光大证券增值行情系统

API接入开发手册

V1.0

**文档版本**

|  |  |
| --- | --- |
| 文档版本 | V1.0 |
| 发布日期 | 2019-12-17 |

**修改记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日期 | 修改人 | 审核人 | 修改说明 |
| V1.0 | 20191217 | 刘政 | 桂林开 | 创建文档 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[前 言](#_Toc19613_WPSOffice_Level1) [5](#_Toc19613_WPSOffice_Level1)

[系统简介](#_Toc10993_WPSOffice_Level2) [5](#_Toc10993_WPSOffice_Level2)

[读者对象](#_Toc16845_WPSOffice_Level2) [5](#_Toc16845_WPSOffice_Level2)

[1. API接口简介](#_Toc10993_WPSOffice_Level1) [6](#_Toc10993_WPSOffice_Level1)

[1.1库文件](#_Toc5817_WPSOffice_Level2) [6](#_Toc5817_WPSOffice_Level2)

[1.2名词解释](#_Toc8614_WPSOffice_Level2) [6](#_Toc8614_WPSOffice_Level2)

[1.2.1交易市场](#_Toc10993_WPSOffice_Level3) [6](#_Toc10993_WPSOffice_Level3)

[1.2.2行情服务](#_Toc16845_WPSOffice_Level3) [6](#_Toc16845_WPSOffice_Level3)

[1.2.3行情订阅](#_Toc5817_WPSOffice_Level3) [7](#_Toc5817_WPSOffice_Level3)

[2.开发流程](#_Toc16845_WPSOffice_Level1) [8](#_Toc16845_WPSOffice_Level1)

[2.1订阅-主推](#_Toc8511_WPSOffice_Level2) [8](#_Toc8511_WPSOffice_Level2)

[2.2请求-应答](#_Toc18697_WPSOffice_Level2) [10](#_Toc18697_WPSOffice_Level2)

[2.3订阅-内存快照](#_Toc1756_WPSOffice_Level2) [11](#_Toc1756_WPSOffice_Level2)

[2.4小结](#_Toc29444_WPSOffice_Level2) [11](#_Toc29444_WPSOffice_Level2)

[3.开发接口](#_Toc5817_WPSOffice_Level1) [12](#_Toc5817_WPSOffice_Level1)

[3.1 连接和订阅](#_Toc11798_WPSOffice_Level2) [12](#_Toc11798_WPSOffice_Level2)

[3.1.1创建句柄（TDR\_Create）](#_Toc8614_WPSOffice_Level3) [12](#_Toc8614_WPSOffice_Level3)

[3.1.2销毁句柄（TDR\_Destroy）](#_Toc8511_WPSOffice_Level3) [12](#_Toc8511_WPSOffice_Level3)

[3.1.3建立动态连接（TDR\_ConnectByDynamic）](#_Toc18697_WPSOffice_Level3) [12](#_Toc18697_WPSOffice_Level3)

[3.1.4建立静态连接（TDR\_ConnectByStatic）](#_Toc1756_WPSOffice_Level3) [13](#_Toc1756_WPSOffice_Level3)

[3.1.5是否连接服务器（TDR\_IsConnected）](#_Toc29444_WPSOffice_Level3) [14](#_Toc29444_WPSOffice_Level3)

[3.1.6断开服务器连接（TDR\_DisConnect）](#_Toc11798_WPSOffice_Level3) [14](#_Toc11798_WPSOffice_Level3)

[3.1.7设置代理服务器（TDR\_SetProxyServer）](#_Toc27885_WPSOffice_Level3) [14](#_Toc27885_WPSOffice_Level3)

[3.1.8订阅指定代码（TDR\_SubscribeByCode）](#_Toc21555_WPSOffice_Level3) [15](#_Toc21555_WPSOffice_Level3)

[3.1.9取消订阅指定代码（TDR\_UnsubscribeByCode）](#_Toc5935_WPSOffice_Level3) [15](#_Toc5935_WPSOffice_Level3)

[3.1.10订阅指定市场（TDR\_ SubscribeByMarket）](#_Toc30530_WPSOffice_Level3) [15](#_Toc30530_WPSOffice_Level3)

[3.1.11取消订阅指定市场（TDR\_UnsubscribeByMarket）](#_Toc24998_WPSOffice_Level3) [16](#_Toc24998_WPSOffice_Level3)

[3.1.12订阅一组代码（TDR\_SubscribeByGroup）](#_Toc12686_WPSOffice_Level3) [16](#_Toc12686_WPSOffice_Level3)

[3.1.13取消订阅一组代码（TDR\_UnsubscribeByGroup）](#_Toc18381_WPSOffice_Level3) [16](#_Toc18381_WPSOffice_Level3)

[3.1.14取消订阅所有订阅（TDR\_ UnsubscribeAll）](#_Toc6556_WPSOffice_Level3) [17](#_Toc6556_WPSOffice_Level3)

[3.2回调接口](#_Toc27885_WPSOffice_Level2) [17](#_Toc27885_WPSOffice_Level2)

[3.2.1行情数据接收回调（ONRECEIVEDATA）](#_Toc13118_WPSOffice_Level3) [17](#_Toc13118_WPSOffice_Level3)

[3.2.2错误信息接收回调（ONERRORMSG）](#_Toc4894_WPSOffice_Level3) [18](#_Toc4894_WPSOffice_Level3)

[3.3业务函数](#_Toc21555_WPSOffice_Level2) [18](#_Toc21555_WPSOffice_Level2)

[3.3.1从内存中获取行情数据（TDR\_GetMarketData）](#_Toc8024_WPSOffice_Level3) [18](#_Toc8024_WPSOffice_Level3)

[3.3.2向服务器请求单一行情数据（TDR\_ReqMarketData）](#_Toc8328_WPSOffice_Level3) [19](#_Toc8328_WPSOffice_Level3)

[3.3.3向服务器请求一组行情数据（TDR\_ReqGroupMarketData）](#_Toc28432_WPSOffice_Level3) [19](#_Toc28432_WPSOffice_Level3)

[3.3.4获取代码表中第一只证券代码（TDR\_GetCodeTableFirst）](#_Toc544_WPSOffice_Level3) [19](#_Toc544_WPSOffice_Level3)

[3.3.5获取代码表中下一只证券代码（TDR\_GetCodeTableNext）](#_Toc11820_WPSOffice_Level3) [20](#_Toc11820_WPSOffice_Level3)

[3.3.6获取指定市场的服务器日期（TDR\_GetMarketDate）](#_Toc5826_WPSOffice_Level3) [20](#_Toc5826_WPSOffice_Level3)

[4.注意事项](#_Toc8614_WPSOffice_Level1) [21](#_Toc8614_WPSOffice_Level1)

[4.1相关限制](#_Toc5935_WPSOffice_Level2) [21](#_Toc5935_WPSOffice_Level2)

