

platformSvr 接口

文件状态： <input checked="" type="checkbox"/> 草稿 <input type="checkbox"/> 正式发布 <input type="checkbox"/> 正在修改	文件标识：	
	当前版本：	V1.0.0.3
	作 者：	Zhiqing.Liao
	完成日期：	2011 年 2 月 22 日
	审 定：	

版本历史

版本/状态	作者	起止日期	备注
V1.0.0.1	Zhiqing.Liao	2011/02/10-2011/02/10	初稿
V1.0.0.2	Zhiqing.Liao	2011/02/21-2011/02/21	不再提供下列接口： 请求查询合约保证金率：ReqQryInstrumentMarginRate 请求查询合约手续费率：ReqQryInstrumentCommissionRate 请求查询交易所：ReqQryExchange 请求查询经纪公司交易算法：ReqQryBrokerTradingAlgos 请求查询经纪公司交易参数：ReqQryBrokerTradingParams 请求查询交易通知：ReqQryTradingNotice
V1.0.0.3	Zhiqing.Liao	2011/02/22-2011/02/22	修改 GetWaitOrderVolume 接口，增加投保参数 修改 GetCloseOrderVolume 接口，增加投保参数 增加 GetPositionKeyList 函数，获取持仓键值列表 增加 GetPositionData 函数，获取指定持仓数据 增加 GetInstrumentMultiple 函数，获取合约乘数
V1.0.0.4	Zhiqing.Liao	2011/07/05-2011/07/07	修改接口数据结构，用 Platform_xxx 代替 ctp 的数据结构，使接口可以扩展

