# DEVELOPERS GUIDE

## HKEX Orion Market Data Platform
## Securities Market & Index Datafeed Products (OMD-C)
## Mainland Market Data Hub (MMDH)

Version 2.1
18 May 2026

# Contents

# 1 INTRODUCTION

This document aims to provide guidelines and suggestions for the development of feed handlers to process messages disseminated from the Mainland Market Data Hub ("MMDH") of the HKEX Orion Market Data Platform Securities Market & Index Datafeed Products ("OMD-C"). All information included in this document is presented for reference only. Clients should design and implement their own MMDH feed handlers that are tailored to their business and technical requirements.

The scope of this document covers packet and message processing, retransmission and request based refresh services, aggregate order book management, logon authentication and exception handling procedures.

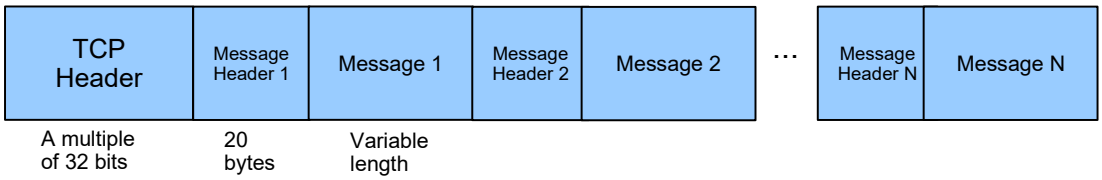
This document provides supplementary information on top of the HKEX OMD-C MMDH interface specification. It shows examples of usage and code snippets to help developers to understand the logic behind the market data disseminated from the OMD-C platform.

Table 1. Acronyms used in this document

| | |
|---|---|
| TCP | Transmission Control Protocol |
| RFS | Request based Refresh Service |
| DR | Disaster Recovery |

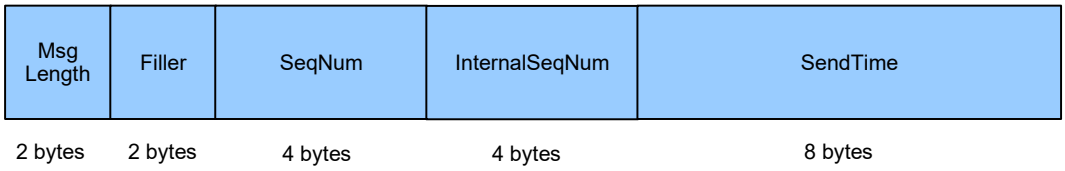
# 2 DATA STRUCTURE

TCP packets are structured into a common packet header followed by zero or more messages. Messages within a packet are laid out sequentially, one after another without any spaces between them.



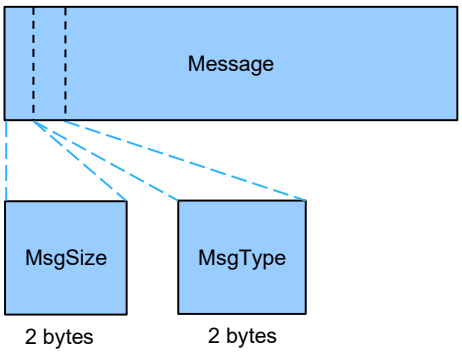

TCP header is padded with trailing zeros to make the header length a multiple of 32 bits. Its minimum size is 20 bytes and maximum of 60 bytes.

The length of a Message Header is 20 bytes. Its structure can be found in the HKEX OMD-C MMDH Binary Interface Specification and it's illustrated below:




MsgLength is the total message length including the message header and the message itself.

Unlike Message Header, the length of each message varies according to message type. Each message starts with a 2-byte message size (MsgSize) and then a 2-byte message type (MsgType).




## 2.1 TCP Header

All packets disseminated from the MMDH feed have a common TCP header. This format is consistent both for messages disseminated by MMDH and client messages sent to the MMDH.

TCP packet consists of a TCP header and a number of messages composed of a 20-byte Message Header and the message content of variable length as described above.

There are no delimiters between TCP header and messages or between messages themselves. One has to use the size of the header and in each individual message to determine the start of each message.

