



INTERFACE SPECIFICATIONS

HKEX Orion Market Data Platform Derivatives Market Datafeed Products (OMD-D)

- Derivatives Lite

Binary Protocol

Version 1.4b
8 Aug 2018

© Copyright 2018 HKEX
All Rights Reserved

DOCUMENT HISTORY

| Version | Date | Changes | | | | |
|---|--|--|----------------|---------|---|--|
| 1.0 | 11 Jul 2017 | First Distribution Issue | | | | |
| 1.1 | 4 Sep 2017 | Revised Edition with the following updates: <table border="1"> <thead> <tr> <th>Effective Date</th> <th>Changes</th> </tr> </thead> <tbody> <tr> <td>Immediate</td> <td>Change of product names - Section 7.4 – Description of Market ID 37 & 38 updated</td> </tr> </tbody> </table> | Effective Date | Changes | Immediate | Change of product names - Section 7.4 – Description of Market ID 37 & 38 updated |
| Effective Date | Changes | | | | | |
| Immediate | Change of product names - Section 7.4 – Description of Market ID 37 & 38 updated | | | | | |
| 1.2 | 28 Sep 2017 | Revised Edition with the following updates: <table border="1"> <thead> <tr> <th>Effective Date</th> <th>Changes</th> </tr> </thead> <tbody> <tr> <td>28 Sep 2017</td> <td>Introduction of Iron Ore Futures Contracts - Appendix 7.4 – Add new market ID#118 for New Iron Ore Futures</td> </tr> </tbody> </table> | Effective Date | Changes | 28 Sep 2017 | Introduction of Iron Ore Futures Contracts - Appendix 7.4 – Add new market ID#118 for New Iron Ore Futures |
| Effective Date | Changes | | | | | |
| 28 Sep 2017 | Introduction of Iron Ore Futures Contracts - Appendix 7.4 – Add new market ID#118 for New Iron Ore Futures | | | | | |
| 1.3 | 7 Feb 2018 | Revised Edition with the following updates: <table border="1"> <thead> <tr> <th>Effective Date</th> <th>Changes</th> </tr> </thead> <tbody> <tr> <td>Immediate</td> <td>Clarification - Appendix 8.1 – Update the information of status code “28” and rename “AHFT” to “AHT”</td> </tr> </tbody> </table> | Effective Date | Changes | Immediate | Clarification - Appendix 8.1 – Update the information of status code “28” and rename “AHFT” to “AHT” |
| Effective Date | Changes | | | | | |
| Immediate | Clarification - Appendix 8.1 – Update the information of status code “28” and rename “AHFT” to “AHT” | | | | | |
| 1.4 | 15 Feb 2018 | Revised Edition with the following updates: <table border="1"> <thead> <tr> <th>Effective Date</th> <th>Changes</th> </tr> </thead> <tbody> <tr> <td>May 2019 (Upon rollout of HKATS / DCASS Upgrade)</td> <td>Data Enrichment for HKATS/DCASS Upgrde <ul style="list-style-type: none"> - Section 3.7.3 – Replace 2-byte filler with a new field “DecimalInStrikePrice” for the strike price in the same message - Section 3.7.4 – Add a new value on existing field “SeriesStatus” and replace 1-byte filler with a new field “PriceMethod” - Section 3.7.4 – Replace 8-byte filler with a new field “DateTimeFirstTrading” - Section 3.8.2 and 3.8.3 – Revise the description - Section 3.8.2 – Rename the existing field “Suspended” to “SuspensionIndicator”, change the existing values and add a new value - Section 3.8.2 – Replace 1-byte filler with a new field “SeriesStatus” - Section 3.8.3 – Revise description for existing field “Suspended” and replace 1-byte filler with a new field “Locked” - Section 3.12.1 – Clarify the content of the Open Interest (366) message and add a note to define a business day </td> </tr> </tbody> </table> | Effective Date | Changes | May 2019 (Upon rollout of HKATS / DCASS Upgrade) | Data Enrichment for HKATS/DCASS Upgrde <ul style="list-style-type: none"> - Section 3.7.3 – Replace 2-byte filler with a new field “DecimalInStrikePrice” for the strike price in the same message - Section 3.7.4 – Add a new value on existing field “SeriesStatus” and replace 1-byte filler with a new field “PriceMethod” - Section 3.7.4 – Replace 8-byte filler with a new field “DateTimeFirstTrading” - Section 3.8.2 and 3.8.3 – Revise the description - Section 3.8.2 – Rename the existing field “Suspended” to “SuspensionIndicator”, change the existing values and add a new value - Section 3.8.2 – Replace 1-byte filler with a new field “SeriesStatus” - Section 3.8.3 – Revise description for existing field “Suspended” and replace 1-byte filler with a new field “Locked” - Section 3.12.1 – Clarify the content of the Open Interest (366) message and add a note to define a business day |
| Effective Date | Changes | | | | | |
| May 2019 (Upon rollout of HKATS / DCASS Upgrade) | Data Enrichment for HKATS/DCASS Upgrde <ul style="list-style-type: none"> - Section 3.7.3 – Replace 2-byte filler with a new field “DecimalInStrikePrice” for the strike price in the same message - Section 3.7.4 – Add a new value on existing field “SeriesStatus” and replace 1-byte filler with a new field “PriceMethod” - Section 3.7.4 – Replace 8-byte filler with a new field “DateTimeFirstTrading” - Section 3.8.2 and 3.8.3 – Revise the description - Section 3.8.2 – Rename the existing field “Suspended” to “SuspensionIndicator”, change the existing values and add a new value - Section 3.8.2 – Replace 1-byte filler with a new field “SeriesStatus” - Section 3.8.3 – Revise description for existing field “Suspended” and replace 1-byte filler with a new field “Locked” - Section 3.12.1 – Clarify the content of the Open Interest (366) message and add a note to define a business day | | | | | |
| 1.4a | 8 May 2018 | Revised Edition with the following updates: <table border="1"> <thead> <tr> <th>Effective Date</th> <th>Changes</th> </tr> </thead> <tbody> <tr> <td>Immediate</td> <td>Introduction of MSCI AC Asia ex Japan Net Total Return Index Futures Contracts - Appendix 8.4 – Add new market ID#108 for New MSCI AxJ Futures</td> </tr> </tbody> </table> | Effective Date | Changes | Immediate | Introduction of MSCI AC Asia ex Japan Net Total Return Index Futures Contracts - Appendix 8.4 – Add new market ID#108 for New MSCI AxJ Futures |
| Effective Date | Changes | | | | | |
| Immediate | Introduction of MSCI AC Asia ex Japan Net Total Return Index Futures Contracts - Appendix 8.4 – Add new market ID#108 for New MSCI AxJ Futures | | | | | |
| 1.4b | 8 Aug 2018 | Revised Edition with the following updates: <table border="1"> <thead> <tr> <th>Effective Date</th> <th>Changes</th> </tr> </thead> <tbody> <tr> <td>May 2019 (Upon rollout of HKATS / DCASS Upgrade)</td> <td>Housekeeping and Clarification <ul style="list-style-type: none"> - Section 3.7.2, 3.7.4, 3.8.1 – Clarify that possible values of the fields “Market” and “InstrumentGroup” are provided in Section 7.4 and 7.3 respectively - Section 3.7.4 – Correct offset for the field “DateTimeFirstTrading” - Section 3.12.1 – Clarify the information provided in Open Interest (366) message </td> </tr> </tbody> </table> | Effective Date | Changes | May 2019 (Upon rollout of HKATS / DCASS Upgrade) | Housekeeping and Clarification <ul style="list-style-type: none"> - Section 3.7.2, 3.7.4, 3.8.1 – Clarify that possible values of the fields “Market” and “InstrumentGroup” are provided in Section 7.4 and 7.3 respectively - Section 3.7.4 – Correct offset for the field “DateTimeFirstTrading” - Section 3.12.1 – Clarify the information provided in Open Interest (366) message |
| Effective Date | Changes | | | | | |
| May 2019 (Upon rollout of HKATS / DCASS Upgrade) | Housekeeping and Clarification <ul style="list-style-type: none"> - Section 3.7.2, 3.7.4, 3.8.1 – Clarify that possible values of the fields “Market” and “InstrumentGroup” are provided in Section 7.4 and 7.3 respectively - Section 3.7.4 – Correct offset for the field “DateTimeFirstTrading” - Section 3.12.1 – Clarify the information provided in Open Interest (366) message | | | | | |

