记录查询地址。

投资者商品群 SPMM 记录查询结构:

```
struct CThostFtdcQryInvestorCommodityGroupSPMMMarginField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///商品群代码
    TThostFtdcSPMMProductIDType CommodityGroupID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易编码
    TThostFtdcClientIDType ClientID;
};
```

4.4.102. ReqQryRULEInstrParameter 方法

RULE 合约保证金参数查询。

函数原型:

```
int ReqQryRULEInstrParameter(
    CThostFtdcQryRULEInstrParameterField *pQryRULEInstrParameter,
    int nRequestID);
```

参数:

pQryRULEInstrParameter: 指向 RULE 合约保证金参数查询地址。

RULE 合约保证金参数查询结构:

```
struct CThostFtdcQryRULEInstrParameterField
{
    ///交易所代码
```

```

    TThostFtdcExchangeIDType ExchangeID;
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.4.103. ReqQryRULEIntraParameter 方法

RULE 品种内对锁仓折扣参数查询。

函数原型：

```

ReqQryRULEIntraParameter(
    CThostFtdcQryRULEIntraParameterField *pQryRULEIntraParameter,
    int nRequestID);

```

参数：

CThostFtdcQryRULEIntraParameterField : 指向 RULE 品种内对锁仓折扣参数地址。

RULE 品种内对锁仓折扣参数查询结构：

```

struct CThostFtdcQryRULEIntraParameterField
{
    /// 交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    /// 品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
};

```

4.4.104. ReqQryRULEInterParameter 方法

RULE 跨品种抵扣参数查询。

函数原型：

```

int ReqQryRULEInterParameter(
    CThostFtdcQryRULEInterParameterField *pQryRULEInterParameter,
    int nRequestID);

```

参数：

CThostFtdcQryRULEInterParameterField : 指向 RULE 跨品种抵扣参数查询参数地址。

RULE 跨品种抵扣参数查询结构：

```

struct CThostFtdcQryRULEInterParameterField

```

```

{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///第一腿构成品种
    TThostFtdcInstrumentIDType Leg1ProdFamilyCode;
    ///第二腿构成品种
    TThostFtdcInstrumentIDType Leg2ProdFamilyCode;
    ///商品群号
    TThostFtdcCommodityGroupIDType CommodityGroupID;
};

```

4.4.105. ReqQryInvestorProdRULEMargin 方法

投资者产品 RULE 保证金查询。

函数原型：

```

int ReqQryInvestorProdRULEMargin(
    CThostFtdcQryInvestorProdRULEMarginField *pQryInvestorProdRULEMargin,
    int nRequestID);

```

参数：

CThostFtdcQryInvestorProdRULEMarginField : 指向投资者产品 RULE 保证金参数地址。

投资者产品 RULE 保证金查询结构：

```

struct CThostFtdcQryInvestorProdRULEMarginField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///商品群号
    TThostFtdcCommodityGroupIDType CommodityGroupID;
};

```

```
};
```

4.5. CthostFtdcMdSpi 接口

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

1.1.1 OnFrontConnected 方法

当客户端与 Kingstar 服务器建立起通信连接时（还未登录前），该方法被调用。

函数原型：

```
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。

4.5.1. OnFrontDisconnected 方法

当客户端与 Kingstar 服务器通信连接断开时，该方法被调用。

函数原型：

```
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

0x2004 服务器主动断开

4.5.2. 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。

函数原型：

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

参数：

nTimeLapse: 距离上次接收报文的时间

4.5.3. OnRspUserLogin 方法

当客户端发出登录请求之后，Kingstar 服务器返回响应时，该方法会被调用，通知客户端登录是否成功。

函数原型：


```
void OnRspUserLogin(
    CThostFtdcRspUserLoginField *pRspUserLogin,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pRspUserLogin: 返回用户登录信息的地址。

用户登录信息结构:

```
struct CThostFtdcRspUserLoginField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///登录成功时间
    TThostFtdcTimeType    LoginTime;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///交易系统名称
    TThostFtdcSystemNameType    SystemName;
    ///前置编号
    TThostFtdcFrontIDType    FrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///最大报单引用
    TThostFtdcOrderRefType    MaxOrderRef;
    ///上期所时间
    TThostFtdcTimeType    SHFETime;
    ///大商所时间
    TThostFtdcTimeType    DCETime;
    ///郑商所时间
    TThostFtdcTimeType    CZCETime;
```

```

        ///中金所时间
        TThostFtdcTimeType    FFEXTime;

        ///能源中心时间
        TThostFtdcTimeType    INETime;
};

```

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

响应信息结构:

```

struct CThostFtdcRspInfoField
{
    ///错误代码
    TThostFtdcErrorIDType ErrorID;

    ///错误信息
    TThostFtdcErrorMsgType    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。

4.5.4. OnRspUserLogout 方法

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

函数原型:

```

void OnRspUserLogout(
    CThostFtdcUserLogoutField *pUserLogout,
    CThostFtdcRspInfoField *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 CThostFtdcUserLogoutField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
};

```

```

    ///用户代码
    TThostFtdcUserIDType UserID;
};

```

4.5.5. OnRspError 方法

针对用户请求的出错通知。

函数原型：

```

void OnRspError(
    CThostFtdcRspInfoField *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 CThostFtdcRspInfoField
{
    ///错误代码
    TThostFtdcErrorIDType ErrorID;
    ///错误信息
    TThostFtdcErrorMsgType ErrorMsg;
};

```

4.5.6. OnRspSubMarketData 方法

当客户端发出订阅行情请求之后，Kingstar 返回响应时，该方法会被调用，通知客户端订阅是否成功。

函数原型：

```

void OnRspSubMarketData(
    CThostFtdcSpecificInstrumentField *pSpecificInstrument,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

pSpecificInstrument： 返回客户端订阅行情的合约结构地址。

合约结构：

```
struct CThostFtdcSpecificInstrumentField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};
```

4.5.7. OnRspUnSubMarketData 方法

当客户端发出退订行情请求之后，Kingstar 返回响应时，该方法会被调用，通知客户端退订是否成功。

函数原型：

```
void OnRspUnSubMarketData (
    CThostFtdcSpecificInstrumentField *pSpecificInstrument,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pSpecificInstrument： 返回客户端订阅行情的合约结构地址。

合约结构：

```
struct CThostFtdcSpecificInstrumentField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};
```

4.5.8. OnRspSubForQuoteRsp 方法

订阅查询应答。

函数原型：

```
void OnRspSubForQuoteRsp(
    CThostFtdcSpecificInstrumentField *pSpecificInstrument,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pSpecificInstrument： 指向查询订阅应答结构。

查询订阅应答结构：

```
struct CThostFtdcSpecificInstrumentField
{
    /// 合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
};
```

4.5.9. OnRspUnSubForQuoteRsp 方法

取消订阅询价应答。

函数原型：

```
void OnRspUnSubForQuoteRsp(
    CThostFtdcSpecificInstrumentField *pSpecificInstrument,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pSpecificInstrument： 指向取消订阅询价应答结构。

取消订阅询价应答结构：

```
struct CThostFtdcSpecificInstrumentField
{
    /// 合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
};
```

4.5.10. OnRtnDepthMarketData 方法

行情推送。当发生行情订阅成功时，Kingstar 服务器会通知客户端，该方法会被调用。

函数原型：

```
void OnRtnDepthMarketData(CThostFtdcDepthMarketDataField *pDepthMarketData);
```

参数：

pDepthMarketData： 返回客户端订阅的行情结构地址。

行情结构：

```
struct CThostFtdcDepthMarketDataField
{
    /// 交易日
    TThostFtdcDateType    TradingDay;
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
```

```
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///最新价
TThostFtdcPriceType LastPrice;
///上次结算价
TThostFtdcPriceType PreSettlementPrice;
///昨收盘
TThostFtdcPriceType PreClosePrice;
///昨持仓量
TThostFtdcLargeVolumeTypePreOpenInterest;
///今开盘
TThostFtdcPriceType OpenPrice;
///最高价
TThostFtdcPriceType HighestPrice;
///最低价
TThostFtdcPriceType LowestPrice;
///数量
TThostFtdcVolumeType Volume;
///成交金额
TThostFtdcMoneyType Turnover;
///持仓量
TThostFtdcLargeVolumeTypeOpenInterest;
///今收盘
TThostFtdcPriceType ClosePrice;
///本次结算价
TThostFtdcPriceType SettlementPrice;
///涨停板价
TThostFtdcPriceType UpperLimitPrice;
///跌停板价
TThostFtdcPriceType LowerLimitPrice;
```

```

///昨虚实度
TThostFtdcRatioType    PreDelta;

///今虚实度
TThostFtdcRatioType    CurrDelta;

///最后修改时间
TThostFtdcTimeType     UpdateTime;

///最后修改毫秒
TThostFtdcMillisecType UpdateMillisec;

///申买价一
TThostFtdcPriceType    BidPrice1;

///申买量一
TThostFtdcVolumeType  BidVolume1;

///申卖价一
TThostFtdcPriceType    AskPrice1;

///申卖量一
TThostFtdcVolumeType  AskVolume1;

///申买价二
TThostFtdcPriceType    BidPrice2;

///申买量二
TThostFtdcVolumeType  BidVolume2;

///申卖价二
TThostFtdcPriceType    AskPrice2;

///申卖量二
TThostFtdcVolumeType  AskVolume2;

///申买价三
TThostFtdcPriceType    BidPrice3;

///申买量三
TThostFtdcVolumeType  BidVolume3;

///申卖价三
TThostFtdcPriceType    AskPrice3;

///申卖量三
TThostFtdcVolumeType  AskVolume3;

```

```

    /// 申买价四
    TThostFtdcPriceType BidPrice4;
    /// 申买量四
    TThostFtdcVolumeType BidVolume4;
    /// 申卖价四
    TThostFtdcPriceType AskPrice4;
    /// 申卖量四
    TThostFtdcVolumeType AskVolume4;
    /// 申买价五
    TThostFtdcPriceType BidPrice5;
    /// 申买量五
    TThostFtdcVolumeType BidVolume5;
    /// 申卖价五
    TThostFtdcPriceType AskPrice5;
    /// 申卖量五
    TThostFtdcVolumeType AskVolume5;
    /// 当日均价
    TThostFtdcPriceType AveragePrice;
};

```

4.6. CthostFtdcMdApi 接口

CthostFtdcMdApi 接口提供给用户的功能主要体现在行情的订阅、退订等功能。

4.6.1. CreateFtdcMdApi 方法

客户端应用程序使用此函数来创建一个“CthostFtdcMdApi”实例。注意：创建实例时不要使用“new”。

函数原型：

```
static CThostFtdcMdApi *CreateFtdcMdApi (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行情系统发布消息的状态。默认值代表当前目录。

返回值：

返回一个指向 CThostFtdcMdApi 实例的指针。

4.6.2. GetApiVersion 方法

获得 API 版本号。只有当与 Kingstar 服务器连接建立后才会取到正确的值。

函数原型:

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

返回值:

返回一个指向日期信息字符串的常量指针。

4.6.3. Release 方法

释放一个 CThostFtdcMdApi 实例。不能使用 delete 方法

函数原型:

```
void Release();
```

4.6.4. Init 方法

使客户端开始与 Kingstar 服务器建立连接，连接成功后可以进行登陆。

函数原型:

```
void Init();
```

4.6.5. Join 方法

客户端等待一个接口实例线程的结束。

函数原型:

```
void Join();
```

4.6.6. GetTradingDay 方法

获得当前交易日。只有当与 Kingstar 服务器连接建立后才会取到正确的值。

函数原型:

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

返回值:

返回一个指向日期信息字符串的常量指针。

4.6.7. RegisterFront 方法

设置行情系统的网络通讯地址，Kingstar 服务器拥有多个通信地址，但用户只需要选择一个通信地址。客户端 RegisterFront 多次时，使用第一个有效的通信地址。

函数原型:

```
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://127.0.0.1:17993"。 "tcp"代表传输协议, "127.0.0.1"

代表服务器地址, "17993"代表服务器端口号。同时 API 也提供了对代理服务器的支持, 包括 socks4、socks5 及 http, 客户端开发时只需通过传递给 API 不同的连接字符串就可实现, 例如 "socks5://127.0.0.1:17993 /user:pass@127.0.0.1:18993" 。

4.6.8. RegisterSpi 方法

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

函数原型:

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

参数:

pSpi: 实现了 CThostFtdcMdSpi 接口的实例指针。

4.6.9. SubscribeMarketData 方法

用户发出订阅行情的请求。

函数原型:

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

参数:

***ppInstrumentID[]:** 指向用户订阅行情的合约结构的地址。

nCount: 行情订阅的合约个数。

4.6.10. UnSubscribeMarketData 方法

用户发出退订行情的请求。

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

参数:

***ppInstrumentID[]:** 指向用户退订行情的合约结构的地址

nCount: 退订行情的合约个数。

4.6.11. SubscribeForQuoteRsp 方法

用户发出订阅询价的请求。

函数原型:

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

参数:

ppInstrumentID[]: 指向用户订阅询价的结构体的地址。

nCount: 询价订阅的个数。

4.6.12. UnSubscribeForQuoteRsp 方法

用户发出退订询价的请求。

函数原型:

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

参数:

ppInstrumentID[]: 指向用户退订询价的结构体的地址。

nCount: 询价退订的个数。

4.6.13. ReqUserLogin 方法

用户发出登陆请求。

函数原型:

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

参数:

pReqUserLoginField: 指向用户登录请求结构体的地址。

用户登录请求结构:

```
struct CThostFtdcReqUserLoginField
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///密码
    TThostFtdcPasswordType    Password;
    ///用户端产品信息
    TThostFtdcProductInfoType    UserProductInfo;
```

```

    ///接口端产品信息
    TThostFtdcProductInfoType InterfaceProductInfo;
    ///协议信息
    TThostFtdcProtocolInfoType ProtocolInfo;
    ///Mac 地址
    TThostFtdcMacAddressType MacAddress;
    ///动态密码
    TThostFtdcPasswordType OneTimePassword;
    ///终端 IP 地址
    TThostFtdcIPAddressType ClientIPAddress;
};

```

4.6.14. ReqUserLogout 方法

用户发出登出请求。

函数原型：

```

int ReqUserLogout(
    CThostFtdcUserLogoutField *pUserLogout,
    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 CThostFtdcUserLogoutField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///用户代码
    TThostFtdcUserIDType UserID;
};

```

4.6.15. RegisterFensUserInfo 方法

注册名字服务器用户信息

函数模型：

```

void RegisterFensUserInfo(CThostFtdcFensUserInfoField * pFensUserInfo);

```

参数：

pFensUserInfo: 指向名字服务器用户信息的指针。

4.6.16. RegisterNameServer 方法

注册名字服务器（门户）

客户端可通过调用此方法注册名字服务器从而获取最优网关以实现快速登录。Kingstar API 支持多次注册名字服务器，也就是说，此函数可以被调用多次，每次调用都能实现从此名字服务器相关的最优网关列表中返回一个网关。需要注意的是，一旦多次调用此方法，后续返回的网关地址会替代上次返回的网关地址。

目前此接口只支持 API 的 windows 版本。

函数模型：

```
void RegisterNameServer(char *pszNameServerAddress);
```

参数：

pszNameServerAddress: 指向名字服务器地址（也叫门户服务器地址）的指针。名字服务器地址格式如下所示：“协议:ip 地址:端口/代理用户名:代理用户密码@代理服务器 ip 地址:代理服务器端口/网关标识/客户号”。其中代理相关字段是可选的。例如：

“tcp://127.0.0.1:11000/A/80001”，“tcp”代表通讯协议，“127.0.0.1”标识名字服务器的 ip 地址，“11000”标识名字服务器的端口，“A”代表网关标识，“80001”标识客户号。

网关标识字段如下所示：

值	描述
‘A’	大智慧
‘B’	文华一键通
‘C’	彭博闪电手
‘D’	托瑞邦泽
‘E’	快期
‘F’	金字塔
‘G’	达钱

4.6.17. LoadExtMdApi 方法

用户加载扩展 API 实例。

函数原型：

```
void *LoadExtMdApi(
    void *pSpi,
    KS_EXTAPI_TYPE nExtApiType);
```

参数：

nExtApiType: 扩展 API 类型。

4.7. CKSCosSpi 接口

CKSCosSpi 实现了条件单事件通知接口。使用 Kingstar 条件单时, 用户必需派生 CKSCosSpi 接口, 编写事件处理方法来处理感兴趣的事件。

4.7.1. OnRspInitInsertConditionalOrder 方法

条件单录入响应

函数模型:

```
void OnRspInitInsertConditionalOrder(
    CKSConditionalOrderOperResultField *pInitInsertConditionalOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pInitInsertConditionalOrder: 指向条件单响应结构的指针。

条件单响应结构如下所示:

```
struct CKSConditionalOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType OrderLocalID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///报单状态
    TThostFtdcOrderStatusType OrderStatus;
    ///开平标志
    TThostFtdcOffsetFlagType CombOffsetFlag;
```

```
///投机套保标志
TThostFtdcHedgeFlagType CombHedgeFlag;
///买卖方向
TThostFtdcDirectionType Direction;
///价格
TThostFtdcPriceType LimitPrice;
///数量
TThostFtdcVolumeType VolumeTotalOriginal;
///交易日
TThostFtdcTradeDateType TradingDay;
///撤销用户
TThostFtdcUserIDType UserID;
///撤销时间
TThostFtdcTimeType CancelTime;
///客户代码
TThostFtdcClientIDType ClientID;
///条件单状态
TTKSConditionalOrderStatusType ConditionalOrderStatus;
///错误信息
TThostFtdcErrorMsgType ErrorMsg;
///价格类别
TThostFtdcOrderPriceTypeType OrderPriceType;
///触发次数
TThostFtdcVolumeType TriggeredTimes;
///条件单类型
TTKSConditionalOrderType OrderType;
///备注
TThostFtdcMemoType Memo;
///有效时间
TThostFtdcTimeType ActiveTime;
///失效时间
TThostFtdcTimeType InActiveTime;
///条件类型
TKSConditionalTypeType ConditionalType;
```

```

        ///条件价
        TThostFtdcPriceType ConditionalPrice;
    };