## 2.2 Message Header

It is a 20-byte Message Header. As illustrated in the diagram above, MsgLength is the aggregated size of the Message Header and the message. SeqNum (message sequence number) is consecutive and strictly increasing from 1 per each logon for the client application to identify missing messages even it is not expected under TCP. InternalSeqNum is an internal MMDH sequence number and is only for data recovery. In the event of intraday reconnection, client application can provide a specific InternalSeqNum in Logon (1101) message during reconnection to specify the point to restart data transmission.

When clients send messages to MMDH, simply put zero for SeqNum and InternalSeqNum in the Message Header unless it is a Logon message for recovering messages from a certain point.

## 2.3 Heartbeats

Heartbeats consist of a 20-byte Message Header with sequence number set to the previous message only. They do not increment the sequence number. The Heartbeat message syntax is identical across MMDH services.

## 2.4 Message

The format of each message varies according to its type. However, regardless of the message type, all messages start with a two-byte message size followed by a two-byte message type.

**MsgSize** Binary integer representing the length of the message (including this field)
**MsgType** Binary integer representing the type of message. Please refer to the HKEX OMD-C MMDH Interface Specification for the full list of message types

The data follows immediately the MsgType within a message.

## 2.5 Message Formats

Please refer to the HKEX OMD-C MMDH Interface Specification for the available message types and their message structures.

# 3 ENDIAN

Almost all binary values are in Little Endian byte order, which means the first byte (lowest address) is the least significant one. The only exception is in Send Key (1105) message, where the *Prime*, *Generator*, *PrimeOrderSubgroup* and *OMDPublicKey* fields are in Big Endian byte order.

In C/C++, one solution is to create a structure containing all the fields from the message header and cast the pointer to a packet, to a pointer to such a structure. For instance:

```
struct MsgHeader
{
        unsigned long mMsgLen;
        char mFiller1;
        char mFiller2;
        unsigned long mSeqNum;
        unsigned long mInternalSeqNum;
        unsigned long long mSendTime;
};
```

Assume the packet is passed as a pointer to const unsigned char, which could look like this:

```
struct MsgHeader* hdr = static_cast<MsgHeader*> (packetPtr);
```

One packet may contain multiple messages. Clients should locate the beginning of each message based on the message length and process each message separately.

# 4 MESSAGE PROCESSING

Each client TCP connection session to MMDH works independently. On connection, MMDH sends a SendKey message (1105) to the client. The key is used to encrypt password or the new password fields during logon.

A session is dedicated to one client per business day. During the day for each logon, message sequence number starts from 1 and strictly increases, therefore unique per each logon session.

## 4.1 Start of Day

OMD-C MMDH is normally ready for a new business day at 6:00am. However, the Exchange has the right to adjust the system ready time in the future according to the different trading situations.

**Client starts at MMDH startup time**

- Client connects to MMDH server (Please refer to **Appendix C** for key exchange diagram)
- MMDH sends a Send Key message (1105), which contains the Diffie-Hellman parameters, a concatenated public key and random Initialisation Vector "IV", to the client
- Client generates the client public key and encrypted password
- Client sends Logon message (1101) to MMDH
- MMDH replies to the client through a Logon Response message (1102)
- MMDH sends Reference Data messages below
  - Market Definition (10)
  - Security Definition (11)
  - Liquidity Provider (13)
  - Currency Rate (14)

**Client starts after OMD-C startup time and data is available in cache**

- Similar to the events described above. Client logons successfully, but MMDH returns SessionStatus =0 (Session Active) through the Logon Response message (1102)
- MMDH sends data within cache
- Normal data flow continues

**Client starts after OMD-C startup time and data is out of range (refresh for latest image required)**

- Client logons successfully, but MMDH returns SessionStatus=101 (Session Active – refresh required) through the Logon Response message (1102)
- Client sends a Refresh Request message (1201) to MMDH
- MMDH responds to the refresh request through Refresh Response message (1202)
- MMDH re-sends all essential messages for client to reconstruct the latest market image
- MMDH sends a Refresh Complete message (203) to signal the end the refresh cycle and to provide the InternalSeqNum of the last message from the Refresh
- Normal data flow continues

## 4.2 Normal Transmission

Normal message transmission is expected between market opens for trading and market is closed for the day. Heartbeats are sent regularly between MMDH Server and client.

When it is a public holiday in Hong Kong only, MMDH will continue to transmit messages for indices of which the dissemination timetable include Hong Kong holiday.