CONTENTS

| | |
|---|-----------|
| DOCUMENT HISTORY | 2 |
| CONTENTS..... | 3 |
| 1. INTRODUCTION..... | 5 |
| 1.1 PURPOSE | 5 |
| 1.2 READING GUIDE..... | 5 |
| 1.3 PRODUCTS | 5 |
| 1.3.1 <i>Derivatives Lite ('D-Lite')</i> | 5 |
| 1.3.2 <i>Summary Table</i> | 6 |
| 2. SYSTEM OVERVIEW..... | 7 |
| 2.1 SCOPE | 7 |
| 2.1.1 <i>Multicast</i> | 7 |
| 2.1.2 <i>Dual Multicast Channels</i> | 7 |
| 2.1.3 <i>Recovery Mechanisms</i> | 7 |
| 2.2 SESSION MANAGEMENT | 7 |
| 2.2.1 <i>Start of Day</i> | 8 |
| 2.2.2 <i>Normal Transmission</i> | 8 |
| 2.2.3 <i>End of Day</i> | 8 |
| 2.2.4 <i>Error Recovery</i> | 8 |
| 2.3 TRADING SESSIONS | 9 |
| 2.3.1 <i>Extended Trading Hours</i> | 9 |
| 2.4 RACE CONDITIONS..... | 9 |
| 3. MESSAGE FORMATS..... | 10 |
| 3.1 DATA TYPES | 10 |
| 3.1.1 <i>Null Values</i> | 10 |
| 3.1.2 <i>Currency Values</i> | 10 |
| 3.1.3 <i>Decimal Values</i> | 10 |
| 3.1.4 <i>Bitmap Values</i> | 10 |
| 3.2 PACKET STRUCTURE | 11 |
| 3.3 PACKET HEADER..... | 12 |
| 3.4 CONTROL MESSAGES..... | 12 |
| 3.4.1 <i>Heartbeat</i> | 12 |
| 3.4.2 <i>Sequence Reset (100)</i> | 12 |
| 3.4.3 <i>Disaster Recovery Signal (105)</i> | 12 |
| 3.5 RETRANSMISSION | 13 |
| 3.5.1 <i>Logon (101)</i> | 13 |
| 3.5.2 <i>Logon Response (102)</i> | 14 |
| 3.5.3 <i>Retransmission Request (201)</i> | 14 |
| 3.5.4 <i>Retransmission Response (202)</i> | 14 |
| 3.6 REFRESH | 15 |
| 3.6.1 <i>Refresh Complete (203)</i> | 15 |
| 3.7 REFERENCE DATA..... | 15 |
| 3.7.1 <i>Commodity Definition (301)</i> | 16 |
| 3.7.2 <i>Class Definition (302)</i> | 17 |
| 3.7.3 <i>Series Definition Base (303)</i> | 18 |

| | | |
|-----------|---|-----------|
| 3.7.4 | <i>Series Definition Extended (304)</i> | 19 |
| 3.7.5 | <i>Combination Definition (305)</i> | 21 |
| 3.8 | STATUS DATA | 22 |
| 3.8.1 | <i>Market Status (320)</i> | 22 |
| 3.8.2 | <i>Series Status (321)</i> | 23 |
| 3.8.3 | <i>Commodity Status (322)</i> | 23 |
| 3.9 | ORDER BOOK DATA | 23 |
| 3.9.1 | <i>Add Order (330)</i> | 24 |
| 3.9.2 | <i>Aggregate Order Book Update (353)</i> | 24 |
| 3.9.3 | <i>Quote Request (336)</i> | 25 |
| 3.10 | TRADE AND PRICE DATA | 25 |
| 3.10.1 | <i>Trade (350)</i> | 25 |
| 3.10.2 | <i>Trade Amendment (356)</i> | 27 |
| 3.10.3 | <i>Trade Statistics (360)</i> | 27 |
| 3.10.4 | <i>Calculated Opening Price (364)</i> | 28 |
| 3.10.5 | <i>Estimated Average Settlement Price (365)</i> | 28 |
| 3.11 | NEWS..... | 29 |
| 3.11.1 | <i>Market Alert (323)</i> | 29 |
| 3.12 | CLEARING INFORMATION | 30 |
| 3.12.1 | <i>Open Interest (366)</i> | 30 |
| 4. | RECOVERY | 31 |
| 4.1 | GAP DETECTION | 31 |
| 4.2 | LINE ARBITRATION | 33 |
| 4.3 | RETRANSMISSION SERVICE..... | 34 |
| 4.4 | REFRESH SERVICE..... | 37 |
| 4.4.1 | <i>Refresh of Market Status</i> | 38 |
| 5. | AGGREGATE ORDER BOOK MANAGEMENT | 39 |
| 6. | AUCTION PERIOD SPECIAL HANDLING | 44 |
| 7. | HKEX DERIVATIVES MARKET | 45 |
| 7.1 | MARKET STATES..... | 45 |
| 7.2 | DEAL SOURCES..... | 47 |
| 7.3 | LIST OF INSTRUMENT CODE | 48 |
| 7.4 | LIST OF MARKET ID..... | 49 |
| 7.5 | LIST OF INDEX CODE FOR INDEX EAS | 49 |

1. INTRODUCTION

1.1 PURPOSE

This document specifies the Binary interface of the datafeed product “Derivatives Lite” (“D-Lite”) under HKEX Orion Market Data Platform (“OMD”) for the Derivatives Market (“OMD-D”).