```

4.7.2. OnRspQueryConditionalOrder 方法

条件单查询返回响应

函数模型:

```

void OnRspQueryConditionalOrder(
    CKSConditionalOrderOperResultField *pQueryConditionalOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pQueryConditionalOrder: 指向条件单响应结构的指针

条件单响应结构如下所示:

```

struct CKSConditionalOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType    ConditionalOrderID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType    OrderLocalID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///报单状态
    TThostFtdcOrderStatusType OrderStatus;
    ///开平标志
    TThostFtdcOffsetFlagType CombOffsetFlag;
    ///投机套保标志

```



```

TThostFtdcHedgeFlagType CombHedgeFlag;
///买卖方向

TThostFtdcDirectionType Direction;
///价格

TThostFtdcPriceType LimitPrice;
///数量

TThostFtdcVolumeType VolumeTotalOriginal;
///交易日

TThostFtdcTradeDateType TradingDay;
///撤销用户

TThostFtdcUserIDType UserID;
///撤销时间

TThostFtdcTimeType CancelTime;
///客户代码

TThostFtdcClientIDType ClientID;
///条件单状态

TTKSConditionalOrderStatusType ConditionalOrderStatus;
///错误信息

TThostFtdcErrorMsgType ErrorMsg;
///价格类别

TThostFtdcOrderPriceTypeType OrderPriceType;
///触发次数

TThostFtdcVolumeType TriggeredTimes;
///条件单类型

TTKSConditionalOrderType OrderType;
///备注

TThostFtdcMemoType Memo;
///有效时间

TThostFtdcTimeType ActiveTime;
///失效时间

TThostFtdcTimeType InActiveTime;
///条件类型

TKSConditionalTypeType ConditionalType;
///条件价

```

```
    TThostFtdcPriceType ConditionalPrice;
};
```

4.7.3. OnRspModifyConditionalOrder 方法

条件单修改响应。

函数模型：

```
void OnRspModifyConditionalOrder(
    CKSConditionalOrderOperResultField *pModifyConditionalOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pModifyConditionalOrder: 指向条件单响应结构的指针

条件单响应结构如下所示：

```
struct CKSConditionalOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType    ConditionalOrderID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType    OrderLocalID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///报单状态
    TThostFtdcOrderStatusType OrderStatus;
    ///开平标志
    TThostFtdcOffsetFlagType CombOffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType CombHedgeFlag;
```

```
///买卖方向
TThostFtdcDirectionType Direction;
///价格
TThostFtdcPriceType LimitPrice;
///数量
TThostFtdcVolumeType VolumeTotalOriginal;
///交易日
TThostFtdcTradeDateType TradingDay;
///撤销用户
TThostFtdcUserIDType UserID;
///撤销时间
TThostFtdcTimeType CancelTime;
///客户代码
TThostFtdcClientIDType ClientID;
///条件单状态
TTKSConditionalOrderStatusType ConditionalOrderStatus;
///错误信息
TThostFtdcErrorMsgType ErrorMsg;
///价格类别
TThostFtdcOrderPriceTypeType OrderPriceType;
///触发次数
TThostFtdcVolumeType TriggeredTimes;
///条件单类型
TTKSConditionalOrderType OrderType;
///备注
TThostFtdcMemoType Memo;
///有效时间
TThostFtdcTimeType ActiveTime;
///失效时间
TThostFtdcTimeType InActiveTime;
///条件类型
TKSConditionalTypeType ConditionalType;
///条件价
TThostFtdcPriceType ConditionalPrice;
```

```
};
```

4.7.4. OnRspPauseConditionalOrder 方法

条件单暂停或激活操作响应。

函数模型：

```
void OnRspPauseConditionalOrder(
    CKSConditionalOrderOperResultField *pPauseConditionalOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pPauseConditionalOrder: 指向条件单响应结构的指针

条件单响应结构如下所示：

```
struct CKSConditionalOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType OrderLocalID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///报单状态
    TThostFtdcOrderStatusType OrderStatus;
    ///开平标志
    TThostFtdcOffsetFlagType CombOffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType CombHedgeFlag;
    ///买卖方向
```

```

    TThostFtdcDirectionType    Direction;
    ///价格
    TThostFtdcPriceType    LimitPrice;
    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///交易日
    TThostFtdcTradeDateType    TradingDay;
    ///撤销用户
    TThostFtdcUserIDType    UserID;
    ///撤销时间
    TThostFtdcTimeType    CancelTime;
    ///客户代码
    TThostFtdcClientIDType    ClientID;
    ///条件单状态
    TTKSConditionalOrderStatusType ConditionalOrderStatus;
    ///错误信息
    TThostFtdcErrorMsgType    ErrorMsg;
    ///价格类别
    TThostFtdcOrderPriceTypeType OrderPriceType;
    ///触发次数
    TThostFtdcVolumeType TriggeredTimes;
    ///条件单类型
    TTKSConditionalOrderType OrderType;
    ///备注
    TThostFtdcMemoType    Memo;
    ///有效时间
    TThostFtdcTimeType    ActiveTime;
    ///失效时间
    TThostFtdcTimeType InActiveTime;
    ///条件类型
    TKSConditionalTypeType    ConditionalType;
    ///条件价
    TThostFtdcPriceType ConditionalPrice;
};

```

4.7.5. OnRspRemoveConditionalOrder 方法

条件单删除响应。

函数模型：

```
void OnRspRemoveConditionalOrder(
    CKSConditionalOrderRspResultField *pRemoveConditionalOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pRemoveConditionalOrder: 指向条件单处理结果信息的指针。

条件单处理结果信息如下所示：

```
struct CKSConditionalOrderRspResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
};
```

4.7.6. OnRspSelectConditionalOrder 方法

条件单选择响应。

函数模型：

```
void OnRspSelectConditionalOrder(
    CKSConditionalOrderRspResultField *pSelectConditionalOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pSelectConditionalOrder: 指向条件单处理结果信息的指针。

条件单处理结果信息如下所示：

```
struct CKSConditionalOrderRspResultField
```

```

{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
};

```

4.7.7. OnRspInsertProfitAndLossOrder 方法

盈损单下单响应。

函数模型：

```

void OnRspInsertProfitAndLossOrder(
    CKSProfitAndLossOrderOperResultField *pInsertProfitAndLossOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

pInsertProfitAndLossOrder: 指向盈损单处理结构的指针。

盈损单处理结构如下所示：

```

struct CKSProfitAndLossOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
    ///操作员
    TThostFtdcUserIDType UserID;
    ///投资者名称
    TThostFtdcPartyNameType InvestorName;
    ///本地报单编号
    TThostFtdcOrderLocalIDType OrderLocalID;
}

```

```

///止损价
TThostFtdcPriceType StopLossPrice;
///止盈价
TThostFtdcPriceType TakeProfitPrice;
///平仓方式
TTKSCloseModeType CloseMode;
///平仓反向加减价位数（平仓方式=1或时有效）
TThostFtdcPriceType Figures;
///行情触发时的最新价
TThostFtdcPriceType LastPrice;
///生成止损止盈单时间
TThostFtdcTimeType ProfitAndLossOrderFormTime;
///生成条件单时间
TThostFtdcTimeType ConditionalOrderFormTime;
///生成委托单时间
TThostFtdcTimeType OrderFormTime;
///止损止盈单状态
TTKSConditionalOrderStatusType ProfitAndLossOrderStatus;
///条件单编号
TTKSConditionalOrderIDType ConditionalOrderID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///客户代码
TThostFtdcClientIDType ClientID;
///合约代码
TThostFtdcInstrumentIDType InstrumentID;
///开平标志
TThostFtdcOffsetFlagType CombOffsetFlag;
///投机套保标志
TThostFtdcHedgeFlagType CombHedgeFlag;
///买卖方向
TThostFtdcDirectionType Direction;
///价格
TThostFtdcPriceType LimitPrice;

```



```

    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///生成盈损价的方式
    TTKSOffsetValueType OffsetValue;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///条件单触发价格类型
    TTKSSpringType SpringType;
    ///浮动止损价
    TThostFtdcPriceType FloatLimitPrice;
    ///开仓成交价格
    TThostFtdcPriceType OpenTradePrice;
    ///止损止盈标志
    TKSProfitAndLossFlagType ProfitAndLossFlag;
};

```

4.7.8. OnRspModifyProfitAndLossOrder 方法

盈损单修改响应。

函数模型：

```

void OnRspModifyProfitAndLossOrder(
    CKSProfitAndLossOrderOperResultField *pModifyProfitAndLossOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

pModifyProfitAndLossOrder: 指向盈损单处理结构的指针。

盈损单处理结构如下所示：

```

struct CKSProfitAndLossOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号

```

```

TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
///操作员
TThostFtdcUserIDType UserID;
///投资者名称
TThostFtdcPartyNameType InvestorName;
///本地报单编号
TThostFtdcOrderLocalIDType    OrderLocalID;
///止损价
TThostFtdcPriceType    StopLossPrice;
///止盈价
TThostFtdcPriceType    TakeProfitPrice;
///平仓方式
TTKSCloseModeType CloseMode;
///平仓反向加减价位数（平仓方式=1或时有效）
TThostFtdcPriceType Figures;
///行情触发时的最新价
TThostFtdcPriceType LastPrice;
///生成止损止盈单时间
TThostFtdcTimeType    ProfitAndLossOrderFormTime;
///生成条件单时间
TThostFtdcTimeType    ConditionalOrderFormTime;
///生成委托单时间
TThostFtdcTimeType    OrderFormTime;
///止损止盈单状态
TTKSConditionalOrderStatusType ProfitAndLossOrderStatus;
///条件单编号
TTKSConditionalOrderIDType    ConditionalOrderID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///客户代码
TThostFtdcClientIDType    ClientID;
///合约代码
TThostFtdcInstrumentIDType    InstrumentID;

```

```

    ///开平标志
    TThostFtdcOffsetFlagType   CombOffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType   CombHedgeFlag;
    ///买卖方向
    TThostFtdcDirectionType   Direction;
    ///价格
    TThostFtdcPriceType   LimitPrice;
    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///生成盈损价的方式
    TTKSOffsetValueType OffsetValue;
    ///业务单元
    TThostFtdcBusinessUnitType   BusinessUnit;
    ///条件单触发价格类型
    TTKSSpringTypeType SpringType;
    ///浮动止损价
    TThostFtdcPriceType   FloatLimitPrice;
    ///开仓成交价格
    TThostFtdcPriceType OpenTradePrice;
    ///止损止盈标志
    TKSProfitAndLossFlagType ProfitAndLossFlag;
};

```

4.7.9. OnRspRemoveProfitAndLossOrder 方法

盈损单删除响应。

函数模型：

```

void OnRspRemoveProfitAndLossOrder(
    CKSProfitAndLossOrderRemoveField *pRemoveProfitAndLossOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

pRemoveProfitAndLossOrder: 指向止损止盈单删除结构的指针。

止损止盈单删除结构如下所示：

```
struct CKSProfitAndLossOrderRemoveField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
};
```

4.7.10. OnRspQueryProfitAndLossOrder 方法

止损止盈单查询响应。

函数模型：

```
void OnRspQueryProfitAndLossOrder(
    CKSProfitAndLossOrderOperResultField *pQueryProfitAndLossOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pQueryProfitAndLossOrder：指向盈损单处理结构的指针。

盈损单处理结构如下所示：

```
struct CKSProfitAndLossOrderOperResultField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
    ///操作员
    TThostFtdcUserIDType UserID;
    ///投资者名称
    TThostFtdcPartyNameType InvestorName;
```

```

///本地报单编号
TThostFtdcOrderLocalIDType    OrderLocalID;
///止损价
TThostFtdcPriceType    StopLossPrice;
///止盈价
TThostFtdcPriceType    TakeProfitPrice;
///平仓方式
TTKSCloseModeType CloseMode;
///平仓反向加减价位数（平仓方式=1或时有效）
TThostFtdcPriceType Figures;
///行情触发时的最新价
TThostFtdcPriceType LastPrice;
///生成止损止盈单时间
TThostFtdcTimeType    ProfitAndLossOrderFormTime;
///生成条件单时间
TThostFtdcTimeType    ConditionalOrderFormTime;
///生成委托单时间
TThostFtdcTimeType    OrderFormTime;
///止损止盈单状态
TTKSConditionalOrderStatusType ProfitAndLossOrderStatus;
///条件单编号
TTKSConditionalOrderIDType    ConditionalOrderID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///客户代码
TThostFtdcClientIDType    ClientID;
///合约代码
TThostFtdcInstrumentIDType    InstrumentID;
///开平标志
TThostFtdcOffsetFlagType    CombOffsetFlag;
///投机套保标志
TThostFtdcHedgeFlagType    CombHedgeFlag;
///买卖方向
TThostFtdcDirectionType    Direction;

```

```

    ///价格
    TThostFtdcPriceType   LimitPrice;
    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///生成盈损价的方式
    TTKSOffsetValueType OffsetValue;
    ///业务单元
    TThostFtdcBusinessUnitType   BusinessUnit;
    ///条件单触发价格类型
    TTKSSpringTypeType SpringType;
    ///浮动止损价
    TThostFtdcPriceType   FloatLimitPrice;
    ///开仓成交价格
    TThostFtdcPriceType OpenTradePrice;
    ///止损止盈标志
    TTKSProfitAndLossFlagType ProfitAndLossFlag;
};

```

4.7.11. OnRtnCOSAskSelect 方法

条件单请求选择通知。

函数模型：

```
void OnRtnCOSAskSelect(CKSCOSAskSelectField *pCOSAskSelect)
```

参数：

pCOSAskSelect: 指向条件单请求选择信息结构的指针。

条件单请求选择信息结构如下：

```

struct CKSCOSAskSelectField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///操作员
    TThostFtdcUserIDType UserID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///序号

```

```

TThostFtdcSequenceNoType SequenceNo;
///条件单编号
TTKSCONDITIONALORDERIDType ConditionalOrderID;
///备注
TThostFtdcMemoType Memo;
///选择方式
TTKSCONDITIONALORDERSELECTTYPEType SelectType;
};

```

4.7.12. OnRtnCOSStatus 方法

条件单状态通知。

函数模型：

```
void OnRtnCOSStatus(CKSCOSStatusField *pCOSStatus)
```

参数：

pCOSStatus：指向条件单状态结构的指针。

条件单状态结构如下所示：

```

struct CKSCOSStatusField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///操作员
    TThostFtdcUserIDType UserID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///序号
    TThostFtdcSequenceNoType SequenceNo;
    ///条件单编号
    TTKSCONDITIONALORDERIDType ConditionalOrderID;
    ///条件单状态
    TTKSCONDITIONALORDERSTATUSTYPE ConditionalOrderStatus;
    ///备注
    TThostFtdcMemoType Memo;
    ///本地报单编号

```

```

TThostFtdcOrderLocalIDType OrderLocalID;
///交易所代码

TThostFtdcExchangeIDType ExchangeID;
///合约代码

TThostFtdcInstrumentIDType InstrumentID;
///报单状态

TThostFtdcOrderStatusType OrderStatus;
///开平标志

TThostFtdcOffsetFlagType CombOffsetFlag;
///投机套保标志

TThostFtdcHedgeFlagType CombHedgeFlag;
///买卖方向

TThostFtdcDirectionType Direction;
///价格

TThostFtdcPriceType LimitPrice;
///数量

TThostFtdcVolumeType VolumeTotalOriginal;
///交易日

TThostFtdcTradeDateType TradingDay;
///撤销用户

TThostFtdcUserIDType CancelUserID;
///撤销时间

TThostFtdcTimeType CancelTime;
///客户代码

TThostFtdcClientIDType ClientID;
///业务单元

TThostFtdcBusinessUnitType BusinessUnit;
///报单编号

TThostFtdcOrderSysIDType OrderSysID;
///今成交数量

TThostFtdcVolumeType VolumeTraded;
///剩余数量

```



```

    TThostFtdcVolumeType VolumeTotal;
    ///委托时间
    TThostFtdcTimeType InsertTime;
    ///激活时间
    TThostFtdcTimeType ActiveTime;
    ///成交价格
    TThostFtdcPriceType TradePrice;
    ///货币代码
    TThostFtdcCurrencyIDType CurrencyID;
};

```

4.7.13. OnRtnPLStatus 方法

盈损单状态通知。

函数模型：

```
void OnRtnPLStatus(CKSPLStatusField *pPLStatus)
```

参数：

pPLStatus：指向盈损单状态结构的指针。

盈损单状态结构如下所示：

```

struct CKSPLStatusField
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///操作员
    TThostFtdcUserIDType UserID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///序号
    TThostFtdcSequenceNoType SequenceNo;
    ///止损止盈单编号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
    ///止损条件单编号
    TTKSConditionalOrderIDType StopLossOrderID;
    ///止盈条件单编号

```

