

level 2 行情系统

<应用程序接口说明>

<v 1.0.2>

<2013 年 2 月>

文档信息

项目名称	Level 2 行情系统项目
文档名称	<level 2 行情系统应用程序接口说明>
版本号	<v1.0.2>
作者	Level 2 行情系统开发组

文档修订历史

版本	日期	修订内容	修订者
1.0.0	08,28,2012	创建	Level 2 行情系统开发组
1.0.1	10,15,2012	加入心跳内容，加入新报文	Level 2 行情系统开发组
1.0.2	02,18,2013	加入期权字段	Level 2 行情系统开发组

目录

1. 简介.....	4
2. 接口类库文件说明.....	4
3. 系统架构.....	5
3.1 接口与其他系统关系.....	5
3.2 通讯模式.....	5
4. 接口开发规范.....	6
4.1 开发流程.....	6
5. 业务与接口对照表.....	6
6. DFITC_L2::DFITCL2Api 使用说明	6
6.1 CreateDFITCL2Api 方法.....	6
6.2 DestoryDFITCL2Api 方法.....	7
6.3 Connect 方法	7
6.4 ReqUserLogin 方法	7
6.5 ReqUserLogout 方法.....	8
6.6 SubscribeMarketData 方法.....	8
6.7 UnSubscribeMarketData 方法	9
6.8 SubscribeAll 方法.....	9
6.9 ReqChangePassword 方法	9
7. DFITCL2Spi 使用说明.....	9
7.1 OnRspUserLogin 方法.....	10
7.2 OnRspUserLogout 方法	11
7.3 OnRspSubscribeMarketData 方法	11
7.4 OnRspUnSubscribeMarketData 方法.....	11
7.5 OnRspSubscribeAll 方法	12
7.6 OnRspUnSubscribeAll 方法.....	12
7.7 OnBestAndDeep 方法.....	12
7.8 OnTenEntrust 方法.....	12
7.9 OnRealtime 方法.....	13
7.10 OnOrderStatistic 方法.....	13
7.11 OnMarchPrice 方法	13
7.12 OnRspModifyPassword 方法	13
7.13 OnHeartBeatLost 方法.....	14
8. 开发样例.....	14

1. 简介

大连飞创深度行情(level2)应用程序接口（API）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深度行情获取功能，行情包括5级深度委托行情、最佳买卖价位上前10笔分笔委托量、加权平均委买价格、加权平均委卖价格、委托买总量、委托卖总量、实时结算价以及分价位成交量统计等。

本文档的主要内容包括：

- 接口类库文件说明
- 接口的系统架构
- 接口开发规范
- 业务与接口的对照表
- 接口定义

2. 接口类库文件说明

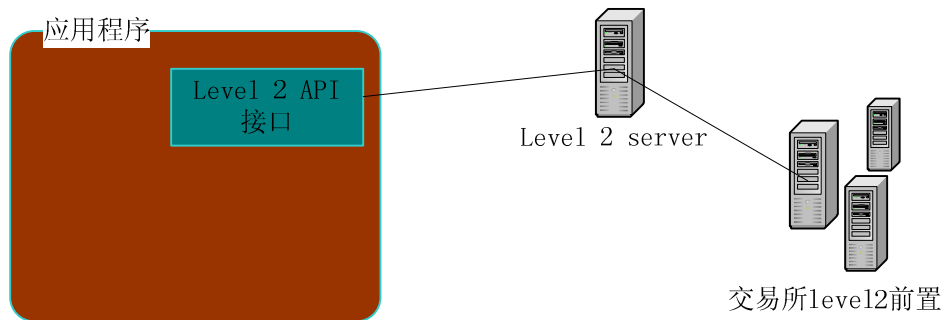
接口类库包含如下文件：

文件名	版本	文件描述
DFITCL2ApiDataType.h	v1.0.2	定义接口所需的接口数据类型的头文件
DFITCL2Api.h	v1.0.2	定义行情接口的头文件
level2Api.dll	v1.0.2	行情接口动态链接库二进制文件
level2Api.lib	v1.0.2	行情接口导入库文件
liblevel2Api.so	v1.0.2	行情接口动态链接库二进制文件(linux)

