# Convergence代码分析：

## 程序的主要功能有：

#### 行情数据的接收，压缩向下转发。

#### 同步历史数据和资讯数据。

## 目录文件分析

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Convergence.cpp | main引导 |  |  |
| CvgCpMgr.cpp | 主管理器 |  |  |
| DeliverMgr.cpp | 异步连接点:   1. 网管查询 2. 转发历史，实时，资讯数据 3. 保存行情数据集 4. 带宽统计 |  |  |
| HisDataHandler.cpp | 异步连接点:  1.保存行情数据，二进制网络报文转为本地结构，更新CMemData |  |  |
| MemData.cpp | 行情表管理  订阅管理 |  |  |
| QuotationTbl.cpp | 行情数据表的访问接口 |  |  |
| SubscriberTbl.cpp | 订阅/取消订阅，订阅用户管理 |  |  |
| InfoPublisher.cpp | 1.线程：监控数据库  2.线程:监控本地目录/文件变化  3.构建自定义数据包 |  |  |
| InfoSender.cpp | //传送器:生成历史数据发送报文 |  |  |
| YLFileMapping.cpp |  |  |  |
| XmlInfoIndexFile.cpp | 读写自定义的XML |  |  |
| Markup.cpp | 读写XML5库 |  |  |
| ServiceHandler.cpp | 登陆，注销，订阅等 |  |  |
| SendCommand.cpp | 支持传送数据到行情服务器 |  |  |

## 逻辑层次结构分析

Convergence.cpp

CvgCpMgr.cpp

DeliverMgr.cpp

HisDataHandler.cpp

MemData.cpp

QuotationTbl.cpp

SubscriberTbl.cpp

**InfoPublisher.cpp**（监控文件目录，监控数据库，调用InfoSender的API，进行数据流转）--------用于推送新数据

InfoSender.cpp(二进制报文)

YLFileMapping.cpp（快速读取文件：内存映射的方式读取）

XmlInfoIndexFile.cpp（读取制定格式的XML）

Markup.cpp

ServiceHandler.cpp（异步连接点子线程：提供登陆，注销，订阅，取消订阅等功能）

SendCommand.cpp（按照条件，截取数据，发送）----------用于数据同步

对于异步连接点子线程（如CServiceHandler），功能可以总结为：网络收数据，处理好后，体系内流转。

## 数据监控与发布

**InfoPublisher.cpp**

|  |  |  |
| --- | --- | --- |
|  | 线程1 | 线程2 |
| 监控 | 50毫秒检测一次目录 | 20秒检测一次数据库 |
| 检测项 | 新增文件 | Mtid大于本地文件保存的索引值（InfoMaxId） |
|  | 一次性读取一个XML文件，循环发送item项，其中发送一条，等待100毫秒 | 一次性取数据，发送一条，更新一次最大索引值 |
|  |  |  |

### 推送规则:

#### 监控文件:

设置好index和content两个目录，使用系统函数监控index目录，当新增文件时，打开文件内容，生成体系内的报文，向下转发。

#### 监控数据库:

在执行程序下，保存有一个InfoMaxId文件，里面保存一个int值，是当前数据库索引的最大值。

## 数据同步

SendCommand.cpp

### 发送命令格式：

#### send [nodeid] [markettype] [code] [begin time] [end time] [cc] [cc] ... [cc]

nodeid: 最大长度为六个字符，代表一个行情服务器

markettype: 长度固定为两个字符，表示市场类型，具体请参考MARKETINFOS中marketAdj的定义，

code: 最大长度为六个字符，代表品种代码,

begin time: yyyymmddhhnn, yyyy表示年份, mm表示月份, dd表示天, hh表示小时，nn表示分钟

end time: yyyymmddhhnn，格式与上相同

cc: 无字符数限制，表示分钟线、日线等类型，可以有多种类型可以选择

暂时只支持每次发送一个品种的数据

### 具体值说明:

#### Markettype:

六个值，值参考如下：

const MarketInfo MARKETINFOS[] =

{

{"SH", "国内股票-上证证券"},

{"SZ", "国内股票-深证证券"},

{"HJ", "黄金"},

{"QH", "期货"},

{"WH", "外汇"},

{"WP", "外盘"},

{"GP", "股票"}, // 资讯使用

{"AM", "大众市场"} // 资讯使用

};

键入命令类似于: send 4056 HJ 6300 201809141112 201809141530 TICK

#### Code

行情系统内部定义的市场板块代码，比如6100，6200，6300等。

#### CC

比如:tick,trend,info,day,min15等等

### 举例说明:

例如:send 4056 HJ 6300 201809141112 201809141530 TICK

strFileName = [\\黄金\*-tick.6300](file:///\\黄金*-tick.6300)，会去指定目录寻找“黄金\*-tick.6300”，这里的code是市场编码。

当前对应数据为：黄金-国际黄金-tick.6300

例如:send 4056 HJ 6300 201809141112 201809141530 Trend

strFileName = [\\黄金\*-trend.6300](file:///\\黄金*-trend.6300)，会去指定目录寻找“黄金\*-trend.6300”，这里的code是市场编码。

当前对应数据为：黄金-国际黄金-trend.6300

例如:send 4056 HJ agtd 201809041112 201809141530 day

strFileName = [\\黄金\*\\day\\agtd.day](file:///\\黄金*\\day\\agtd.day)，会去指定目录，遍历含有“黄金“的目录下，da目录下的，agtd.day,即这里的code是合约编码。

例子:send 4056 HJ 6300 201809141112 201809141530 info

不会用命令里面的词来合成搜索路径

对于资讯来讲，HJ对应XML里面的“marketType”，6300对应XML里面的“productCode”，productCode要么为6300，要么为FFFFFF，要么为\*,命令词之间相隔只能有一个空格隔开。

send 100235 HJ 6100 201501062000 201501062100 info

send 100235 HJ 6300 201501062000 201501062100 info

send 100235 QH 6100 201501062000 201501062100 info

send 100235 QH 6300 201501062000 201501062100 info

send 100235 QH \* 201501062000 201501062100 info

### 注意事项：

1. 命令不区分大小写。send必须小写。
2. Tick，trend都是以板块
3. 命令的词与词之间相隔一个space,多了space算错误。
4. 在控制台按照格式输入即可，日志覆盖输入不不影响。

|  |  |  |
| --- | --- | --- |
| 三类数据 |  |  |
| 分时图 | Tick ，trend | 会用命令里面的词来组成搜索路径 |
| 资讯数据 | Info | 不会用命令里面的词来合成搜索路径 |
| 历史周期数据 | Day，week…… | 会用命令里面的词来组成搜索路径 |
|  |  |  |
|  |  |  |

hisdata=D:\YLink\hqServer\hqData\data

tickdata=D:\YLink\hqServer\hqData\dat

infodata=E:\info\index

infocontent=E:\info\content

storeType=0;//0启动文件推送 1：启动数据库推送

infodb\_ODBC= //数据库ODBC