[4.2编程建议](#_Toc30530_WPSOffice_Level2) [21](#_Toc30530_WPSOffice_Level2)

[附录](#_Toc8511_WPSOffice_Level1) [22](#_Toc8511_WPSOffice_Level1)

[附1交易市场nMarketId定义](#_Toc18697_WPSOffice_Level1) [22](#_Toc18697_WPSOffice_Level1)

[附2证券类型uType定义](#_Toc1756_WPSOffice_Level1) [22](#_Toc1756_WPSOffice_Level1)

[附2.1证券类型uType基础定义](#_Toc24998_WPSOffice_Level2) [22](#_Toc24998_WPSOffice_Level2)

[附2.2证券类型uType细分定义](#_Toc383_WPSOffice_Level3) [22](#_Toc383_WPSOffice_Level3)

[附3行情服务nServiceId在各市场定义](#_Toc21435_WPSOffice_Level3) [24](#_Toc21435_WPSOffice_Level3)

[附3.1上交所](#_Toc12686_WPSOffice_Level2) [24](#_Toc12686_WPSOffice_Level2)

[附3.2深交所](#_Toc29444_WPSOffice_Level1) [24](#_Toc29444_WPSOffice_Level1)

[附4行情数据pData结构体](#_Toc11798_WPSOffice_Level1) [24](#_Toc11798_WPSOffice_Level1)

[附4.1上交所](#_Toc18381_WPSOffice_Level2) [24](#_Toc18381_WPSOffice_Level2)

[附4.1.1上交所指数快照](#_Toc26078_WPSOffice_Level3) [24](#_Toc26078_WPSOffice_Level3)

[附4.1.2上交所逐笔成交](#_Toc5211_WPSOffice_Level3) [25](#_Toc5211_WPSOffice_Level3)

[附4.1.3上交所委托队列](#_Toc7143_WPSOffice_Level3) [25](#_Toc7143_WPSOffice_Level3)

[附4.1.4上交所十档快照（Level-2）](#_Toc18220_WPSOffice_Level3) [26](#_Toc18220_WPSOffice_Level3)

[附4.1.5上交所五档行情快照（Level-1）](#_Toc20176_WPSOffice_Level3) [27](#_Toc20176_WPSOffice_Level3)

[附4.2深交所](#_Toc6556_WPSOffice_Level2) [28](#_Toc6556_WPSOffice_Level2)

[附4.2.1深交所指数快照](#_Toc28158_WPSOffice_Level3) [28](#_Toc28158_WPSOffice_Level3)

[附4.2.2深交所逐笔成交](#_Toc14036_WPSOffice_Level3) [28](#_Toc14036_WPSOffice_Level3)

[附4.2.3深交所委托队列](#_Toc20256_WPSOffice_Level3) [28](#_Toc20256_WPSOffice_Level3)

[附4.2.4深交所逐笔委托](#_Toc15362_WPSOffice_Level3) [29](#_Toc15362_WPSOffice_Level3)

[附4.2.5深交所十档行情快照（Level-2）](#_Toc27000_WPSOffice_Level3) [29](#_Toc27000_WPSOffice_Level3)

[附4.2.6深交所五档行情快照（Level-1）](#_Toc1138_WPSOffice_Level3) [30](#_Toc1138_WPSOffice_Level3)

[附4.2.7深圳千档行情](#_Toc27885_WPSOffice_Level1) [31](#_Toc27885_WPSOffice_Level1)

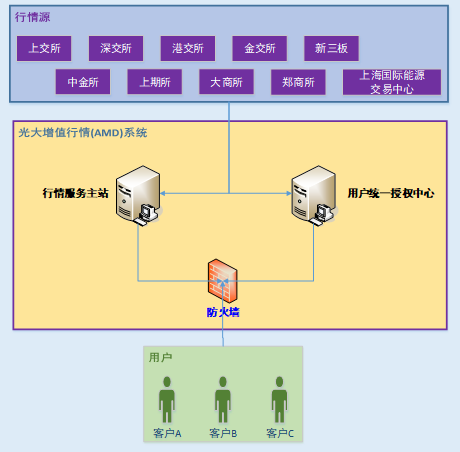
[附5代码状态nStatus字典项定义](#_Toc21555_WPSOffice_Level1) [32](#_Toc21555_WPSOffice_Level1)

[附6错误码（函数返回值）](#_Toc5935_WPSOffice_Level1) [33](#_Toc5935_WPSOffice_Level1)

前 言

系统简介

光大增值行情系统（以下简称AMD系统）是光大证券自主开发的一套高性能行情分发系统。系统通过API接口和个性化客户端等方式为客户提供多市场、多品种的标准和特色行情数据服务。



读者对象

本指南主要适用于以下人员：

* 开发人员。支持C++、C#、Python等语言接入增值行情系统。

# API接口简介

AMD系统提供了C++、C#和python语言的Demo用例，各语言可以通过导入API接口动态链接库文件实现接入增值行情系统。

1.1库文件

Windows系统中，增值行情API接口包含3个动态链接库及对应的静态库文件。

|  |  |
| --- | --- |
| 功能描述 | 库文件 |
| 数据通信 | sipuicom.dll、sipuicom.lib |
| 登录管理 | sip\_mid.dll、sip\_mid.lib |
| 数据接口 | sip\_uid.dll、sip\_ uid.lib |

Linux系统中，增值行情API接口包含3个库文件

|  |  |
| --- | --- |
| 功能描述 | 库文件 |
| 数据通信 | sipuicom.so |
| 登录管理 | sip\_mi.so |
| 数据接口 | sip\_ui.so |

1.2名词解释

本小节介绍AMD系统的一些基本名词概念。

### 1.2.1交易市场

AMD系统目前提供多个交易市场的行情数据服务。在API接口中，各交易市场的定义参见文末附录“附1”。

### 1.2.2行情服务

各个交易市场提供的行情服务类型和行情数据的存储结构均不同。参照附录“附3”。

例如：深圳市场共提供7种行情服务，深圳市场下的每种行情服务对应一个数据类型。

|  |  |  |
| --- | --- | --- |
| nServiceId | 说明 | 说明 |
| 0x00 | 指数 | 行情数据结构体见“附4.2.1” |
| 0x01 | 逐笔成交 | 行情数据结构体见“附4.2.2” |
| 0x02 | 委托队列 | 行情数据结构体见“附4.2.3” |
| 0x03 | 逐笔委托 | 行情数据结构体见“附4.2.4” |
| 0x04 | 十档行情（Level-2） | 行情数据结构体见“附4.2.5” |
| 0x06 | 五档行情（Level-1） | 行情数据结构体见“附4.2.6” |
| 0x08 | 千档行情（基于Level-2） | 行情数据结构体见“附4.2.7” |

其中，“千档行情”是AMD系统提供的特色行情服务。“千档行情”是根据“逐笔委托”和“逐笔成交”计算出来的特色行情数据。“千档行情”更新频率为1秒一帧，优于深交所3秒一帧的快照。