This document is the Transmission Specification(s) of the relevant Datafeed(s) under the Market Data Vendor Licence Agreement, the Market Data End-User License Agreement and/or the Market Data Application Service Provider Licence (“License Agreement”). Please refer to the summary table at Section 1.3 for the information applicable to the Datafeed(s) under your License Agreement.

HKEX endeavors to ensure the accuracy and reliability of the information provided in this interface specification, but takes no responsibility for any errors or omissions or for any losses arising from decisions, action, or inaction based on this information. The Licensee shall not use such interface specifications for any purpose other than as expressly permitted under the License Agreement. No part of this document may be copied, distributed, transmitted, transcribed, stored in a retrieval system, translated into any human or computer language, or disclosed to third parties without written permission from HKEX-IS.

1.2 READING GUIDE

The chapters following this introduction are:

| | |
|------------|----------------------------------|
| Chapter 2: | System Overview |
| Chapter 3: | Message Formats |
| Chapter 4: | Recovery |
| Chapter 5: | Aggregated Order Book Management |
| Chapter 6: | Auction Period Special Handling |
| Chapter 7: | HKEX Derivatives Market |

1.3 PRODUCTS

The datafeed product described in this section is available by splitting into two separate sets of multicast group channels for the Stock Options Market and the Non-Stock Options Markets to better serve OMD-D clients on bandwidth consumption planning. Messages are applicable to datafeeds irrespective of whether it belongs to Stock Options Market or non-Stock Options Markets. For the avoidance of doubt, Stock Options Market (“SOM”) refer to the market for trading the stock option that is a financial contract based on a single underlying stock which is traded on SEHK and cleared through the SEHK Options Clearing House Limited (SEOCH). Non-Stock Options Markets (“Non-SOM”) refer to the markets of trading HKEX derivatives products on HKFE. For the latest list of derivatives products on HKFE, please refer to the [HKEX website](#).

1.3.1 Derivatives Lite (‘D-Lite’)

D-Lite comprises a conflated feed to provide Level 2 book showing up to 5 aggregated price levels and snapshots of order depth information for active series which refer to the series eligible for Volatility Control Mechanism (VCM series) for the Derivatives Market.

1.3.1.1 Complimentary Feed to D-Lite – Derivatives Trades (‘DT’)

DT is a streaming trades feed with all Trade and Trade Amendments and offered complementarily to D-Lite clients.

1.3.2 Summary Table

● The information supplied in the corresponding sub-section applies to D-Lite

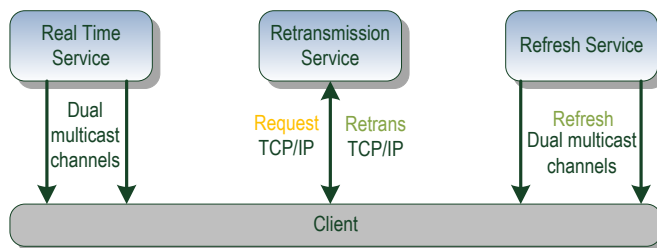
▲ Complimentary service to the Datafeed(s)

| Section | Message Formats | Derivatives Lite (D-Lite) |
|---------|--|---------------------------|
| 3.1 | Data Types | ● |
| 3.2 | Packet Structure | ● |
| 3.3 | Packet Header | ● |
| 3.4 | Control Messages | ● |
| 3.5 | Retransmission | ● |
| 3.6 | Refresh | ● |
| 3.7 | Reference Data | ● |
| 3.8 | Status Data | ● |
| 3.9.1 | Add Order (330) | ● |
| 3.9.4 | Aggregate Order Book Update (353) | ● |
| 3.9.6 | Quote Request (336) | ● |
| 3.10.1 | Trade (350) | ▲ |
| 3.10.2 | Trade Amendment (356) | ▲ |
| 3.10.3 | Trade Statistics (360) | ● |
| 3.10.5 | Calculated Opening Price (364) | ● |
| 3.10.6 | Estimated Average Settlement Price (365) | ● |
| 3.11.1 | Market Alert (323) | ● |
| 3.12.1 | Open Interest (366) | ● |
| 5 | Aggregate Order Book Management | ● |

2. SYSTEM OVERVIEW

2.1 SCOPE

Figure 1: Access to Market Data



OMD provides market data represented in an efficient binary message format for all instruments traded on the Derivatives Market. It has been designed for high throughput and low latency.

2.1.1 Multicast

Messages are published in a one-to-many fashion using the IP multicast and UDP transport protocols. Multicast is not a connection-oriented protocol. Data is sent strictly in one direction from server to clients.

2.1.2 Dual Multicast Channels

Due to the inherently unreliable nature of the UDP transport, packets may be lost or delivered out-of-order. To mitigate the risk of packet loss, the messages are duplicated and sent over two separate multicast channels (dual channels). Technically, a multicast channel corresponds to a multicast group.

Each pair of dual multicast channels has a unique identifier, which is referred to as the ChannelID.

More details regarding the configuration parameters (including IP addresses, port numbers corresponding to the multicast channels) will be found in a Connectivity Guide which will be provided at a later stage.

2.1.3 Recovery Mechanisms

OMD provides two recovery mechanisms:

- A retransmission server provides on request gap-fill retransmission of lost messages. The retransmission requests and gap-fill replies are point-to-point (TCP/IP connection).
- A refresh server provides snapshots of the market state at regular intervals throughout the business day. Snapshots are sent using multicast on separate channels for the real time messages.

2.2 SESSION MANAGEMENT

Each multicast channel maintains its own session. A session is limited to one business day. During this day the message sequence number is strictly increasing and therefore unique within the channel.

OMD-D does not operate on non-trading days of the Hong Kong Derivative Market where a trading day ends at the close of the after hour trading session. HKEX may perform system testing on Saturdays, Sundays or days when OMD-D is not in operation. Clients should treat data transmitted via OMD-D on those days as non production data and disregard them.

2.2.1 Start of Day

Housekeeping and system maintenance work may take place overnight until 6:00am. In this regard, Clients are advised to make connection to OMD Derivatives Market ("OMD-D") at or after 6:00am every business day to ensure that the data received from OMD-D are good for the start of the day. Please also refer to the OMD-D Developer's Guide for more information.

On each channel the first message at the start of the business day is the Sequence Reset message. The Sequence Reset message carries sequence number 1. On receipt of this message, the client must clear all cached data for all instruments.