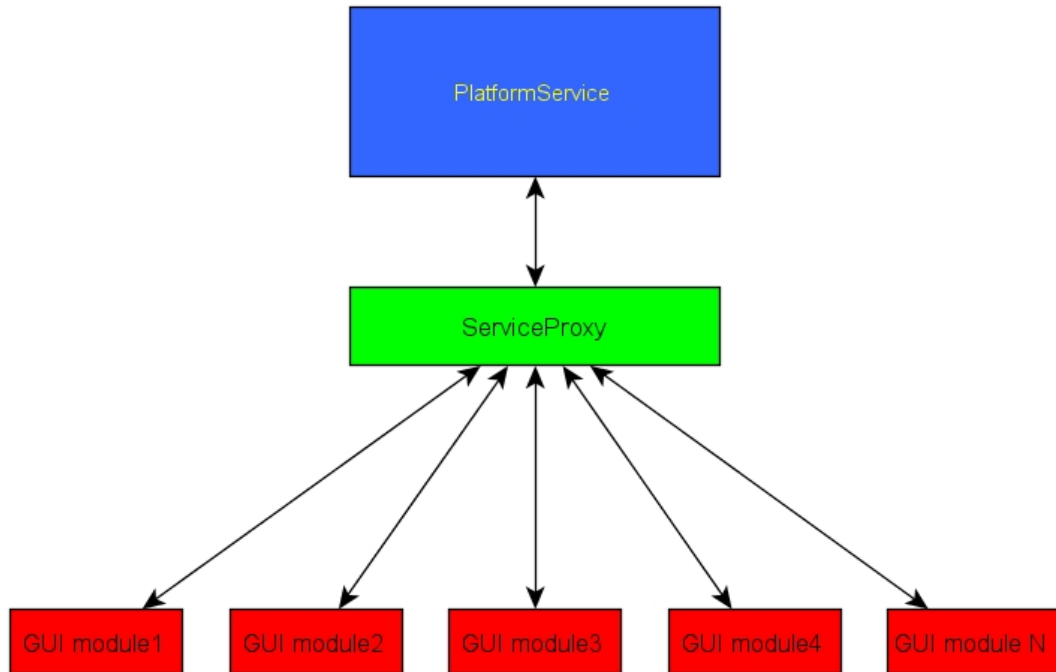
目 录

一.	platformSvr 接口示意图	6
二.	platformSvr 接口描述	7
1.	接口类.....	7
2.	接口分类.....	7
3.	头文件.....	7
三.	同步调用接口.....	8
1.	平台接口.....	8
(1).	平台启动接口 (OK)	8
(2).	平台关闭接口 (OK)	8
(3).	判断登录密码是否合法 (OK)	8
(4).	设置交易前置机地址 (OK)	8
(5).	设置行情前置机地址 (OK)	8
(6).	设置 CTP 登录信息 (OK)	9
(7).	登录 CTP (OK)	9
(8).	登出 CTP (OK)	9
(9).	获取底层服务的当前状态 (OK)	9
2.	合约信息接口.....	10
(1).	获取合约列表 (OK)	10
(2).	获取指定合约信息 (OK)	10
(3).	设置指定合约信息 (OK)	10
(4).	获取合约品种代码 (OK)	10
(5).	获取合约产品类型代码 (OK)	11
(6).	获取合约品种类型 (OK)	11
(7).	获取合约手续费率 (OK)	11
(8).	获取合约手续费率并触发查询 (OK)	11
(9).	设置合约手续费率 (OK)	12
(10).	获取合约保证金率 (OK)	12
(11).	获取合约保证金率并触发查询 (OK)	12
(12).	设置合约保证金率 (OK)	12
(13).	获取合约乘数 (OK)	13
3.	资金接口.....	13
(1).	获取交易资金账户信息 (OK)	13
(2).	设置交易资金账户信息 (OK)	13
4.	成交记录接口.....	13
(1).	获取全部成交单 (OK)	13
5.	持仓接口.....	14
(1).	获取指定合约的持仓明细 (OK)	14
(2).	获取持仓记录键值列表 (OK)	14
(3).	获取指定合约、方向、投保的持仓数据 (OK)	14
6.	报单接口.....	15

(1).	获取全部报单 (OK)	15
(2).	获取挂单数量 (OK)	15
(3).	获取指定合约指定方向的平仓挂单的平仓量 (OK)	15
7.	行情接口	15
(1).	获取指定合约行情 (OK)	15
(2).	获取指定合约的最新价 (OK)	16
(3).	获取指定合约旧行情 (OK)	16
8.	其它接口	16
(1).	获取交易所时间	16
(2).	获取当前登入的投资者个人信息	16
(3).	请求查询投资者结算结果响应	17
(4).	获取连接服务器和登录时间	17
四.	异步调用接口	18
1.	用户口令更新请求	18
2.	资金账户口令更新请求	18
3.	报单录入请求	18
4.	预埋单录入请求	18
5.	预埋撤单录入请求	18
6.	报单操作请求	19
7.	查询最大报单数量请求	19
8.	投资者结算结果确认	19
9.	请求删除预埋单	19
10.	请求删除预埋撤单	19
11.	请求查询报单	20
12.	请求查询成交	20
13.	请求查询投资者持仓	20
14.	请求查询资金账户	20
15.	请求查询投资者	20
16.	请求查询合约	21
17.	请求查询行情	21
18.	请求查询投资者结算结果	21
19.	请求查询投资者持仓明细	21
20.	请求查询投资者持仓明细(组合)	21
21.	请求查询客户通知	22
22.	请求查询结算信息确认	22
23.	请求查询转帐银行	22
24.	请求查询转帐流水	22
25.	请求查询签约银行	22
26.	请求查询银期签约关系	23
27.	请求查询预埋单	23
28.	请求查询预埋撤单	23
29.	期货发起查询银行余额请求	23
30.	期货发起银行资金转期货请求	23
31.	期货发起期货资金转银行请求	24

32.	订阅行情.....	24
33.	取消行情.....	24
五.	业务定制与回调接口.....	25
1.	定制业务数据.....	25
2.	取消业务数据的定制.....	25
3.	业务回调.....	25

一. platformSvr接口示意图



二. platformSvr接口描述

1. 接口类

platformSvr 通过 CServiceProxy 类向其它模块提供接口。

为便于使用，通过宏 PLATFORM_SVR() 来使用 platformSvr 的接口：

```
#define PLATFORM_SVR() (CServiceProxy::GetSingleInstance())
```