### 1.2.3行情订阅

用户根据业务需求，通过“订阅”或“取消订阅”的方式，来获取行情服务。用户每订阅一个标的一种行情数据类型的服务，程序就会有内存中开辟一块内存空间，用于接收服务端传过来的实时行情数据。

# 2.开发流程

本章节主要介绍三种行情接口的使用方式。

1）订阅-主推。通过回调函数获取行情数据。（推荐方式）

2）请求-应答。通过TDR\_ReqXXX类接口，向服务器请求行情数据。

3）订阅-内存快照。通过TDR\_GetXXX类接口，从本地内存获取最新行情。

2.1订阅-主推



1. 创建句柄。TDR\_Create创建句柄时可以指定系统消息回调函数以及实时数据回调函数。
2. 建立连接，支持以动态方式（TDR\_ConnectByDynamic）或静态方式（TDR\_ConnectByStatic）建立连接。
3. 订阅行情，可以订阅单支代码（TDR\_SubscribeByCode）、一组代码（TDR\_SubscribeByGroup）、指定市场（TDR\_SubscribeByGroup）或全市场代码（TDR\_SubscribeByAll）的指定类型行情。
4. 回调推送，服务端会将客户订阅的行情数据，实时推送到回调函数中。
5. 用户在回调函数中获取到行情数据。
6. 结束时，依次取消订阅（TDR\_UnsubscribeByXXX）、断开连接（TDR\_DisConnect）、销毁句柄（TDR\_Destroy）。

注意

通过API建立连接的方式有两种：静态连接（TDR\_ConnectByStatic）和动态连接（TDR\_ConnectByDynamic）

* 动态连接。同时指定了信息服务器以及行情分发服务器的地址。
* 静态连接。只指定了信息服务器的地址，由系统通过一定的负载均衡算法选择在信息服务器中注册的其中一个行情分发服务器。
* 为了支持负载均衡，建议使用动态连接方式建立连接。

订阅行情可以订阅指定代码、订阅指定市场、订阅一组代码的某一行情服务（参照3.1小节）。

2.2请求-应答



1. 创建句柄。
2. 建立连接。
3. 用户通过调用TDR\_GetMarketData函数，向服务端发起同步请求。行情结果保存在pData指针中。
4. 结束时，依次断开连接，销毁句柄。

2.3订阅-内存快照



1. 创建句柄。
2. 建立连接。
3. 订阅行情。程序会自动开辟内存空间，用于接收从服务端推送过来的实时行情数据。
4. 用户主动调用TDR\_GetMarketData函数，从内存中获取到最新的行情数据。
5. 结束时，依次取消订阅、断开连接、销毁句柄。

2.4小结

1）“订阅-主推”和“订阅-内存快照”使用异步通信，在使用前均需要订阅行情。

2）“请求-应答”使用同步通信。不需要预先订阅行情服务。

# 3.开发接口

关于API的导出函数，参见demo用例的的tdr.h头文件。不再赘述。

## 3.1 连接和订阅

### 3.1.1创建句柄（TDR\_Create）

**函数原型：**

\_\_stdcall long **TDR\_Create**(const char\* slibpath, ONRECEIVEDATA funData, void\* pParamData, ONERRORMSG funError, void\* pParamMsg, bool bSnapshot, bool bAutoReRss);

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| slibpath | 库目录，不含库文件名，如果放在path中，可为NULL |  |
| funData | 数据到达回调函数 | 回调指针，参照3.2.1小节 |
| pParamData | 数据到达回调函数的调用者参数 | 回调参数，参照3.2.1小节 |
| funError | 错误消息回调函数 | 回调指针，参照3.2.2小节 |
| pParamMsg | 错误消息回调函数的调用者参数 | 回调参数，参照3.2.2小节 |
| bSnapshot | 是否使用快照,true/false |  |
| bAutoReRss | 是否使用自动重订阅true/false |  |

**返回：**

创建成功返回句柄值（正整数）。

**用法说明：**

参照DEMO。

### 3.1.2销毁句柄（TDR\_Destroy）

**函数原型：**

\_\_stdcall int TDR\_Destroy (long h);

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | 传入待销毁的句柄 |  |

**返回：**

返回0表示成功。

### 3.1.3建立动态连接（TDR\_ConnectByDynamic）

**函数原型：**

\_\_stdcall int TDR\_ConnectByStatic(long h, const char\* sSvrIP, int nSvrPort, const char\* sUserName, const char\* sPassword, const char\* sInfosvrIP, int nInfoSvrPort, int nLoginMode, int nTimeOutSec)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sIP | IP地址 |  |
| nPort | 端口号 |  |
| nIPType | 内网或外网标识。SIP\_SVR\_LAN:内网SIP\_SVR\_WAN:外网 |  |
| nWanType | 网络类型移动、电信、网通 |  |
| sUserName | 用户名 |  |
| sPassword | 密码 |  |
| nLoginMode | 登录模式 | 普通登录-0; 级联登录-1 |
| nTimeOutSec | 登陆超时(秒) |  |

**返回：**

返回0表示成功。

注意

登录模式（nLoginMode）有两种：普通登录（UI\_LOGIN\_NORMAL）和级联登录（UI\_LOGIN\_UPLINK）。

* 普通登录。UI\_LOGIN\_NORMAL 普通登录后，用户需要再主动订阅后才能获取订阅行情。
* 级联登录。UI\_LOGIN\_UPLINK 级联登录后无需订阅，直接获取账号数据权限内的市场全量行情。

### 3.1.4建立静态连接（TDR\_ConnectByStatic）

**函数原型：**

\_\_stdcall int TDR\_ConnectByDynamic(long h, const char\* sIP, int nPort, int nIPType, int nWanType, const char\* sUserName, const char\* sPassword, int nLoginMode, int nTimeOutSec);

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sSvrIP | ip地址 |  |
| nSvrPort | 端口号 |  |
| sUserName | 用户名 |  |
| sPassword | 密码 |  |
| sInfosvrIP | Info\_SVR。IP地址Info\_SVR上存储市场信息、证券代码信息、用户信息等 |  |
| nInfoSvrPort | Info\_SVR。Port地址 |  |
| nLoginMode | 登录模式 | 普通登录-0; 级联登录-1 |
| nTimeOutSec | 登陆超时(秒) |  |

**返回：**

返回0表示成功。

注意

登录模式（nLoginMode）有两种：普通登录（UI\_LOGIN\_NORMAL）和级联登录（UI\_LOGIN\_UPLINK）。

* 普通登录。UI\_LOGIN\_NORMAL 普通等录需要订阅后才能获取行情。
* 级联登录。UI\_LOGIN\_UPLINK 级联登录后不需要订阅，直接获取行情。

