

MDL 客户端使用说明

通联数据股份公司 2023 年 03 月

CONTENTS

1. MDL 简介	1
2. 安装与运行.....	1
2.1. 硬件需求	1
2.2. WINDOWS 环境	1
2.3. LINUX 环境	2
3. 配置文件.....	3
3.1. 监听端口	3
3.2. 行情服务器.....	3
3.3. 回补服务	4
3.4. 文件存储	4
4. 用户获取数据方式	5
4.1. 从文件获取.....	5
4.2. 通过 REDIS 订阅获取	6
4.3. 通过 MDL SDK 获取.....	6
4.4. 通过 WEBSOCKET 接口获取.....	7
5. 注意事项.....	8
5.1. 并发连接数.....	8
5.2. 数据缺失处理	9
5.3. 用户程序快速响应	9
5.4. 级联转发	9

1. MDL 简介

MDL 客户端用于远程接收 MDL 行情服务器发送的实时行情数据，接收到的数据会以 append 方式实时写入到文件，同时以 redis/websocket 协议对外发布，用户可以自由选择从文件或 redis/websocket 中获取数据(详见 3.2)。












2. 安装与运行

2.1. 硬件需求

MDL 客户端对硬件的需求不高，2 核 2G 内存即可流畅运行，Windows 环境建议 4G+内存。网络带宽建议 10Mbps 以上，如果订阅的消息数据量较大(如沪深 L2 行情)，建议 30Mbps 以上带宽。

2.2. Windows 环境

支持的 Windows 版本为 Win7 及以上(64 位)。本软件为绿色安装方式，安装包解压后即可使用，内容如下：

名称	修改日期	类型	大小
 feeder_handler.cfg	2021/3/18 13:35	CFG 文件	4 KB
 feeder_handler.cfg.hiszl2	2021/3/18 13:47	HISZL2 文件	4 KB
 feeder_handler.exe	2021/3/15 14:42	应用程序	17,793 KB
 libmysql.dll	2020/6/10 13:55	应用程序扩展	5,070 KB
 msvcp60.dll	2019/9/26 11:21	应用程序扩展	397 KB
 msvcp100.dll	2019/9/26 11:21	应用程序扩展	412 KB
 msvcp120.dll	2019/9/26 11:21	应用程序扩展	445 KB
 msucr100.dll	2019/9/26 11:21	应用程序扩展	756 KB
 msucr120.dll	2019/9/26 11:21	应用程序扩展	949 KB
 msvcr120.dll	2019/9/26 11:21	应用程序扩展	675 KB
 Readme.txt	2021/3/18 14:06	文本文档	1 KB

其中 feeder_handler.exe 是主程序，feeder_handler.cfg 是配置文件，feeder_handler.log 和 feeder_handler.trace.log 是日志文件。接收到的数据默认会保存在 msg_backup 目录下，每天自动创建一个子文件夹。

有两种运行模式，用户可以任选一种使用：

1. 命令行窗口运行。启动一个 cmd 命令行窗口，cd 到客户端所在目录，运行 `feeder_handler.exe -c` 即可启动程序。
2. 以服务模式运行。命令行窗口中执行 `feeder_handler -i` 命令(需要管理员权限)把本程序注册为系统服务，之后即可在系统的服务管理器中启停 feeder_handler 服务。机器重启后会自动启动服务。

2.3. Linux 环境

支持 64 位 Linux 系统，建议 Ubuntu 16.04+。

程序目录中有以下 3 个文件:

```
feeder_handler // 可执行文件
feeder_handler.cfg // 配置文件
libmdl_api.so // 动态库
```

运行方式与 Windows 下类似:

1. 命令行窗口运行。cd 到程序目录后运行 `./feeder_handler -c`
2. 配置为服务运行。在不同的 OS 版本上配置方式略有不同，以 Ubuntu 16.04 环境为例:

在 `/lib/systemd/system` 目录下创建一个文本文件 `mdl-forward.service`，内容如下:

```
Description=MDL forward

After=network.target

[Service]

User=root

WorkingDirectory=/datayes/mdl/bin #可执行文件所在路径

Type=forking

ExecStart=/datayes/mdl/bin/feeder_handler -d

KillMode=process

TimeoutStopSec=45

Restart=on-failure

StandardOutput=null

StandardError=null

[Install]

WantedBy=multi-user.target
```