支持 windows 32（Microsoft VisualC++6.0），linux 32/ 64 集成开发环境。

3. 系统架构

3.1 接口与其他系统关系



说明：

- 本 API 接口用于应用程序获取大连商品交易所 level2 深度行情；
- 应用程序可以是人工交易客户端或者程序化交易平台；
- 交易 API 接口负责与 level2 server 连接。

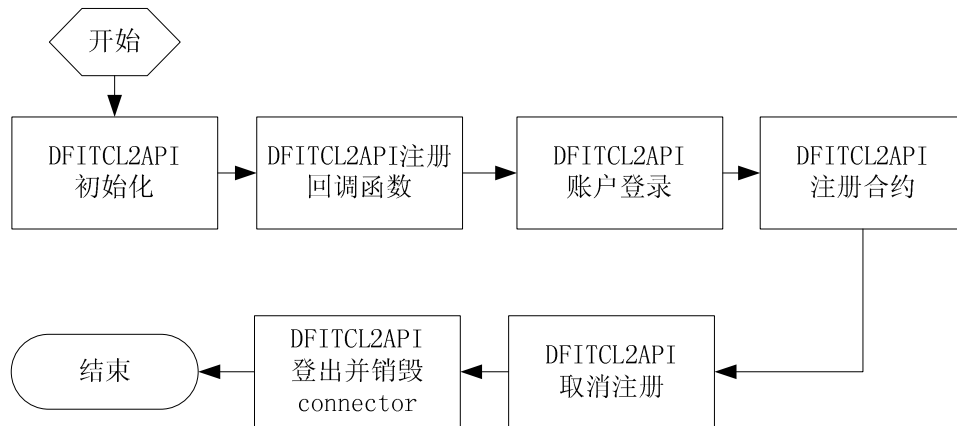
3.2 通讯模式

API 与 level 2 server 的通讯协议是建立在 TCP/UDP 协议上的通讯协议，其中：

- 控制信息只能通过 TCP 传输，此 TCP 连接是长连接；
- 行情信息可以通过 TCP/UDP 两种方式获得，API 中的传输方式选择只针对行情接收方式。

4. 接口开发规范

4.1 开发流程



说明:

使用 windows api 时，编译时需要 level2Api.lib，运行时需要 level2Api.dll

使用 linux api 时，编译和运行时都需要指定 liblevel2Api.so 的路径

在 centos 5.5 32/64 位，举例

g++ -L 来指定编译时 liblevel2Api.so 的路径

export LD_LIBRARY_PATH 指定运行时 liblevel2Api.so 的路径，或者将其加入/usr/lib(/lib64)

5. 业务与接口对照表

6. DFITC_L2::DFITCL2Api 使用说明

6.1 CreateDFITCL2Api 方法

NEW_CONNECTOR ()

该方法产生一个行情 api 实例。

函数原型:

```
static DFITCL2Api *CreateDFITCL2Api();
```

成功: 返回一个指向类型为 DFITC_L2::DFITCL2Api 的对象的指针。

失败: 返回空指针。

6.2 DestoryDFITCL2Api 方法

DELETE_CONNECTOR (p)

该方法销毁一个 api 实例。

函数原型:

```
static int DestoryDFITCL2Api(DFITCL2Api * & L2Api);
```

参数:

L2Api:指向 api 实例的指针。

返回值:

成功: 返回 0

失败: 返回非 0 值

6.3 Connect 方法

该方法进行连接 level 2 server 操作, 注册回调函数集。

函数原型:

```
virtual int Connect(char * pszSvrAddr, DFITCL2Spi * pSpi,unsigned int RecvQuoteType)
```

参数:

pszSvrAddr:level2 服务器地址。

pSpi:回调函数集指针。

RecvQuoteType:接收行情方式, 0 为 udp, 1 为 tcp

返回值:

0: 连接到 level 2 server 成功

非 0: 连接到 level 2 server 失败

6.4 ReqUserLogin 方法

用户发出请求登录。

函数原型:

```
int ReqUserLogin(struct DFITCUserLoginField *pUserLoginData);
```

参数:

pUserLoginData: 指向用户登录请求结构的地址。用户请求登录结构:

```
struct DFITCReqUserLoginField
{
    long lRequestID;           //请求 id
    DFITCAccountIDType accountID; //资金帐号
    DFITCPasswdType passwd;    //密码
}
```

返回值:

- 0: 发送登录请求成功。
- 非 0: 发送登录请求失败。

6.5 ReqUserLogout 方法

用户发出退出请求。

函数原型:

```
int ReqUserLogout( struct DFITCUserLogoutField *pUserLogoutData );
```

参数:

pUserLogoutData:指向用户退出请求结构的地址。用户请求退出结构:

```
struct DFITCUserLogoutField
{
    long lRequestID;           //请求 ID
    DFITCAccountIDType accountID; //资金帐号
};
```

返回值:

- 0: 发送登出请求成功。
- 非 0: 发送登出请求失败。

6.6 SubscribeMarketData 方法

该方法发出订阅某个/某些合约行情请求。

函数原型:

```
virtual int SubscribeMarketData(char *ppInstrumentID[], int nCount)
```

参数:

- ppInstrument:指针数组, 每个指针指向一个合约.
- nCount:合约个数.

6.7 UnSubscribeMarketData 方法

该方法发出退订某个/某些合约行情请求。

函数原型：

```
virtual int UnSubscribeMarketData(char *ppInstrumentID[], int nCount)
```

参数：

同上。

6.8 SubscribeAll 方法

该方法发出订阅全部行情。

函数原型：

```
virtual int SubscribeAll();
```

参数：

无

返回值：

0：成功

非 0：失败

6.9 ReqChangePassword 方法

函数原型：

```
virtual int ReqChangePassword(DFITCPasswdChangeField * pReqUserPasswdChangeField)
```

参数：

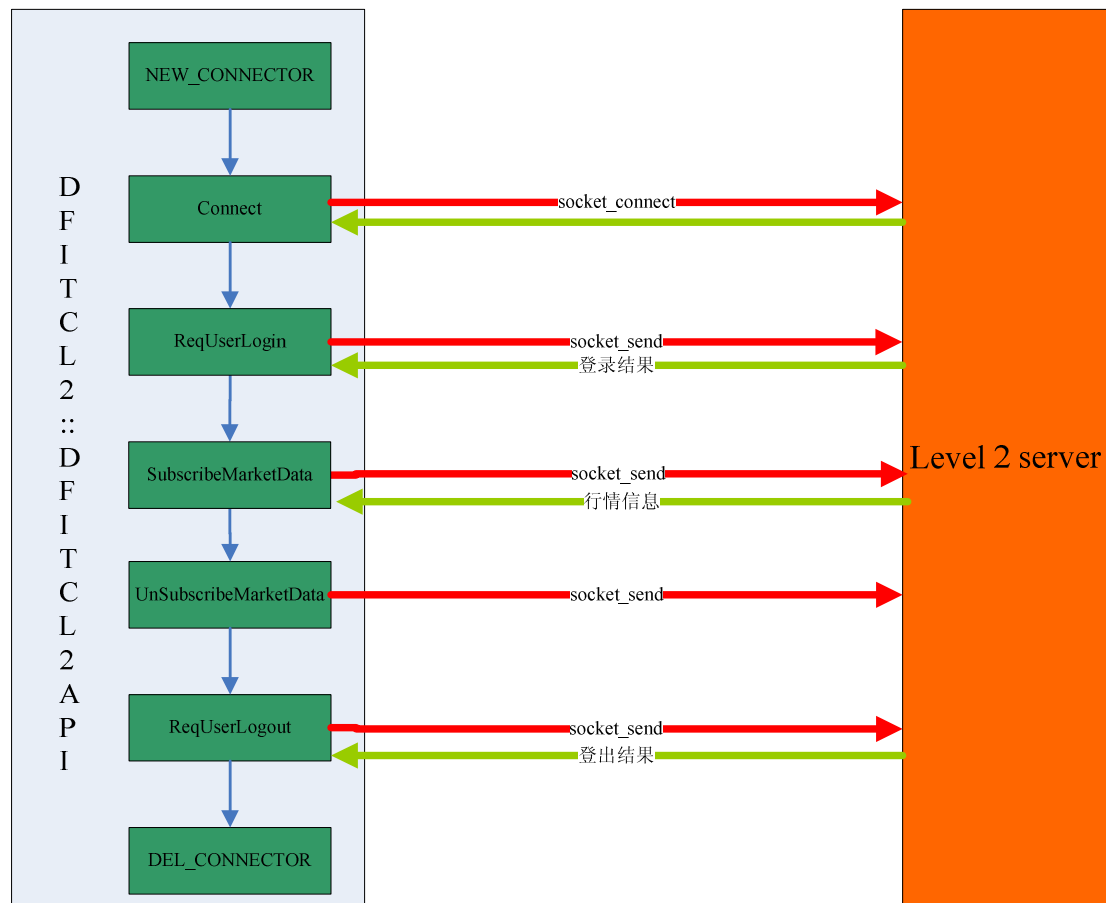
pReqUserPasswdChangeField：指向用户修改密码请求结构的地址。用户请求登录结构：

```
struct DFITCPasswdChangeField
{
    long lRequestID;
    char AccountID[30]; // 帐户名称
    char OldPassword[30]; // 当前密码
    char NewPassword[30]; // 新密码
};
```