### 3.1.5是否连接服务器（TDR\_IsConnected）

**函数原型：**

\_\_stdcall bool TDR\_IsConnected(long h)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |

**返回：**

返回true表示已经连接服务器，返回false表示未连接服务器

### 3.1.6断开服务器连接（TDR\_DisConnect）

**函数原型：**

\_\_stdcall int TDR\_DisConnect(long h)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |

**返回：**

返回0表示成功断开服务器连接。

### 3.1.7设置代理服务器（TDR\_SetProxyServer）

**函数原型：**

\_\_stdcall int TDR\_SetProxyServer(long h, unsigned short int sutype, const char\* sip, unsigned short int suport, const char\* suser, const char\* spass )

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | UI\_Create创建的句柄 |  |
| sutype | 代理类型 |  |
| sip | 代理服务器IP |  |
| suport | 代理服务器端口 |  |
| suser | 连接代理服务器的用户名 |  |
| spass | 连接代理服务器的密码 |  |

**返回：**

返回0表示成功。

### 3.1.8订阅指定代码（TDR\_SubscribeByCode）

**函数原型：**

\_\_stdcall int TDR\_SubscribeByCode(long h, const char\* sMarket, const char\* sCode, unsigned char ucMode, int nServiceId )

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| sCode | 证券代码 |  |
| ucMode | 订阅模式 | 最新订阅-0; 增量订阅-1; |
| nServiceId | 服务编码 | 参照“附3” |

**返回：**

返回0表示成功。

注意

订阅模式有两种：最新订阅（RSS\_MODE\_NEW）和增量订阅（RSS\_MODE\_INC）

* 最新订阅。服务器只推送最新数据包。即，服务器一方面接收交易所的行情源数据，一方面发送行情数据给客户端。当接收速率大于发送速率时，服务器上可能会发生行情队列的积压。此时，服务器只推送最新的数据包。
* 增量订阅。服务器推送订阅的所有数据包。即，某一时刻服务器的行情队列中存在n条尚未推送到客户端的行情数据，服务器将行情按照时间顺序依次推送给客户端。

### 3.1.9取消订阅指定代码（TDR\_UnsubscribeByCode）

**函数原型：**

\_\_stdcall int TDR\_UnsubscribeByCode(long h, const char\* sMarket, const char\* sCode, int nServiceId)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| sCode | 证券代码 |  |
| nServiceId | 服务编码 | 参照“附3” |

**返回：**

返回0表示成功。

### 3.1.10订阅指定市场（TDR\_ SubscribeByMarket）

**函数原型：**

\_\_stdcall int TDR\_SubscribeByMarket(long h, const char\* sMarket, unsigned char ucMode, int nServiceId)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| ucMode | 订阅模式 | 最新订阅-0; 增量订阅-1;  同3.1.7小节 |
| nServiceId | 服务编码 | 参照“附3” |

**返回：**

返回0表示成功。

### 3.1.11取消订阅指定市场（TDR\_UnsubscribeByMarket）

**函数原型：**

\_\_stdcall int TDR\_UnsubscribeByMarket(long h, const char\* sMarket, int nServiceId)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| nServiceId | 服务编码 | 参照“附3” |

**返回：**

返回0表示成功。

### 3.1.12订阅一组代码（TDR\_SubscribeByGroup）

**函数原型：**

\_\_stdcall int TDR\_SubscribeByGroup(long h, const char\* sMarket, const char\* sCodes, unsigned char ucMode, int nServiceId)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码SH、SZ、SHFE、 | 参照“附1”宏定义 |
| sCodes | 订阅的证券代码多支证券以逗号分隔""比如600000000001000012 |  |
| ucMode | 订阅模式 | 最新订阅-0; 增量订阅-1;  同3.1.7小节 |
| nServiceId | 服务编码 | 参照“附3” |

**返回：**

返回0表示成功。

### 3.1.13取消订阅一组代码（TDR\_UnsubscribeByGroup）

**函数原型：**

\_\_stdcall int TDR\_UnsubscribeByGroup(long h, const char\* sMarkets, const char\* sCodes, int nServiceId)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarkets | 市场代码 | 参照“附1”宏定义 |
| sCodes | 订阅的证券代码多支证券以逗号分隔""比如600000000001000012 |  |
| nServiceId | 服务编码 | 参照“附3” |

**返回：**

返回0表示成功。

### 3.1.14取消订阅所有订阅（TDR\_ UnsubscribeAll）

**函数原型：**

\_\_stdcall int TDR\_UnsubscribeAll(long h)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |

**返回：**

返回0表示成功。

## 3.2回调接口

### 3.2.1行情数据接收回调（ONRECEIVEDATA）

**函数原型：**

typedef void(\_\_stdcall \*ONRECEIVEDATA)(void\* pUserParam, int nDate, int nMarketId, const char\* sCode, const char\* sName, unsigned int uType, int nServiceId, void\* pData, int nLen);

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| pUserParam | 用户自定义参数,由用户调用TDR\_Create时传入 |  |
| nDate | 日期 |  |
| nMarketId | 市场代码，参见tdr.h中对市场进行的宏定义 | 参照“附1” |
| sCode | 证券代码 |  |
| sName | 证券名称 |  |
| uType | 证券类型 |  |
| nServiceId | 服务数据ID，比如行情、逐笔成交等 | 参照“附3” |
| pData | 行情数据内容 |  |
| nLen | 行情数据长度 |  |

**返回：**

返回0表示成功。

**用法说明：**

创建句柄时注册该回调函数。在建立连接后，会向该函数主推行情数据。

### 3.2.2错误信息接收回调（ONERRORMSG）

**函数原型：**

typedef void (\_\_stdcall \*ONERRORMSG)(void\* pUserParam, int nError, int nErrSource, unsigned int uData);

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| pUserParam | 用户自定义参数,由用户调用TDR\_Create时传入 |  |
| nError | 错误编号 |  |
| nErrSource | 错误来源 |  |
| uData | 错误数据 |  |

**返回：**

返回0表示成功。

**用法说明：**

创建句柄时注册该回调函数。在建立连接后，会向该函数主推错误信息。

## 3.3业务函数

### 3.3.1从内存中获取行情数据（TDR\_GetMarketData）

**函数原型：**

\_\_stdcall int TDR\_GetMarketData(long h, const char\* sMarket, const char\* sCode, int nServiceId, void\* pData, int nDataSize)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| sCode | 证券代码 |  |
| nServiceId | 服务ID号 | 参照“附3” |
| pData | 输出快照数据地址 |  |
| nDataSize | 快照数据结构体长度 |  |

**返回：**

返回0表示成功。

**用法说明：**

订阅行情服务后，电脑内存中便会缓存每支代码的行情数据。调用该函数能获取到最新的行情数据。如果尚未订阅行情，该函数取到的数值为空或随机数；取消订阅后，该函数依然可以获取到的数据不是实时行情，而是取消订阅前的内存中最后保留的那一帧行情数据。