2. 接口分类

platformSvr 的接口分两类。

一类是调用接口，供其它模块调用的接口。

另一类是业务定制与回调接口，其它模块定制感兴趣的业务事件，platformSvr 通过回调函数通知相应模块。

3. 头文件

调用 platformSvr 接口的模块，需包含头文件 `platformSvr/ServiceProxy.hpp`

三. 同步调用接口

1. 平台接口

(1). 平台启动接口 (OK)

接口名称	Start
接口描述	启动 platformSvr 平台
调用方法	PLATFORM_SVR().Start()
返回值	无
备注	OK

(2). 平台关闭接口 (OK)

接口名称	Stop
接口描述	关闭 platformSvr 平台
调用方法	PLATFORM_SVR().Stop()
返回值	无
备注	OK

(3). 判断登录密码是否合法 (OK)

接口名称	IsValidPW
接口描述	判断登录密码是否合法
调用方法	PLATFORM_SVR().IsValidPW()
返回值	Int 1: 合法; 0: 不合法
备注	OK

(4). 设置交易前置机地址 (OK)

接口名称	SetTradeFrontAddr
接口描述	设置交易前置机地址
调用方法	PLATFORM_SVR().SetTradeFrontAddr(const std::string& addr)
返回值	无
备注	OK

(5). 设置行情前置机地址 (OK)

接口名称	SetQuotFrontAddr
接口描述	设置行情前置机地址

调用方法	PLATFORM_SVR().SetQuotFrontAddr()
返回值	无
备注	OK

(6). 设置CTP登录信息 (OK)

接口名称	SetParticipantInfo
接口描述	设置 CTP 登录信息
调用方法	PLATFORM_SVR().SetParticipantInfo(const std::string& BrokerID, const std::string& UserID, const std::string& UserPW)
返回值	无
备注	OK

(7). 登录CTP (OK)

接口名称	Login
接口描述	登录 CTP
调用方法	PLATFORM_SVR().Login(const std::string& InvestorID, const std::string& InvestorPW)
返回值	无
备注	OK

(8). 登出CTP (OK)

接口名称	Logout()
接口描述	登出 CTP
调用方法	PLATFORM_SVR().Logout()
返回值	无
备注	OK

(9). 获取底层服务的当前状态 (OK)

接口名称	GetCurrentStatus
接口描述	获取底层服务的当前状态
调用方法	PLATFORM_SVR().GetCurrentStatus(int* pQuotStatus=NULL, int* pTradeStatus=NULL)
返回值	返回值为 ServiceStatus 型，表示平台是否可用；如果 pQuotStatus 和 pTradeStatus 不为空，则分别返回行情和交易接口的状态
备注	OK

2. 合约信息接口

(1). 获取合约列表 (OK)

接口名称	GetInstrumentList
接口描述	获取合约列表
调用方法	PLATFORM_SVR().GetInstrumentList(std::set<std::string> &outData)
返回值	返回值>=0 表示成功, 返回合约数量, 同时 outData 中返回各合约名称的数组 返回值<0 表示失败
备注	OK

(2). 获取指定合约信息 (OK)

接口名称	GetInstrumentInfo
接口描述	获取指定合约信息
调用方法	PLATFORM_SVR().GetInstrumentInfo(const std::string& InstrumentID, PlatformStru_InstrumentInfo & outData)
返回值	返回 0 表示成功, outData 中返回合约信息 返回-1 表示失败
备注	OK

(3). 设置指定合约信息 (OK)

接口名称	SetInstrumentInfo
接口描述	设置指定合约信息
调用方法	PLATFORM_SVR().SetInstrumentInfo(const std::string& InstrumentID, PlatformStru_InstrumentInfo & outData)
返回值	无
备注	OK

(4). 获取合约品种代码 (OK)

接口名称	GetProductID
接口描述	获取全部合约的品种代码
调用方法	PLATFORM_SVR().GetProductID(std::map<std::string, std::set<std::string> >& outData)
返回值	返回合约数量, outData 中返回全部合约的品种代码
备注	OK

(5). 获取合约产品类型代码 (OK)