Reliable transmission is guaranteed by the TCP/IP protocol and gaps in transmission is not expected to happen as long as the TCP connection is intact.

### 4.2.1 Process Data Message

Message carrying information about a particular instrument has a Security Code field. This field is the unique instrument identifier. The Security name, ISIN code, etc. are only carried in the Security Definition (11) message, so client applications should refer to Security Definition based on the Security Code for an instrument's attributes when needed. The Security Code, once allocated for an instrument, does not change.

#### 4.2.1.1 Process Fields Carrying Decimal Value

Message with fields carrying decimal value are usually defined with a fixed decimal place value. For example, PreviousClosingPrice in Security Definition (11) message is defined with 3 implied decimal places. There is exception to this, however, for some of the fields carrying decimal value are defined with variable decimal place value which is provided in a separate field in the same message. For example, CallPrice in Security Definition (11) messages is defined with variable decimal place value which is provided in DecimalsInCallPrice in the same message. Clients need to make use of value defined in DecimalsInCallPrice to obtain the actual CallPrice. For example, CallPrice = 1234567 with DecimalsInCallPrice = 5 will imply a Call Price of $12.34567 for the CBBC security.

### 4.2.2 Process Control Message (Heartbeats)

Heartbeats are disseminated at regular time intervals. Clients can use heartbeats to check if the feed is alive. If there is no heartbeat for longer than a configurable time (please refer to MMDH Interface Specification Section 3.3.1 for unicast heartbeat interval currently set in MMDH), then it indicates that there is an outage at the exchange side.

Note that MMDH sends heartbeats only when there is no market data being disseminated. When there is market data on the line, no heartbeat will be sent.

Heartbeats consist of a message header with length set to the message header length and do not increment the sequence number. SeqNum in packet header is set to the sequence number of the previous message.

Clients should repeatedly send a heartbeat message to the Server at all times to maintain the TCP connection. The heartbeat should be periodic – as defined by the HeartBtInterval field received in the Logon Response (1102) message. Clients should set SeqNum and InternalSeqNum to zero and SendTime to current time.

When receiving heartbeat packet, clients should ignore this packet in gap detection. Otherwise, clients may fail to detect the actual message gap.

Table Gap Detection Example

| Time | Packet sent from OMD | Packet received by Client | Remark |
|---|---|---|---|
| T1 | 101 | 101 | |
| T2 | 102 | 102 | |
| T3 | 103 | | Packet with seqNum 103 is lost |
| T4 | 103 (Heartbeat) | 103 (Heartbeat) | If client receives heartbeat message but cannot find the corresponding packet with same sequence number, it should be a message gap and client should recover the lost message |
| T5 | 104 | 104 | |
| T6 | 105 | 105 | |
| T7 | 106 | 106 | |

## 4.3 Recovery

For large scale data loss, client can reconnect to MMDH providing the InternalSeqNum to indicate the last message received. If MMDH responds with a Logon Response message (1102) with the SessionStatus=101, then client should recover by using Refresh Request.

### 4.3.1 Refresh Service

The MMDH feed provides a refresh facility, which allows clients to start intraday or recover from significant data loss.

Refresh provides a snapshot of the market on request basis. Not all the messages available from the live feed can be recovered from the refresh service which serves the purpose of providing sufficient information for reconstructing an up-to-date image of the market.

When a logon response with SessionStatus=101 is received, client should submit a Refresh Request message (1201) to MMDH. If accepted, MMDH sends the refresh data to the client via the same TCP connection. At the end of a refresh cycle, a Refresh Complete (203) message will arrive. Client now has the latest market information thereupon continue to receive normal data flow.

### 4.3.2 Refresh Snapshot

Please refer to the HKEX OMD-C MMDH Interface Specification for the coverage of refresh messages.

## 4.4 Password Management

Client needs to change the logon password when Logon Response message with SessionStatus set to 2 (“Session password due to expire”) received. Client should disconnect the current active session and start a new connection to MMDH Server, send Logon message with EncryptedNewPasswordLen and EncryptedNewPassword parameters to change the password.

Client is recommended to perform this password change action outside trading hours. Any additional Logon message after successful Logon will force MMDH Server to logout active session and cause interruption to receiving market data.

