.对外行情接口文件说明

KsFtQtPub.dll v6 对外行情接口 dll

KsFtQtPub.lib v6 对外行情接口 lib

KsFtQtPub.h   v6 对外行情接口头文件

其他 dll 为依赖的底层 dll

UseQtPubDllDemo.exe   测试程序

2.对外接口定义如下:

```
const int MAX_QUOTA_STATUS_LEN = 2;

const int MAX_DATE_LEN = 9;

const int MAX_EXCHCODE_LEN = 6;

const int MAX_VARI_LEN = 32;


typedef struct ksftquota_pubdata_item_tag

{

    int  contract_id;                    //由交易品种和交割期算出来的 id,对
应：id 号;

    int  upd_serial;                 //行情更新序号，对应：序号
    int  upd_date;                //行情日期(保留)
    int  pre_upd_date;                //行情上次更新日期(保留)
    int  pre_upd_serial;                  //上次更新时的序号(保留)
    char     sys_recv_time[MAX_DATE_LEN];       //行情服务器收到行情的时间，行情服务器唯一维护(保留)

    char     exchCode[MAX_EXCHCODE_LEN];        //交易所代码
    char     varity_code[MAX_VARI_LEN];      //品种代码
    char     deliv_date[MAX_DATE_LEN];       //交割期
    char     chgStatus[MAX_QUOTA_STATUS_LEN]; //对应：状态
                         //1-2bit 表示：买入;3-4bit 表示：卖出;
                         //5-6bit 表示：最新;7-8bit 不用;
                         //00->新行情       01->低于以前的行情
                         //11->高于以前的行情       00->与以前相平

    double openPrice;                //开盘价
    double lastPrice;                //最新价
    double highestPrice;                //最高价
    double lowestPrice;                 //最低价
    int  doneVolume;                 //成交量
    double chgPrice;                 //涨跌
    double upperLimitPrice;             //涨停板
```

```
double lowerLimitPrice;          //跌停板
double hisHighestPrice;          //历史最高价
double hisLowestPrice;                //历史最低价
int  openInterest;           //净持仓
double preSettlePrice;           //昨日结算
double preClosePrice;            //昨日收盘
double settlePrice;          //今日结算
double turnover;             //成交金额

double bidPrice1;            //买入价 1
int  bidVolume1;             //买入量 1
double bidPrice2;            //买入价 2
int  bidVolume2;             //买入量 2
double bidPrice3;            //买入价 3
int  bidVolume3;             //买入量 3
double bidPrice4;            //买入价 4
int  bidVolume4;             //买入量 4
double bidPrice5;            //买入价 5
int  bidVolume5;             //买入量 5

double askPrice1;            //卖出价 1
int  askVolume1;             //卖出量 1
```

```
    double askPrice2;            //卖出价 2
    int  askVolume2;             //卖出量 2
    double askPrice3;            //卖出价 3
    int  askVolume3;             //卖出量 3
    double askPrice4;            //卖出价 4
    int  askVolume4;             //卖出量 4
    double askPrice5;            //卖出价 5
    int  askVolume5;             //卖出量 5
}KSFT_QUOTA_PUBDATA_ITEM;



//功能：启动行情接收
//参数:
//udpPort[in]:接收 udp 行情的广播端口
//errorMsg[out]:错误消息，缓冲区大小必须大于等于 256 个字节
//返回:
//true:成功
//false:失败，可以从 errorMsg 中获取错误原因
//特别说明:在程序启动的时候调用一次就可以了
KSFTQTPUB_API bool WINAPI KSFTHQPUB_Start(unsigned short udpPort, char* errorMsg);
```

//功能：关闭行情接收,并且释放内部资源

KSFTQTPUB_API void WINAPI KSFTHQPUB_Stop();

//功能：获取以 KSFT_QUOTA_PUBDATA_ITEM 数组存放的行情信息,可能一次返回一条或者多条行情

//参数:

//dataBuf[out]:存放 KSFT_QUOTA_PUBDATA_ITEM 格式的行情数组缓冲

//bufSize[in]:KSFT_QUOTA_PUBDATA_ITEM 数组大小(以字节为单位)

//timeOut[in]:超时时间,单位毫秒

//errorMsg[out]:错误消息，缓冲区大小必须大于等于 256 个字节

//返回:

//0:接收超时,没有行情数据

//>0:表示 dataBuf 中存储了 KSFT_QUOTA_PUBDATA_ITEM 结构的行情数据的个数

//<0:调用错误，可以通过 errorMsg 获得错误信息

//特别说明:在 KSFTHQPUB_Start 成功后，不断调用来获取行情信息，一般建议单独开一个线程获取行情信息

KSFTQTPUB_API int WINAPI KSFTHQPUB_GetQuota(unsigned char* dataBuf, int bufSize, int timeOut, char* errorMsg);

3.测试程序使用说明：

如果想在屏幕上显示收到的行情数据

运行 cmd 进入 dos 窗口

运行：UseQtPubDllDemo.exe 32020

其中 32020 是行情广播端口

通过 ctrl+c 终止运行

如果想将收到的行情数据落在文件中

运行 cmd 进入 dos 窗口

运行：UseQtPubDllDemo.exe 32020 >quota_info.txt

其中 32020 是行情广播端口

quota_info.txt 为收到的行情数据

通过 ctrl+c 终止运行

4.示例程序如下：

```
#include "KsFtQtPub.h"

#pragma comment(lib,"D:/sendbuf/v6 对外行情接口/ksftqtpub.lib")

...
```

```
//用来在屏幕上显示行情的函数
void ShowQuotaInfo(KSFT_QUOTA_PUBDATA_ITEM* quotaData, int
quotaCount)
{
    for (int i = 1; i <= quotaCount; ++i)
    {
        cout<<"index["<<i<<"]"<<","
            <<quotaData->contract_id<<","
            <<quotaData->upd_serial<<","
            <<quotaData->sys_recv_time<<","
            <<quotaData->exchCode<<","
            <<quotaData->varity_code<<","
            <<quotaData->deliv_date<<","
            <<quotaData->openPrice<<","
            <<quotaData->lastPrice<<","
            <<quotaData->highestPrice<<","
            <<quotaData->lowestPrice<<","
            <<quotaData->doneVolume<<","
            <<quotaData->chgPrice<<","
            <<quotaData->upperLimitPrice<<","
            <<quotaData->lowerLimitPrice<<","
```

```
                <<quotaData->hisHighestPrice<<","
                <<quotaData->hisLowestPrice<<","
                <<quotaData->openInterest<<","
                <<quotaData->preSettlePrice<<","
                <<quotaData->preClosePrice<<","
                <<quotaData->settlePrice<<","
                <<quotaData->turnover<<","
                <<quotaData->bidPrice1<<","
                <<quotaData->bidVolume1<<","
                <<quotaData->askPrice1<<","
                <<quotaData->askVolume1
                <<endl;
            quotaData++;
        }
}

//调用主程序
int main(int argc, char* argv[])
{
    int udpPort = 32010;
    char errorMsg[256] = "";
    bool procRtn = false;
    //启动行情接收
    procRtn = KSFTHQPUB_Start(udpPort,errorMsg);
    if (!procRtn)
    {
        cout<<errorMsg<<endl;
        return -1;
    }

    int timeOut = 2000;//超时时间 2000ms
    const int MAX_QUOTA_ITEM_COUNT = 50;
    KSFT_QUOTA_PUBDATA_ITEM
quotaData[MAX_QUOTA_ITEM_COUNT];

    while (true)
    {
        //接收行情，可能同时返回多条行情,函数返回值会告诉返回了
几条行情
        int     quotaCount     =     KSFTHQPUB_GetQuota((unsigned
char*)quotaData,
sizeof(KSFT_QUOTA_PUBDATA_ITEM)*MAX_QUOTA_ITEM_CO
UNT,
            timeOut, errorMsg);
        if (quotaCount < 0)
        {
            //接收发生错误了
            cout<<errorMsg<<endl;
        }
        else if (quotaCount > 0)
        {
            //接收到数据
            cout<<"recv quotaCount = "<<quotaCount<<endl;
            ShowQuotaInfo(quotaData, quotaCount);
        }
        else
        {
            //接收数据超时,没有行情数据
            cout<<"no quota data!"<<endl;
        }
    }

    KSFTHQPUB_Stop();
    return 0;
}
```