接口名称	GetProductClassType
接口描述	获取指定合约的产品类型代码
调用方法	Char PLATFORM_SVR().GetProductClassType(const std::string& InstrumentID)
返回值	失败返回-1, 成功返回合约的产品类型代码
备注	产品类型代码取值如下: #define THOST_FTDC_PC_Futures '1' ///期货 #define THOST_FTDC_PC_Options '2' ///期权 #define THOST_FTDC_PC_Combination '3' ///组合 #define THOST_FTDC_PC_Spot '4' ///即期 #define THOST_FTDC_PC_EFP '5' ///期转现 OK

(6). 获取合约品种类型 (OK)

接口名称	GetProductClassType
接口描述	获取合约的品种类型
调用方法	Char PLATFORM_SVR().GetProductClassType (std::map<std::string>)
返回值	返回品种类型, -1 为失败
备注	OK

(7). 获取合约手续费率 (OK)

接口名称	GetCommissionRate
接口描述	获取合约手续费率
调用方法	PLATFORM_SVR().GetCommissionRate(const std::string& InstrumentID, PlatformStru_InstrumentCommissionRate & outData)
返回值	返回值为 0 表示成功, <0 表示失败 成功时 outData 返回指定合约的手续费率
备注	OK

(8). 获取合约手续费率并触发查询 (OK)

接口名称	GetCommissionRate_TriggerReq
接口描述	获取合约手续费率, 如果没有, 则触发查询
调用方法	PLATFORM_SVR().GetCommissionRate_TriggerReq (const std::string& InstrumentID, PlatformStru_InstrumentCommissionRate & outData)
返回值	返回值为 0 表示成功, <0 表示失败

	成功时 outData 返回指定合约的手续费率
备注	底层自动管理费率查询，上层应用程序不要轻易使用此接口 OK

(9). 设置合约手续费率 (OK)

接口名称	SetCommissionRate
接口描述	设置合约手续费率
调用方法	PLATFORM_SVR().SetCommissionRate(const std::string& InstrumentID, PlatformStru_InstrumentCommissionRate & outData)
返回值	无
备注	OK

(10). 获取合约保证金率 (OK)

接口名称	GetMarginRate
接口描述	获取合约保证金率，如果没有，则触发查询
调用方法	PLATFORM_SVR().GetMarginRate(const std::string& InstrumentID, PlatformStru_InstrumentMarginRate & outData)
返回值	返回值为 0 表示成功，<0 表示失败 成功时 outData 返回指定合约的保证金费率
备注	OK

(11). 获取合约保证金率并触发查询 (OK)

接口名称	GetMarginRate_TriggerReq
接口描述	获取合约保证金率
调用方法	PLATFORM_SVR().GetMarginRate_TriggerReq (const std::string& InstrumentID, PlatformStru_InstrumentMarginRate & outData)
返回值	返回值为 0 表示成功，<0 表示失败 成功时 outData 返回指定合约的保证金费率
备注	底层自动管理费率查询，上层应用程序不要轻易使用此接口 OK

(12). 设置合约保证金率 (OK)

接口名称	SetMarginRate
接口描述	设置合约保证金率
调用方法	PLATFORM_SVR().SetMarginRate(const std::string& InstrumentID, PlatformStru_InstrumentMarginRate & outData)

返回值	无
备注	OK

(13). 获取合约乘数 (OK)

接口名称	GetInstrumentMultiple
接口描述	获取合约乘数
调用方法	PLATFORM_SVR().GetInstrumentMultiple (const std::string& InstrumentID)
返回值	返回合约乘数，失败返回-1
备注	OK

3. 资金接口

(1). 获取交易资金账户信息 (OK)

接口名称	GetTradingAccount
接口描述	获取交易资金账户信息
调用方法	PLATFORM_SVR().GetTradingAccount(DataRspQryTradingAccount& outData)
返回值	返回值为 0 表示成功，<0 表示失败 成功时 outData 返回数据
备注	(OK)

(2). 设置交易资金账户信息 (OK)

接口名称	SetTradingAccount
接口描述	设置交易资金账户信息
调用方法	PLATFORM_SVR().SetTradingAccount(const PlatformStru_TradingAccountInfo & inData)
返回值	无
备注	(OK)