# 5 RACE CONDITIONS

The real-time order/trade data and reference data are disseminated via separate channels to MMDH, so users need to be aware of the possibility of race conditions.

For example, the Trading Session Status (20) message marking the trading session as halted, but real time data for the same market may continue to arrive for a short time afterwards.

# 6 AGGREGATE ORDER BOOK MANAGEMENT

Book updates are sent by OMD-C via Aggregate Order Book (53) messages. Each message may contain any combination of new, changed or deleted entries for a book or clear the whole book. The nature of an entry is defined by its UpdateAction.

Table 11. Actions on Aggregate Order Book Messages

| Action | Description |
|---|---|
| New | Create/Insert a new price level |
| Delete | Remove a price level |
| Change | Update aggregate quantity at a price level |
| Clear | Clear the whole book of a particular security code |

### General Rules

- All entries within an Aggregate Order Book message must be applied one by one sequentially.
- Clients must adjust the price levels of entries which are lower than the deleted or inserted entry in action. Note - Implicit level adjustments must be carried out after each single entry in Aggregate Order Book message.
- If a new book entry causes the bottom entry of a book to be shifted out of the book (i.e. beyond 10 tick levels),
  1. If the shifted out entry is within 10 price levels, OMD-C will send an explicit deletion message (Explicit delete). Client application should delete the entry accordingly.
  2. If the shifted out entry is outside the 10 Price level, OMD-C will **not** send any explicit deletion message. Clients applications must be able to identify entries fallen out of 10 price levels to delete (Implicit delete).
  3. If the book shrinks again, MMDH will resend the entries that have temporarily fallen out with the latest update.
- MMDH always provides sufficient information for client applications to build an order book of top 10 tick levels. In any event such as order cancellation or trade execution which empties the queue at a price within the top 10 ticks, MMDH will send out messages to delete the price concerned and at the same time to send new price(s) from those originally below the top 10 ticks, if any, to complement the top 10 order book.
- If a clear aggregate order book message is received, client should clear all entries of the order book. In normal circumstances, MMDH will resend all outstanding order book entries after the "Clear" message.

Please refer to *Section 6 – Aggregate Order Book Management in the HKEX OMD-C MMDH Interface Specification* for different scenarios on how OMD-C sends Aggregate Order Book message.

You may also refer to *APPENDIX A – Pseudo code for processing Aggregate Order Book Message* for example on handling Order Book messages from OMD-C server

# 7 EXCEPTION HANDLING

Listed below are some common exception handling procedures that clients must be capable of doing when subscribing to MMDH:

- Late connection
- Client application restarts
- MMDH restarts before market open
- MMDH component and Site Failure
- MMDH-Index component and Site Failure
- Special Trading Condition

In order to facilitate clients' verification of their exception handling ability, some of the exceptional scenarios are included in the Certification Test which clients have to pass in order to get on board. Emergency drills are also held in a production-like environment on a regular basis for clients to test their systems and practise their operations on the handling of other exceptional scenarios such as disaster recovery site failover. Please refer to the following sub-sections for the availability of test/drill session for each of the exceptional scenarios. Client notice will be issued to announce the schedule and coverage of a regular rehearsal in advance.

## 7.1 Late Connection

Please refer to section *4.3 Recovery* for recovery procedures.

Clients can test late connection in the Certification Test environment as they wish.

## 7.2 Client Application Restarts

Similar to "Late Connection" described above. Client may probably require to recover data in cache by providing the last received sequence number as the 'InternalSeqNum' during logon. Normal data flow continues thereafter.

This exceptional scenario is covered in the Certification Test.

## 7.3 MMDH Terminates Client Connection

If a Logout (1103) message arrived, please check for the SessionStatus.

Please refer to *Section 5.6 and 5.8 of the HKEX OMD-C MMDH Interface Specification*.

This exceptional scenario is covered in the Certification Test.

## 7.4 MMDH Restart Before Market Open

In case of OMD-C/MMDH performs start-of-day twice (errors encountered during first start-of-day), clients will be informed to reconnect to MMDH. Clients should discard all reference data received in the first start-of-day and process messages in the second start-of-day.

This exceptional scenario is covered in the Certification Test.

## 7.5 MMDH Restarts After Market Open (Intraday Restart)

When MMDH fails during trading hours, MMDH may be recovered by Intraday Restart where MMDH will be shut down and then restarted. When MMDH is shutdown, no live data can be received and clients will be disconnected. Clients will be informed if Intraday Restart needs to take place.