### 3.3.2向服务器请求单一行情数据（TDR\_ReqMarketData）

**函数原型：**

\_\_stdcall int TDR\_ReqMarketData(long h, const char\* sMarket, const char\* sCode, int nServiceId, void\* pData, int nDataSize)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| sCode | 证券代码 |  |
| nServiceId | 服务ID号 | 参照“附3” |
| pData | 输出快照数据地址 |  |
| nDataSize | 快照数据结构体长度 |  |

**返回：**

返回0表示成功。

**用法说明：**

使用该函数不需要提前订阅行情，可以直接向服务器请求行情数据。

### 3.3.3向服务器请求一组行情数据（TDR\_ReqGroupMarketData）

**函数原型：**

\_\_stdcall int TDR\_ReqGroupMarketData(long h, const char\* sMarket, const char\* sCode, int nServiceId, void\* pDataArray, int nDataSize, int nCount)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| sCode | 订阅的证券代码多支证券以逗号分隔""比如000001,000002,000012 |  |
| nServiceId | 服务ID号 | 参照“附3” |
| pDataArray | 输出快照数据数组地址 |  |
| nDataSize | 快照数据结构体长度 |  |
| nCount | 快照数量(必须和sCode中证券代码的数量匹配) |  |

**返回：**

返回0表示成功。

### 3.3.4获取代码表中第一只证券代码（TDR\_GetCodeTableFirst）

**函数原型：**

\_\_stdcall bool TDR\_GetCodeTableFirst(long h, const char\* sMarket, tPOSMAP\* pos, Security\_Code\* pcode)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| pos | 代码表中位置 |  |
| pcode | 输出的证券代码信息 |  |

**返回：**

返回true表示成功。

### 3.3.5获取代码表中下一只证券代码（TDR\_GetCodeTableNext）

**函数原型：**

\_\_stdcall bool TDR\_GetCodeTableNext(long h, const char\* sMarket, tPOSMAP\* pPos, Security\_Code\* pcode)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| sMarket | 市场代码 | 参照“附1”宏定义 |
| pPos | 代码表中位置 |  |
| pcode | 输出的证券代码信息 |  |

**返回：**

返回true表示成功。

### 3.3.6获取指定市场的服务器日期（TDR\_GetMarketDate）

**函数原型：**

\_\_stdcall int TDR\_GetMarketDate(long h, int nMarket, int\* nDate)

**输入参数：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数说明 | 备注 |
| h | TDR\_Create返回的句柄 |  |
| nMarket | 市场编码 | 参照“附1” |
| nDate | 返回值为0时，输出交易日期 | YYMMDD |

**返回：**

返回0表示成功。错误码见附录6。

# 4.注意事项

4.1相关限制

* 在普通模式下，用户进行指定标的和数据类型行情订阅，系统对用户账号设置了最大订阅行情标的数不超过50个。如果行情需求大于50个，可以更换级联模式登录。

4.2编程建议

* 回调函数中，不可以有同步的操作，外部调用者切勿在回调函数中做大耗时操作，否则会引起数据发送/接收阻塞。

# 附录

## 附1交易市场nMarketId定义

| 市场编号 | 宏定义 | 交易市场 | 备注 |
| --- | --- | --- | --- |
| 1 | SH | 上海证券交易所 |  |
| 2 | SZ | 深圳证券交易所 |  |
| 3 | CFFEX | 中金所 |  |
| 4 | CZCE | 郑商所 |  |
| 5 | DCE | 大商所 |  |
| 6 | SHFE | 上期所 |  |
| 7 | ZZZS | 中证指数市场 | 暂不支持 |
| 8 | SHOP | 上交所期权市场 |  |
| 9 | HK | 香港市场 | 暂不支持 |
| 11 | SZOP | 深交所期权 | 暂不支持 |
| 12 | SGE | 上海黄金市场 |  |

## 附2证券类型uType定义

### 附2.1证券类型uType基础定义

| 类型名称 | 宏定义 | 类型值 |
| --- | --- | --- |
| 指数 | ID\_BASECLASS\_INDEX | 0x0000 |
| 股票 | ID\_BASECLASS\_SHARES | 0x0100 |
| 基金 | ID\_BASECLASS\_FUND | 0x0200 |
| 债券 & 可转债 | ID\_BASECLASS\_BOND | 0x0300 |
| 回购 | ID\_BASECLASS\_REPO | 0x0400 |
| 权证 | ID\_BASECLASS\_QZ | 0x0500 |
| 期货 | ID\_BASECLASS\_FUTURES | 0x0600 |
| 期权 | ID\_BASECLASS\_OPTION | 0x0700 |
| 其他 | ID\_BASECLASS\_OTHER | 0xFF00 |

### 附2.2证券类型uType细分定义

| 类型名称 | 宏定义 | 类型值 |
| --- | --- | --- |
| 交易所指数 | ID\_BT\_INDEX | 0x0001 |
| A股 | ID\_BT\_SHARES\_A | 0x0101 |
| 中小板股 | ID\_BT\_SHARES\_S | 0x0102 |
| 创业板股 | ID\_BT\_SHARES\_G | 0x0103 |
| B股 | ID\_BT\_SHARES\_B | 0x0104 |
| H股 | ID\_BT\_SHARES\_H | 0x0105 |
| 优先股 | ID\_BT\_SHARES\_OPS | 0x0106 |
| 新三板 | ID\_BT\_SHARES\_NEEQ | 0x0108 |
| 其它股票 | ID\_BT\_SHARES\_OTHER | 0x01FF |
| 未上市开放基金 | ID\_BT\_FUND\_OPEN | 0x0201 |
| 上市开放基金 | ID\_BT\_FUND\_LOF | 0x0202 |
| 交易型开放式指数基金 | ID\_BT\_FUND\_ETF | 0x0203 |
| 封闭式基金 | ID\_BT\_FUND\_CEF | 0x0204 |
| 其它基金 | ID\_BT\_FUND\_OTHER | 0x02FF |
| 政府债券 | ID\_BT\_BOND\_NA | 0x0301 |
| 企业债券 | ID\_BT\_BOND\_CORP | 0x0302 |
| 金融债券 | ID\_BT\_BOND\_FIN | 0x0303 |
| 可转债券 | ID\_BT\_BOND\_CON | 0x0304 |
| 债券预发行 | ID\_BT\_BOND\_WI | 0x0305 |
| 其它债券 | ID\_BT\_BOND\_OTHER | 0x03FF |
| 国债回购 | ID\_BT\_REPO\_NA | 0x0401 |
| 企债回购 | ID\_BT\_REPO\_CORP | 0x0402 |
| 买断式债券回购 | ID\_BT\_REPO\_ORP | 0x0403 |
| 报价回购 | ID\_BT\_REPO\_QRP | 0x0404 |
| 质押回购 | ID\_BT\_REPO\_DW | 0x0405 |
| 其它回购 | ID\_BT\_REPO\_OTHER | 0x04FF |
| 企业发行权证 | ID\_BT\_QZ\_CIW | 0x0501 |
| 备兑权证 | ID\_BT\_QZ\_COV | 0x0502 |
| 其它权证 | ID\_BT\_QZ\_OTHER | 0x05FF |
| 指数期货 | ID\_BT\_FUTURES\_IDX | 0x0601 |
| 商品期货 | ID\_BT\_FUTURES | 0x0602 |
| 股票期货 | ID\_BT\_FUTURES\_SHA | 0x0603 |
| 债券期货 | ID\_BT\_FUTURES\_FBD | 0x0604 |
| 其它期货 | ID\_BT\_FUTURES\_OTHER | 0x06FF |
| 个股期权 | ID\_BT\_OPTION\_STOCK | 0x0701 |
| ETF期权 | ID\_BT\_OPTION\_ETF | 0x0702 |
| 其它期权 | ID\_BT\_OPTION\_OTHER | 0x07FF |