7. DFITCL2Spi 使用说明

使用者要新设计一个类继承自 class DFITCL2Spi，并实现其全部回调函数集接口成员函数。下图为应用程序使用 api 简要流程。

注意,如果回调函数耗时过长,需要将从回调函数中取得的行情信息放入一个缓冲区中处理,不然 udp 接收行情有可能产生行情丢失, tcp 接收行情有可能产生行情滞后。



7.1 OnRspUserLogin 方法

当应用程序调用 ReqUserLogin 方法时, OnRspUserLogin 会被调用, 通知用户登录是否成功, 登录错误时, 会返回登录错误原因在 ErrorFiled.ErrMsg 中。

函数原型:

```
virtual void OnRspUserLogin(struct ErrorRtnFiled * pErrorFiled)
```

参数: 参数一, 登录返回信息

struct ErrorRtnFiled

```
{
    long LocalOrderID;
    int  ErrorID; // 0 为登录成功, 非 0 为登录失败
    char ErrorMsg[128]; // 登录失败时的失败原因信息
};
```

7.2 OnRspUserLogout 方法

当应用程序调用 ReqUserLogout 方法时，OnRspUserLogout 会被调用，通知用户登出是否成功，登出错误时，会返回登录错误原因在 ErrorFiled.ErrMsg 中。

通知用户退出状态。

函数原型：

```
virtual void OnRspUserLogout(struct ErrorRtnFiled * pErrorFiled)
```

参数：参数一，登出返回信息

struct ErrorRtnFiled

```
{  
    long LocalOrderID;  
    int  ErrorID; // 0 为登出成功，非 0 为登录失败  
    char ErrorMsg[128]; // 登出失败时的失败原因信息  
};
```

7.3 OnRspSubscribeMarketData 方法

当应用程序调用 SubscribeMarketData 方法时，OnRspSubscribeMarketData 会被调用，通知用户订阅是否成功，订阅错误时，会返回登录错误原因在 ErrorFiled.ErrMsg 中。

函数原型：

```
virtual void OnRspSubscribeMarketData(struct ErrorRtnFiled * pErrorFiled)
```