The messages sent at start of day are:

- Commodity Definition, Class Definition, Series Definition Base, Series Definition Extend and Combination, messages for all tradeable series, including Combo Series
- Latest series quotation and trade snapshot.
- Latest market and series status snapshot
- Previous day open interest & settlement information for all tradeable series if available
- The latest Market message, COP ,EAS

If a client starts listening after the start of business day and misses the Sequence Reset message and reference data, it must use the refresh service to recover and synchronize with the real time channels.

2.2.2 Normal Transmission

Normal message transmission is expected between when the market opens for trading and when the market is closed. Heartbeats are sent every at regular intervals (currently set at every 2 seconds) on each channel when there is no activity. HKEX may adjust this interval.

2.2.3 End of Day

OMD will typically shut down following the end of after hours trading at around midnight Hong Kong time. Later shutdown is possible to cater for special circumstances such as capital adjustment effective the following trading day. Shutdown time is not rigid and the Exchange has the right to adjust this time according to the different trading situations.

Clients should wait for a Market Status (320) message marking end of day before disconnecting.

2.2.4 Error Recovery

2.2.4.1 System Component Failure

If a system component fails that leads to a small amount of packet loss and requires a failover or restart, there will be a short interruption in multicast dissemination from either Line A or Line B. The system is deployed in an active-active configuration with Line A and Line B being generated independently and so line arbitration will allow the client to continue receiving messages – see section 4 for more information about recovery.

2.2.4.2 Disaster Recovery

In the unlikely event of a disaster recovery situation at the primary site, OMD will be brought up at the disaster recovery (DR) site.

During the interruption, no data will be sent including heartbeats.

A Disaster Recovery (DR) Signal message indicating the DR status will also be sent on its dedicated channel when OMD is brought up – see section 3 for more information about the DR Signal message. See *Developers Guide for more details*.

IP addresses and ports that have been provided for the disaster site's retransmission service should be used. See *Connectivity Guide* for more details.

2.3 TRADING SESSIONS

Normally, trading is conducted in auction trading session(s) and continuous trading session(s) every trading day. However, there are situations where there is only half day trading with fewer trading session(s) (Christmas eve, New Year eve and Chinese New Year eve), or a trading session is suspended due to a typhoon etc. OMD is not affected by the number of trading sessions and will continue to provide real time data as long as the Exchange's trading system is available.

2.3.1 Extended Trading Hours

Extended trading hours may apply to certain products. Trading hours of individual products are listed on the HKEX website.

2.4 RACE CONDITIONS

Due to the nature of the dissemination protocol the real time order/trade data and reference data are disseminated via separate channels so users need to be aware that there is a race condition.

As an example suppose a Market Status message is sent showing a change to state 'Closed', however for a very short time after this message the regular order and trade information for this series may continue to arrive.

3. MESSAGE FORMATS

3.1 DATA TYPES

The following table lists all the data types used by OMD.

| Format | Description |
|--------|--|
| String | ASCII characters which are left aligned and padded with spaces, unless otherwise specified. |
| UInt8 | 8 bit unsigned integer. |
| UInt16 | Little-Endian encoded 16 bit unsigned integer. |
| UInt32 | Little-Endian encoded 32 bit unsigned integer. |
| UInt64 | Little-Endian encoded 64 bit unsigned integer. |
| Int16 | Little-Endian encoded 16 bit signed integer. |
| Int32 | Little-Endian encoded 32 bit signed integer. |
| Int64 | Little-Endian encoded 64 bit signed integer. |
| Binary | Unicode encoding used for Chinese characters which are left aligned and padded with binary null. |

3.1.1 Null Values

From time to time certain fields cannot be populated and specific values are used to represent null. This is currently used within Int32 fields of the Trade (350) message, the Aggregate Order Book Update (353) message, the Trade Statistics (360) message, the Calculated Opening Price (364) message, the Trade Amendment (356) message as well as the Add Order messages. Null values may also be used within the Int64 field of the EAS (365) message.

| Format | Null representation (Hex 2's complement) |
|--------|--|
| Int32 | 0x80000000 |
| Int64 | 0x8000000000000000 |

3.1.2 Currency Values

See the ISO-3166 Currency Codes for a full list of possible data values. Currently the system uses the following codes; 'HKD', 'USD', 'CNY'. HKEX may add or delete currency code(s), whenever applicable, in the future.

3.1.3 Decimal Values

Decimal values are sent as integers. This is done for efficiency - for example, a price value sent as "12345" and with 3 decimal places should be interpreted as "12.345". See individual fields for number of decimal places used.

3.1.4 Bitmap Values

Certain fields within the Add Order (330) and Trade (350) messages are bitmap representations which can denote a number of properties within a single field.

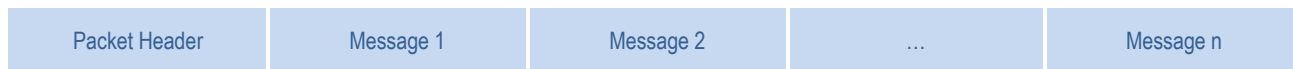
For example, with the 'DealType' field shown below, if a value of 3 is received this indicates that the trade is both "Printable" and a 'Occurred at Cross'.

| Offset | Field | Format | Len | Description | Values |
|--------|----------|--------|-----|------------------|---|
| 33 | DealType | UInt8 | 1 | Deal Type Bitmap | <ul style="list-style-type: none"> 0 None 1 Printable (see note) 2 Occurred at Cross 4 Reported Trade |

Note : The Printable property does not apply to Reported Trade. For other trades, i.e. trades matched by the trading system, it will be OFF for trades on Combo Series to avoid double counting of the trade quantity to market turnover.

3.2 PACKET STRUCTURE

Multicast 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.



The maximum length of a packet is 1500 bytes which includes the multicast headers, packet header and messages.

The packet header provides information including the total packet length, the number of messages within the packet, the sequence number of the first message and a send timestamp.

A packet will only ever contain complete messages. A single message will never be fragmented across packets. The format of each message within a packet will vary according to message type. However, regardless of the message type, each message will start with a two-byte message size (MsgSize) followed by a two-byte message type (MsgType). These are described in the following table.