```

TTKSConditionalOrderIDType    TakeProfitOrderID;
///盈损单状态

TTKSConditionalOrderStatusType ProfitAndLossOrderStatus;
///止损价

TThostFtdcPriceType    StopLossPrice;
///止盈价

TThostFtdcPriceType    TakeProfitPrice;
///生成盈损价的方式

TTKSOffsetValueType OffsetValue;
///开仓成交价格

TThostFtdcPriceType OpenTradePrice;
///备注

TThostFtdcMemoType    Memo;
///本地报单编号

TThostFtdcOrderLocalIDType    OrderLocalID;
///交易所代码

TThostFtdcExchangeIDType ExchangeID;
///合约代码

TThostFtdcInstrumentIDType    InstrumentID;
///报单状态

TThostFtdcOrderStatusType OrderStatus;
///开平标志

TThostFtdcOffsetFlagType    CombOffsetFlag;
///投机套保标志

TThostFtdcHedgeFlagType    CombHedgeFlag;
///买卖方向

TThostFtdcDirectionType    Direction;
///价格

TThostFtdcPriceType    LimitPrice;
///数量

TThostFtdcVolumeType VolumeTotalOriginal;
///交易日

```

```

TThostFtdcTradeDateType   TradingDay;
///撤销用户
TThostFtdcUserIDType   CancelUserID;
///撤销时间
TThostFtdcTimeType   CancelTime;
///客户代码
TThostFtdcClientIDType   ClientID;
///业务单元
TThostFtdcBusinessUnitType   BusinessUnit;
///报单编号
TThostFtdcOrderSysIDType   OrderSysID;
///今成交数量
TThostFtdcVolumeType   VolumeTraded;
///剩余数量
TThostFtdcVolumeType   VolumeTotal;
///委托时间
TThostFtdcTimeType   InsertTime;
///激活时间
TThostFtdcTimeType   ActiveTime;
///成交价格
TThostFtdcPriceType   TradePrice;
///货币代码
TThostFtdcCurrencyIDType   CurrencyID;
};

```

4.8. CKSCosApi 接口

4.8.1. ReqInitInsertConditionalOrder 方法

条件单录入请求

函数模型:

```

Int ReqInitInsertConditionalOrder(
    CKSConditionalOrderInitInsert *pConditionalOrderInitInsert,
    int nRequestID)

```

参数:

pConditionalOrderInitInsert: 指向条件单录入结构的指针

条件单录入结构如下:

```
struct CKSConditionalOrderInitInsert
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///客户代码
    TThostFtdcClientIDType ClientID;
    ///买卖方向
    TThostFtdcDirectionType Direction;
    ///开平标志
    TThostFtdcOffsetFlagType CombOffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType CombHedgeFlag;
    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///价格
    TThostFtdcPriceType LimitPrice;
    ///价格类别
    TTKSOrderPriceTypeType OrderPriceType;
    ///条件类型
    TTKSConditionalTypeType ConditionalType;
    ///条件价
    TThostFtdcPriceType ConditionalPrice;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
```

```

    ///触发次数
    TThostFtdcVolumeType TriggeredTimes;
    ///条件单类型
    TTKSConditionalOrderType OrderType;
    ///有效时间
    TThostFtdcTimeType ActiveTime;
    ///失效时间
    TThostFtdcTimeType InActiveTime;
    ///货币代码
    TThostFtdcCurrencyIDType CurrencyID;
};

```

4.8.2. ReqQueryConditionalOrder 方法

条件单查询请求。

函数模型：

```

int ReqQueryConditionalOrder(
    CKSConditionalOrderQuery *pConditionalOrderQuery,
    int nRequestID)

```

参数：

pConditionalOrderQuery：指向条件单查询结构的指针

条件单查询结构如下：

```

struct CKSConditionalOrderQuery
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
};

```

4.8.3. ReqModifyConditionalOrder 方法

条件单修改请求。

函数模型：

```
int ReqModifyConditionalOrder(
    CKSConditionalOrderModify *pConditionalOrderModify,
    int nRequestID)
```

参数:

pConditionalOrderModify: 指向修改条件单结构的指针

修改条件单结构如下:

```
struct CKSConditionalOrderModify
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///客户代码
    TThostFtdcClientIDType ClientID;
    ///买卖方向
    TThostFtdcDirectionType Direction;
    ///开平标志
    TThostFtdcOffsetFlagType CombOffsetFlag;
    ///投机套保标志
    TThostFtdcHedgeFlagType CombHedgeFlag;
    ///数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///价格
    TThostFtdcPriceType LimitPrice;
    ///价格类别
    TTKSOrderPriceTypeType OrderPriceType;
    ///条件类型
    TTKSConditionalTypeType ConditionalType;
```

```

///条件价
TThostFtdcPriceType ConditionalPrice;
///条件单编号
TTKSConditionalOrderIDType ConditionalOrderID;
///触发次数
TThostFtdcVolumeType TriggeredTimes;
///条件单类型
TTKSConditionalOrderType OrderType;
///有效时间
TThostFtdcTimeType ActiveTime;
///失效时间
TThostFtdcTimeType InActiveTime;
///货币代码
TThostFtdcCurrencyIDType CurrencyID;
};

```

4.8.4. ReqRemoveConditionalOrder 方法

条件单删除请求。

函数模型：

```

int ReqRemoveConditionalOrder(
    CKSConditionalOrderRemove *pConditionalOrderRemove,
    int nRequestID)

```

参数：

pConditionalOrderRemove：指向条件单删除结构的指针

条件单删除结构如下：

```

struct CKSConditionalOrderRemove
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
}

```

```
};
```

4.8.5. ReqStateAlterConditionalOrder 方法

条件单暂停或激活请求。

函数模型：

```
int ReqStateAlterConditionalOrder(
    CKSConditionalOrderStateAlter *pConditionalOrderStateAlter,
    int nRequestID)
```

参数：

pConditionalOrderStateAlter： 指向条件单暂停或激活结构的指针

条件单暂停或激活结构如下：

```
struct CKSConditionalOrderStateAlter
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
    ///暂停或激活操作标志
    TTKSConditionalOrderStateAlterType ConditionalOrderStateAlter;
};
```

4.8.6. ReqSelectConditionalOrder 方法

选择条件单请求

函数模型：

```
int ReqSelectConditionalOrder(
    CKSConditionalOrderSelect *pConditionalOrderSelect,
    int nRequestID)
```

参数：

pConditionalOrderSelect： 指向条件单选择结构的指针。

条件单选择结构如下：

```
struct CKSConditionalOrderSelect
{
```



```

    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///条件单编号
    TTKSConditionalOrderIDType ConditionalOrderID;
    ///选择结果
    TTKSConditionalOrderSelectResultType SelectResult;
};

```

4.8.7. ReqInsertProfitAndLossOrder 方法

止损止盈单录入请求

函数模型：

```

int ReqInsertProfitAndLossOrder(
    CKSProfitAndLossOrderInsert *pProfitAndLossOrderInsert,
    int nRequestID)

```

参数：

pProfitAndLossOrderInsert: 指向止损止盈单录入结构的指针。

止损止盈单录入结构如下：

```

struct CKSProfitAndLossOrderInsert
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType OrderLocalID;
    ///止损价
    TThostFtdcPriceType StopLossPrice;
    ///止盈价
    TThostFtdcPriceType TakeProfitPrice;
    ///平仓方式
    TTKSCloseModeType CloseMode;
    //平仓反向加减价位数（平仓方式=1或时有效）

```

```

    TThostFtdcPriceType FiguresPrice;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///业务单元
    TThostFtdcBusinessUnitType    BusinessUnit;
    ///生成盈损价的方式
    TTKSOffsetValueType OffsetValue;
    ///条件单触发价格类型
    TTKSSpringTypeType SpringType;
    ///浮动止损价
    TThostFtdcPriceType    FloatLimitPrice;
    ///止损单触发次数
    TThostFtdcVolumeType TriggeredTimes;
    ///货币代码
    TThostFtdcCurrencyIDType    CurrencyID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///止损止盈标志
    TKSProfitAndLossFlagType ProfitAndLossFlag;
};

```

4.8.8. ReqModifyProfitAndLossOrder 方法

止损止盈单修改请求。

函数模型：

```

int ReqModifyProfitAndLossOrder(
    CKSProfitAndLossOrderModify *pProfitAndLossOrderModify,
    int nRequestID)

```

参数：

pProfitAndLossOrderModify：指向止损止盈单修改结构的指针。

止损止盈单修改结构如下：

```

struct CKSProfitAndLossOrderModify
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;

```

```

    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
    ///止损价
    TThostFtdcPriceType StopLossPrice;
    ///止盈价
    TThostFtdcPriceType TakeProfitPrice;
    ///平仓方式
    TTKSCloseModeType CloseMode;
    //平仓反向价位数（平仓方式=1 或时有效）
    TThostFtdcPriceType FiguresPrice;
    ///生成盈损价的方式
    TTKSOffsetValueType OffsetValue;
    ///条件单触发价格类型
    TTKSSpringType SpringType;
    ///浮动止损价
    TThostFtdcPriceType FloatLimitPrice;
    ///触发次数
    TThostFtdcVolumeType TriggeredTimes;
    ///止损止盈标志
    TTKSProfitAndLossFlagType ProfitAndLossFlag;
};

```

4.8.9. ReqRemoveProfitAndLossOrder 方法

止损止盈单删除请求。

函数模型：

```

int ReqRemoveProfitAndLossOrder(
    CKSProfitAndLossOrderRemove *pProfitAndLossOrderRemove,
    int nRequestID)

```

参数：

pProfitAndLossOrderRemove: 指向止损止盈单删除结构的指针

止损止盈单删除结构如下所示：

```

struct CKSProfitAndLossOrderRemove
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
    ///本地报单编号
    TThostFtdcOrderLocalIDType    OrderLocalID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///业务单元
    TThostFtdcBusinessUnitType    BusinessUnit;
};

```

4.8.10. ReqQueryProfitAndLossOrder 方法

止损止盈单查询请求。

函数模型：

```

int ReqQueryProfitAndLossOrder(
    CKSProfitAndLossOrderQuery *pProfitAndLossOrderQuery,
    int nRequestID)

```

参数：

pProfitAndLossOrderQuery： 指向止损止盈单查询结构的指针。

止损止盈单查询结构如下所述：

```

struct CKSProfitAndLossOrderQuery
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///止损止盈单号
    TTKSProfitAndLossOrderIDType ProfitAndLossOrderID;
    ///本地报单编号

```

```

TThostFtdcOrderLocalIDType    OrderLocalID;
///交易所代码

TThostFtdcExchangeIDType ExchangeID;
///业务单元

TThostFtdcBusinessUnitType    BusinessUnit;

};

```

4.9. CKSOptionSpi 接口

CKSOptionSpi 实现了条件期权单事件通知接口。使用 Kingstar 期权单时，用户必需派生 CKSOptionSpi 接口，编写事件处理方法来处理感兴趣的事件。

4.9.1. OnRspQryIndexPrice 方法

查询现货指数响应

函数模型：

```

void OnRspQryIndexPrice (
    CKSIndexPriceField *pIndexPrice,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

pIndexPrice：指向查询现货指数响应结构的指针。

查询现货指数响应结构如下所示：

```

struct CKSIndexPriceField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///指数现货收盘价
    TThostFtdcPriceType    ClosePrice;
    ///安全系数
    TThostFtdcRatioType    GuarantRatio;
};

```

4.9.2. OnRspQryOptionInstrGuard 方法

查询股指现货指数响应

函数模型:

```
void OnRspQryOptionInstrGuard (
    CKSOptionInstrGuardField *pOptionInstrGuard,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pOptionInstrGuard: 指向查询股指现货指数响应结构的指针。

查询股指现货指数响应结构如下所示:

```
struct CKSOptionInstrGuardField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///最低保障系数分子
    TThostFtdcVolumeType MiniNumerator;
    ///最低保障系数分母
    TThostFtdcVolumeType MiniDenominator ;
    ///期权合约最小保证金
    TThostFtdcMoneyType MiniMargin;
};
```

4.9.3. OnRspQryExecOrderVolume 方法

查询宣告数量请求响应

函数模型:

```
void OnRspQryExecOrderVolume (
    CKSExecOrderVolumeField *pExecOrderVolume,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
```

```
bool bIsLast)
```

参数:

pExecOrderVolume: 指向查询宣告数量请求响应结构的指针。

查询宣告数量请求响应结构如下所示:

```
struct CKSExecOrderVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType   InstrumentID;
    ///产品代码
    TThostFtdcInstrumentIDType   ProductID;
    ///投机套保标志
    TThostFtdcHedgeFlagType   HedgeFlag;
    ///可申请执行量
    TThostFtdcVolumeTypeExecVolume;
    ///可申请放弃量
    TThostFtdcVolumeTypeActionVolume;
    ///已申请执行量
    TThostFtdcVolumeTypeExecedVolume;
    ///已申请放弃量
    TThostFtdcVolumeTypeActionedVolume;
    ///买卖标志
    TThostFtdcDirectionType   Direction;
};
```

4.9.4. OnRspQryStockOptionAssignment 方法

查询个股行权指派信息请求响应

函数模型:

```
void OnRspQryStockOptionAssignment (
    CKSStockOptionAssignmentField *pStockOptionAssignment,
    CThostFtdcRspInfoField *pRspInfo,
```

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

参数:

pStockOptionAssignment: 指向查询个股行权指派信息请求响应结构的指针。

查询个股行权指派信息请求响应结构如下所示:

```
struct CKSStockOptionAssignmentField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///买卖
    TThostFtdcDirectionType   Direction;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///合约名称
    TThostFtdcInstrumentNameType InstrumentName;
    ///产品代码
    TThostFtdcInstrumentIDType  ProductID;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///上日持仓
    TThostFtdcVolumeType YdPosition;
    ///行权指派合约数量
    TThostFtdcVolumeType AssInsVo;
    ///行权标的证券数量
    TThostFtdcVolumeType AssProVol;
    ///行权指派应付金额
    TThostFtdcMoneyType   FeePay;
    ///期权C/P标志
    TThostFtdcOptionsTypeType OptionsType;
```



```

    ///行权交收日
    TThostFtdcDateType DeliveryDay;
    ///个股合约标识码
    TThostFtdcInstrumentIDType StockID;
};

```

4.9.5. OnRspQryInvestorTradeLevel 方法

查询客户交易级别响应

函数模型:

```

void OnRspQryInvestorTradeLevel (
    CKSInvestorTradeLevelField *pInvestorTradeLevel,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pInvestorTradeLevel: 指向查询客户交易级别响应结构的指针。

查询客户交易级别响应结构如下所示:

```

struct CKSInvestorTradeLevelField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///交易级别
    TKSTradeLevelType TradeLevel;
};

```

4.9.6. OnRspQryPurchaseLimitAmt 方法

查询个股限购额度响应

函数模型:

```

void OnRspQryPurchaseLimitAmt (

```

```

    CKSPurchaseLimitAmtField *pPurchaseLimitAmt,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pPurchaseLimitAmt: 指向查询个股限购额度响应结构的指针。

查询个股限购额度响应结构如下所示:

```

struct CKSPurchaseLimitAmtField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///限购额度
    TThostFtdcMoneyType PurLimitAmt;
};

```

4.9.7. OnRspQryPositionLimitVol 方法

查询个股限仓额度响应

函数模型:

```

void OnRspQryPositionLimitVol (
    CKSPositionLimitVolField *pPositionLimitVol,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pPositionLimitVol: 指向查询个股限仓额度响应结构的指针。

查询个股限仓额度响应结构如下所示:

```

struct CKSPositionLimitVolField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;

```

```

    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///限仓类型
    TKSLimitFlagType LimitFlag;
    ///持仓方向
    TThostFtdcOptionsTypeType OptionsType;
    ///控制范围
    TKSControlRangeType ControlRange;
    ///限仓额度
    TThostFtdcVolumeType PosiLimitVol;
};

```

4.9.8. OnRspQryPurchaseLimitAmtS 方法

查询结算系统个股限购额度响应。

函数原型：

```

void OnRspQryPurchaseLimitAmtS(
    CKSPurchaseLimitAmtSField *pPurchaseLimitAmt,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pPurchaseLimitAmt: 指向结算系统个股限购额度结构的指针。

结算系统个股限购额度结构：

```

struct CKSPurchaseLimitAmtSField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///客户资产%

```

```

TThostFtdcMoneyType Fund10P;
///日均资产%
TThostFtdcMoneyType AverageFund20P;
///当前限购额度
TThostFtdcMoneyType CurrPurLimitAmt;
///可设最大限购额度
TThostFtdcMoneyType MaxPurLimitAmt;
};

```

4.9.9. OnRspQryPositionLimitVolS 方法

查询结算系统个股限仓额度响应。

函数原型：

```

void OnRspQryPositionLimitVolS(
    CKSPositionLimitVolSField *pPositionLimitVol,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pPositionLimitVol: 指向结算系统个股限仓额度结构的指针。

结算系统个股限仓额度结构：

```

struct CKSPositionLimitVolSField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///控制范围
    TKSControlRangeType ControlRange;
    ///限仓类型
    TKSLimitFlagType LimitFlag;
    ///当前持仓限额
    TThostFtdcVolumeType CurrPosiLimitVol;
    ///最高可调额度
    TThostFtdcVolumeType MaxPosiLimitVol;
};

```

4.9.10. OnRspPurchaseLimitAmtUpdate 方法

个股限购额度变更响应。

函数原型：

```
void OnRspPurchaseLimitAmtUpdate(
    CKSPurchaseLimitAmtField *pPurchaseLimitAmt,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pPurchaseLimitAmt: 指向个股限购额度变更结构的指针。

个股限购额度变更结构：

```
struct CKSPurchaseLimitAmtField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///限购额度
    TThostFtdcMoneyType PurLimitAmt;
};
```

4.9.11. OnRspPositionLimitVolUpdate 方法

个股限仓额度变更响应。

函数原型：