启用服务: `systemctl enable mdl-forward.service`

相关启动停止等命令: `systemctl start/stop/restart/status mdl-forward`

3. 后台 Daemon 模式运行。直接不带参数运行，或加-d 参数都可以，如：

```
./feeder_handler 或 ./feeder_handler -d
```

不同于服务模式，此种模式在**机器重启后不会自动启动程序**。

3. 配置文件

配置文件名称为 feeder_handler.cfg，json 格式，可以用文本编辑器打开编辑。默认的配置大部分都不用改，**唯一必须要改的是"UserName"**，见 3.2。

3.1. 监听端口

配置文件中的"Publishers"部分指定了客户端对外暴露的协议和端口，用户可以从这些端口获取到实时数据。默认监听的端口是 9012(tcp 协议)、9379(redis 协议)和 9020(websocket 协议)，**须确保这些端口未被其他应用占用**，否则程序会启动失败，此时也可以打开配置文件把有冲突的端口号改为其他未用端口号即可。

3.2. 行情服务器

配置文件中的"UpStreams"部分包括了行情服务器地址和订阅的消息等设置，其中"Address"为行情服务器地址，格式为 *IP 或域名:port*，支持配置多个地址，中间以分号分隔。如配置了多个地址，在其中一个地址连不通时，程序会尝试连接另一个地址。

如我们在电信和联通线路各提供一个服务地址: mdl01.datayes.com:19010 和 mdl02.datayes.com:19010，因此"Address"可配置为：

```
"Address": "mdl01.datayes.com:19010; mdl02.datayes.com:19010"
```

如果确定自己的网络运营商是电信/联通，可以只保留对应的地址，删除另一个地址，避免跨网络连接导致延迟升高。

"UserName"的值即是用户账号 token，**用户需要把默认值改为自己的 token 才能正常登录服务器。**

"Services"部分是向服务器订阅的消息配置，为方便起见，默认配置已加入了大部分消息，但实际能收到的消息数据取决于服务端的权限设置。

"UpStreams"下的每一段配置代表一个到服务器的 tcp 连接, 其中的"Address"和"UserName"需要同步修改。

"EnableServerSelect"选项决定是否启用自动连接质量检测(2.13.113+版本支持), 默认为 false。如为 true, 且 Address 中配置了多于一个地址, 则自动在连接前对所有地址进行 3 秒钟的连接质量检测, 并取最优地址连接。

"DisconnectBackupTime"选项配置断开备服务器并重连至主服务器的时间, 格式为"时分", 即 500 代表 05:00, 1630 代表 16:30。当"Address"配置了多于一个地址, 并且当前连接的不是第一个地址, 则会在指定的时间断开当前连接, 如果"EnableServerSelect"为 true 就会启动连接质量检测并连到最快的服务器, 否则会重连至第一个服务器。

"SendMacAuth"选项决定是否在登录服务器成功后发送本机 MAC 地址用于身份验证, 默认不启用。当订阅上海 Level 2 行情时, 按交易所要求需要启用。MAC 地址需事先注册到服务器端用于验证。

3.3. 回补服务

客户端在网络波动或盘中重启导致逐笔数据丢失, 可开启回补功能, 回补消息会插入正常推送的行情队列中。如收到消息的序列号小于当前最大序列号, 就可以判断为是回补消息。

- "RebuildService"部分配置回传服务器的地址和消息等, "UpStreams"配置同行情服务器
- 回补服务器地址 "Address": "mdl01.datayes.com:19018;mdl02.datayes.com:19018";
- "UserName" 需要替换成用户自己的 token;
- "SaveChannels" 序列化回传通道到文件, 只在程序正常退出时生效, 重启后可回补逐笔丢失的数据, 默认开启;
- "DelaySeconds" 回补延时时间, 客户端检查到有消息不连续情况, 会在 "DelaySeconds"后向服务器发送回补请求, 需要等待服务端数据加载就绪, 默认 120s。

3.4. 文件存储

"msg_backup"部分是行情数据文件存储设置, 默认的保存目录是 msg_backup, 格式为 csv。如不需要生成文件, 可以删除这部分配置或改名。

DiskSpaceReservedInGB 和 FolderCountReserve 定义了数据清理策略。行情数据会以日为单位自动保存在 msg_backup 下的相应子目录, 每天下午收盘后, 程序会检查磁盘剩余空间和数据子目录个数, 如果剩余磁

盘空间小于 *DiskSpaceReservedInGB* 并且目录个数多于 *FolderCountReserve*, 则会删除较旧的目录直到目录数等于 *FolderCountReserve*。