Table 1: MsgSize and MsgType Fields

| Field | Format | Len | Description |
|---------|--------|-----|---|
| MsgSize | UInt16 | 2 | Message length (including this field) |
| MsgType | UInt16 | 2 | Type of message. The valid values for MsgType are below: 100 Sequence Reset 101 Logon 102 Logon Response 201 Retransmission Request 202 Retransmission Response 203 Refresh Complete 301 Commodity Definition 302 Class Definition 303 Series Definition Base 304 Series Definition Extended 305 Combination Definition 320 Market Status 321 Series Status 322 Commodity Status 330 Add Order 353 Aggregate Order Book Update 336 Quote Request 350 Trade 356 Trade Amendment 360 Trade Statistics 364 Calculated Opening Price 365 Estimated Average Settlement Price 323 Market Alert 366 Open Interest |

3.3 PACKET HEADER

All packets will begin with a common packet header.

| Offset | Field | Format | Len | Description |
|---------------|----------|--------|-----|---|
| 0 | PktSize | UInt16 | 2 | Size of the packet (including this field) |
| 2 | MsgCount | UInt8 | 1 | Number of messages included in the packet |
| 3 | Filler | String | 1 | |
| 4 | SeqNum | UInt32 | 4 | Sequence number of the first message in the packet |
| 8 | SendTime | UInt64 | 8 | UTC Timestamp. The number of nanoseconds since <i>January 1, 1970, 00:00:00 GMT</i> , precision is provided to the nearest millisecond. |
| Packet length | | | 16 | |

3.4 CONTROL MESSAGES

3.4.1 Heartbeat

Heartbeats consist of a packet header with MsgCount set to 0. They do not carry a sequence number and therefore do not increment the sequence number of the multicast channel. SeqNum is set to the sequence number of the previous message sent on the channel.

The Heartbeat message will be identical for all the services.

3.4.2 Sequence Reset (100)

The Sequence Reset message is sent on each multicast channel at start of day. It may also be sent when there is a need for the rectification of stock reference data before market open.

The client must ignore the sequence number of the Sequence Reset message itself, and set the next expected sequence number to NewSeqNo. The client may receive multiple sequence reset messages from all channels. Whenever the Sequence Reset message is received, clients must clear all cached data for all instruments traded in the Derivatives Market and then subscribe to the refresh channels to receive the current state of the market.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|----------|--------|-----|----------------------|--------------------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 100 Sequence Reset |
| 4 | NewSeqNo | UInt32 | 4 | New sequence number. | Always set to 1 |
| Total Length | | | 8 | | |

3.4.3 Disaster Recovery Signal (105)

The Disaster Recovery (DR) Signal message is sent on a dedicated multicast channel (DR channel) whenever site failover is triggered. In normal situation, the dedicated DR channel only carries Heartbeat till end of business day.

When site failover begins, DR Signal is sent with "DRStatus=1" indicating that the DR process has been activated. Clients should then clear all cached market data and prepare their own system for the site failover. When the site failover process finishes, DR Signal will be sent with "DRStatus=2" thereupon clients could start rebuild the latest market image from the refresh service. The same DR Signal will be sent periodically until end of business day.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|----------|--------|-----|-----------------------------|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 105 DR Message |
| 4 | DRStatus | UInt32 | 4 | Status during site failover | 1 – DR in progress 2 – DR completed |
| Total Length | | | 8 | | |

3.5 RETRANSMISSION

Refer to Retransmission service for details on the retransmission messages.

Note that when the Logon (101) or Retransmission Request (201) messages are sent to the OMD server, the client must also include a packet header as shown below.

Also note that the same header is used by the RTS server when sending either Logon Response (102) or Retransmission Response (202) messages to clients. Again in this case the SeqNum and SendTime fields are not relevant and can be discarded.

| Offset | Field | Format | Len | Values | Notes |
|--------------|----------|--------|-----|--------|---|
| 0 | PktSize | UInt16 | 2 | 32 | 16 bytes for this header plus 16 bytes for either the Logon (101) or Retransmission Request (201) message |
| 2 | MsgCount | UInt8 | 1 | 1 | One message only |
| 3 | Filler | String | 1 | | Empty Filler |
| 4 | SeqNum | UInt32 | 4 | 0 | This field is not used |
| 8 | SendTime | UInt64 | 8 | 0 | This field is not used |
| Total Length | | | 16 | | |

After this header, the fields for either Logon (101) or Retransmission Request (201) should follow.

Please note that Retranmission service is not available for Add Order (330) message which is a snapshot message refreshed every second.

3.5.1 Logon (101)

The Logon message enables client authentication. This is not required for multicast channels and is only used for retransmission requests.

Normal operation: Client sends a Logon message containing username to the OMD, which responds with a Logon Response message with the SessionStatus set to 0 (Session Active).

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|----------|--------|-----|--|-----------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 101 Logon |
| 4 | Username | String | 12 | Username to log on, padded with binary null characters | |
| Total Length | | | 16 | | |

3.5.2 Logon Response (102)

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|---------------|--------|-----|-----------------------|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 102 Logon Response |
| 4 | SessionStatus | UInt8 | 1 | Status of the session | 0 Session Active 5 Invalid username or IP address 100 User already connected |
| 5 | Filler | String | 3 | | |
| Total Length | | | 8 | | |

3.5.3 Retransmission Request (201)

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|-------------|--------|-----|---|-----------------------------------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 201 Retransmission Request |
| 4 | ChannelID | UInt16 | 2 | Multicast Channel ID to which the retransmission relates | |
| 6 | Filler | String | 2 | | |
| 8 | BeginSeqNum | UInt32 | 4 | Beginning of sequence | |
| 12 | EndSeqNum | UInt32 | 4 | Message sequence number of last message in range to be resent | |
| Total Length | | | 16 | | |

3.5.4 Retransmission Response (202)

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|---------------|--------|-----|---|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 202 Retransmission Response |
| 4 | ChannelID | UInt16 | 2 | Multicast Channel ID with which the retransmission relates | |
| 6 | RetransStatus | UInt8 | 1 | Status of the Retransmission response | 0 Request accepted 1 Unknown/Unauthorized channel ID 2 Messages not available 100 Exceeds maximum sequence range 101 Exceeds maximum requests in a day |
| 7 | Filler | String | 1 | | |
| 8 | BeginSeqNum | UInt32 | 4 | Beginning of sequence | |
| 12 | EndSeqNum | UInt32 | 4 | Message sequence number of last message in range to be resent | |
| Total Length | | | 16 | | |

3.6 REFRESH

Refer to Refresh service for details on the Refresh Complete message.