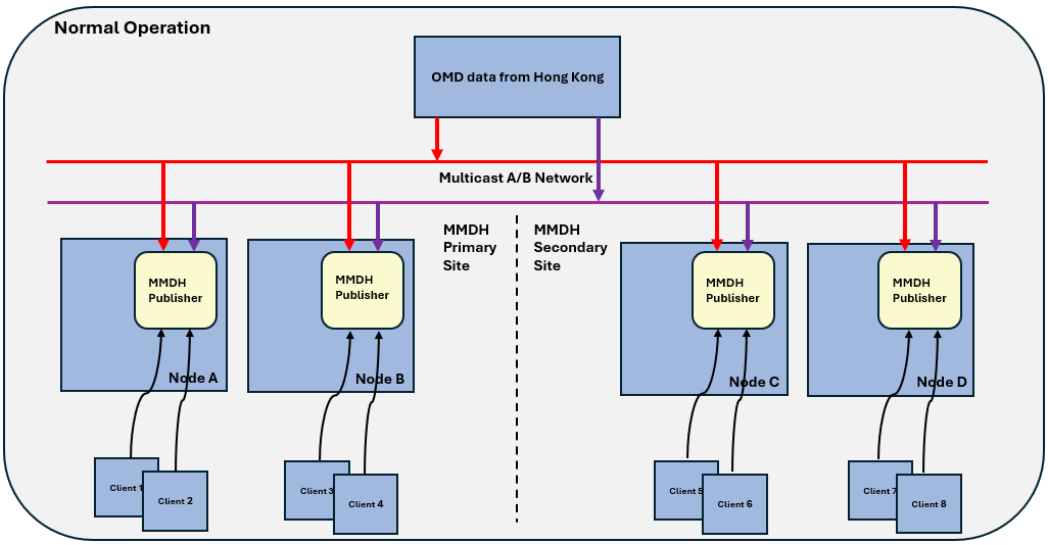
Clients will be notified when MMDH is subsequently restarted. Upon receipt our notification, all clients should clear all their internal cache and reconnect to MMDH afresh in the same way as their systems start up late and connect to MMDH only after the cash market has opened. In this situation, clients need to go through the refresh service (see 4.3.1) to establish the latest market status.

The duration between MMDH's shutdown and the subsequent restart could be a timespan of an hour and clients need to observe announcement made.

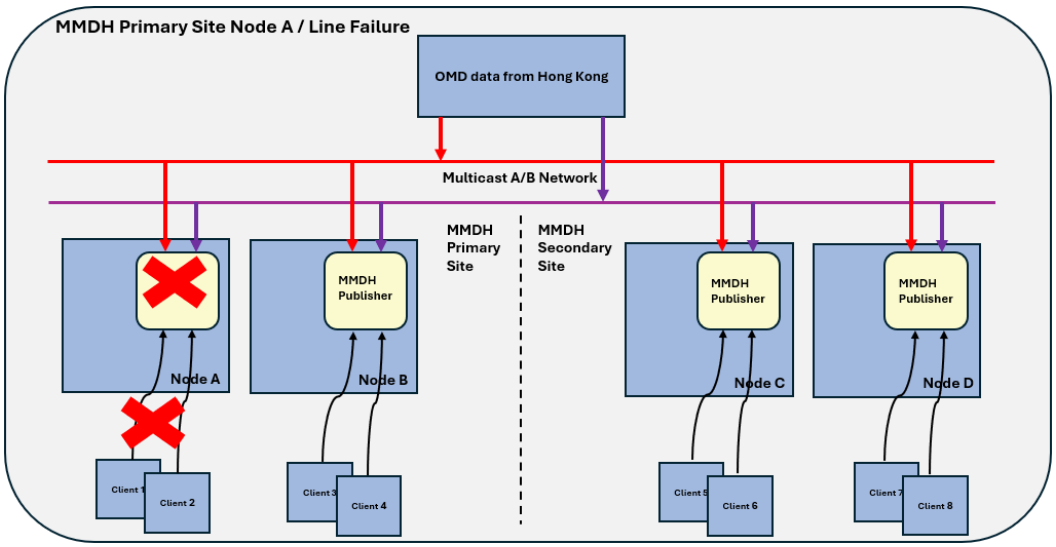
This exceptional scenario is among the possible scenarios to be covered in a regular emergency drill.

