

# **XQN-10G 行情系统**

**<API 使用说明>**

**<2017 年 4 月>**

## 目录

1. 简介.....	2
2. 接口类库文件说明.....	2
3. 接口开发规范.....	3
3.1 命名空间.....	3
3.2 开发流程.....	3
3.3 DFITCMdSpi 接口 .....	3
4. DFITCMdAPI 使用参考手册.....	3
4.1 DFITCMdApi 接口.....	3
4.1.1 CreateDFITCMdApi 方法.....	3
4.1.2 Init 方法 .....	4
4.2 DFITCMdSpi 接口 .....	4
4.2.1 OnMarketData 方法.....	4
5. 开发样例.....	5

## 1. 简介

XQN-10G 行情系统是一个基于 C++ 的类库，通过使用和扩展类库提供的功能来实现相关的行情接收功能。

本文档的主要内容包括：

- 接口类库文件说明
- 接口开发规范
- 接口参考手册

## 2. 接口类库文件说明

接口类库包含如下文件：

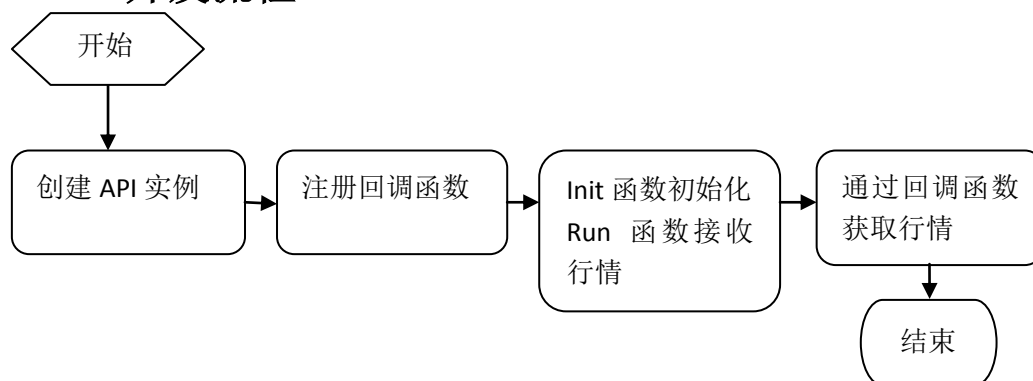
文件名	文件描述
DFITCApiStructXQN.h	定义接口所需的结构体的头文件
DFITCMdApiXQN.h	定义行情接口的头文件
libDFITCMdApiXQN.so	Linux 版本行情 API 的动态链接库文件

## 3. 接口开发规范

### 3.1 命名空间

XQN-10G 行情 API 的命名空间为“**DFITCAPIXQN**”，使用该接口时，请添加命名空间的引用：  
using namespace DFITCAPIXQN;

### 3.2 开发流程



说明:

- 初始化将根据给定的地址和端口接收组播行情数据并发送给用户，用户可以通过回调函数来获取行情数据并执行所需的其它操作。
- 如果在回调函数中有耗时过多的操作,将影响后续回报处理的速度，通常 Spi 应迅速返回，可放入一个缓冲区或队列，以免影响下一次数据的接收。

### 3.3 DFITCMdSpi 接口

- DFITCMdSpi 接口定义了事件通知接口，开发人员必须正确继承并实现 DFITCMdSpi 接口，编写对应的事件处理方法。

## 4. DFITCMdAPI 使用参考手册

### 4.1 DFITCMdApi 接口

#### 4.1.1 CreateDFITCMdApi 方法

产生一个 DFITCMdApi 实例

函数原型:

```
DFITCMDAPIXQN::DFITCmApi * DFITCMDAPIXQN::DFITCmApi::CreateDFITCmApi()
```

**返回值:**

返回一个指向 DFITCmApi 实例的指针。

### 4.1.2 Init 方法

该方法主要实现 socket 连接相关工作，以及绑定 cpu 核心的设置。

**函数原型:**

```
int Init(const char *ip, long port, int cpu_core, const char *local_addr,  
DFITCMDAPIXQN::DFITCmSpi *pSpi)
```

**参数:**

ip: 组播地址 例如"226.100.100.100"

port:端口号 例如"11003"

local\_addr:本地收取行情的网卡 ip 例如"172.16.26.46"

cpu\_core:绑定 cpu 核心号，例如该参数赋值 3，则会将该线程绑定在 cpu 第 3 个核心中

pSpi:类 DFITCmSpi 对象实例

### 4.1.3 Run 方法

该方法需要在 Init 方法之后调用，循环接收行情。

**函数原型:**

```
int Run()
```

## 4.2 DFITCmSpi 接口

DFITCmSpi 实现了事件通知接口，用户要实现此类接口，编写事件处理方法来处理用户感兴趣的事件。

### 4.2.1 OnMarketData 方法

行情消息应答方法，有行情返回时，该方法将被调用。

**函数原型:**

```
void OnMarketData(struct DFITCMarketDataFieldXQN * pMarketDataField)
```

**参数:**

pMarketDataField: 返回行情相关信息:

```
struct APISTRUCT DFITCMarketDataFieldXQN  
{  
    char Market[3];           //市场说明
```

```
char Status;           //字段状态
char Instrument[7];     //合约代码
char UpdateTime[9];     //更新时间
int UpdateMillisec;     //最后更新时间毫秒
double LastPrice;       //最新价
int Volume;             //成交量
double Turnover;        //成交金额
double OpenInterest;    //持仓量
double BidPrice;        //买一价
int BidVolume;          //买一量
double AskPrice;        //卖一价
int AskVolume;          //卖一量
};
#pragma pack()
```

#### 字段说明:

“市场说明”字段用“SQ”来表示上海期货交易所的行情。目前只有“SQ”一个值，以后可能会加入“DL”(大商所)、“ZZ”(郑商所)和“ZJ”(中金所)等值。

“字段状态”是一个1字节的数据，目前只有低两位(bit)有意义：最低1bit为1表示“最新价”、“成交量”、“成交金额”和“持仓量”是最新的数值；最低1bit为0表示“最新价”、“成交量”、“成交金额”和“持仓量”为0(即本笔行情这四个字段没有变化)；倒数第二个1bit为1表示“买一价”、“买一量”、“卖一价”和“卖一量”是最新的数值；倒数第二个1bit为0表示“买一价”、“买一量”、“卖一价”和“卖一量”为0(即本笔行情这四个字段没有变化)。

## 5. 开发样例

具体样例请查看发布包中的/demo/xqnSocket2ApiDemo.cpp文件。该demo文件实现了一个简单的行情接收并打印的功能。主函数中，首先调用CreateDFITCMdApi()方法创建了一个DFITCMdApi的实例，再创建一个回调类，并实现了回调函数OnMarketData。之后再进行Init操作，进行初始化。再调用Run函数，开始接收行情数据，并存入一个队列中。另外一个线程负责从队列中读取内容并打印到日志文件中。

在libs目录下执行make命令，编译该cpp文件产生可执行文件xqnSocket2ApiDemo，按如下命令运行可执行文件：

```
./xqnSocket2ApiDemo -ip 226.100.100.100 -port 11003 -local 172.16.26.46
```

该命令表示，组播地址为226.100.100.100，端口为11003，本地收取行情的网卡ip为172.16.26.46。运行后将得到一个日志文件，其中记录的行情各个字段的信息。

API仅供参考，用户也可以根据自己的需要编写demo程序，实际使用中请在程序中加入组播行情接收内容，尽可能提高接收行情的效率。