3.6.1 Refresh Complete (203)

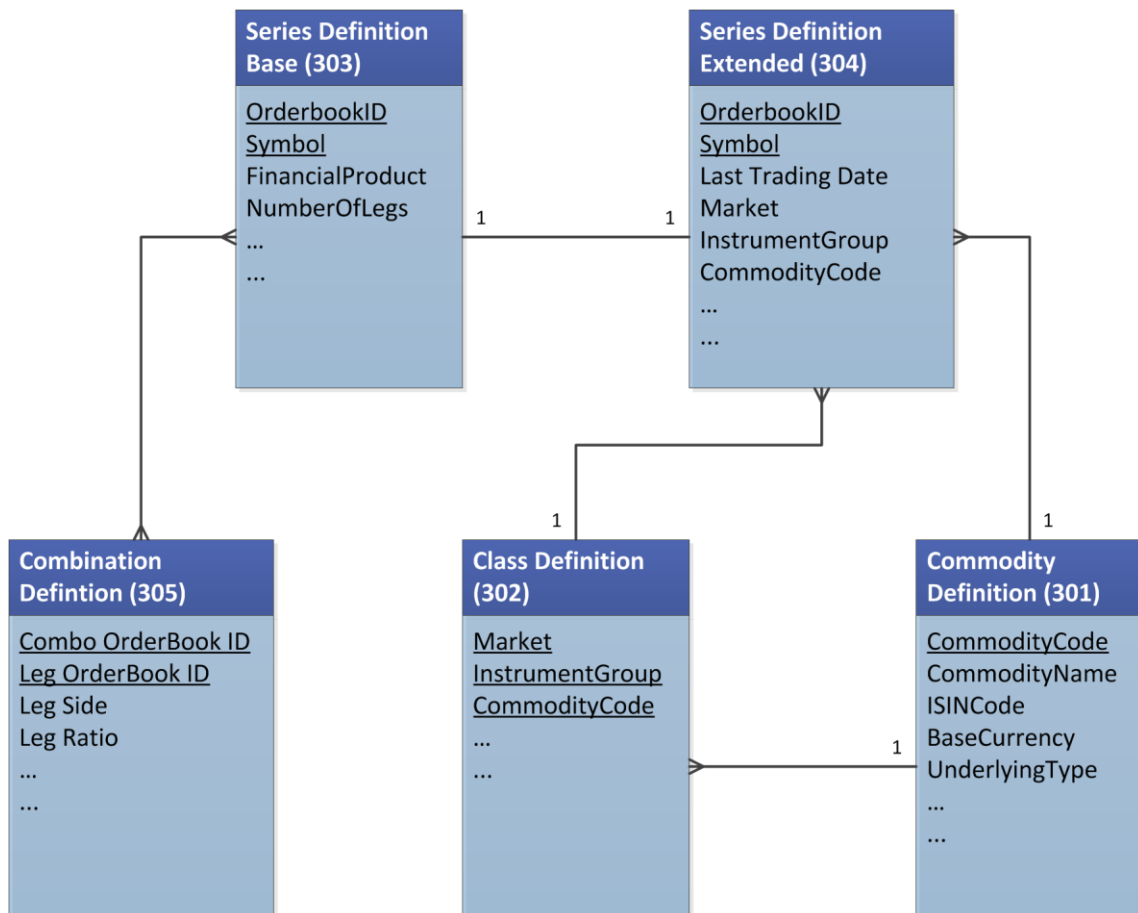
This message is published to mark the end of a refresh cycle or the end of a snapshot cycle of Add Order (330) messages, see section 0 for a full description of refresh.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|------------|--------|-----|---|----------------------|
| 0 | MsgSize | Uint16 | 2 | Size of the message | |
| 2 | MsgType | Uint16 | 2 | Type of message. | 203 Refresh Complete |
| 4 | LastSeqNum | Uint32 | 4 | Sequence number with which the refresh is synchronized. | Numerical |
| Total Length | | | 8 | | |

3.7 REFERENCE DATA

Static Reference data is organized into 5 messages which are shown in the entity relationship diagram below. The underlined field(s) form the primary key for each message type.



3.7.1 Commodity Definition (301)

Describes individual commodities available from the OMD-D system.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|--------------------------|--------|-----|--|--|
| 0 | MsgSize | Uint16 | 2 | Size of the message | |
| 2 | MsgType | Uint16 | 2 | Type of message. | 301 Commodity Definition |
| 4 | CommodityCode | Uint16 | 2 | Numerical identifier of the Underlying. This is the unique identifier of the message. The Series Definition Extended and the Class Definition are retrieved through this field which links them to the Commodity Definition. eg. 2005 (HKB). | |
| 6 | DecimalInUnderlyingPrice | Uint16 | 2 | Number of implicit decimals in the underlying price received from external sources. | |
| 8 | ISINCode | String | 12 | A code which uniquely identifies a specific securities issue (International Securities Identification Number). For more information about ISIN code, see the international standard ISO 3166. | |
| 20 | BaseCurrency | String | 3 | Defines the trading currency for the instrument or the currency for the underlying. The representation of the currency follows the S.W.I.F.T. handbook and ISO 3166 standard. | See Currency Values in section 3.1.2 for full details. |
| 23 | UnderlyingPriceUnit | Uint8 | 1 | The price unit for the underlying | 1 Price 2 Yield * 3 Points 4 Yield Diff * 5 IMM Index * 6 Basis Points * 7 Inverted Yield * 8 Percentage of Nominal * 9 Dirty Price * |
| 24 | CommodityName | String | 32 | Descriptive Name of the underlying Eg. Hang Seng Index | |
| 56 | NominalValue | Int64 | 8 | Nominal Value of the Commodity | Applicable for 3-Year EFN Futures only |
| 64 | UnderlyingCode | String | 20 | Underlying Code of the Commodity | |
| 84 | UnderlyingType | Uint8 | 1 | Type of the underlying | 1 Stock 2 Currency 3 Interest rate 4 Energy * 5 Soft and Agrics * 6 Metal 7 Stock Index 8 Currency Index * 9 Interest Rate Index * 10 Energy Index * 11 Softs and Agrics Index * 12 Metal Index * |
| 85 | EffectiveTomorrow | Uint8 | 1 | This declaration is for series to be traded the next day | 0 False 1 True |
| 86 | CommodityID | String | 6 | Commodity ID of the underlying E.g. HSB | |
| 92 | Filler | String | 2 | | |
| Total Length | | | 94 | | |

Remark *: denotes that the value is not currently in use for existing products. However, please note that HKEX may use the value anytime when introducing new products