4. 成交记录接口

(1). 获取全部成交单 (OK)

接口名称	GetAllTrades
接口描述	获取全部成交单
调用方法	PLATFORM_SVR().GetAllTrades (std::map<TradeKey, PlatformStru_TradeInfo >& outData)
返回值	返回值为 0 表示成功，<0 表示失败 成功时 outData 返回数据
备注	(OK)

5. 持仓接口

(1). 获取指定合约的持仓明细 (OK)

接口名称	GetPositionDetails
接口描述	获取指定合约的持仓明细
调用方法	<pre>Int GetPositionDetails(std::multimap<std::string, PositionDetailStru> &PositionDetailMap); int GetPositionDetails(const std::string& InstrumentID, std::multimap<std::string, PositionDetailStru> &PositionDetailMap); int GetPositionDetails(std::list< PositionDetailStru > &PositionDetailList)</pre>
返回值	int: 返回值为 0 表示成功, <0 表示失败 成功时 outData 返回数据 或直接返回一个 map
备注	有三种使用方法 (OK)

(2). 获取持仓记录键值列表 (OK)

接口名称	GetPositionKeyList
接口描述	获取持仓记录键值列表
调用方法	<pre>int GetPositionKeyList(std::list<PositionKey> &PositionKeyList)</pre>
返回值	int: 返回值为,返回持仓记录的数量 成功时 outData 返回数据
备注	(OK)

(3). 获取指定合约、方向、投保的持仓数据 (OK)

接口名称	GetPositionData
接口描述	获取指定合约、方向、投保的持仓数据
调用方法	<pre>int GetPositionData(const std::string InstrumentID, const char Direction, const char HedgeFlag, PositionStru& PositionData)</pre>
返回值	成功返回, 失败返回-1 成功时 outData 返回数据
备注	(OK)

6. 报单接口

(1). 获取全部报单 (OK)

接口名称	GetAllOrders
接口描述	获取全部报单
调用方法	PLATFORM_SVR().GetAllOrders(std::map<OrderKey, PlatformStru_OrderInfo >& outData)
返回值	返回值为 0 表示成功, <0 表示失败 成功时 outData 返回数据
备注	(OK)

(2). 获取挂单数量 (OK)

接口名称	GetWaitOrderVolume
接口描述	获取挂单数量
调用方法	PLATFORM_SVR().GetWaitOrderVolume(const std::string &strInstrumentID, const char Direction, const char CloseType, const char HedgeFlag)
返回值	返回挂单数量
备注	(OK)

(3). 获取指定合约指定方向的平仓挂单的平仓量 (OK)

接口名称	GetCloseOrderVolume
接口描述	获取指定合约指定方向的平仓挂单的平仓量
调用方法	PLATFORM_SVR().GetCloseOrderVolume(const std::string &strInstrumentID, const char& Direction, int& CloseVolume, int& CloseTodayVolume, int& CloseYesterdayVolume, const char HedgeFlag)
返回值	返回表示成功, <0 表示失败。成功时 CloseVolume 返回平仓单(含强平和本地强平)的手数, CloseTodayVolume 返回平今单手数, CloseYesterdayVolume 表示平昨单手数
备注	(OK)

7. 行情接口

(1). 获取指定合约行情 (OK)

接口名称	GetQuotInfo
------	-------------

接口描述	获取指定合约行情
调用方法	PLATFORM_SVR().GetQuotInfo(const std::string& InstrumentID, PlatformStru_DepthMarketData & outData)
返回值	返回值为 0 表示成功, <0 表示失败 成功时 outData 返回指定合约的行情数据
备注	(OK)

(2). 获取指定合约的最新价 (OK)

接口名称	GetCurPrice
接口描述	获取指定合约最新价
调用方法	PLATFORM_SVR().GetCurPrice (const std::string& InstrumentID)
返回值	返回值为最新价, 失败时为 0.0
备注	(OK)

(3). 获取指定合约旧行情 (OK)