4. 用户获取数据方式

4.1. 从文件获取

客户端收到的数据会实时以 `append` 模式写入到文件中, 每种消息一个文件, 文件名格式为 `mdl_<ServiceID>_<MessageID>_<Index>.csv`

名称	修改日期	类型	大小
mdl_6_12_0.csv	2019/5/6 17:00	CSV 文件	375,888 KB
mdl_6_19_0.csv	2019/5/6 16:59	CSV 文件	2,621 KB
mdl_6_28_0.csv	2019/5/6 17:00	CSV 文件	5,198,516...
mdl_6_28_1.csv	2019/5/6 17:00	CSV 文件	1,505,065...
mdl_6_28_2.csv	2019/5/6 17:00	CSV 文件	1,469,189...
mdl_6_29_0.csv	2019/5/6 16:59	CSV 文件	235,173 KB
mdl_6_30_0.csv	2019/5/6 16:59	CSV 文件	5,267 KB
mdl_6_31_0.csv	2019/5/6 17:00	CSV 文件	854,338 KB
mdl_6_33_0.csv	2019/5/6 15:30	CSV 文件	2,672,570...
mdl_6_34_0.csv	2019/5/6 13:35	CSV 文件	1 KB
mdl_6_35_0.csv	2019/5/6 14:43	CSV 文件	2 KB
mdl_6_36_0.csv	2019/5/6 15:30	CSV 文件	2,808,490...
mdl_6_40_0.csv	2019/5/6 16:59	CSV 文件	649 KB

ServiceID 和 *MessageID* 指定了消息类型, 见附录。Index 一般为 0, 存储主要的行情数据; 个别类型的消息会同时生成 0、1、2 等多个数据文件, 分别保存此类消息中的不同数据。

文件内容格式为 csv, 各字段间以逗号分隔。

沪 Level 2 快照数据生成两个文件:

mdl_4_4_0.csv - 十档行情主体数据, 每个证券一行

mdl_4_4_1.csv - 50 笔委托数据, 每个证券两行(买卖各一行)

深 Level 2 快照数据生成 3 个文件:

mdl_6_28_0.csv - 十档行情主体数据, 每个证券一行

mdl_6_28_1.csv - 50 笔委卖数据, 每个证券一行

mdl_6_28_2.csv - 50 笔委买数据, 每个证券一行

4.2. 通过 Redis 订阅获取

可以把客户端程序看作一个轻量级的 redis server，通过 9379 端口对外提供服务，支持部分 redis 命令，如 get/mget/subscribe 等。用户可以通过标准的 redis 连接方式连到 9379 端口执行 redis 命令。

使用 `redis subscribe` 命令可以订阅实时推送的行情数据，格式为 `csv(GBK 编码)`，各字段定义可参考相应消息的数据文件，如：

[illegible]

图中 10.24.21.61 即客户端程序所在机器的 ip 地址, redis key 的格式为

mdl.<ServiceID>.<MessageID>.<证券代码>

也支持 `get` 和 `mget` 命令获取数据, 如: `mget mdl.5.2.000001 mdl.3.8.600000`

ServiceID 和 MessageID 请参考附录。

subscribe 命令 channel key 中的证券代码可以为*, 即订阅此类所有消息, 如 mdl.3.4.*代表沪 L1 所有证券。

对于沪深 L2 第一档前 50 笔委托数据, 缺省情况下无法通过 `redis subscribe` 方式获取到。如需获取, 2.13.88 以上版本可以在配置文件的 `publishers` 中 `redis_server` 下新增 `"EnableL2Orders": true` 配置项以启用此功能。

4.3. 通过 MDL SDK 获取

如有需要可以向通联索取 MDL SDK (C++), 在自己的程序中使用 SDK 来连接到客户端 9012 端口, 或直接连到通联行情服务器相应端口订阅数据。

4.4. 通过 WebSocket 接口获取

客户端(v2.13.148+)缺省会在 9020 端口提供 WebSocket 订阅服务, 用户以标准 WebSocket 协议(RFC 6455) 连接到端口, 握手成功后, 可以发送请求订阅行情消息, 请求为 json 格式:

```
{
  "format": "csv",
  "subscribe": [ "3.8.600000", "5.2.000001" ]
}
```

其中:

- *"format"*指定行情数据的格式, 可以是 *"csv"*或 *"json"*
- *"subscribe"*指定要订阅的具体行情类别, 类别格式为: *<大类 ID>.<小类 ID>.<证券代码>*, 如 *3.8.600000* 代表上交所股票(浦发银行); 多个类别也可以拼成以逗号分隔的字符串, 如: *"subscribe": "3.8.600000,5.2.000001"*