```
void OnRspPositionLimitVolUpdate(
    CKSPositionLimitVolField *pPositionLimitVol,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pPositionLimitVol: 指向个股限仓额度变更结构的指针。

个股限仓额度变更结构：

```
struct CKSPositionLimitVolField
{
    ///经纪公司代码
```

```

TThostFtdcBrokerIDType BrokerID;
///投资者代码
TThostFtdcInvestorIDType InvestorID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///产品代码
TThostFtdcInstrumentIDType ProductID;
///控制范围
TKSControlRangeType ControlRange;
///限仓类型
TKSLimitFlagType LimitFlag;
///持仓限仓
TThostFtdcVolumeType PosiLimitVol;
};

```

4.9.12. OnRspQryHistoryOrder 方法

查询历史报单响应

函数模型:

```

void OnRspQryHistoryOrder (
    CThostFtdcOrderField *pOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pOrder: 指向查询历史报单响应结构的指针。

查询历史报单响应结构如下所示:

```

struct CKSHistoryOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;

```

```
///本地报单编号
TThostFtdcOrderLocalIDType    OrderLocalID;
///申报时间
TThostFtdcTimeType    InsertTime;
///业务单元
TThostFtdcBusinessUnitType    BusinessUnit;
///成交价格
TThostFtdcPriceType    TradePrice;
///成交金额
TThostFtdcMoneyType    TradeAmount;
///成交数量
TThostFtdcVolumeType    VolumeTraded;
///冻结解冻金额
TThostFtdcMoneyType    FrozenAmount;
///币种代码
TThostFtdcCurrencyIDType    CurrencyID;
///序号
TThostFtdcSequenceNoType    SequenceNo;
///买卖方向
TThostFtdcDirectionType    Direction;
///交易所代码
TThostFtdcExchangeIDType    ExchangeID;
///用户端产品信息
TThostFtdcProductInfoType    UserProductInfo;
///请求编号
TThostFtdcRequestIDType    RequestID;
///报单引用
TThostFtdcOrderRefType    OrderRef;
///前置编号
TThostFtdcFrontIDType    FrontID;
///会话编号
TThostFtdcSessionIDType    SessionID;
///委托价格
TThostFtdcPriceType    OrderPrice;
```

```

    ///委托来源
    TThostFtdcOrderSourceType OrderSource;
    ///委托日期
    TThostFtdcDateType    InsertDate;
    ///委托时间
    TThostFtdcTimeType    OrderTime;
    ///委托数量
    TThostFtdcVolumeType VolumeTotalOriginal;
    ///委托状态
    TThostFtdcOrderStatusType OrderStatus;
    ///产品代码
    TThostFtdcInstrumentIDType    ProductID;
    ///产品名称
    TThostFtdcProductNameType    ProductName;
    ///产品类型
    TThostFtdcProductClassTypeProductClass;
    ///业务标志
    TKSSOBusinessFlagType    BusinessFlag;
    ///业务类型名称
    TThostFtdcFunctionNameType    FunctionName;
};

```

4.9.13. OnRspQryHistoryTrade 方法

查询历史成交响应

函数模型:

```

void OnRspQryHistoryTrade (
    CThostFtdcTradeField *pTrade,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pTrade: 指向查询历史成交响应结构的指针。

查询历史成交响应结构如下所示:

```

struct CKSHistoryTradeField

```



```

{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///客户姓名
    TThostFtdcIndividualNameType CustomerName;
    ///本地报单编号
    TThostFtdcOrderLocalIDType OrderLocalID;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///产品名称
    TThostFtdcProductType ProductName;
    ///成交数量
    TThostFtdcVolumeType VolumeTraded;
    ///成交金额
    TThostFtdcMoneyType TradeAmount;
    ///成交日期
    TThostFtdcDateType TradeDate;
    ///成交时间
    TThostFtdcTimeType TradeTime;
    ///业务单元
    TThostFtdcBusinessUnitType BusinessUnit;
    ///手续费
    TThostFtdcMoneyType Commission;
    ///备注
    TThostFtdcMemoType Memo;
    ///报盘时间
    TThostFtdcTimeType TraderOfferTime;
    ///成交价格
    TThostFtdcPriceType TradePrice;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///期权类型
    TThostFtdcOptionsTypeOptionsType;
    ///备兑标志

```

```

TThostFtdcHedgeFlagType HedgeFlag;
///权利仓数量
TThostFtdcVolumeType RoyaltyVolume;
///义务仓数量
TThostFtdcVolumeType ObligationVolume;
///权利仓金额
TThostFtdcMoneyType RoyaltyAmount;
///义务仓金额
TThostFtdcMoneyType ObligationAmount;
///成交编号
TThostFtdcTradeIDType TradeID;
///业务标志
TKSSOBusinessFlagType BusinessFlag;
///业务类型名称
TThostFtdcFunctionNameType FunctionName;
}

```

4.9.14. OnRspQryStockOptionHistoryAssignment 方法

查询个股历史行权指派明细请求响应

函数模型:

```

void OnRspQryStockOptionHistoryAssignment (
    CKSHistoryAssignmentField *pHistoryAssignment,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pHistoryAssignment: 指向查询个股历史行权指派明细请求响应结构的指针。

查询个股历史行权指派明细请求响应结构如下所示:

```

struct CKSHistoryAssignmentField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易日期
    TThostFtdcTradeDateType TradingDay;
    ///产品代码

```

```

    TThostFtdcInstrumentIDType    ProductID;
    ///合约代码

    TThostFtdcInstrumentIDType    InstrumentID;
    ///客户代码

    TThostFtdcClientIDType    ClientID;
    ///标的证券代码

    TThostFtdcInstrumentIDType    StockInstr;
    ///投保标记

    TThostFtdcHedgeFlagType    HedgeFlag;
    ///期权类型

    TThostFtdcOptionsTypeTypeOptionsType;
    ///持仓方向

    TKSSOPosiDirectionType PosiDirection;
    ///执行价

    TThostFtdcPriceType    StrikePrice;
    ///行权指派数量

    TThostFtdcVolumeType ExecVol;
    ///标的证券应收付数量

    TThostFtdcVolumeType IOVol;
    ///应收付金额

    TThostFtdcMoneyType IOAmt;
    ///行权交收日

    TThostFtdcDateType    DelivDate;
};

```

4.9.15. OnRspQryStockOptionDelivDetail 方法

查询个股行权交割明细请求响应

函数模型:

```

void OnRspQryStockOptionDelivDetail (
    CKSSODelivDetailField *pSODelivDetail,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pSODelivDetail: 指向查询个股行权交割明细请求响应结构的指针。

查询个股行权交割明细请求响应结构如下所示:

```
struct CKSSODelivDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///标的证券代码
    TThostFtdcInstrumentIDType StockInstr;
    ///应收/应付/扣券证券数量
    TThostFtdcVolumeType IOVol;
    ///实收付数量
    TThostFtdcVolumeType IOVolInFact;
    ///结算价
    TThostFtdcPriceType   SettlementPrice;
    ///扣券面值/结算金额
    TThostFtdcMoneyType SettlementAmt;
    ///行权交收日
    TThostFtdcDateType   DelivDate;
    ///业务类型名称
    TThostFtdcFunctionNameType   FunctionName;
};
```

4.9.16. OnRspAutoExecOrderAction 方法

查询自动行权执行操作请求响应

函数模型:

```
void OnRspAutoExecOrderAction (
    CKSAutoExecOrderActionField *pAutoExecOrderAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pAutoExecOrderAction: 指向查询自动行权执行操作请求响应结构的指针。

查询自动行权执行操作请求响应结构如下所示:

```
struct CKSAutoExecOrderActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///自动行权阈值(0-不自动行权,非-代表超过阈值会自动行权(20代表%))
    TThostFtdcVolumeType RangeVol;
};
```

4.9.17. OnRspCombActionInsert 方法

申请个股组合录入请求响应。

函数原型:

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

参数:

pInputCombAction: 指向个股组合录入请求响应结构的指针。

个股组合录入请求响应结构:

```
struct CKSInputCombActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///策略代码
    TKSCombStrategyIDType   StrategyID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID1;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID2;
```

```

    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID3;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID4;
    ///组合引用
    TThostFtdcOrderRefType    CombActionRef;
    ///用户代码
    TThostFtdcUserIDType    UserID;
    ///买卖方向
    TThostFtdcDirectionType    Direction1;
    ///买卖方向
    TThostFtdcDirectionType    Direction2;
    ///买卖方向
    TThostFtdcDirectionType    Direction3;
    ///买卖方向
    TThostFtdcDirectionType    Direction4;
    ///强拆标记
    TKSCombActionType    CombActionFlag;
    ///数量
    TThostFtdcVolumeType    Volume;
    ///组合拆分标记
    TThostFtdcCombDirectionType    CombDirection;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///交易所组合编号
    TThostFtdcTradeIDType    ComTradeID;
};

```

4.9.18. OnRspQryInvestorCombinePosition 方法

查询个股组合持仓明细应答。

函数原型：

```

void OnRspQryInvestorCombinePosition(
    CKSInvestorPositionCombineDetailField *pInvestorPositionCombineDetail,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInvestorPositionCombineDetail: 指向个股组合持仓明细结构的指针。

个股组合持仓明细结构:

```
struct CKSInvestorPositionCombineDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易编码
    TThostFtdcClientIDType    ClientID;
    ///组合编号
    TThostFtdcTradeIDType     ComTradeID;
    ///策略代码
    TKSCombStrategyIDType     StrategyID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID1;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID2;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID3;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID4;
    ///占用保证金
    TThostFtdcMoneyType Margin;
    ///距离自动拆分日天数
    TThostFtdcVolumeTypeCombActionVolume;
    ///投保标记
    TThostFtdcHedgeFlagType  HedgeFlag;
    ///持仓量
    TThostFtdcVolumeTypeTotalAmt;
    ///买卖方向
    TThostFtdcDirectionType  Direction1;
    ///买卖方向
    TThostFtdcDirectionType  Direction2;
    ///买卖方向
    TThostFtdcDirectionType  Direction3;
    ///买卖方向
    TThostFtdcDirectionType  Direction4;
```

```

    ///组合买卖方向
    TThostFtdcDirectionType    CombDirection;
};

```

4.9.19. OnRspQryCombActionVolume 方法

个股可组合可拆分手数请求响应。

函数原型:

```

void OnRspQryCombActionVolume(
    CKSCombActionVolumeField *pCombActionVolume,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pCombActionVolume: 指向个股可组合可拆分手数结构的地址。

个股可组合可拆分手数结构:

```

struct CKSCombActionVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///最大可组合数量
    TThostFtdcVolumeTypeMaxCombVolume;
    ///最大可拆分数
    TThostFtdcVolumeTypeMaxActionVolume;
};

```

4.9.20. OnRspQuoteUpdate 方法

上证做市商报价单修改请求响应。

函数原型:

```

void OnRspQuoteUpdate(
    CKSInputQuoteUpdateField *pInputQuoteUpdate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pInputQuoteUpdate: 指向上证做市商报价单修改请求结构的指针。

上证做市商报单修改请求结构:

```
struct CKSInputQuoteUpdateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///报价引用
    TThostFtdcOrderRefType   QuoteRef;
    ///请求编号
    TThostFtdcRequestIDType  RequestID;
    ///前置编号
    TThostFtdcFrontIDTypeFrontID;
    ///会话编号
    TThostFtdcSessionIDType  SessionID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///报价编号
    TThostFtdcOrderSysIDType QuoteSysID;
    ///用户代码
    TThostFtdcUserIDType   UserID;
    ///合约代码
    TThostFtdcInstrumentIDType   InstrumentID;
    ///买价格
    TThostFtdcPriceType   BidPrice;
    ///卖价格
    TThostFtdcPriceType   AskPrice;
    ///报价修改引用
    TThostFtdcOrderRefType   QuoteUpdateRef;
    ///衍生卖报单引用
    TThostFtdcOrderRefType   AskOrderUpdateRef;
    ///衍生买报单引用
    TThostFtdcOrderRefType   BidOrderUpdateRef;
    ///应价编号
    TThostFtdcOrderSysIDType ForQuoteSysID;
};
```

4.9.21. OnRtnCombAction 方法

个股组合拆分委托通知。

函数原型:

```
void OnRtnCombAction(CKSCombActionField *pCombAction);
```

参数:

pCombAction: 指向个股组合拆分委托结构的指针。

个股组合拆分委托结构:

```
struct CKSCombActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///完整合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///策略代码
    TKSCombStrategyIDType     StrategyID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID1;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID2;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID3;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID4;
    ///组合引用
    TThostFtdcOrderRefType    CombActionRef;
    ///用户代码
    TThostFtdcUserIDType      UserID;
    ///买卖方向
    TThostFtdcDirectionType   Direction1;
    ///买卖方向
    TThostFtdcDirectionType   Direction2;
    ///买卖方向
    TThostFtdcDirectionType   Direction3;
    ///买卖方向
    TThostFtdcDirectionType   Direction4;
    ///强拆标记
    TKSCombActionType         CombActionFlag;
    ///数量
    TThostFtdcVolumeType      Volume;
    ///组合拆分标记
```

```

TThostFtdcCombDirectionType CombDirection;
///投机套保标志
TThostFtdcHedgeFlagType HedgeFlag;
///本地申请组合编号
TThostFtdcOrderLocalIDType ActionLocalID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///会员代码
TThostFtdcParticipantIDType ParticipantID;
///客户代码
TThostFtdcClientIDType ClientID;
///合约在交易所的代码
TThostFtdcExchangeInstIDType ExchangeInstID;
///交易所交易员代码
TThostFtdcTraderIDType TraderID;
///安装编号
TThostFtdcInstallIDType InstallID;
///组合状态
TThostFtdcOrderActionStatusType ActionStatus;
///报单提示序号
TThostFtdcSequenceNoType NotifySequence;
///交易日
TThostFtdcDateType TradingDay;
///结算编号
TThostFtdcSettlementIDType SettlementID;
///序号
TThostFtdcSequenceNoType SequenceNo;
///前置编号
TThostFtdcFrontIDType FrontID;
///会话编号
TThostFtdcSessionIDType SessionID;
///用户端产品信息
TThostFtdcProductInfoType UserProductInfo;
///状态信息
TThostFtdcErrorMsgType StatusMsg;
///交易所组合编号
TThostFtdcTradeIDType ComTradeID;
///报单来源
TThostFtdcOrderSourceType OrderSource;
};

```

4.10. CKSOptionApi 接口

4.10.1. ReqQryIndexPrice 方法

查询股指现货指数请求

函数模型:

```
int ReqQryIndexPrice (
    CKSQryIndexPriceField *pIndexPrice,
    int nRequestID)
```

参数:

pIndexPrice: 指向查询股指现货指数请求结构的指针

查询股指现货指数请求结构如下:

```
struct CKSQryIndexPriceField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType   InstrumentID;
};
```

4.10.2. ReqQryOptionInstrGuard 方法

查询期权合约保障系数请求

函数模型:

```
int ReqQryOptionInstrGuard (
    CKSQryOptionInstrGuardField *pQryOptionInstrGuard,
    int nRequestID)
```

参数:

pQryOptionInstrGuard: 指向查询期权合约保障系数请求结构的指针

查询期权合约保障系数请求结构如下:

```
struct CKSQryOptionInstrGuardField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
```

```

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
};

```

4.10.3. ReqQryExecOrderVolume 方法

查询宣告数量请求

函数模型:

```

int ReqQryExecOrderVolume (
    CKSQryExecOrderVolumeField *pQryExecOrderVolume,
    int nRequestID)

```

参数:

pQryExecOrderVolume: 指向查询宣告数量请求结构的指针

查询宣告数量请求结构如下:

```

struct CKSQryExecOrderVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///买卖标志
    TThostFtdcDirectionType Direction;
};

```

4.10.4. ReqQryStockOptionAssignment 方法

查询个股行权指派信息

函数模型:

```
int ReqQryStockOptionAssignment (
    CKSQryStockOptionAssignmentField *pQryStockOptionAssignment,
    int nRequestID)
```

参数:

pQryStockOptionInsMarginRate: 指向查询个股行权指派信息结构的指针

查询个股行权指派信息结构如下:

```
struct CKSQryStockOptionAssignmentField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///买卖
    TThostFtdcDirectionType   Direction;
};
```

4.10.5. ReqQryInvestorTradeLevel 方法

查询客户交易级别

函数模型:

```
int ReqQryInvestorTradeLevel (
    CKSQryInvestorTradeLevelField *pQryInvestorTradeLevel,
    int nRequestID)
```

参数:

pQryInvestorTradeLevel: 指向查询客户交易级别结构的指针

查询客户交易级别结构如下:

```
struct CKSQryInvestorTradeLevelField
{
```

```

    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
};

```

4.10.6. ReqQryPurchaseLimitAmtT 方法

查询交易系统个股限购额度

函数模型:

```

int ReqQryPurchaseLimitAmtT (
    CKSQryOptLimitField *pQryPurchaseLimitAmt ,
    int nRequestID)

```

参数:

pQryPurchaseLimitAmt: 指向查询交易系统个股限购额度结构的指针

查询交易系统个股限购额度结构如下:

```

struct CKSQryOptLimitField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.10.7. ReqQryPositionLimitVolT 方法

查询交易系统个股限仓额度

函数模型:

```

int ReqQryPositionLimitVolT (
    CKSQryOptLimitField *pQryPositionLimitVol,
    int nRequestID)

```

参数:

pQryPositionLimitVol: 指向查询交易系统个股限仓额度结构的指针

查询交易系统个股限仓额度结构如下:

```
struct CKSQryOptLimitField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.10.8. ReqQryPurchaseLimitAmtS 方法

查询结算系统个股限购额度。

函数原型：

```
int ReqQryPurchaseLimitAmtS(
    CKSQryOptLimitField *pQryPurchaseLimitAmt,
    int nRequestID);
```

参数：

pQryPurchaseLimitAmt: 指向查询结算系统个股限购额度结构的指针。

查询结算系统个股限购额度结构：

```
struct CKSQryOptLimitField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.10.9. ReqQryPositionLimitVolS 方法

查询结算系统个股限仓额度。

函数原型：

```
int ReqQryPositionLimitVolS(
    CKSQryOptLimitField *pQryPositionLimitVol,
    int nRequestID);
```

参数：

pQryPositionLimitVol: 指向查询结算系统个股限仓额度结构的指针。

查询结算系统个股限仓额度结构：

```
struct CKSQryOptLimitField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.10.10. ReqPurchaseLimitAmtUpdate 方法

个股限购额度变更。

函数原型：

```
int ReqPurchaseLimitAmtUpdate(
    CKSPurchaseLimitAmtField *pPurchaseLimitAmt,
    int nRequestID);
```

参数：

pPurchaseLimitAmt: 指向个股限购额度变更结构的指针。

个股限购额度变更结构：

```
struct CKSPurchaseLimitAmtField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///限购额度
    TThostFtdcMoneyType PurLimitAmt;
};
```

4.10.11. ReqPositionLimitVolUpdate 方法

个股限仓额度变更。

函数原型：

```
int ReqPositionLimitVolUpdate(
    CKSPositionLimitVolField *pPositionLimitVol,
    int nRequestID);
```

参数:

pPositionLimitVol: 指向个股限仓额度变更结构的指针。

个股限仓额度变更结构:

```
struct CKSPositionLimitVolField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///控制范围
    TKSControlRangeType ControlRange;
    ///限仓类型
    TKSLimitFlagType LimitFlag;
    ///持仓限仓
    TThostFtdcVolumeType PosiLimitVol;
};
```

4.10.12. ReqQryHistoryOrder 方法

查询历史报单

函数模型:

```
int ReqQryHistoryOrder (
    CKSQryHistoryOrderField *pQryHistoryOrder,
    int nRequestID)
```

参数:

pQryHistoryOrder: 指向查询历史报单结构的指针

查询历史报单结构如下:

```
struct CKSQryHistoryOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
```

```

    TThostFtdcInstrumentIDType    InstrumentID;
    ///本地报单编号

    TThostFtdcOrderLocalIDType    OrderLocalID;
    ///品种代码

    TThostFtdcInstrumentIDType    ProductID;
    ///开始日期

    TThostFtdcDateType    OrderDataStart;
    ///结束日期

    TThostFtdcDateType    OrderDataEnd;
};

```

4.10.13. ReqQryHistoryTrade 方法

查询历史成交

函数模型:

```

int ReqQryHistoryTrade (
    CKSQryHistoryTradeField *pQryHistoryTrade,
    int nRequestID)

```

参数:

pQryHistoryTrade: 指向查询历史成交结构的指针

查询历史成交结构如下:

```

struct CKSQryHistoryTradeField
{
    ///经纪公司代码

    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType    InvestorID;
    ///品种代码

    TThostFtdcInstrumentIDType    ProductID;
    ///开始日期

    TThostFtdcDateType    TradeDataStart;
    ///结束日期

    TThostFtdcDateType    TradeDataEnd;
};

```

4.10.14. ReqQryStockOptionHistoryAssignment 方法

查询个股历史行权指派明细请求

函数模型:

```
int ReqQryStockOptionHistoryAssignment (
    CKSQryHistoryAssignmentField *pQryHistoryAssignment,
    int nRequestID)
```

参数:

pQryHistoryAssignment: 指向查询个股历史行权指派明细请求结构的指针

查询个股历史行权指派明细请求结构如下:

```
struct CKSQryHistoryAssignmentField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投保标记
    TThostFtdcHedgeFlagType HedgeFlag;
    ///期权类型
    TThostFtdcOptionsTypeType OptionsType;
    ///交割月
    TThostFtdcDateType DeliveryMonth;
    ///持仓方向
    TKSSOPosiDirectionType PosiDirection;
};
```

4.10.15. ReqQryStockOptionDelivDetail 方法

查询个股行权交割明细请求

函数模型:

```
int ReqQryStockOptionDelivDetail (
    CKSQrySODelivDetailField *pQrySODelivDetail,
    int nRequestID)
```

参数:

pQrySODelivDetail: 指向查询个股行权交割明细请求结构的指针

查询个股行权交割明细请求结构如下:

```
struct CKSQrySODelivDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///产品代码
    TThostFtdcInstrumentIDType ProductID;
    ///交割月
    TThostFtdcDateType   DeliveryMonth;
    ///个股交收查询类型
    TKSSODelivModeType DelivMode;
};
```

4.10.16. ReqAutoExecOrderAction 方法

查询自动行权执行操作请求

函数模型:

```
int ReqAutoExecOrderAction (
    CKSAutoExecOrderActionField *pAutoExecOrderAction,
    int nRequestID)
```

参数:

pAutoExecOrderAction: 指向查询自动行权执行操作请求结构的指针

查询自动行权执行操作请求结构如下:

```
struct CKSAutoExecOrderActionField
{
    ///经纪公司代码
```

```

    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码

    TThostFtdcInvestorIDType  InvestorID;
    ///自动行权阈值(0-不自动行权,非-代表超过阈值会自动行权(20代表%))

    TThostFtdcVolumeType RangeVol;

};

```

4.10.17. ReqCombActionInsert 方法

个股组合拆分委托请求。

函数原型：

```

int ReqCombActionInsert(
    CKSInputCombActionField *pInputCombAction,
    int nRequestID);

```

参数：

pInputCombAction: 指向个股组合拆分委托请求结构的指针。

个股组合拆分委托请求结构：

```

struct CKSInputCombActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///策略代码
    TKSCombStrategyIDType     StrategyID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID1;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID2;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID3;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID4;
    ///组合引用
    TThostFtdcOrderRefType     CombActionRef;
    ///用户代码
    TThostFtdcUserIDType       UserID;
    ///买卖方向
    TThostFtdcDirectionType    Direction1;

```

```

    ///买卖方向
    TThostFtdcDirectionType    Direction2;
    ///买卖方向
    TThostFtdcDirectionType    Direction3;
    ///买卖方向
    TThostFtdcDirectionType    Direction4;
    ///强拆标记
    TKSCombActionType    CombActionFlag;
    ///数量
    TThostFtdcVolumeType    Volume;
    ///组合拆分标记
    TThostFtdcCombDirectionType    CombDirection;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///交易所组合编号
    TThostFtdcTradeIDType    ComTradeID;
};

```

4.10.18. ReqQryInvestorCombinePosition 方法

查询个股组合持仓明细。

函数原型：

```

int ReqQryInvestorCombinePosition(
    CKSQryInvestorPositionCombineDetailField *pQryInvestorCombinePosition,
    int nRequestID);

```

参数：

pQryInvestorCombinePosition： 指向查询个股组合持仓明细结构的指针。

查询个股组合持仓明细结构：

```

struct CKSQryInvestorPositionCombineDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///产品代码

```

```

        TThostFtdcInstrumentIDType    ProductID;
        ///交易所组合编号
        TThostFtdcTradeIDType    ComTradeID;
    };

```

4.10.19. ReqQryCombActionVolume 方法

查询个股可组合可拆分手数请求。

函数原型：

```

int ReqQryCombActionVolume(
    CKSQryCombActionVolumeField *pQryCombActionVolume,
    int nRequestID);

```

参数：

pQryCombActionVolume: 指向查询个股可组合可拆分手数请求结构的指针

查询个股可组合可拆分手数请求结构：

```

struct CKSQryCombActionVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///策略代码
    TKSCombStrategyIDType    StrategyID;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID1;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID2;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID3;
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID4;
    ///投保标记
    TThostFtdcHedgeFlagType    HedgeFlag;

```



```

    ///组合拆分标记
    TThostFtdcCombDirectionType   CombDirection;
    ///买卖方向
    TThostFtdcDirectionType       Direction1;
    ///买卖方向
    TThostFtdcDirectionType       Direction2;
    ///买卖方向
    TThostFtdcDirectionType       Direction3;
    ///买卖方向
    TThostFtdcDirectionType       Direction4;
    ///强拆标记
    TKSCombActionType             CombActionFlag;
    ///组合编号
    TThostFtdcTradeIDType         ComTradeID;
};

```

4.10.20. ReqQuoteUpdate 方法

上证做市商报价单修改请求。

函数原型：

```

int ReqQuoteUpdate(
    CKSInputQuoteUpdateField *pInputQuoteUpdate,
    int nRequestID);

```

参数：

pInputQuoteUpdate: 指向上证做市商报价单修改请求结构的指针。

上证做市商报价单修改请求结构：

```

struct CKSInputQuoteUpdateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///报价引用
    TThostFtdcOrderRefType   QuoteRef;
    ///请求编号
    TThostFtdcRequestIDType  RequestID;
    ///前置编号

```

```

TThostFtdcFrontIDType FrontID;
///会话编号
TThostFtdcSessionIDType SessionID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///报价编号
TThostFtdcOrderSysIDType QuoteSysID;
///用户代码
TThostFtdcUserIDType UserID;
///合约代码
TThostFtdcInstrumentIDType InstrumentID;
///买价格
TThostFtdcPriceType BidPrice;
///卖价格
TThostFtdcPriceType AskPrice;
///报价修改引用
TThostFtdcOrderRefType QuoteUpdateRef;
///衍生卖报单引用
TThostFtdcOrderRefType AskOrderUpdateRef;
///衍生买报单引用
TThostFtdcOrderRefType BidOrderUpdateRef;
///应价编号
TThostFtdcOrderSysIDType ForQuoteSysID;
};

```

4.11. CKSVocSpi 接口

CKSVocSpi 实现了用户感兴趣的其他事件通知接口。使用时，用户必需派生 CKSVocSpi 接口，编写事件处理方法来处理感兴趣的事件。

4.11.1. OnRspQryInvestorOpenPosition 方法

请求查询投资者开盘前持仓明细响应。当客户端发出请求请求查询投资者开盘前持仓明细指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

void OnRspQryInvestorOpenPosition(
    CThostFtdcInvestorPositionDetailField *pInvestorPositionDetail,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInvestorPositionDetail : 指向投资者开盘前持仓明细结构的地址。

投资者开盘前持仓明细结构:

```
struct CThostFtdcInvestorPositionDetailField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
    /// 投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    /// 买卖
    TThostFtdcDirectionType Direction;
    /// 开仓日期
    TThostFtdcDateType OpenDate;
    /// 成交编号
    TThostFtdcTradeIDType TradeID;
    /// 数量
    TThostFtdcVolumeType Volume;
    /// 开仓价
    TThostFtdcPriceType OpenPrice;
    /// 交易日
    TThostFtdcDateType TradingDay;
    /// 结算编号
    TThostFtdcSettlementIDType SettlementID;
    /// 成交类型
    TThostFtdcTradeTypeType TradeType;
    /// 组合合约代码
    TThostFtdcInstrumentIDType CombInstrumentID;
    /// 交易所代码
    TThostFtdcExchangeIDType ExchangeID;
```

```

///逐日盯市平仓盈亏
TThostFtdcMoneyType CloseProfitByDate;
///逐笔对冲平仓盈亏
TThostFtdcMoneyType CloseProfitByTrade;
///逐日盯市持仓盈亏
TThostFtdcMoneyType PositionProfitByDate;
///逐笔对冲持仓盈亏
TThostFtdcMoneyType PositionProfitByTrade;
///投资者保证金
TThostFtdcMoneyType Margin;
///交易所保证金
TThostFtdcMoneyType ExchMargin;
///保证金率
TThostFtdcRatioType MarginRateByMoney;
///保证金率(按手数)
TThostFtdcRatioType MarginRateByVolume;
///昨结算价
TThostFtdcPriceType LastSettlementPrice;
///结算价
TThostFtdcPriceType SettlementPrice;
///平仓量
TThostFtdcVolumeType CloseVolume;
///平仓金额
TThostFtdcMoneyType CloseAmount;
};

```

4.11.2. OnRspQryInvestorOpenCombinePosition 方法

请求查询投资者开盘前组合持仓明细响应。

函数原型：

```

void OnRspQryInvestorPositionCombineDetail(
    CThostFtdcInvestorPositionCombineDetailField *pInvestorPositionCombineDetail,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,

```

```
bool bIsLast);
```

参数:

pInvestorPositionCombineDetail: 指向请求查询投资者开盘前组合持仓明细响应结构。

请求查询投资者开盘前组合持仓明细响应结构

```
struct CThostFtdcInvestorPositionCombineDetailField
```

```
{
    ///交易日
    TThostFtdcDateType    TradingDay;
    ///开仓日期
    TThostFtdcDateType    OpenDate;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///结算编号
    TThostFtdcSettlementIDType SettlementID;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///组合编号
    TThostFtdcTradeIDType    ComTradeID;
    ///撮合编号
    TThostFtdcTradeIDType    TradeID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///买卖
    TThostFtdcDirectionType    Direction;
    ///持仓量
    TThostFtdcVolumeType    TotalAmt;
    ///投资者保证金
    TThostFtdcMoneyType    Margin;
```

```

    ///交易所保证金
    TThostFtdcMoneyType  ExchMargin;
    ///保证金率
    TThostFtdcRatioType  MarginRateByMoney;
    ///保证金率(按手数)
    TThostFtdcRatioType  MarginRateByVolume;
    ///单腿编号
    TThostFtdcLegIDType  LegID;
    ///单腿乘数
    TThostFtdcLegMultipleType  LegMultiple;
    ///组合持仓合约编码
    TThostFtdcInstrumentIDType  CombInstrumentID;
    ///成交组号
    TThostFtdcTradeGroupIDType  TradeGroupID;
};

```

4.11.3. OnRspBulkCancelOrder 方法

批量订单撤销响应

函数原型:

```

void  OnRspBulkCancelOrder(
    CThostFtdcBulkCancelOrderField *pBulkCancelOrder,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pBulkCancelOrder: 指向批量报单结构的地址

批量报单结构如下:

```

struct CThostFtdcBulkCancelOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType  BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;

```

```

    ///用户代码
    TThostFtdcUserIDType      UserID;
    ///报单类型
    TThostFtdcOrderTypeType   OrderType;
    ///报单个数
    TThostFtdcVolumeType nCount;
    ///报单集合
    CThostOrderKeyField OrderKey[MAX_ORDER_COUNT];
}

```

4.11.4. OnRspQryCloseStrategy 方法

平仓策略查询响应

函数原型：

```

void OnRspQryCloseStrategy(
    CKSCloseStrategyResultField *pCloseStrategy,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数：

pCloseStrategy: 指向平仓策略查询响应结构的地址

平仓策略查询响应结构如下：

```

struct CKSCloseStrategyResultField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易所名称
    TThostFtdcExchangeNameType ExchangeName;
    ///平仓策略
    TKSCloseStrategyType CloseStrategy;
};

```

4.11.5. OnRspQryCombStrategy 方法

组合策略查询响应

函数原型：

```
void OnRspQryCombStrategy(
    CKSCombStrategyResultField *pCombStrategy,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pCombStrategy: 指向组合策略查询响应结构的地址

组合策略查询响应结构如下:

```
struct CKSCombStrategyResultField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///策略代码
    TKSStrategyIDType StrategyID;
    ///组合类型
    TKSCombTypeType CombType;
    ///组合买卖方向
    TThostFtdcDirectionType CombDirection;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType CombOffsetFlag;
    ///数量
    TThostFtdcVolumeType Volume;
    ///买卖方向一
    TThostFtdcDirectionType Direction1;
    ///开平标志一
    TThostFtdcOffsetFlagType OffsetFlag1;
    ///数量比例一
    TThostFtdcVolRatioType VolRatio1;
    ///计算符号一
    TKSCalcFlagType CalcFlag1;
    ///保证金比例一
    TThostFtdcMoneyRatioType MarginRatio1;
    ///买卖方向二
```



```

    TThostFtdcDirectionType    Direction2;
    ///开平标志二
    TThostFtdcOffsetFlagType    OffsetFlag2;
    ///数量比例二
    TThostFtdcVolRatioType      VolRatio2;
    ///计算符号二
    TKSCalcFlagType CalcFlag2;
    ///保证金比例二
    TThostFtdcMoneyRatioType    MarginRatio2;
    ///买卖方向三
    TThostFtdcDirectionType    Direction3;
    ///开平标志三
    TThostFtdcOffsetFlagType    OffsetFlag3;
    ///数量比例三
    TThostFtdcVolRatioType      VolRatio3;
    ///计算符号三
    TKSCalcFlagType CalcFlag3;
    ///保证金比例三
    TThostFtdcMoneyRatioType    MarginRatio3;
    ///买卖方向四
    TThostFtdcDirectionType    Direction4;
    ///开平标志四
    TThostFtdcOffsetFlagType    OffsetFlag4;
    ///数量比例四
    TThostFtdcVolRatioType      VolRatio4;
    ///计算符号四
    TKSCalcFlagType CalcFlag4;
    ///保证金比例四
    TThostFtdcMoneyRatioType    MarginRatio4;
    ///组合合约代码
    TThostFtdcInstrumentIDType    CombInstrumentID;
};

```

4.11.6. OnRspQryOptionCombStrategy 方法

期权组合策略查询响应

函数原型:

```
void OnRspQryOptionCombStrategy(
    CKSOptionCombStrategyResultField *pOptionCombStrategy,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数:

pOptionCombStrategy: 指向期权组合策略查询响应结构的地址

期权组合策略查询响应结构如下:

```
struct CKSOptionCombStrategyResultField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///策略代码
    TKSSStrategyIDType StrategyID;
    ///组合买卖方向
    TThostFtdcDirectionType CombDirection;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType CombOffsetFlag;
    ///数量
    TThostFtdcVolumeType Volume;
    ///买卖方向一
    TThostFtdcDirectionType Direction1;
    ///期权类型一
    TThostFtdcOptionsTypeType OptionsType1;
    ///执行价类型一
    TThostFtdcStrikeModeType StrikeMode1;
    ///开平标志一
    TThostFtdcOffsetFlagType OffsetFlag1;
    ///数量比例一
    TThostFtdcVolRatioType VolRatio1;
```

```

///保证金比例一
TThostFtdcMoneyRatioType  MarginRatio1;
///计算符号一
TKSCalcFlagType  CalcFlag1;
///买卖方向二
TThostFtdcDirectionType  Direction2;
///期权类型二
TThostFtdcOptionsTypeType  OptionsType2;
///执行价类型二
TThostFtdcStrikeModeType  StrikeMode2;
///开平标志二
TThostFtdcOffsetFlagType  OffsetFlag2;
///数量比例二
TThostFtdcVolRatioType  VolRatio2;
///保证金比例二
TThostFtdcMoneyRatioType  MarginRatio2;
///计算符号二
TKSCalcFlagType  CalcFlag2;
///买卖方向三
TThostFtdcDirectionType  Direction3;
///期权类型三
TThostFtdcOptionsTypeType  OptionsType3;
///执行价类型三
TThostFtdcStrikeModeType  StrikeMode3;
///开平标志三
TThostFtdcOffsetFlagType  OffsetFlag3;
///数量比例三
TThostFtdcVolRatioType  VolRatio3;
///保证金比例三
TThostFtdcMoneyRatioType  MarginRatio3;
///计算符号三
TKSCalcFlagType  CalcFlag3;
///买卖方向四
TThostFtdcDirectionType  Direction4;

```

```

    ///期权类型四
    TThostFtdcOptionsTypeType OptionsType4;
    ///执行价类型四
    TThostFtdcStrikeModeType StrikeMode4;
    ///开平标志四
    TThostFtdcOffsetFlagType OffsetFlag4;
    ///数量比例四
    TThostFtdcVolRatioType VolRatio4;
    ///保证金比例四
    TThostFtdcMoneyRatioType MarginRatio4;
    ///计算符号四
    TKSCalcFlagType CalcFlag4;
};

```

4.11.7. OnRspQryTransferInfo 方法

客户转帐信息查询响应

函数原型:

```

void OnRspQryTransferInfo (
    CKSTransferInfoResultField *pResultField,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pResultField: 指向客户转帐信息查询响应结构的地址

客户转帐信息查询响应结构如下:

```

struct CKSTransferInfoResultField
{
    ///入金开始时间
    TThostFtdcTimeType DepositBegin;
    ///入金结束时间
    TThostFtdcTimeType DepositEnd;
    ///出金开始时间
    TThostFtdcTimeType WithdrawBegin;
    ///出金结束时间

```

```

TThostFtdcTimeType   WithdrawEnd;
///出金次数

TThostFtdcLargeVolumeTypeWithdrawVolume;
///出金限额

TThostFtdcMoneyType   WithdrawLimit;
///出金单笔限额

TThostFtdcMoneyType   WithdrawLimitByTrade;
///可提资金

TThostFtdcMoneyType   Available;
///已出次数

TThostFtdcLargeVolumeTypeWithdrawedVolume;
///已出金额

TThostFtdcMoneyType   WithdrawedAmount;
///币种代码

TThostFtdcCurrencyIDType   CurrencyID;
};

```

4.11.8. OnRspQryKSTradingNotice 方法

交易通知查询响应

函数原型:

```

void OnRspQryKSTradingNotice (
    CKSTradingNoticeField *pTradingNotice,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pTradingNotice: 指向交易通知查询响应结构的地址

交易通知查询响应结构如下:

```

struct CKSTradingNoticeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;

```

```

    ///信息类型
    TKSInfoTypeType InfoType;
    ///信息摘要
    TThostFtdcDigestType InfoDigest;
    ///消息正文
    TThostFtdcContentType FieldContent;
    ///发送日期
    TThostFtdcTimeType SendDate;
    ///发送时间
    TThostFtdcTimeType SendTime;
    ///序列号
    TThostFtdcSequenceNoType SequenceNo;
    ///确认标志
    TKSConfirmFlagType ConfirmFlag;
};

```

4.11.9. OnRtnKSTradingNotice 方法

交易通知

函数原型:

```
void OnRtnKSTradingNotice (CKSTradingNoticeField *pTradingNoticeInfo)
```

参数:

pTradingNoticeInfo: 指向交易通知结构的地址

交易通知结构如下:

```

struct CKSTradingNoticeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///信息类型
    TKSInfoTypeType InfoType;
    ///信息摘要
    TThostFtdcDigestType InfoDigest;
    ///消息正文

```

```

    TThostFtdcContentType FieldContent;
    ///发送日期
    TThostFtdcTimeType    SendDate;
    ///发送时间
    TThostFtdcTimeType    SendTime;
    ///序列号
    TThostFtdcSequenceNoType SequenceNo;
    ///确认标志
    TKSConfirmFlagType ConfirmFlag;
};

```

4.11.10. OnRspQryUserProductUrl 方法

用户端产品资源查询应答

函数原型:

```

void OnRspQryUserProductUrl (
    CKSUserProductUrlField *pUserProductUrl,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pUserProductUrl: 指向用户端产品资源查询结构的地址

用户端产品资源查询结构如下:

```

struct CKSUserProductUrlField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///用户端产品信息
    TThostFtdcProductInfoType    UserProductInfo;
    ///用户端产品最新版本号
    TKSProductVersionType    LastProductVersion;
    ///用户端产品可用最低版本号
    TKSProductVersionType    MinUseProductVersion;
    ///用户端产品可升级最低版本号
    TKSProductVersionType    MinUpdateVersion;
}

```

```

    /// 用户端产品最新版本安装包大小
    TThostFtdcLargeVolumeType  InstallPackSize;
    /// 用户端产品最新版本发布日期
    TThostFtdcTradeDateType  PublishDate;
    /// 用户端产品最新版本说明
    TThostFtdcUserProductMemoType  ProductMemo;
    /// 用户端产品资源 Url 地址
    TThostFtdcFunctionUrlType Url;
}

```

4.11.11. OnRspQryMaxCombActionVolume 方法

最大组合拆分单量查询请求响应。

函数原型：

```

void OnRspQryMaxCombActionVolume(
    CKSMaxCombActionVolumeField *pMaxCombActionVolume,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pMaxCombActionVolume： 指向最大组合拆分单量结构的指针。

最大组合拆分单量结构：

```

struct CKSMaxCombActionVolumeField
{
    /// 经纪公司代码
    TThostFtdcBrokerIDType  BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    /// 最大可组合数量
    TThostFtdcVolumeType  MaxCombVolume;
    /// 最大可拆分数
    TThostFtdcVolumeType  MaxActionVolume;
};

```

4.11.12. OnRspQryKSInstrCommRate 方法

请求查询期权合约手续费响应。

函数原型：

```

void OnRspQryKSInstrCommRate(

```



```

    CKSInstrCommRateField *pInstrCommRate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pInstrCommRate: 指向查询期权合约手续费结构的指针

查询期权合约手续费:

```

struct CKSInstrCommRateField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 投资者范围
    TThostFtdcInvestorRangeType InvestorRange;
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
    /// 开平手续费率
    TThostFtdcRatioType OpenCloseRatioByMoney;
    /// 开平手续费
    TThostFtdcRatioType OpenCloseRatioByVolume;
    /// 平今手续费率
    TThostFtdcRatioType CloseTodayRatioByMoney;
    /// 平今手续费
    TThostFtdcRatioType CloseTodayRatioByVolume;
    /// 执行手续费率
    TThostFtdcRatioType StrikeRatioByMoney;
    /// 执行手续费
    TThostFtdcRatioType StrikeRatioByVolume;
    /// 交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    /// 投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
};

```

4.11.13. OnRspQryKSETFOptionInstrCommRate 方法

请求查询 ETF 期权合约手续费响应。

函数原型:

```
void OnRspQryKSETFOptionInstrCommRate(
```

```

    CKSETFOptionInstrCommRateField * pETFOptionInstrCommRate,
    CThostFtdcRspInfoField * pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pETFOptionInstrCommRate: 指向查询 ETF 期权合约手续费结构的指针。

查询 ETF 期权合约手续费结构:

```

struct CKSETFOptionInstrCommRateField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 投资者范围
    TThostFtdcInvestorRangeType InvestorRange;
    /// 经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    /// 投资者代码
    TThostFtdcInvestorIDType InvestorID;
    /// 买开仓手续费率
    TThostFtdcRatioType BuyOpenRatioByMoney;
    /// 买开仓手续费
    TThostFtdcRatioType BuyOpenRatioByVolume;
    /// 买平仓手续费率
    TThostFtdcRatioType BuyCloseRatioByMoney;
    /// 买平仓手续费
    TThostFtdcRatioType BuyCloseRatioByVolume;
    /// 行权手续费率
    TThostFtdcRatioType StrikeRatioByMoney;
    /// 行权手续费
    TThostFtdcRatioType StrikeRatioByVolume;
    /// 卖开仓手续费率
    TThostFtdcRatioType SellOpenRatioByMoney;
    /// 卖开仓手续费
    TThostFtdcRatioType SellOpenRatioByVolume;
    /// 卖平仓手续费率
    TThostFtdcRatioType SellCloseRatioByMoney;
    /// 卖平仓手续费
    TThostFtdcRatioType SellCloseRatioByVolume;
    /// 备兑开仓手续费率
    TThostFtdcRatioType CoveredOpenRatioByMoney;

```

```

    ///备兑开仓手续费
    TThostFtdcRatioType    CoveredOpenRatioByVolume;
    ///备兑平仓手续费率
    TThostFtdcRatioType    CoveredCloseRatioByMoney;
    ///备兑平仓手续费
    TThostFtdcRatioType    CoveredCloseRatioByVolume;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.11.14. OnRspQryKSIInstrumentMarginRate 方法

请求查询合约保证金率响应。

函数原型：

```

void OnRspQryKSIInstrumentMarginRate(
    CKSIInstrumentMarginRateField *pInstrumentMarginRate,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pInstrumentMarginRate：指向查询合约保证金率结构的指针。

查询合约保证金率结构：

```

struct CKSIInstrumentMarginRateField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投资者范围
    TThostFtdcInvestorRangeType    InvestorRange;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///投机套保标志
    TThostFtdcHedgeFlagType    HedgeFlag;
    ///多头保证金率
    TThostFtdcRatioType    LongMarginRatioByMoney;
    ///多头保证金费
    TThostFtdcMoneyType    LongMarginRatioByVolume;
    ///空头保证金率
    TThostFtdcRatioType    ShortMarginRatioByMoney;

```

```

    ///空头保证金费
    TThostFtdcMoneyType ShortMarginRatioByVolume;
    ///是否相对交易所收取
    TThostFtdcBoolType IsRelative;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.11.15. OnRspWithdrawCreditApply 方法

客户每日出金额度申请响应。

函数原型：

```

void OnRspWithdrawCreditApply(
    CKSInputWithdrawCreditApplyField *pWithdrawCreditApply,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pWithdrawCreditApply：指向客户每日出金额度申请结构的指针。

客户每日出金额度申请结构：

```

struct CKSInputWithdrawCreditApplyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///当日出金额度
    TThostFtdcMoneyType WithdrawCredit;
    ///当日出金额度操作类型
    TKSFOCreditApplyType CreditApplyFlag;
    ///流水号
    TThostFtdcTradeSerialNoType CreditSerial;
};

```

4.11.16. OnRspQryWithdrawCredit 方法

客户每日出金额度查询响应。

函数原型：

```

void OnRspQryWithdrawCredit(
    CKSRspQryWithdrawCreditField *pRspQryWithdrawCredit,

```

```

    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pRspQryWithdrawCredit: 指向客户每日出金额度查询结构的指针。

客户每日出金额度查询结构:

```

struct CKSRspQryWithdrawCreditField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///当日出金额度
    TThostFtdcMoneyType   WithdrawCredit;
    ///当日已出金额度
    TThostFtdcMoneyType   WithdrawedCredit;
    ///最后更新日期
    TThostFtdcDateType   UpdateDate;
    ///设置日期
    TThostFtdcDateType   SetDate;
    ///设置时间
    TThostFtdcTimeType   SetTime;
};

```

4.11.17. OnRspQryWithdrawCreditApply 方法

客户每日出金额度申请查询响应。

函数原型:

```

void OnRspQryWithdrawCreditApply(
    CKSRspQryWithdrawCreditApplyField *pRspQryWithdrawCreditApply,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pRspQryWithdrawCreditApply: 指向客户每日出金额度申请查询结构的指针。

客户每日出金额度申请查询结构:

```

struct CKSRspQryWithdrawCreditApplyField
{
    ///经纪公司代码

```

```

    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///流水号
    TThostFtdcTradeSerialNoType CreditSerial;
    ///申请出金额度
    TThostFtdcMoneyType        WithdrawCredit;
    ///当日出金额度处理状态
    TKSFOCreditStatusType      DealStatus;
    ///申请日期
    TThostFtdcDateType         ApplyDate;
    ///申请时间
    TThostFtdcTimeType         ApplyTime;
    ///审核日期
    TThostFtdcDateType         ConfirmDate;
    ///审核时间
    TThostFtdcTimeType         ConfirmTime;
};

```

4.11.18. OnRspQryKSOOptionMargin 方法

请求查询个股单张保证金响应。

函数原型：

```

void OnRspQryKSOOptionMargin(
    CKSOOptionMarginField *pOptionMargin,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pOptionMargin： 指向查询个股单张保证金请求结构的指针。

查询个股单张保证金请求结构：

```

struct CKSOOptionMarginField
{
    ///合约代码
    TThostFtdcInstrumentIDType    InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType      ExchangeID;
    ///个股单张保证金
    TThostFtdcMoneyType           OptionMargin;
};

```

4.11.19. OnRspQryKSTradingParams 方法

请求查询 KS 交易参数响应。

函数原型：

```
void OnRspQryKSTradingParams(
    CKSTradingParamsField *pTradingParams,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pTradingParams: 指向查询 KS 交易参数请求结构的指针。

查询 KS 交易参数请求结构：

```
struct CKSTradingParamsField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///实值额是否计入执行冻结保证金参数
    TKSExecFrozenMarginParamsType   ExecFrozenMarginParams;
};
```

4.11.20. OnRspQryKSTradingAccount 方法

请求查询 KS 资金账户响应。

函数原型：

```
void OnRspQryKSTradingAccount(
    CKSTradingAccountField *pTradingAccount,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pTradingAccount: 指向 KS 资金账户结构的指针。

KS 资金账户结构：

```
struct CKSTradingAccountField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///资金帐号
```

```

TThostFtdcAccountIDType AccountID;
///币种代码
TThostFtdcCurrencyIDType CurrencyID;
///交易日
TThostFtdcDateType TradingDay;
///上日结存
TThostFtdcMoneyType PreBalance;
///可用资金
TThostFtdcMoneyType Available;
///可取资金
TThostFtdcMoneyType WithdrawQuota;
///买冻结
TThostFtdcMoneyType BuyFrozenMargin;
///卖冻结
TThostFtdcMoneyType SellFrozenMargin;
///冻结的手续费
TThostFtdcMoneyType FrozenCommission;
///手续费
TThostFtdcMoneyType Commission;
///买保证金
TThostFtdcMoneyType BuyMargin;
///卖保证金
TThostFtdcMoneyType SellMargin;
///平仓盈亏
TThostFtdcMoneyType CloseProfit;
///浮动盈亏
TThostFtdcMoneyType PositionProfit;
///在途资金
TThostFtdcMoneyType Transit;
///未交割盈亏
TThostFtdcMoneyType UndeliveredProfit;
///今日入金
TThostFtdcMoneyType Deposit;
///今日出金
TThostFtdcMoneyType Withdraw;
///动态权益
TThostFtdcMoneyType Balance;
///冻结资金
TThostFtdcMoneyType FrozenCash;
///总保证金

```



```

TThostFtdcMoneyType TotalMargin;
///总交易所保证金
TThostFtdcMoneyType ExchangeMargin;
///质押金额
TThostFtdcMoneyType Mortgage;
///信用金额
TThostFtdcMoneyType Credit;
///风险级别
TKSRiskLevelType RiskLevel;
///风险度
TThostFtdcRatioType RishRatio1;
///风险度
TThostFtdcRatioType RishRatio2;
///当日权利金收入
TThostFtdcMoneyType RoyaltyIn;
///当日权利金支出
TThostFtdcMoneyType RoyaltyOut;
///冻结权利金
TThostFtdcMoneyType FrozenRoyalty;
///期权市值
TThostFtdcMoneyType OptionMarket;
///市值动态权益
TThostFtdcMoneyType MarketBalance;
///执行冻结保证金
TThostFtdcVolumeType StrikeFrozenMargin;
///实时保证金
TThostFtdcMoneyType TimeMargin;
///优惠保证金
TThostFtdcMoneyType DiscountMargin;
///实体账户货币质入
TThostFtdcMoneyType EntityFundMortgageIn;
///实体账户货币质出
TThostFtdcMoneyType EntityFundMortgageOut;
///日初质押资金
TThostFtdcMoneyType PreMortgage;
///质押余额
TThostFtdcMoneyType FundMortgageAvailable;
///借用资金
TThostFtdcMoneyType BorrowCash;
///借出资金

```