3.7.2 Class Definition (302)

Describes individual instrument classes available from the OMD-D system. The key of a Class Definition is composed by Country, Market, Instrument Group and Commodity Code.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------|-----------------------|--------|-----|---|---|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 302 Class Definition |
| 4 | Country | UInt8 | 1 | Country Identifier | |
| 5 | Market | UInt8 | 1 | Market Code | See section 7.4 for a list of possible values |
| 6 | InstrumentGroup | UInt8 | 1 | Instrument Group. This field together with the Commodity, forms the unique identifier of the message | See section 7.3 for a list of possible values |
| 7 | Modifier | UInt8 | 1 | Expiration date modified. Value is incremented by one each time the instrument is involved in an issue, split, etc. Note that the modifier value can be different for Call and Put options that have the same expiration and strike price. | 0-255 |
| 8 | CommodityCode | UInt16 | 2 | Numerical identifier of the Underlying This field, together with the InstrumentGroup forms the unique identifier of the message The Commodity Definition and the Series Definition Extended are retrieved through this field which links them to the Class Definition Eg. eg. 2005 (HKB). | |
| 10 | Filler | String | 2 | | |
| 12 | PriceQuotationFactor | Int32 | 4 | Implies the contracted value of the product / series | Decimal places determined from Class Definition field "DecimalInContractSize" |
| 16 | ContractSize | UInt32 | 4 | Number of Underlying entities per contract. | Decimal places determined from Class Definition field "DecimalInContractSize" |
| 20 | DecimalInStrikePrice | UInt16 | 2 | Number of implicit decimals in the strike price. | |
| 22 | DecimalInContractSize | UInt16 | 2 | Number of implicit decimals in the Contract Size and the Price Quotation Factor fields. | |
| 24 | DecimalInPremium | UInt16 | 2 | The number of decimals used in Price fields | |
| 26 | RankingType | UInt16 | 2 | This identifies how the instrument is ranked. | 1 Price, Time 2 Inverted Price, Time * 3 Price, Traders before MM, Time * 4 Inverted Price, Traders before MM, Time * 5 Price, MM before Traders, Time * 6 Inverted Price, MM before Traders, Time * 7 Price, Bait before Normal Orders, Time * 8 Inverted Price, Bait before Normal Orders, Time * 11 Price, Own Orders, Time * 12 Inverted Price, Own Orders, Time * |
| 28 | Tradable | UInt8 | 1 | Defines if the instrument is a tradable instrument or not. | 1 Yes 2 No |

| Offset | Field | Format | Len | Description | Values |
|--------------|----------------------|--------|-----|---|---|
| 29 | PremiumUnit4Price | UInt8 | 1 | The premium unit that describes the price unit in the order. | <ul style="list-style-type: none"> 1 Price 2 Yield * 3 Points 4 Yield Diff * 5 IMM Index * 6 Basis Points * 7 Inverted Yield * 8 Percentage of Nominal * 9 Dirty Price * |
| 30 | BaseCurrency | String | 3 | Defines the trading currency for the instrument or the currency for the underlying. The representation of the currency follows the S.W.I.F.T. handbook and ISO 3166 standard. | See Currency Values in section 3.1.2 for full details. |
| 33 | InstrumentClassID | String | 14 | The ASCII representation of the instrument class. | |
| 47 | InstrumentClassName | String | 32 | The full ASCII representation. name_short | |
| 79 | IsFractions | String | 1 | Is the premium internally represented as fractions? | <ul style="list-style-type: none"> Y Yes N No |
| 80 | SettlementCurrencyID | String | 32 | Full descriptive name of the Settlement Currency. The representation of the currency follows the S.W.I.F.T. handbook and ISO 3166 standard. | See Currency Values in section 3.1.2 for full details. |
| 112 | Effective Tomorrow | UInt8 | 1 | This declaration is for series to be traded the next day | <ul style="list-style-type: none"> 0 False 1 True |
| 113 | TickStepSize | Int32 | 4 | Minimum Fluctuation of the product / series | Decimal places determined from Class Definition field 'DecimalInPremium' |
| 117 | Filler | String | 1 | | |
| Total Length | | | 118 | | |

Remark *: denotes that the value is not currently in use for existing products. However, please note that HKEX may use the value anytime when introducing new products

3.7.3 Series Definition Base (303)

Describes basic series information.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------|-------------|--------|-----|--|----------------------------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 303 Series Definition Base |
| 4 | OrderbookID | UInt32 | 4 | <p>Numerical identifier of the order book</p> <p>This is the unique identifier for the series</p> <p>The Combination Definition is retrieved through this field which links it to the Series Definition Base</p> | |
| 8 | Symbol | String | 32 | Short Name | |

| Offset | Field | Format | Len | Description | Values |
|--------------|-----------------------|--------|-----|--|--|
| 40 | FinancialProduct | UInt8 | 1 | Financial Product | 1 Option 2 Forward * 3 Future 4 FRA * 5 Cash * 6 Payment * 7 Exchange Rate * 8 Interest Rate Swap * 9 REPO * 10 Synthetic Box Leg/Reference * 11 Standard Combination 12 Guarantee * 13 OTC General * 14 Equity Warrant * 15 Security Lending * |
| 41 | NumberOfDecimalsPrice | UInt16 | 2 | The number of decimals used in Price fields | |
| 43 | NumberOfLegs | UInt8 | 1 | Number of legs in the series There can be up to 256 legs per series | |
| 44 | StrikePrice | Int32 | 4 | In general, it is the price at which a specific options series can be exercised. Zero implies the Strike Price is not applicable, e.g. for futures contracts. For Combo Series, this field may not have meaning but can be used with other fields such as CommodityCode, ExpirationDate, InstrumentGroup and Modifier to differentiate the series from the others. | Decimal places determined from "DecimallnStrikePrice". Not applicable for Combo Series. |
| 48 | ExpirationDate | String | 8 | Expiry date of the series | YYYYMMDD |
| 56 | DecimallnStrikePrice | UInt16 | 2 | Number of implicit decimals in the strike price. | |
| 58 | PutOrCall | UInt8 | 1 | Identifies whether the series is a put or call type | 0 Undefined 1 Call 2 Put |
| 59 | Filler | String | 1 | | |
| Total Length | | | 60 | | |

Remark *: denotes that the value is not currently in use for existing products. However, please note that HKEX may use the value anytime when introducing new products

3.7.4 Series Definition Extended (304)

Describes series static data.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------|-------------|--------|-----|---------------------|--------------------------------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 304 Series Definition Extended |
| 4 | OrderBookID | UInt32 | 4 | Orderbook ID | 0 If Not Available |