接口名称	GetOldQuotInfo
接口描述	获取指定合约旧行情
调用方法	PLATFORM_SVR().GetOldQuotInfo (const std::string& InstrumentID, PlatformStru_DepthMarketData & outData)
返回值	返回值为 0 表示成功, <0 表示失败 成功时 outData 返回指定合约的行情数据
备注	(OK)

8. 其它接口

(1). 获取交易所时间

接口名称	GetExchangeTime
接口描述	获取交易所时间
调用方法	PLATFORM_SVR().GetExchangeTime(ExchangeTime& outData)
返回值	返回值为 0 表示成功, <0 表示失败 成功时 outData 返回数据
备注	

(2). 获取当前登入的投资者个人信息

接口名称	GetInvestorInfo
接口描述	获取当前登入的投资者个人信息

调用方法	PLATFORM_SVR().GetInvestorInfo(DataRspQryInvestor& outData)
返回值	返回值为 0 表示成功，<0 表示失败 成功时 outData 返回数据
备注	

(3). 请求查询投资者结算结果响应

接口名称	GetLastSettlementInfo
接口描述	请求查询投资者结算结果响应
调用方法	PLATFORM_SVR().GetLastSettlementInfo(std::string& outData)
返回值	返回值为 0 表示成功，<0 表示失败 成功时 outData 返回数据
备注	

(4). 获取连接服务器和登录时间

接口名称	GetConnectLoginTime
接口描述	获取连接服务器和登录时间
调用方法	PLATFORM_SVR().GetConnectLoginTime(SYSTEMTIME& quotconn, SYSTEMTIME& traderconn, SYSTEMTIME& quotlogin, SYSTEMTIME& traderlogin)
返回值	无
备注	

四. 异步调用接口

1. 用户口令更新请求

接口名称	ReqUserPasswordUpdate
接口描述	用户口令更新请求
调用方法	PLATFORM_SVR().ReqUserPasswordUpdate(PlatformStru_UserPasswordUpdate &pUserPasswordUpdate, int nRequestID=0)
备注	

2. 资金账户口令更新请求

接口名称	ReqTradingAccountPasswordUpdate
接口描述	资金账户口令更新请求
调用方法	PLATFORM_SVR().ReqTradingAccountPasswordUpdate(PlatformStru_TradingAccountPasswordUpdate &pTradingAccountPasswordUpdate, int nRequestID=0)
备注	

3. 报单录入请求

接口名称	ReqOrderInsert
接口描述	报单录入请求
调用方法	PLATFORM_SVR().ReqOrderInsert(PlatformStru_InputOrder &pInputOrder, int nRequestID=0)
备注	

4. 预埋单录入请求

接口名称	ReqParkedOrderInsert
接口描述	预埋单录入请求
调用方法	PLATFORM_SVR().ReqParkedOrderInsert(PlatformStru_ParkedOrder &pParkedOrder, int nRequestID=0)
备注	

5. 预埋撤单录入请求

接口名称	ReqParkedOrderAction
------	----------------------

接口描述	预埋撤单录入请求
调用方法	PLATFORM_SVR().ReqParkedOrderAction(PlatformStru_ParkedOrderAction &pParkedOrderAction, int nRequestID=0)
备注	

6. 报单操作请求

接口名称	ReqOrderAction
接口描述	报单操作请求
调用方法	PLATFORM_SVR().ReqOrderAction(PlatformStru_InputOrderAction &pInputOrderAction, int nRequestID=0)
备注	

7. 查询最大报单数量请求

接口名称	ReqQueryMaxOrderVolume
接口描述	查询最大报单数量请求
调用方法	PLATFORM_SVR().ReqQueryMaxOrderVolume(PlatformStru_QueryMaxOrderVolume &pQueryMaxOrderVolume, int nRequestID=0)
备注	

8. 投资者结算结果确认

接口名称	ReqSettlementInfoConfirm
接口描述	投资者结算结果确认
调用方法	PLATFORM_SVR().ReqSettlementInfoConfirm()
备注	

9. 请求删除预埋单

接口名称	ReqRemoveParkedOrder
接口描述	请求删除预埋单
调用方法	PLATFORM_SVR().ReqRemoveParkedOrder(PlatformStru_RemoveParkedOrder &pRemoveParkedOrder, int nRequestID=0)
备注	