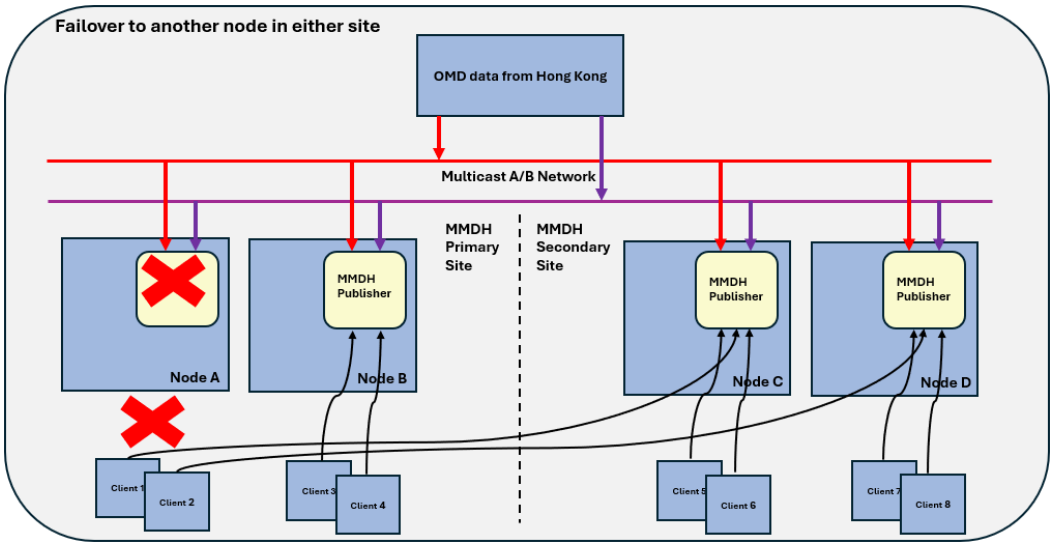
## 7.6 MMDH Component and Site Failure

MMDH Primary Site and Secondary Site operate in an active-active model. With the active-active model where Primary Site and Secondary Site are simultaneously running, all 4 MMDH nodes are disseminating same data.




In case no live data can be received (e.g. missing of 3 consecutive heartbeat messages) from the current connected node (e.g. Primary Site node A), client may reconnect to the node in another Site (e.g. Secondary Site node C) by specifying the last received sequence number (i.e. the "InternalSeqNum" of the last good message received) as the 'InternalSeqNum' in the logon message.






The same password should be used to logon in both MMDH Primary Site and Secondary Site, even if the password has been changed in one of the sites.

Client will be individually notified for the IP address of MMDH node A or node B in the MMDH Primary Site and node C or Node D in the MMDH Secondary Site.

These exceptional scenario are covered in the Certification Test. Site failure scenario is also covered in the regular emergency drill.

## 7.7 MMDH-Index Component and Site Failure

Index data is disseminated from a dedicated MMDH-Index component and is separated from the MMDH component. As a result, MMDH-Index component is independent to MMDH component.

The infrastructure design and operation of MMDH-Index component are exactly the same as the MMDH component (i.e. operate in an active-active model) and therefore its component failure arrangement follows the same as illustrated in Section 7.6 above.

## 7.8 Special Trading Condition

Clients' system should have the flexibility to handle the variations of MMDH data transmission pattern due to the following special trading conditions:

**Hong Kong Holiday which is not a Holiday in Mainland**

On Hong Kong holidays which are not holidays in the Mainland, only MMDH-Index component will be operated as usual for the clients to connect and subscribe. However, only index reference and real-time index data of non-Hong Kong indices will be disseminated.

On the other hand, MMDH component will not be brought up and clients will not be able to connect to MMDH component to receive the market data of Securities Standard (SS), Odd Lot (OLO) and Stock Connect Market (SCM).