此请求在 WebSocket 连接存续过程中可以多次发送, 订阅不同的行情消息, 每次请求的订阅信息会取代之
前请求, 不会自动叠加。

订阅的行情条目数上限取决于配置, 默认 300 条, 超出的条目会被自动截断。

server 端接到并处理订阅请求, 返回成功或失败消息, 格式也是 json:

```
{
  "result": "success",
  "subscribed_messages": "5.2.000001,3.8.600000"
}
```

其中:

- *"result"*是订阅结果, 可能是 *"success"*或 *"failed"*, 当为 *"failed"*时, 会返回一个 *"reason"*属性说明具体原因, 如 *"result": "failed", "reason": "invalid token"*
- *"subscribed_messages"*包含成功订阅的消息类别

订阅成功的行情数据会实时推送到 client, 格式如下:

CSV:

```
{
  "sid":5, "mid":2, "lt":153303662,
  "data":"000001, 平安银行, 15:33:03.000, 2, 12.790, 12.670, 12.710, 12.470, ..."
```

```
}  
  
json:  
{  
  "sid":5, "mid":2, "lt":153703671,  
  "data": {  
    "SecurityID":"000001", "SecurityName":"平安银行", "TurnNum":34911,  
    "LastPrice":12.590, "Volume":44621440,  
    "Turnover":562008050.350, "UpdateTime":153703000,  
    ...  
  }  
}
```

其中:

- *"sid"*是消息大类
- *"mid"*是子类
- *"lt"*是消息的 LocalTime，是通联行情接收机上生成消息时的本地时间，格式为：时分秒毫秒，如 153303662 = 15:33:03.662
- *"data"*包含具体的行情数据，如果以 csv 方式订阅，是一个 csv 字符串；如果以 json 方式订阅，则是一个 json 对象

以上类别信息和 csv 格式等数据，可以参考附录和 csv 文件。

其他

- a) 如果 server 启用了空闲检测机制(默认启用), client 成功握手后,需要在一定时间内发送订阅请求,一般是 10 秒,如果超时未订阅任何行情会被断开连接
- b) client 需要能够响应 server 定时发送的 ping 请求 (浏览器会自动响应)

5. 注意事项

5.1. 并发连接数

服务器对来自同一账号(token)的客户端有数量限制，一个转发账号可用于运行多个客户端，但通联服务器

默认配置的上限仅适用于一个客户端，如需同时运行多个客户端(非级联)，请联系客服调整连接数上限，否则在断线重连时，连接数可能在短时间内超限而被服务器拒绝登录。

有任何配置相关的改动，请重新启动客户端(feeder_handler)使改动生效；2.13.178 版本以上，权限变更无需重启，客户端会自动重连。

5.2. 数据缺失处理

客户端程序建议在盘中交易时段保持运行以避免漏接数据(如已配置为服务则会在后台自动运行)。

如果因网络波动连接断开或盘中重启的情况，期间漏掉的逐笔数据支持自动回补机制(详见 3.3 节)，如服务器宕机或程序异常退出等特殊情况下暂不支持回补，可以联系通联客服，在盘后人工发送完整的数据文件。

5.3. 用户程序快速响应

无论用户选择用 sdk 方式还是 redis/websocket 方式连接到 feeder_handler 程序，都要保证接收数据的线程处理速度足够快，耗时操作建议放到其他线程执行，否则可能导致上游消息积压，超出缓存上限而强制断开用户连接。

出现这种情况时可以在日志中看到"Close slow client" 或"close low speed client"的错误记录。

5.4. 级联转发

当下游应用的连接数较多时，可能会占用较多系统资源导致转发性能下降，这种情况可以通过级联转发的方式，配置和运行第二个或更多的客户端，连接到第一个客户端订阅行情，达到分散转发压力和扩展下游应用数量的目的。具体方式为：

- 1). 复制整个客户端目录到其他机器/目录下
- 2). 修改新目录中的 feeder_handler.cfg 配置文件，把所有 Address 改为第一个客户端的 tcp 发布地址(如都在一台机器就是: 127.0.0.1:9012)，并确保自身的发布端口未被占用
- 3). 删除或改名配置文件中的"msg_backup"部分以禁止生成文件(可选)
- 4). 启动新目录中的 feeder_handler

注意：转发数据到外网是不允许的。