10. 请求删除预埋撤单

接口名称	ReqRemoveParkedOrderAction
接口描述	请求删除预埋撤单
调用方法	PLATFORM_SVR().ReqRemoveParkedOrderAction(PlatformStru_RemoveParkedOrderAction &pRemoveParkedOrderAction, int nRequestID=0)

	PlatformStru_RemoveParkedOrderAction &pRemoveParkedOrderAction, int nRequestID=0)
备注	

11. 请求查询报单

接口名称	ReqQryOrder
接口描述	请求查询报单
调用方法	PLATFORM_SVR().ReqQryOrder(PlatformStru_QryOrder &pQryOrder, int nRequestID=0)
备注	

12. 请求查询成交

接口名称	ReqQryTrade
接口描述	请求查询成交
调用方法	PLATFORM_SVR().ReqQryTrade(PlatformStru_QryTrade &pQryTrade, int nRequestID=0)
备注	

13. 请求查询投资者持仓

接口名称	ReqQryInvestorPosition
接口描述	请求查询投资者持仓
调用方法	PLATFORM_SVR().ReqQryInvestorPosition(PlatformStru_QryInvestorPosition &pQryInvestorPosition, int nRequestID=0)
备注	

14. 请求查询资金账户

接口名称	ReqQryTradingAccount
接口描述	请求查询资金账户
调用方法	PLATFORM_SVR().ReqQryTradingAccount(PlatformStru_QryTradingAccount &pQryTradingAccount, int nRequestID=0)
备注	

15. 请求查询投资者

接口名称	ReqQryInvestor
接口描述	请求查询投资者
调用方法	PLATFORM_SVR().ReqQryInvestor(

	PlatformStru_QryInvestor &pQryInvestor, int nRequestID=0)
备注	

16. 请求查询合约

接口名称	ReqQryInstrument
接口描述	请求查询合约
调用方法	PLATFORM_SVR().ReqQryInstrument(PlatformStru_QryInstrument &pQryInstrument, int nRequestID=0)
备注	

17. 请求查询行情

接口名称	ReqQryDepthMarketData
接口描述	请求查询行情
调用方法	PLATFORM_SVR().ReqQryDepthMarketData(PlatformStru_QryDepthMarketData &pQryDepthMarketData, int nRequestID=0)
备注	

18. 请求查询投资者结算结果

接口名称	ReqQrySettlementInfo
接口描述	请求查询投资者结算结果
调用方法	PLATFORM_SVR().ReqQrySettlementInfo(PlatformStru_QrySettlementInfo &pQrySettlementInfo, int nRequestID=0)
备注	

19. 请求查询投资者持仓明细

接口名称	ReqQryInvestorPositionDetail
接口描述	请求查询投资者持仓明细
调用方法	PLATFORM_SVR().ReqQryInvestorPositionDetail(PlatformStru_QryInvestorPositionDetail &pQryInvestorPositionDetail, int nRequestID=0)
备注	

20. 请求查询投资者持仓明细(组合)

接口名称	ReqQryInvestorPositionCombineDetail
接口描述	请求查询投资者持仓明细（组合）

调用方法	PLATFORM_SVR().ReqQryInvestorPositionCombineDetail(PlatformStru_QryInvestorPositionCombineDetail &pQryInvestorPositionCombineDetail, int nRequestID=0)
备注	

21. 请求查询客户通知

接口名称	ReqQryNotice
接口描述	请求查询客户通知
调用方法	PLATFORM_SVR().ReqQryNotice(PlatformStru_QryNotice &pQryNotice, int nRequestID=0)
备注	

22. 请求查询结算信息确认

接口名称	ReqQrySettlementInfoConfirm
接口描述	请求查询结算信息确认
调用方法	PLATFORM_SVR().ReqQrySettlementInfoConfirm(PlatformStru_QrySettlementInfoConfirm &pQrySettlementInfoConfirm, int nRequestID=0)
备注	