# APPENDIX A – Pseudo code for processing Aggregate Order Book Message

An example shows how clients process Aggregate Order Book Message and update the internal order book.

```
OrderBook mOrderBook;

void processAggregateOrderBook(AggregateOB aggregateOB) {

    switch(aggregateOB.getAction())

    case ADD:
      int tickLevel = getTickLevel(mOrderBook, aggregateOB.getPrice());

      insertOB(mOrderBook, tickLevel, aggregateOB);

      //If Price level > 10, delete those order from OBMap
      deleteOBExceedMaxPriceLevel(mOrderBook);

    case Update:
       int tickLevel = getTickLevel(mOrderBook, aggregateOB);
       updateOB(mOrderBook, tickLevel, aggregateOB);

    case Delete:
        int tickLevel = getTickLevel(mOrderBook, aggregateOB);
       deleteOB(mOrderBook, tickLevel, aggregateOB);
       updateOBPriceLeve(mOrderBook)
    case Clear:
      clearOB(mOrderBook);

}
void insertOB(mOrderBook, tickLevel, newAggregateOB) {
    newAggregateOB.setTickLevel(tickLevel);
    mOrderBook.add(tickLevel-1, newAggregateOB);

    for (int i=tickLevel; i < mOrderBook.getSize(); i++) {
        AggregateOB aggregateOB = mOrderBook.get(i);
        aggregateOB.updateTickLevel(mOrderBook);
        aggregateOB.updatePriceLevel(mOrderBook);
    }
}
```

# APPENDIX B – Pseudo code for MMDH logon

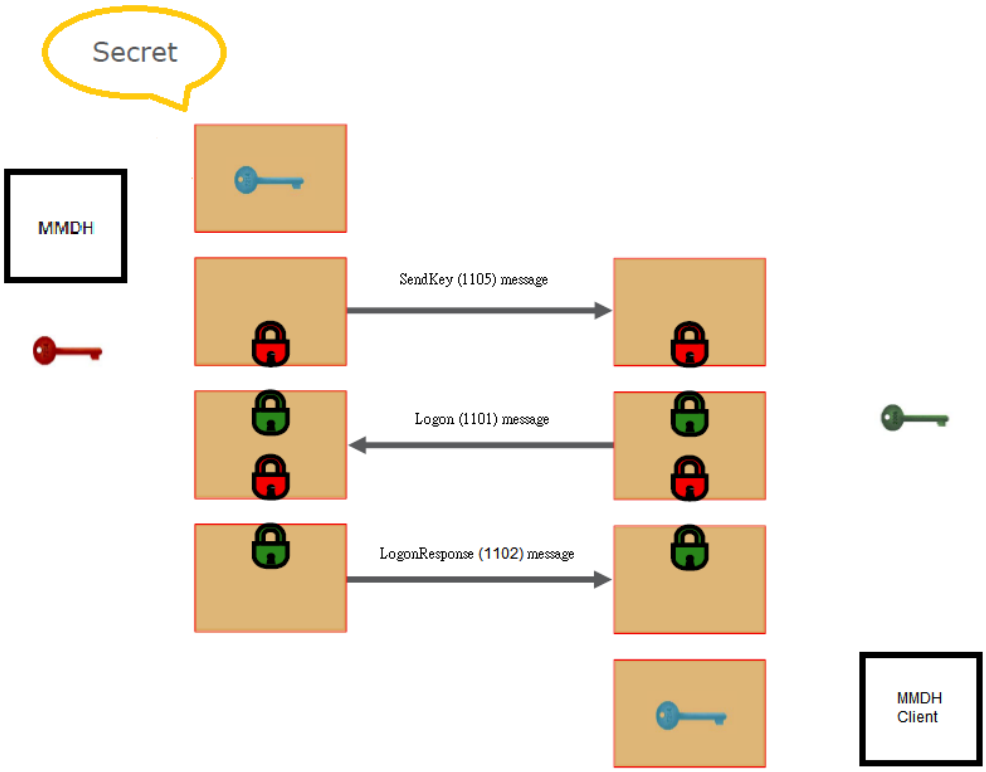
An example shows how to logon to MMDH.

```
switch (msgType) {
   case SEND_KEY_TYPE:
   {
    // Make use of Prime, Generator, PrimeOrderSubgroup, and OMDPublicKey fields
    // in Send Key message
    processSendKeyMsg();

     // Generate Diffie-Hellman public and private keys
     generateKeyPair();

     // Create shared key using generated private key and received OMDPublicKey field
     computeSharedKey();

     // Calculate SHA-256 message digest
     calcDigest();

     // Use AES to encrypt password
     aesEncrypt(password, passwordLen, encryptedPassword);

     // If change password, use AES to encrypt new password
     aesEncrypt(newPassword, newPasswordLen, encryptedNewPassword);

     break;
  }

  case LOGON_RESPONSE_TYPE:
  {
     processLogonResponseMsg();
     break;
  }

  default:
     break;
}
```

# APPENDIX C – Diffie – Hellman Key Exchange

The Diffie-Hellman Key Exchange is used before MMDH client logons to OMD-C system.