参数：参数一，订阅返回信息

struct ErrorRtnFiled

```
{  
    long LocalOrderID;  
    int  ErrorID; // 0 为订阅成功，非 0 为订阅失败  
    char ErrorMsg[128]; // 订阅失败时的失败原因信息  
};
```

7.4 OnRspUnSubscribeMarketData 方法

当应用程序调用 UnSubscribeMarketData 方法时，OnRspUnSubscribeMarketData 会被调用，通知用户取消订阅是否成功。

函数原型：

```
virtual void OnRspUnSubscribeMarketData(struct ErrorRtnFiled * pErrorFiled)
```

7.5 OnRspSubscribeAll 方法

当应用程序调用 SubscribeAll 方法时，OnRspSubscribeAll 会被调用，通知用户全部订阅是否成功，全部订阅错误时，会返回全部订阅错误原因在 ErrorFiled.ErrMsg 中。

函数原型：

```
virtual void OnRspSubscribeAll(struct ErrorRtnFiled * pErrorFiled)
```

参数： 参数一，登录返回信息

struct ErrorRtnFiled

```
{  
    long LocalOrderID;  
    int  ErrorID; // 0 为全部订阅成功，非 0 为失败  
    char ErrorMsg[128]; // 全部订阅失败时的失败原因信息  
};
```

7.6 OnRspUnSubscribeAll 方法

当应用程序调用 ReqUnSubscribeAll 方法时，OnRspUnSubscribeAll 会被调用，通知用户取消全部订阅是否成功。

函数原型：

```
virtual void OnRspUnSubscribeAll(struct ErrorRtnFiled * pErrorFiled)
```

7.7 OnBestAndDeep 方法

当 api 接收到最优与深度行情时，该方法会被调用，通知应用程序基本行情与五档深度行情信息。

函数原型：

```
virtual void OnBestAndDeep(MDBestAndDeep * const pQuote)
```

参数： 参数一，基本行情，五档行情

7.8 OnTenEntrust 方法

当 api 接收到最优价十笔委托行情时，该方法会被调用，通知应用程序最优价十笔委托行情信息。

函数原型：

```
virtual void OnTenEntrust(MDTenEntrust * const pQuote)
```

参数：参数一，最优价十笔委托行情

7.9 OnRealtime 方法

当 api 接收到实时结算价行情时，该方法会被调用，通知应用程序实时结算价行情信息。

函数原型：

```
virtual void OnRealtime(MDRealTimePrice * const pQuote)
```

参数：参数一，实时结算价行情

7.10 OnOrderStatistic 方法

当 api 接收到委托统计行情域，加权平均及委托总量行情时，该方法会被调用，通知应用程序该行情信息。

函数原型：

```
virtual void OnOrderStatistic(MDOrderStatistic * const pQuote)
```

参数：参数一，委托统计行情域

7.11 OnMarchPrice 方法

当 api 接收到分价位成交行情时，该方法会被调用，通知应用程序分价位成交行情信息。

函数原型：

```
virtual void OnMarchPrice(MDMarchPriceQty * const pQuote)
```

参数：参数一，分价位成效行情

7.12 OnRspModifyPassword 方法

当应用程序调用 ReqModifyPassword 方法时，OnRspModifyPassword 会被调用，通知用户修改密码是否成功，修改密码失败时，会返回修改密码错误原因在 ErrorFiled.ErrMsg 中。

通知用户退出状态。

函数原型：

```
virtual void OnRspModifyPassword(struct ErrorRtnFiled * pErrorFiled)
```

参数：参数一，登出返回信息

struct ErrorRtnFiled

```
{  
    long LocalOrderID;  
    int  ErrorID; // 0 为修改密码成功，非 0 为登录失败  
    char ErrorMsg[128]; // 修改密码失败时的失败原因信息  
};
```

7.13 OnHeartBeatLost 方法

当应用程序接收不到从 level 2 server 发来的心跳包时，OnHeartBeatLost 会被调用，通知用户心跳丢失

函数原型：

```
virtual void OnHeartBeatLost() {}
```

8. 开发样例

具体样例请查看发布包中 demo 工程。