```

TThostFtdcMoneyType LendCash;
///特殊产品总冻结保证金
TThostFtdcMoneyType SpecProductFrozenMargin;
///特定产品买冻结保证金
TThostFtdcMoneyType SpecProductBuyFrozenMargin;
///特定产品卖冻结保证金
TThostFtdcMoneyType SpecProductSellFrozenMargin;
///特定产品买持仓保证金
TThostFtdcMoneyType SpecProductBuyMargin;
///特定产品卖持仓保证金
TThostFtdcMoneyType SpecProductSellMargin;
///特殊产品持仓保证金
TThostFtdcMoneyType SpecProductMargin;
///特殊产品交易所保证金
TThostFtdcMoneyType SpecProductExchangeMargin;
///特殊产品冻结手续费
TThostFtdcMoneyType SpecProductFrozenCommission;
///特殊产品手续费
TThostFtdcMoneyType SpecProductCommission;
///特殊产品平仓盈亏
TThostFtdcMoneyType SpecProductCloseProfit;
///特殊产品浮动盈亏
TThostFtdcMoneyType SpecProductPositionProfit;
///质押账户货币质入
TThostFtdcMoneyType FundMortgageIn;
///质押账户货币质出
TThostFtdcMoneyType FundMortgageOut;
///追加保证金
TThostFtdcMoneyType AddMargin;
///期权浮动盈亏
TThostFtdcMoneyType OptionPositionProfit;
///期权平仓盈亏
TThostFtdcMoneyType OptionCloseProfit;
};

```

4.11.21. OnRspQryWithdrawReservedAccount 方法

请求查询预约出金账户响应。

函数原型：

```

void OnRspQryWithdrawReservedAccount(
    CKSWithdrawReservedAccountField *pWithdrawReservedAccount,

```

```

    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pWithdrawReservedAccount: 指向预约出金账户结构的指针。

预约出金账户结构:

```

struct CKSWithdrawReservedAccountField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///银行简称
    TThostFtdcBankAccountType BankName;
    ///银行账号
    TThostFtdcBankAccountType BankAccount;
    ///银行代码
    TThostFtdcBankIDType   BankID;
    ///开户银行
    TThostFtdcLedgerManageBankType  Bank;
    ///币种代码
    TThostFtdcCurrencyIDType  CurrencyID;
};

```

4.11.22. OnRspWithdrawReservedApply 方法

预约出金申请响应。

函数原型:

```

void OnRspWithdrawReservedApply(
    CKSWithdrawReservedApplyField *pWithdrawReservedApply,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数:

pWithdrawReservedApply: 指向预约出金申请结构的指针。

预约出金申请结构:

```

struct CKSWithdrawReservedApplyField
{
    ///经纪公司代码

```

```

    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///资金密码
    TThostFtdcPasswordType     Password;
    ///出金金额
    TThostFtdcMoneyType        Withdraw;
    ///银行账号
    TThostFtdcBankAccountType  BankAccount;
    ///银行代码
    TThostFtdcBankIDType       BankID;
    ///币种代码
    TThostFtdcCurrencyIDType   CurrencyID;
    ///流水号
    TThostFtdcTradeSerialNoType Serial;
};

```

4.11.23. OnRspWithdrawReservedCancel 方法

预约出金撤销申请响应。

函数原型：

```

void OnRspWithdrawReservedCancel(
    CKSWithdrawReservedCancelField *pWithdrawReservedCancel,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);

```

参数：

pWithdrawReservedCancel： 指向预约出金撤销申请结构的指针。

预约出就撤销申请结构：

```

struct CKSWithdrawReservedCancelField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///资金密码
    TThostFtdcPasswordType     Password;
    ///流水号
    TThostFtdcTradeSerialNoType Serial;
};

```

4.11.24. OnRspQryWithdrawReservedApply 方法

请求查询预约出金响应。

函数原型:

```
void OnRspQryWithdrawReservedApply(
    CKSRspWithdrawReservedApplyField *pRspWithdrawReservedApply,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数:

pRspWithdrawReservedApply: 指向查询预约出金结构的指针。

查询预约出金结构:

```
struct CKSRspWithdrawReservedApplyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///银行简称
    TThostFtdcBankAccountType BankName;
    ///银行账号
    TThostFtdcBankAccountType BankAccount;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///出金金额
    TThostFtdcMoneyType Withdraw;
    ///处理状态
    TKSSStatusMsgType StatusMsg;
    ///处理结果
    TKSResultMsgType ResultMsg;
    ///操作日期
    TThostFtdcDateType ApplyDate;
    ///操作时间
    TThostFtdcTimeType ApplyTime;
    ///流水号
    TThostFtdcTradeSerialNoType Serial;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
};
```

4.11.25. OnRspOptionSelfCloseAction 方法

期权自对冲操作响应。

函数原型：

```
void OnRspOptionSelfCloseAction(
    CKSOptionSelfCloseActionField *pOptionSelfCloseAction,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast);
```

参数：

pOptionSelfCloseAction： 指向期权自对冲操作结构的指针。

期权自对冲操作结构：

```
struct CKSOptionSelfCloseActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///自对冲操作引用
    TThostFtdcOrderActionRefType    SelfCloseActionRef;
    ///自对冲引用
    TThostFtdcOrderRefType    SelfCloseRef;
    ///请求编号
    TThostFtdcRequestIDType    RequestID;
    ///前置编号
    TThostFtdcFrontIDTypeFrontID;
    ///会话编号
    TThostFtdcSessionIDType    SessionID;
    ///期权自对冲操作标志
    TThostFtdcActionFlagType    ActionFlag;
    ///自对冲类型
    TKSOptSelfCloseFlagType    SelfCloseFlag;
```

```

///会员代码
TThostFtdcParticipantIDType    ParticipantID;
///客户代码
TThostFtdcClientIDType    ClientID;
///交易用户代码
TThostFtdcUserIDType    UserID;
///业务单元
TThostFtdcBusinessUnitType    BusinessUnit;
///交易所代码
TThostFtdcExchangeIDType    ExchangeID;
///期权自对冲系统编号
TKSOptionSelfCloseSysIDType    OptionSelfCloseSysID;
///IP 地址
TThostFtdcIPAddressType    IPAddress;
///Mac 地址
TThostFtdcMacAddressType    MacAddress;
///操作本地编号
TThostFtdcOrderLocalIDType    ActionLocalID;
///流水号
TThostFtdcTradeSerialNoType    Serial;
///对冲编号
TThostFtdcOrderLocalIDType    OptionSelfCloseLocalID;
///对冲标志
TThostFtdcHedgingFlagType    HedgingFlag;
};

```

4.11.26. OnRspQryDepositAndWithdraw 方法

用户端出入金查询应答

函数原型:

```

void OnRspQryDepositAndWithdraw(
    CKSRspDepositAndWithdrawField *pRspQryDepositAndWithdraw,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)

```

参数:

pRspQryDepositAndWithdraw: 指向用户端出入金查询结构的地址

用户端出入金响应结构如下:

```
struct CKSRspDepositAndWithdrawField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType      BrokerID;
    ///日期
    TThostFtdcDateType          Date;
    ///流水号
    TThostFtdcTradeSerialNoType Serial;
    ///借方（出金）
    TThostFtdcMoneyType          Withdraw;
    ///贷方（入金）
    TThostFtdcMoneyType          Deposit;
    ///摘要
    TThostFtdcDigestType         Digest;
    ///币种
    TThostFtdcCurrencyIDType     CurrencyID;
    ///银行名称
    TThostFtdcBankAccountType    BankName;
}
```

4.11.27. OnRspQryInvestorPortfMarginRatio 方法

投资者新型组合保证金系数查询响应。当客户端发出投资者新型组合保证金系数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
OnRspQryInvestorPortfMarginRatio(CThostFtdcInvestorPortfMarginRatioField
    *pInvestorPortfMarginRatio, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
    bIsLast);
```

参数:

pInvestorPortfMarginRatio: 投资者新型组合保证金系数地址。

投资者新型组合保证金参数结构:

```
struct CThostFtdcInvestorPortfMarginRatioField
{
    ///投资者范围
    TThostFtdcInvestorRangeType    InvestorRange;
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType    ExchangeID;
    ///会员对投资者收取的保证金和交易所对投资者收取的保证金的比例
    TThostFtdcRatioType    MarginRatio;
    ///交易编码
    TThostFtdcClientIDType    ClientID;
};
```

4.11.28. OnRspQryInvestorSPMMCombDiscountParam 方法

投资者 SPMM 组合优惠分类参数查询响应。当客户端发出投资者 SPMM 组合优惠分类参数查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型:

```
void OnRspQryInvestorSPMMCombDiscountParam(
    CKSInvestorSPMMCombDiscountParamField
    *pInvestorSPMMCombDiscountParam, CThostFtdcRspInfoField *pRspInfo,
    int nRequestID, bool bIsLast);
```

参数:

pInvestorSPMMCombDiscountParam: 投资者 SPMM 组合优惠分类参数地址。

投资者 SPMM 组合优惠分类参数结构:

```
struct CKSInvestorSPMMCombDiscountParamField
{
```

```

///投资者代码
TThostFtdcInvestorIDType InvestorID;
///交易所代码
TThostFtdcExchangeIDType ExchangeID;
///交易编码
TThostFtdcClientIDType ClientID;
///是否启用
TThostFtdcBoolType IsSPMM;
///优惠分类
TThostFtdcDiscountClassificationType DiscountClassification;
///商品群代码
TThostFtdcSPMMProductIDType CommodityGroupID;
///期权纯粹价格风险比例
TThostFtdcMoneyType OptionsPureRiskRatio;
///跨期优惠系数
TThostFtdcMoneyType IntertemporalCommodityRate;
///跨品种优惠系数
TThostFtdcMoneyType InterCommodityRate;
///最小保证金比例
TThostFtdcSPMMDiscountRatioType MiniMarginRatio;
};

```

4.11.29. OnRspQryInvestorProdRULEDetail 方法

投资者 RULE 明细查询响应。当客户端发出投资者 RULE 明细查询指令后，Kingstar 服务器返回响应时，该方法会被调用。

函数原型：

```

OnRspQryInvestorProdRULEDetail(
    CKSInvestorProdRULEDetailField *pInvestorProdRULEDetail,
    CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool bIsLast);

```

参数：

CKSInvestorProdRULEDetailField ：投资者 RULE 明细查询参数地址。

投资者 RULE 明细查询参数结构：

```

struct CKSInvestorProdRULEDetailField
{
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///交易编码
    TThostFtdcClientIDType     ClientID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
    ///币种代码
    TThostFtdcCurrencyIDType  CurrencyID;
    ///客户品种保证金
    TThostFtdcMoneyType  ClientMargin;
    ///交易所品种保证金
    TThostFtdcMoneyType  ExchMargin;
    ///交易所品种内对锁保证金
    TThostFtdcMoneyType  ExchIntraProdMargin;
    ///交易所品种间对锁保证金
    TThostFtdcMoneyType  ExchInterProdMargin;
    ///交易所品种裸仓保证金
    TThostFtdcMoneyType  ExchSingleMargin;
    ///交易所交割月裸仓保证金
    TThostFtdcMoneyType  ExchDeliverySingleMargin;
    ///交易所非组合合约保证金
    TThostFtdcMoneyType  ExchNonCombMargin;
    ///交易所持仓附加保证金
    TThostFtdcMoneyType  ExchAddOnMargin;
    ///委托冻结保证金
    TThostFtdcMoneyType  FrozenMargin;
    ///交易所委托冻结保证金
    TThostFtdcMoneyType  ExchFrozenMargin;
    ///品种内对锁保证金
    TThostFtdcMoneyType  IntraProdMargin;

```

```

    ///品种间对锁保证金
    TThostFtdcMoneyType  InterProdMargin;
    ///品种裸仓保证金
    TThostFtdcMoneyType  SingleMargin;
    ///交割月裸仓保证金
    TThostFtdcMoneyType  DeliverySingleMargin;
    ///非组合合约保证金
    TThostFtdcMoneyType  NonCombMargin;
    ///持仓附加保证金
    TThostFtdcMoneyType  AddOnMargin;
};

```

4.12. CKSVocApi 接口

4.12.1. ReqQueryInvestorOpenPosition 方法

请求查询开盘前的持仓明细

函数原型:

```

int ReqQueryInvestorOpenPosition (
    CThostFtdcQryInvestorPositionDetailField *pQryInvestorPositionDetail,
    int nRequestID);

```

参数:

pQryInvestorPositionDetail: 指向投资者开盘前持仓明细请求结构的地址。

投资者开盘前持仓明细结构:

```

struct CThostFtdcQryInvestorPositionDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType  BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
};

```

4.12.2. ReqQueryInvestorOpenCombinePosition 方法

查询开盘前的组合持仓明细

函数模型:

```
int ReqQueryInvestorOpenCombinePosition (
    CThostFtdcQryInvestorPositionCombineDetailField
    *pQryInvestorPositionCombineDetail,
    int nRequestID);;
```

参数:

pQryInvestorPositionCombineDetail: 指向投资者开盘前组合持仓明细请求结构的地址。

开盘前组合持仓明细请求结构:

```
struct CThostFtdcQryInvestorPositionCombineDetailField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///组合持仓合约编码
    TThostFtdcInstrumentIDType CombInstrumentID;
};
```

4.12.3. ReqBulkCancelOrder 方法

批量撤单请求

函数模型:

```
int ReqBulkCancelOrder(
    CThostFtdcBulkCancelOrderField *pBulkCancelOrder,
    int nRequestID)
```

参数:

pBulkCancelOrder: 指向批量报单请求结构的地址。

批量报单请求结构如下:

```
struct CThostFtdcBulkCancelOrderField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
```

```

    ///投资者代码
    TThostFtdcInvestorIDType    InvestorID;
    ///用户代码
    TThostFtdcUserIDType        UserID;
    ///报单类型
    TThostFtdcOrderTypeType     OrderType;
    ///报单个数
    TThostFtdcVolumeType nCount;
    ///报单集合
    CThostOrderKeyField OrderKey[MAX_ORDER_COUNT];
}

```

4.12.4. ReqQryCloseStrategy 方法

平仓策略查询请求

函数模型：

```

int ReqQryCloseStrategy(
    CKSCloseStrategy *pCloseStrategy,
    int nRequestID)

```

参数：

pCloseStrategy: 指向平仓策略请求结构的地址。

平仓策略请求结构如下：

```

struct CKSCloseStrategy
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
};

```

4.12.5. ReqQryCombStrategy 方法

组合策略查询请求

函数模型：

```

int ReqQryCombStrategy(
    CKSCombStrategy *pCombStrategy,

```

```
int nRequestID)
```

参数:

pCombStrategy: 指向组合策略请求结构的地址。

组合策略请求结构如下:

```
struct CKSCombStrategy
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易类别(1:期货;2:期权;空:全部)
    TThostFtdcProductClassType ProductClass;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///策略代码(SP/SPD)
    TKSSStrategyIDType StrategyID;
    ///互换单标志
    TThostFtdcBoolType IsSwapOrder;
};
```

4.12.6. ReqQryOptionCombStrategy 方法

期权组合策略查询请求

函数模型:

```
int ReqQryOptionCombStrategy(
    CKSOptionCombStrategy *pOptionCombStrategy,
    int nRequestID)
```

参数:

pOptionCombStrategy: 指向期权组合策略请求结构的地址。

期权组合策略请求结构如下:

```
struct CKSOptionCombStrategy
{
    ///营业部代码
    TThostFtdcBrokerIDType BrokerID;
```

```

    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///策略代码
    TKSSStrategyIDType StrategyID;
    ///组合买卖方向
    TThostFtdcDirectionType CombDirection;
    ///组合开平标志
    TThostFtdcCombOffsetFlagType CombOffsetFlag;
};

```

4.12.7. ReqQryTransferInfo 方法

客户转帐信息查询请求

函数模型：

```

int ReqQryTransferInfo (
    CKSTransferInfo *pTransferInfo,
    int nRequestID)

```

参数：

pTransferInfo：指向客户转帐信息请求结构的地址。

客户转帐信息请求结构如下：

```

struct CKSTransferInfo
{
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
};

```

4.12.8. ReqQryKSTradingNotice 方法

交易通知查询请求

函数模型：

```

int ReqQryKSTradingNotice (
    CKSQryTradingNoticeField *pQryTradingNotice,
    int nRequestID)

```


参数:

pQryTradingNotice: 指向交易通知请求结构的地址。

交易通知请求结构如下:

```
struct CKSQryTradingNoticeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///信息类型
    TKSInfoTypeType InfoType;
    ///确认标志
    TKSConfirmFlagType ConfirmFlag;
    ///开始时间
    TThostFtdcTimeType   TimeStart;
    ///结束时间
    TThostFtdcTimeType   TimeEnd;
};
```

4.12.9. ReqQryUserProductUrl 方法

用户端产品资源查询请求

函数模型:

```
int ReqQryUserProductUrl (
    CKSQryUserProductUrlField *pQryUserProductUrl,
    int nRequestID)
```

参数:

pQryUserProductUrl: 指向用户端产品资源请求结构的地址。

用户端产品资源请求结构如下:

```
struct CKSQryUserProductUrlField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
```

```

///用户端产品信息
TThostFtdcProductInfoType  UserProductInfo;
};

```

4.12.10. ReqQryMaxCombActionVolume 方法

最大组合拆分单量查询请求。

函数原型：