## 附3行情服务nServiceId在各市场定义

### 附3.1上交所

|  |  |  |
| --- | --- | --- |
| nServiceId | 说明 | 说明 |
| 0x00 | 指数 | 行情数据结构体见“附4.1.1” |
| 0x01 | 逐笔成交 | 行情数据结构体见“附4.1.2” |
| 0x02 | 委托队列 | 行情数据结构体见“附4.1.3” |
| 0x04 | 十档行情（Level-2） | 行情数据结构体见“附4.1.4” |
| 0x05 | 五档行情（Level-1） | 行情数据结构体见“附4.1.5” |

### 附3.2深交所

|  |  |  |
| --- | --- | --- |
| nServiceId | 说明 | 说明 |
| 0x00 | 指数 | 行情数据结构体见“附4.2.1” |
| 0x01 | 逐笔成交 | 行情数据结构体见“附4.2.2” |
| 0x02 | 委托队列 | 行情数据结构体见“附4.2.3” |
| 0x03 | 逐笔委托 | 行情数据结构体见“附4.2.4” |
| 0x04 | 十档行情（Level-2） | 行情数据结构体见“附4.2.5” |
| 0x06 | 五档行情（Level-1） | 行情数据结构体见“附4.2.6” |

## 附4行情数据pData结构体

### 附4.1上交所

#### 附4.1.1上交所指数快照

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| nOpenIndex | int | 今开盘指数 |
| nHighIndex | int | 最高指数 |
| nLowIndex | int | 最低指数 |
| nLastIndex | int | 最新指数 |
| iTotalVolume | long long | 参与计算相应指数的交易数量 |
| iTurnover | long long | 参与计算相应指数的成交金额 |
| nPreCloseIndex | int | 前盘指数 |

#### 附4.1.2上交所逐笔成交

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTradeIndex | Int | 成交序号 |
| nTradeChannel | Int | 成交通道 |
| nTradeTime | Int | 成交时间 |
| nTradePrice | Int | 成交价格 |
| iTradeQty | long long | 成交数量 |
| iTradeMoney | long long | 成交金额(元) |
| iTradeBuyNo | long long | 买方订单号 |
| iTradeSellNo | long long | 卖方订单号 |
| cTradeBSflag | char | 内外盘标识 |
| sRes[3] | char | 保留字段1 |

#### 附4.1.3上交所委托队列

委托队列由两个数据组成：队列头和队列体。

队列头保存了队列体的数量，值通常是2(买一价队列和卖一价队列)。队列体是一个变长数组，数组的长度就是队列头的数值。

* 上交所委托队列

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| tHead | T\_OrderQueueHead | 成交序号 |
| tItem[0] | T\_OrderQueueItem | 成交通道 |

* 2.上交所委托队列（T\_OrderQueueHead）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nItem | int | 队列体数量，值通常是2 |

* 3.上交所队列体（T\_OrderQueueItem）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 订单时间(HHMMSSmmmm) |
| nSide | int | 买卖方向('B':Bid |
| nPrice | int | 成交价格 |
| nOrders | int | 订单数量 |
| nABItems | int | 明细个数 |
| nABVolume[200] | int | 订单明细 |

#### 附4.1.4上交所十档快照（Level-2）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| nStatus | int | 状态 |
| uPreClose | unsigned int | 前收盘价 |
| uOpen | unsigned int | 开盘价 |
| uHigh | unsigned int | 最高价 |
| uLow | unsigned int | 最低价 |
| uMatch | unsigned int | 最新价 |
| uAskPrice[10] | unsigned int | 申卖价 |
| uAskVol[10] | unsigned int | 申卖量 |
| uBidPrice[10] | unsigned int | 申买价 |
| uBidVol[10] | unsigned int | 申买量 |
| uNumTrades | unsigned int | 成交笔数 |
| iVolume | long long | 成交总量 |
| iTurnover | long long | 成交总金额 |
| iTotalBidVol | long long | 委托买入总量 |
| iTotalAskVol | long long | 委托卖出总量 |
| uWeightedAvgBidPrice | unsigned int | 加权平均委买价格 |
| uWeightedAvgAskPrice | unsigned int | 加权平均委卖价格 |
| nIOPV | int | IOPV净值估值 |
| nYieldToMaturity | int | 到期收益率 |
| uHighLimited | unsigned int | 涨停价 |
| uLowLimited | unsigned int | 跌停价 |
| sPrefix[4] | char | 证券信息前缀 |
| nSyl1 | int | 市盈率1，2位小数 股票：价格/上年每股利润 债券：每百元应计利息 |
| nSyl2 | int | 市盈率2 ，2位小数 股票：价格/本年每股利润 债券：到期收益率 基金：每百份的IOPV 或净值 权证：溢价率 |
| nSD2 | int | 升跌2（对比上一笔） |
| sTradingPhraseCode[8] | char | 该字段为8位字符串，左起每位表示特定的含义，无定义则填空格。第1位：‘S’表示启动（开市前）时段，‘C’表示集合竞价时段，‘T’表示连续交易时段，‘B’表示休市时段，‘E’表示闭市时段，‘P’表示产品停牌。 |
| nPreIOPV | int | 基金 |