For the key exchange mechanism, please refer to Section 3.4.1 Send Key and 3.4.2 Logon of the HKEX OMD-C MMDH Interface Specification for details.




At the end of a logon attempt, MMDH will give a SessionStatus via Logon Response (1102) message to the client. MMDH client should verify the session status code.

The same key exchange mechanism is also used when an MMDH client wants to change a new password.

# DOCUMENT HISTORY

<table>
<tr><th>Version</th><th>Date of Issue</th><th>Comments</th></tr>
<tr><td>V1.0</td><td>31 Dec 2012</td><td>First Distribution Issue</td></tr>
<tr><td>V1.1</td><td>27 May 2013</td><td>Amendment to section 4.2.2 Process Control Message (Heartbeat) and section 7.5 MMDH Component Failover</td></tr>
<tr><td>V1.2</td><td>12 Jul 2013</td><td>Amendment to sections 4.2 & 4.2.2 for Heartbeat Message</td></tr>
<tr><td>V1.3</td><td>9 Oct 2013</td><td>Adding new section 4.4 for Password Management and Section 7.7 for Special Trading Condition</td></tr>
<tr><td>V1.4</td><td>19 Dec 2014</td><td>Section 7<br>• Elaborate exceptional scenarios (newly added Section 7.5 and renumbering of subsequent subsections) and state the arrangements to facilitate client tests on the exceptional scenarios</td></tr>
<tr><td>V1.5</td><td>6 Feb 2015</td><td>Section 7.5<br>• Add notes on handling Intra-day Restart</td></tr>
<tr><td>V1.6</td><td>27 Mar 2017</td><td><table><tr><th>Effective Date</th><th>Changes</th></tr><tr><td>Immediate</td><td>Clarifications<br>• Section 4.1 – Revise description to clarify Start of Day</td></tr></table></td></tr>
<tr><td>V1.7</td><td>9 Mar 2018</td><td><table><tr><th>Effective Date</th><th>Changes</th></tr><tr><td>30 April 2018</td><td>OMD-C Reference Data Enrichment :<br>• Section 4.2.1.1 – New section to highlight the handling of some new fields with variable decimal place value defined in another field</td></tr><tr><td>Immediate</td><td>Clarifications:<br>• Section 5 – Revise the example of race conditions</td></tr></table></td></tr>
<tr><td>V1.8</td><td>6 Sep 2019</td><td><table><tr><th>Effective Date</th><th>Changes</th></tr><tr><td>Immediate</td><td>Clarifications:<br>• Section 7.7 – Additional information for logon password to be used in site failover scenario</td></tr></table></td></tr>
<tr><td>V1.9</td><td>24 Apr 2023</td><td><table><tr><th>Effective Date</th><th>Changes</th></tr><tr><td>Immediate</td><td>Housekeeping<br>• Sections 7, 7.4 and 7.5– Remove the wording “HKEX”</td></tr></table></td></tr>
<tr><td>V2.0</td><td>26 Sep 2025</td><td><table><tr><th>Effective Date</th><th>Changes</th></tr><tr><td>December 2025 tentatively</td><td>Implementation of MMDH Enhanacement<br>• Sections 7.5, 7.6 and 7.7– Update exception handling after MMDH Primary Site and Secondary Site operate in Active-Active model</td></tr></table></td></tr>
<tr><td>V2.1</td><td>18 May 2026</td><td><table><tr><th>Effective Date</th><th>Changes</th></tr><tr><td>September 2026 tentatively</td><td>Implementation of Enhanacement on Index Datafeed in MMDH<br>• Section 7.7 – Add a new section for MMDH-Index Component and Site Failure<br>• Section 7.8 – Update the special trading scenario</td></tr></table></td></tr>
</table>