```

int ReqQryMaxCombActionVolume(
    CKSQryMaxCombActionVolumeField *pQryMaxCombActionVolume,
    int nRequestID);

```

参数：

pQryMaxComb.ActionVolume： 指向最大组合拆分单量查询结构的指针。

最大组合拆分单量查询结构：

```

struct CKSQryMaxCombActionVolumeField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType  BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType  ExchangeID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///投保标记
    TThostFtdcHedgeFlagType  HedgeFlag;
    ///买卖方向
    TThostFtdcDirectionType  Direction;
    ///组合拆分标记
    TThostFtdcCombDirectionType  CombDirection;
};

```

4.12.11. ReqQryKSInstrCommRate 方法

请求查询合约手续费。

函数原型：

```

int ReqQryKSInstrCommRate(
    CKSQryInstrCommRateField *pQryInstrCommRate,
    int nRequestID);

```

参数:

pQryInstrCommRate: 指向查询合约手续费请求结构的指针。

查询合约手续费请求结构:

```
struct CKSQryInstrCommRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
};
```

4.12.12. ReqQryKSETFOptionInstrCommRate 方法

请求查询 ETF 期权合约手续费。

函数原型:

```
int ReqQryKSETFOptionInstrCommRate(
    CKSQryETFOptionInstrCommRateField *pQryETFOptionInstrCommRate,
    int nRequestID);
```

参数:

PQryETFOptionInstrCommRate: 指向查询 ETF 期权合约手续费请求结构的指针。

查询 ETF 期权合约手续费请求结构:

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

4.12.13. ReqQryKSIInstrumentMarginRate 方法

请求查询合约保证金率。

函数原型：

```
int ReqQryKSIInstrumentMarginRate(
    CKSQryInstrumentMarginRateField *pQryInstrumentMarginRate,
    int nRequestID);
```

参数：

pQryInstrumentMarginRate： 指向查询合约保证金率请求结构的指针。

查询合约保证金率请求结构：

```
struct CKSQryInstrumentMarginRateField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///投机套保标志
    TThostFtdcHedgeFlagType HedgeFlag;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.12.14. ReqWithdrawCreditApply 方法

客户每日出金额度申请。

函数原型：

```
int ReqWithdrawCreditApply(
    CKSInputWithdrawCreditApplyField *pWithdrawCreditApply,
    int nRequestID);
```

参数：

pWithdrawCreditApply： 指向客户每日出金额度申请结构的指针。

客户每日出金额度申请结构：

```
struct CKSInputWithdrawCreditApplyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
```

```

    ///当日出金额度
    TThostFtdcMoneyType WithdrawCredit;
    ///当日出金额度操作类型
    TKSFOCreditApplyType CreditApplyFlag;
    ///流水号
    TThostFtdcTradeSerialNoType CreditSerial;
};

```

4.12.15. ReqQryWithdrawCredit 方法

客户每日出金额度查询。

函数原型：

```

int ReqQryWithdrawCredit(
    CKSQryWithdrawCreditField *pQryWithdrawCredit,
    int nRequestID);

```

参数：

pQryWithdrawCredit：指向客户每日出金额度查询结构的指针。

客户每日出金额度查询结构：

```

struct CKSQryWithdrawCreditField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
};

```

4.12.16. ReqQryWithdrawCreditApply 方法

客户每日出金额度申请查询。

函数原型：

```

int ReqQryWithdrawCreditApply(
    CKSQryWithdrawCreditApplyField *pQryWithdrawCreditApply,
    int nRequestID);

```

参数：

pQryWithdrawCreditApply：指向客户每日出金额度申请查询结构的指针。

客户每日出金额度申请查询结构：

```

struct CKSQryWithdrawCreditApplyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
};

```

```

    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///查询开始日期
    TThostFtdcDateType         QryCreditStart;
    ///查询结束日期
    TThostFtdcDateType         QryCreditEnd;
    ///当日出金额度处理状态
    TKSFOCreditStatusType      DealStatus;
};

```

4.12.17. ReqKSEMailLogin 方法

高华认证邮箱登陆请求。

函数原型：

```

int ReqKSEMailLogin(
    CKSReqEMailLoginField *pReqEMailLoginField,
    int nRequestID);

```

参数：

pReqEMailLoginField： 指向高华认证邮箱登录请求结构的指针。

高华认证邮箱登陆请求结构：

```

struct CKSReqEMailLoginField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///电子邮箱
    TThostFtdcUOAEMailType   UOAEMail;
    ///动态密码
    TThostFtdcPasswordType   OneTimePassword;
    ///用户端产品信息
    TThostFtdcProductInfoType UserProductInfo;
    ///接口端产品信息
    TThostFtdcProductInfoType InterfaceProductInfo;
    ///Mac地址
    TThostFtdcMacAddressType  MacAddress;
    ///终端IP地址
    TThostFtdcIPAddressType   ClientIPAddress;
};

```

4.12.18. ReqQryKSOOptionMargin 方法

请求查询个股单张保证金。

函数原型:

```
int ReqQryKSOOptionMargin(
    CKSQryOptionMarginField *pQryOptionMargin,
    int nRequestID);
```

参数:

pQryOptionMargin: 指向查询个股单张保证金请求结构的指针。

查询个股单张保证金请求结构:

```
struct CKSQryOptionMarginField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDTypeInstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.12.19. ReqQryKSTradingParams 方法

请求查询 KS 交易参数。

函数原型:

```
int ReqQryKSTradingParams(
    CKSQryTradingParamsField *pQryTradingParams,
    int nRequestID);
```

参数:

pQryTradingParams: 指向查询 KS 交易参数请求结构的指针。

查询 KS 交易参数请求结构:

```
struct CKSQryTradingParamsField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
};
```

4.12.20. ReqQryKSTradingAccount 方法

请求查询 KS 资金账户

函数原型:

```
int ReqQryKSTradingAccount(
    CKSQryTradingAccountField *pQryTradingAccount,
    int nRequestID);
```

参数:

pQryTradingAccount: 指向查询 KS 资金账户请求结构的指针。

查询 KS 资金账户请求结构:

```
struct CKSQryTradingAccountField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
};
```

4.12.21. ReqQryWithdrawReservedAccount 方法

预约出金账户查询。

函数原型:

```
int ReqQryWithdrawReservedAccount(
    CKSQryWithdrawReservedAccountField *pQryWithdrawReservedAccount,
    int nRequestID);
```

参数:

pQryWithdrawReservedAccount: 指向查询预约出金账户请求结构的指针。

查询预约出金账户请求结构:

```
struct CKSQryWithdrawReservedAccountField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType   BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///密码
    TThostFtdcPasswordType   Password;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
};
```


4.12.22. ReqWithdrawReservedApply 方法

预约出金申请。

函数原型：

```
int ReqWithdrawReservedApply(
    CKSWithdrawReservedApplyField *pWithdrawReservedApply,
    int nRequestID);
```

参数：

pWithdrawReservedApply： 指向预约出金申请结构的指针。

预约出金申请结构：

```
struct CKSWithdrawReservedApplyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///资金密码
    TThostFtdcPasswordType Password;
    ///出金金额
    TThostFtdcMoneyType Withdraw;
    ///银行账号
    TThostFtdcBankAccountType BankAccount;
    ///银行代码
    TThostFtdcBankIDType BankID;
    ///币种代码
    TThostFtdcCurrencyIDType CurrencyID;
    ///流水号
    TThostFtdcTradeSerialNoType Serial;
};
```

4.12.23. ReqWithdrawReservedCancel 方法

预约出金撤销申请。

函数原型：

```
int ReqWithdrawReservedCancel(
    CKSWithdrawReservedCancelField *pWithdrawReservedCancel,
    int nRequestID);
```

参数：

pWithdrawReservedCancel： 指向预约出金撤销申请结构的指针。

预约出金撤销申请结构：

```

struct CKSWithdrawReservedCancelField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///资金密码
    TThostFtdcPasswordType    Password;
    ///流水号
    TThostFtdcTradeSerialNoType    Serial;
};

```

4.12.24. ReqQryWithdrawReservedApply 方法

预约出金查询。

函数原型：

```

int ReqQryWithdrawReservedApply(
    CKSQryWithdrawReservedApplyField *pQryWithdrawReservedApply,
    int nRequestID);

```

参数：

pQryWithdrawReservedApply: 指向预约出金查询结构指针。

预约出金查询结构：

```

struct CKSQryWithdrawReservedApplyField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType  InvestorID;
    ///资金密码
    TThostFtdcPasswordType    Password;
    ///开始日期
    TThostFtdcDateType        DateStart;
    ///结束日期
    TThostFtdcDateType        DateEnd;
    ///流水号
    TThostFtdcTradeSerialNoType    Serial;
    ///币种代码
    TThostFtdcCurrencyIDType    CurrencyID;
};

```

4.12.25. ReqOptionSelfCloseAction 方法

期权自对冲操作请求。

函数原型：

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

参数：

pOptionSelfCloseAction： 指向期权自对冲操作请求结构的指针。

期权自对冲操作请求结构：

```
struct CKSOptionSelfCloseActionField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///自对冲操作引用
    TThostFtdcOrderActionRefType SelfCloseActionRef;
    ///自对冲引用
    TThostFtdcOrderRefType SelfCloseRef;
    ///请求编号
    TThostFtdcRequestIDType RequestID;
    ///前置编号
    TThostFtdcFrontIDType FrontID;
    ///会话编号
    TThostFtdcSessionIDType SessionID;
    ///期权自对冲操作标志
    TThostFtdcActionFlagType ActionFlag;
    ///自对冲类型
    TKSOptSelfCloseFlagType SelfCloseFlag;
    ///会员代码
    TThostFtdcParticipantIDType ParticipantID;
    ///客户代码
    TThostFtdcClientIDType ClientID;
    ///交易用户代码
    TThostFtdcUserIDType UserID;
    ///业务单元
```

```

    TThostFtdcBusinessUnitType BusinessUnit;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///期权自对冲系统编号
    TKSOptionSelfCloseSysIDType OptionSelfCloseSysID;
    ///IP地址
    TThostFtdcIPAddressType IPAddress;
    ///Mac地址
    TThostFtdcMacAddressType MacAddress;
    ///操作本地编号
    TThostFtdcOrderLocalIDType ActionLocalID;
    ///流水号
    TThostFtdcTradeSerialNoType Serial;
    ///对冲编号
    TThostFtdcOrderLocalIDType OptionSelfCloseLocalID;
    ///对冲标志
    TThostFtdcHedgingFlagType HedgingFlag;
};

```

4.12.26. ReqQryDepositAndWithdraw 方法

用户端出入金查询请求

函数模型:

```

int ReqQryDepositAndWithdraw (
    CKSQryDepositAndWithdrawField *pQryDepositAndWithdraw,
    int nRequestID)

```

参数:

pQryDepositAndWithdraw: 指向用户端出入金查询请求结构的地址。

用户端出入金查询请求结构如下:

```

struct CKSQryDepositAndWithdrawField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///开始日期
    TThostFtdcDateType DateStart;
    ///结束日期
    TThostFtdcDateType DateEnd;
}

```

```

    ///出入金方向1入金2出金
    TThostFtdcFundDirectionType    FundDirection;
    ///币种代码
    TThostFtdcCurrencyIDType        CurrencyID;
};

```

备注：其中CurrencyID传入RMB代表人民币，USD代表美元，HKD代表港币

4.12.27. ReqQryInvestorPortfMarginRatioAddClientID 方法

投资者新型组合保证金系数查询带交易编码函数模型：

```

int ReqQryInvestorPortfMarginRatioAddClientID(
    CKSQryInvestorPortfMarginRatioAddClientIDField
    *pQryInvestorPortfMarginRatioAddClientID,int nRequestID)

```

参数：

pQryInvestorPortfMarginRatioAddClientID：投资者新型组合保证金系数查询带交易编码的地址。

投资者新型组合保证金系数查询带交易编码结构如下：

```

struct CKSQryInvestorPortfMarginRatioAddClientIDField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType    BrokerID;
    ///投资者代码
    TThostFtdcInvestorIDType   InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType   ExchangeID;
    ///交易编码
    TThostFtdcClientIDType      ClientID;
};

```

4.12.28. ReqQryInvestorSPMMCombDiscountParam 方法

投资者 SPMM 组合优惠分类参数查询：

```

ReqQryInvestorSPMMCombDiscountParam(
    CKSQryInvestorSPMMCombDiscountParamField
    *pQryInvestorSPMMCombDiscountParam, int nRequestID)

```

参数：

pQryInvestorSPMMCombDiscountParam: 投资者 SPMM 组合优惠分类参数查询的地址。

投资者 SPMM 组合优惠分类参数查询结构如下:

```
struct CKSQryInvestorSPMMCombDiscountParamField
{
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///交易编码
    TThostFtdcClientIDType ClientID;
};
```

4.12.29. ReqQryInvestorProdRULEDetail 方法

投资者 RULE 明细查询请求:

```
ReqQryInvestorProdRULEDetail(
    CKSQryInvestorProdRULEDetailField *pQryInvestorProdRULEDetail,
    int nRequestID)
```

参数:

CKSQryInvestorProdRULEDetailField : 投资者 RULE 明细查询的地址。

投资者 RULE 明细查询结构如下:

```
struct CKSQryInvestorProdRULEDetailField
{
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
    ///投资者代码
    TThostFtdcInvestorIDType InvestorID;
    ///品种代码
    TThostFtdcInstrumentIDType ProdFamilyCode;
};
```

4.13. CKSVocMdSpi 接口

CKSVocMdSpi 实现了事件通知接口。该 Spi 中函数应答结构添加了交易所代码字段。

4.13.1. OnRspSubKSMarketData 方法

当客户端发出订阅行情请求之后, Kingstar 返回响应时, 该方法会被调用, 通知客户端订阅是否成功。

函数原型：

```
void OnRspSubKSMarketData(
    CKSSpecificInstrumentField *pSpecificInstrument,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pSpecificInstrument： 返回客户端订阅行情的合约结构地址。

合约结构：

```
struct CKSSpecificInstrumentField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    /// 交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};
```

4.13.2. OnRspUnSubKSMarketData 方法

当客户端发出退订行情请求之后，Kingstar 返回响应时，该方法会被调用，通知客户端退订是否成功。

函数原型：

```
void OnRspUnSubKSMarketData (
    CKSSpecificInstrumentField *pSpecificInstrument,
    CThostFtdcRspInfoField *pRspInfo,
    int nRequestID,
    bool bIsLast)
```

参数：

pSpecificInstrument： 返回客户端退订行情的合约结构地址。

合约结构：

```
struct CKSSpecificInstrumentField
{
    /// 合约代码
    TThostFtdcInstrumentIDType InstrumentID;
```

```

    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

4.14. CKSVocMdApi 接口

CKSVocMdApi 接口提供给用户的功能主要体现在行情的订阅、退订等功能。该函数请求结构添加了交易所代码字段。

4.14.1. SubscribeKSMarketData 方法

用户发出订阅行情的请求。

函数原型：

```

int SubscribeKSMarketData(
    CKSSpecificInstrumentField *ppInstrumentID[],
    int nCount);

```

参数：

ppInstrumentID： 指向投资者订阅行情请求结构的地址。

订阅行情请求结构：

```

struct CKSSpecificInstrumentField
{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

nCount： 行情订阅的合约个数。

4.14.2. UnSubscribeKSMarketData 方法

用户发出退订行情的请求。

```

int UnSubscribeKSMarketData(
    CKSSpecificInstrumentField *ppInstrumentID[],
    int nCount);

```

参数：

ppInstrumentID： 指向投资者退订行情请求结构的地址。

退订行情请求结构：

```

struct CKSSpecificInstrumentField

```



```

{
    ///合约代码
    TThostFtdcInstrumentIDType InstrumentID;
    ///交易所代码
    TThostFtdcExchangeIDType ExchangeID;
};

```

nCount: 退订行情的合约个数。

4.14.3. ReqKSEMailLogin 方法

高华认证邮箱登陆请求。

函数原型:

```

int ReqKSEMailLogin(
    CKSReqEMailLoginField *pReqEMailLoginField,
    int nRequestID);

```

参数:

pReqEMailLoginField: 指向高华认证邮箱登陆请求结构的指针。

高华认证邮箱登陆请求结构:

```

struct CKSReqEMailLoginField
{
    ///经纪公司代码
    TThostFtdcBrokerIDType BrokerID;
    ///电子邮箱
    TThostFtdcUOAEMailType UOAEMail;
    ///动态密码
    TThostFtdcPasswordType OneTimePassword;
    ///用户端产品信息
    TThostFtdcProductInfoType UserProductInfo;
    ///接口端产品信息
    TThostFtdcProductInfoType InterfaceProductInfo;
    ///Mac地址
    TThostFtdcMacAddressType MacAddress;
    ///终端IP地址
    TThostFtdcIPAddressType ClientIPAddress;
};

```

5. 第五章 示例代码

参见开发包内的 Demo 文件夹。

6. 第六章 Q&A 及问题反馈方式

6.1. 多实例的编程要求

此版本 API 支持 256 个实例。

对于不同实例用同一个客户号登录的情况，受多点登录的限制（柜台中设置有多点登录数量，一般为 5 个，具体数量需咨询所接的期货公司）。多点登录时相同 IP、MAC 及目录算一个点。比如，程序在相同 IP、MAC 及同一个目录下运行，通过 KSFT_API 方式连接到柜台，此时会被判断为同一个点，这样后面的登录就会踢掉前面的登录。

若接口在使用过程有任何问题，请将详细信息反馈至电子邮箱：Mingming.shen@sungard.com 及 Yanyun.liu@sungard.com。谢谢。