#### 附4.1.5上交所五档行情快照（Level-1）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| nStatus | int | 状态 |
| uPreClose | unsigned int | 前收盘价 |
| uOpen | unsigned int | 开盘价 |
| uHigh | unsigned int | 最高价 |
| uLow | unsigned int | 最低价 |
| uMatch | unsigned int | 最新价 |
| uAskPrice[5] | unsigned int | 申卖价 |
| uAskVol[5] | unsigned int | 申卖量 |
| uBidPrice[5] | unsigned int | 申买价 |
| uBidVol[5] | unsigned int | 申买量 |
| uNumTrades | unsigned int | 成交笔数 |
| iVolume | long long | 成交总量 |
| iTurnover | long long | 成交总金额 |
| uHighLimited | unsigned int | 涨停价 |
| uLowLimited | unsigned int | 跌停价 |
| sTradingPhaseCode[8] | char | 该字段为8位字符串，左起每位表示特定的含义，无定义则填空格。第1位：‘S’表示启动（开市前）时段，‘C’表示集合竞价时段，‘T’表示连续交易时段，‘B’表示休市时段，‘E’表示闭市时段，‘P’表示产品停牌。 |
| nPreIOPV | int | 基金T-1日收盘时刻IOPV |
| nIOPV | int | 基金IOPV |
| uClosePrice | unsigned int | 今日收盘价 |
| uResv1 | unsigned int | 保留字段1 |
| uResv2 | unsigned int | 保留字段2 |
| uResv3 | unsigned int | 保留字段3 |

### 附4.2深交所

#### 附4.2.1深交所指数快照

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| nOpenIndex | int | 今开盘指数 |
| nHighIndex | int | 最高指数 |
| nLowIndex | int | 最低指数 |
| nLastIndex | int | 最新指数 |
| iTotalVolume | long long | 参与计算相应指数的交易数量 |
| iTurnover | long long | 参与计算相应指数的成交金额 |
| nPreCloseIndex | int | 前盘指数 |

#### 附4.2.2深交所逐笔成交

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| uSetno | unsigned int | 证券集代码 |
| uRecno | unsigned int | 成交索引 |
| uBuyOrderRecNo | unsigned int | 买方委托索引 |
| uSellOrderRecNo | unsigned int | 卖方委托索引 |
| uPrice | unsigned int | 成交价格 |
| uTradeQty | unsigned int | 成交数量 |
| cOrderKind | char | 成交类别 |
| sResv1[3] | char | 保留字段1 |
| cFunctionCode | char | 成交代码 |
| sResv2[3] | char | 保留字段2 |
| nTradeTime | int | 成交时间 |

#### 附4.2.3深交所委托队列

委托队列由两个数据组成：队列头和队列体；

队列头保存了数据的个数。队列体是一个变长数组，数组的长度就是队列头的数值。

* 深交所委托队列

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| tHead | T\_OrderQueueHead | 成交序号 |
| tItem[0] | T\_OrderQueueItem | 成交通道 |

* 深交所队列头（T\_OrderQueueHead）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nItem | int | 数据个数 |

* 深交所队列体（T\_OrderQueueItem）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 订单时间(HHMMSSmmmm) |
| nSide | int | 买卖方向('B':Bid |
| nPrice | int | 成交价格 |
| nOrders | int | 订单数量 |
| nABItems | int | 明细个数 |
| nABVolume[200] | int | 订单明细 |

#### 附4.2.4深交所逐笔委托

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| uSetno | unsigned int | 证券集代码 |
| uRecno | unsigned int | 委托编号 |
| uPrice | unsigned int | 委托价格 |
| uOrderQty | unsigned int | 委托数量 |
| cOrderKind | char | 委托类别 1：市价委托，2：限价委托，U：本方最优 |
| sResv1[3] | char | 保留字段1 |
| cFunctionCode | char | 委托方向 B:买入， |
| sResv2[3] | char | 保留字段2 |
| nOrderEntryTime | int | 委托时间 |

#### 附4.2.5深交所十档行情快照（Level-2）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| nStatus | int | 状态，字典项请参照附录5 |
| uPreClose | unsigned int | 前收盘价 |
| uOpen | unsigned int | 开盘价 |
| uHigh | unsigned int | 最高价 |
| uLow | unsigned int | 最低价 |
| uMatch | unsigned int | 最新价 |
| uAskPrice[10] | unsigned int | 申卖价 |
| uAskVol[10] | unsigned int | 申卖量 |
| uBidPrice[10] | unsigned int | 申买价 |
| uBidVol[10] | unsigned int | 申买量 |
| uNumTrades | unsigned int | 成交笔数 |
| iVolume | long long | 成交总量 |
| iTurnover | long long | 成交总金额 |
| iTotalBidVol | long long | 委托买入总量 |
| iTotalAskVol | long long | 委托卖出总量 |
| uWeightedAvgBidPrice | unsigned int | 加权平均委买价格 |
| uWeightedAvgAskPrice | unsigned int | 加权平均委卖价格 |
| nIOPV | int | IOPV净值估值 |
| nYieldToMaturity | int | 到期收益率 |
| uHighLimited | unsigned int | 涨停价 |
| uLowLimited | unsigned int | 跌停价 |
| sPrefix[4] | char | 证券信息前缀 |
| nSyl1 | int | 市盈率1，2位小数 股票：价格/上年每股利润 债券：每百元应计利息 |
| nSyl2 | int | 市盈率2 ，2位小数 股票：价格/本年每股利润 债券：到期收益率 基金：每百份的IOPV 或净值 权证：溢价率 |
| nSD2 | int | 升跌2（对比上一笔） |
| sTradingPhraseCode[8] | char | 该字段为8位字符串，左起每位表示特定的含义，无定义则填空格。第1位：‘S’表示启动（开市前）时段，‘C’表示集合竞价时段，‘T’表示连续交易时段，‘B’表示休市时段，‘E’表示闭市时段，‘P’表示产品停牌。 |
| nPreIOPV | int | 基金 |

#### 附4.2.6深交所五档行情快照（Level-1）

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| nStatus | int | 状态，字典项请参照附录5 |
| uPreClose | unsigned int | 前收盘价 |
| uOpen | unsigned int | 开盘价 |
| uHigh | unsigned int | 最高价 |
| uLow | unsigned int | 最低价 |
| uMatch | unsigned int | 最新价 |
| uAskPrice[5] | unsigned int | 申卖价 |
| uAskVol[5] | unsigned int | 申卖量 |
| uBidPrice[5] | unsigned int | 申买价 |
| uBidVol[5] | unsigned int | 申买量 |
| uNumTrades | unsigned int | 成交笔数 |
| iVolume | long long | 成交总量 |
| iTurnover | long long | 成交总金额 |
| uHighLimited | unsigned int | 涨停价 |
| uLowLimited | unsigned int | 跌停价 |
| sTradingPhaseCode[8] | char | 该字段为8位字符串，左起每位表示特定的含义，无定义则填空格。第1位：‘S’表示启动（开市前）时段，‘C’表示集合竞价时段，‘T’表示连续交易时段，‘B’表示休市时段，‘E’表示闭市时段，‘P’表示产品停牌。 |
| nPreIOPV | int | 基金T-1日收盘时刻IOPV |
| nIOPV | int | 基金IOPV |
| uClosePrice | unsigned int | 今日收盘价 |
| uResv1 | unsigned int | 保留字段1 |
| uResv2 | unsigned int | 保留字段2 |
| uResv3 | unsigned int | 保留字段3 |