23. 请求查询转帐银行

接口名称	ReqQryTransferBank
接口描述	请求查询转帐银行
调用方法	PLATFORM_SVR().ReqQryTransferBank(PlatformStru_QryTransferBank &pQryTransferBank, int nRequestID=0)
备注	

24. 请求查询转帐流水

接口名称	ReqQryTransferSerial
接口描述	请求查询转帐流水
调用方法	PLATFORM_SVR().ReqQryTransferSerial(PlatformStru_QryTransferSerial &pQryTransferSerial, int nRequestID=0)
备注	

25. 请求查询签约银行

接口名称	ReqQryContractBank
------	--------------------

接口描述	请求查询签约银行
调用方法	PLATFORM_SVR().ReqQryContractBank(PlatformStru_QryContractBank &pQryContractBank, int nRequestID=0)
备注	

26. 请求查询银期签约关系

接口名称	ReqQryAccountregister
接口描述	请求查询银期签约关系
调用方法	PLATFORM_SVR().ReqQryAccountregister(PlatformStru_QryAccountRegister &pQryAccountregister, int nRequestID=0)
备注	

27. 请求查询预埋单

接口名称	ReqQryParkedOrder
接口描述	请求查询预埋单
调用方法	PLATFORM_SVR().ReqQryParkedOrder(PlatformStru_QryParkedOrder &pQryParkedOrder, int nRequestID=0)
备注	

28. 请求查询预埋撤单

接口名称	ReqQryParkedOrderAction
接口描述	请求查询预埋撤单
调用方法	PLATFORM_SVR().ReqQryParkedOrderAction(PlatformStru_QryParkedOrderAction &pQryParkedOrderAction, int nRequestID=0)

29. 期货发起查询银行余额请求

接口名称	ReqQueryBankAccountMoneyByFuture
接口描述	期货发起查询银行余额请求
调用方法	PLATFORM_SVR().ReqQueryBankAccountMoneyByFuture(PlatformStru_ReqQueryAccount &pReqQueryAccount, int nRequestID=0)

30. 期货发起银行资金转期货请求

接口名称	ReqFromBankToFutureByFuture
接口描述	期货发起银行资金转期货请求
调用方法	PLATFORM_SVR().ReqFromBankToFutureByFuture(

	PlatformStru_ReqTransfer &pReqTransfer, int nRequestID=0)
--	--

31. 期货发起期货资金转银行请求

接口名称	ReqFromFutureToBankByFuture
接口描述	期货发起期货资金转银行请求
调用方法	PLATFORM_SVR().ReqFromFutureToBankByFuture(PlatformStru_ReqTransfer &pReqTransfer, int nRequestID=0)

32. 订阅行情

接口名称	SubscribeQuot
接口描述	订阅行情
调用方法	PLATFORM_SVR().SubscribeQuot(const std::string &InstrumentID)

33. 取消行情

接口名称	UnSubscribeQuot
接口描述	取消行情
调用方法	PLATFORM_SVR().UnSubscribeQuot(const std::string &InstrumentID)

五. 业务定制与回调接口

1. 定制业务数据

接口名称	SubscribeBusinessData
接口描述	定制业务数据
调用方法	PLATFORM_SVR().SubscribeBusinessData(BusinessID BID, GUIModuleID GID, GuiCallBackFunc callback, bool sendCurrentInfo=false)
备注	

2. 取消业务数据的定制

接口名称	UnSubscribeBusinessData
接口描述	取消业务数据的定制
调用方法	PLATFORM_SVR().UnSubscribeBusinessData(BusinessID BID, GUIModuleID GID);
备注	

3. 业务回调

接口名称	由 GUI 模块在定制业务时指定
接口描述	1. GUI 模块定制业务时，指定对应的回调函数 2. 回调函数类型为：GuiCallBackFunc typedef int (*GuiCallBackFunc)(const AbstractBusinessData&) 3. 业务事件发生时，platformSvr 回调对应函数，参数为： AbstractBusinessData+事件数据
备注	由于回调函数是在 platformSvr 线程中执行的，在回调函数中，禁止有阻塞的操作，以免影响 platformSvr 线程。