#### 附4.2.7深圳千档行情

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 说明 |
| nTime | int | 时间(HHMMSSmmmm) |
| uMatch | unsigned int | 最新价 |
| uAskPrice[10] | unsigned int | 申卖价 |
| uAskVol[10] | unsigned int | 申卖量 |
| uBidPrice[10] | unsigned int | 申买价 |
| uBidVol[10] | unsigned int | 申买量 |
| iTotalBidAmount | long long | 总买量 |
| iTotalAskAmount | long long | 总卖量 |
| iTotalVolume | long long | 总成交量 |

## 附5代码状态nStatus字典项定义

沪、深市场的五档行情和十档行情中，AMD系统的“证券交易状态nStatus”字段字典值及与交易所发布状态值的转换关系如下表所示：

注：AMD系统默认初始化状态和未识别的状态置为数字0。

|  |  |  |
| --- | --- | --- |
| nStatus字典项 | 说明 | 上交所状态名称 |
| Y - ASCII码值89 | 新产品 | .ADD |
| R - ASCII码值82 | 交易间，禁止任何交易活动 | .BETW |
| P - ASCII码值80 | 休市，例如：午餐休市。无撮合和市场内部信息披露 | .BREAK |
| C - ASCII码值67 | 闭市，自动计算闭市价格 | .CLOSE |
| Z - ASCII码值90 | 产品待删除 | .DEL |
| V - ASCII码值86 | 交易结束 | .ENDTR |
| T - ASCII码值84 | 固定价格集合竞价 | .FCALL |
| W - ASCII码值87 | 暂停，除了自有订单和交易的查询之外，任何交易活动都被禁止 | .HALT |
| X - ASCII码值88 | 停牌（SUSP 和HALT 的区别在于在 SUSP 时可以撤单） | .SUSP |
| I - ASCII码值73 | 盘中集合竞价 | .ICALL |
| N - ASCII码值78 | 盘中集合竞价订单簿平衡（OBB） | .IOBB |
| L - ASCII码值76 | 盘中集合竞价 PreOBB | .IPOBB |
| I - ASCII码值73 | 开市集合竞价 | .OCALL |
| M - ASCII码值77 | 开市集合竞价 OBB | .OOBB |
| K - ASCII码值75 | 开市集合竞价订单簿平衡（OBB）前期时段 | .OPOBB |
| S - ASCII码值83 | 非交易支持非交易服务 | .NOTRD |
| U - ASCII码值85 | 盘后处理 | .POSTR |
| F - ASCII码值70 | 盘前处理 | .PRETR |
| E - ASCII码值69 | 启动 | .START |
| 字母O - ASCII码值79 | 连续自动撮合 | .TRADE |
| Q - ASCII码值81 | 连续交易和集合竞价交易的波动性中断（VOLA ） | .VOLA |
| A - ASCII码值65 | 收盘集合竞价 | .CCALL |

|  |  |  |
| --- | --- | --- |
| nStatus字典项 | 说明 | 深交所状态名称 |
| E - ASCII码值69 | 开市前 | S |
| I - ASCII码值73 | 开盘集合竞价 | O |
| 字母O - ASCII码值79 | 连续竞价阶段 | T |
| P - ASCII码值80 | 休市 | B |
| A - ASCII码值65 | 收盘集合竞价 | C |
| C - ASCII码值67 | 已闭市 | E |
| H - ASCII码值72 | 临时停牌 | H |
| U - ASCII码值85 | 盘后交易 | A |
| V - ASCII码值86 | 波动性中断 | V |
| X - ASCII码值88 | 全天停牌 | 第1位值是‘1’ |

## 附6错误码（函数返回值）

|  |  |  |
| --- | --- | --- |
| 错误码 | 宏定义 | 说明 |
| 0x00 | SE\_OK | 成功 |
| 0x01 | SE\_ERR | 未知错误 |
| 0x02 | SE\_ERRHANDLE | 无效句柄 |
| 0x03 | SE\_ARGS | 参数错误 |
| 0x04 | SE\_ERRDATA | 错误的数据或数据无效 |
| 0x05 | SE\_EXP | 错误的表达式 |
| 0x07 | SE\_FRMERR | 报文错误 |
| 0x08 | SE\_UNINITIONAL | 未初始化 |
| 0x0F | SE\_NOTSUPPORT | 不支持的命令 |
| 0x10 | SE\_NOCONNECT | 网络未连接 |
| 0x11 | SE\_TIMEOUT | 网络超时 |
| 0x12 | SE\_CONNECTFULL | 服务器连接满 |
| 0x13 | SE\_NETERR | 网络错误 |
| 0x20 | SE\_OP\_NOUSER | 无此用户 |
| 0x21 | SE\_OP\_NOPOWER | 无此权限 |
| 0x22 | SE\_OP\_NOACTIVE | 帐号不活动 |
| 0x23 | SE\_OP\_PASSWORD | 密码错误 |
| 0x30 | SE\_NOKEY | 主键不存在 |
| 0x31 | SE\_KEYEXIST | 主键已经存在 |
| 0x32 | SE\_NOTDEFITEM | 没有定义的抽象数据类型 |
| 0x40 | SE\_SUBSCRIPT\_FULL | 订阅满 |
| 0x41 | SE\_NOTSUBSCRIPT | 没有订阅 |
| 0x42 | SE\_INFOSVRNOTRUN | info\_svr没有运行或者连接失败! |
| 0x43 | SE\_USERMAXCONNECTED | 用户已达到最大同时登录数，不能再连接 |
| 0x44 | SE\_IPMASK | IP被屏蔽 |
| 0x45 | SE\_VER | 版本错误 |
| 0x46 | SE\_NOTUPLINK | 不容许级联 |
| 0x47 | SE\_PGMCONNECTERR | 加入可靠多播错误 |
| 0x48 | SE\_READBUFOVERFLOW | 接收缓冲溢出 |
| 0x49 | SE\_NOPGMRECVPOWER | 无PGM读取权限 |
| 0x51 | SE\_LIBUI | UI库错误 |
| 0x52 | SE\_LIBMI | MI库错误 |
| 0x53 | SE\_NOSERVER | 没有服务器，动态连接时，当没有sip\_svr运行时会返回这个错误代码. |
| 0x54 | SE\_NOMARKET | 无此市场 |
| 0x55 | SE\_NOCODE | 无此编码，当不能转换为抽象ID时，返回此错误 |
| 0x56 | SE\_USERNAMENOTDIGIT | 用户名非数字格式 |
| 0x57 | SE\_OUTSIDEUSERS | 超出最大用户数,TDR\_SubscribeTrand可能会返回这个错误 |
| 0x58 | SE\_NOSNAPSHOT | 没有快照，是指创建时不使用快照，Get接口会返回这个错误码 |
| 0x59 | SE\_LIBERR | 库错误，一般是没有找到库或者使用了错误的库 |