| Offset | Field | Format | Len | Description | Values |
|--------|-------------------|--------|-----|--|--|
| 8 | Symbol | String | 32 | Symbol This is the unique identifier of the message | |
| 40 | Country | UInt8 | 1 | Country Identifier | |
| 41 | Market | UInt8 | 1 | Market Code | See section 7.4 for a list of possible values |
| 42 | InstrumentGroup | UInt8 | 1 | Instrument Group | See section 7.3 for a list of possible values |
| 43 | Modifier | UInt8 | 1 | Expiration date modified. Value is incremented by one each time the instrument is involved in an issue, split, etc. Note that the modifier value can be different for Call and Put options that have the same expiration and strike price. | 0-255 |
| 44 | CommodityCode | UInt16 | 2 | Numerical identifier of the Underlying This is the unique commodity identifier. The Commodity Definition and the Class Definition are retrieved through this field which links them to the Series Definition Extended Eg. HSI | |
| 46 | ExpirationDate | UInt16 | 2 | Expiry date of the series | |
| 48 | StrikePrice | Int32 | 4 | In general, it is the price at which a specific options series can be exercised. Zero implies the Strike Price is not applicable, e.g. for futures contracts. For Combo Series, this field may not have meaning but can be used with other fields such as CommodityCode, ExpirationDate, InstrumentGroup and Modifier to differentiate the series from the others. | Decimal places determined from Class Definition field "DecimalInStrikePrice". Not applicable for Combo Series. |
| 52 | ContractSize | Int64 | 8 | Number of Underlying entities per contract. | Decimal places determined from Class Definition field "DecimalInContractSize" 0 If Not Available |
| 60 | ISINCode | String | 12 | A code which uniquely identifies a specific securities issue (International Securities Identification Number). For more information about ISIN code, see the international standard ISO 3166. | 0 If Not Available |
| 72 | SeriesStatus | UInt8 | 1 | The actual status of the series. | 0 If Not Available 1 Active (both expired and not expired) 2 Suspended (temporarily stopped) 3 Issued 4 Delisted 5 Locked (suspended for trading and post-trade operations) |
| 73 | EffectiveTomorrow | UInt8 | 1 | This declaration is for next day series | 0 False 1 True |

| Offset | Field | Format | Len | Description | Values |
|--------------|----------------------|--------|-----|--|---|
| 74 | PriceQuotationFactor | Int32 | 4 | Implies the contracted value of the product / series | Decimal places determined from Class Definition field "DecimalInContractSize" |
| 78 | PriceMethod | UInt8 | 1 | Specifies the pricing method used for the combo series | 0 If Not Applicable 1 Net Price 2 Net Value |
| 79 | Filler | String | 1 | | |
| 80 | EffectiveExpDate | String | 8 | The effective expiration date is the actual expiration date of the series and will normally be the same as expiration_date_n in the series binary code. The effective expiration date can be changed during the lifetime of the series whereas expiration_date_n will continue to hold the original expiration date. | YYYYMMDD " " 8 blank spaces if not available |
| 88 | DateTimeLastTrading | Int64 | 8 | The last trading date/time of the Series in UTC timestamp (nanoseconds since 1970) precision to the nearest second | UTC Timestamp |
| 96 | DateTimeFirstTrading | Int64 | 8 | The first trading date/time of the Series in UTC timestamp (nanoseconds since 1970) precision to the nearest second For series with a non-zero value in this field, the series will not be tradable on the days before the Date in this field Time in this field can be ignored | 0 – Not Applicable UTC Timestamp |
| Total Length | | | 104 | | |

3.7.5 Combination Definition (305)

Describes a combination orderbook.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------|------------------|--------|-----|--|----------------------------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 305 Combination Definition |
| 4 | ComboOrderbookID | UInt32 | 4 | Numerical identifier of the combination order book | |
| 8 | LegOrderbookID | UInt32 | 4 | This is the orderbook identification number of the leg | |
| 12 | Filler | String | 3 | | |
| 15 | LegSide | String | 1 | Identifies whether the leg within the combination order book is the same side as that defined for the leg in the OrderBook definition Possible values: As Defined or Opposite | B As Defined C Opposite |
| 16 | LegRatio | Int32 | 4 | Relative numbers of bid and ask contracts between the combo legs. | |
| Total Length | | | 20 | | |

3.8 STATUS DATA

3.8.1 Market Status (320)

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|----------------------|--------|-----|---|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 320 Market Status |
| 4 | StateLevel | UInt16 | 2 | Indicates the level which a state applies to | 1 Market 2 Instrument Type 3 Instrument Class 4 Instrument Series 5 Underlying * 99 End of Business Day |
| 6 | Market | UInt8 | 1 | Market Code Populated only if StateLevel = 1, 2 or 3 | See section 7.4 for a list of possible values 0 If Not Available |
| 7 | Instrument | UInt8 | 1 | Instrument Group Populated only if StateLevel = 2, 3 | See section 7.3 for a list of possible values 0 If Not Available |
| 8 | OrderbookID | UInt32 | 4 | Orderbook ID Populated only if StateLevel = 4 | 0 If Not Available |
| 12 | CommodityCode | UInt16 | 2 | Commodity Populated only if StateLevel = 3 or 5 eg. 2005 (HKB). | 0 If Not Available |
| 14 | Filler | String | 2 | | |
| 16 | ActualStartDate | String | 8 | UTC Start Date | YYYYMMDD |
| 24 | ActualStartTime | String | 6 | UTC Start Time. If specified it is a warning and defines the next planned state. | HHMMSS |
| 30 | PlannedStartDate | String | 8 | UTC next planned Date. If specified it is a warning and defines the next planned state. If not specified it is a state change. | YYYYMMDD |
| 38 | PlannedStartTime | String | 6 | UTC next planned time. If specified it is a warning and defines the next planned state. If not specified it is a state change. | HHMMSS |
| 44 | SecondsToStateChange | UInt16 | 2 | Number of seconds to the next state change | 0 If Not Available |
| 46 | State | UInt16 | 2 | Numeric identification of the State Type. | See full list of states within section 7.1 If StateLevel is 4 0 End of the Current State If StateLevel is 1, 2 or 3 0 If Not Available |
| 48 | Priority | UInt8 | 1 | The priority of the State, either the Trading Session State or Instrument Session State. The State Priority is a number between 1-255. | 0 If Not Available |
| 49 | Filler | String | 3 | | |
| Total Length | | | 52 | | |

Remark *: denotes that the value is not currently in use for existing products. However, please note that HKEX may use the value anytime when introducing new products

The Market Status (320) message can be used to derive the active state of a series. Readers should refer to the Developer's Guide for a fuller description.

3.8.2 Series Status (321)

The Series Status message is generated whenever there is a change to suspension indicator or series status, or when the date/time of last trading is changed.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|---------------------|--------|-----|--|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 321 Series Status |
| 4 | OrderbookID | UInt32 | 4 | Numerical identifier of the order book | |
| 8 | SuspensionIndicator | UInt8 | 1 | Indicates if the series is suspended or not. | <ul style="list-style-type: none"> 1 Suspended for trading 2 Not suspended 3 Locked (suspended for trading and post-trade operations) |
| 9 | SeriesStatus | UInt8 | 1 | The actual status of the series. | <ul style="list-style-type: none"> 0 If Not Available 1 Active (both expired and not expired) 2 Suspended (temporarily stopped) 3 Issued 4 Delisted 5 Locked (suspended for trading and post-trade operations) |
| 10 | Filler | String | 2 | | |
| Total Length | | | 12 | | |

3.8.3 Commodity Status (322)

The Commodity Status message is generated whenever a commodity status changes.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|---------------|--------|-----|---|---|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 322 Commodity Status |
| 4 | CommodityCode | UInt16 | 2 | Underlying definitions are defined by each exchange. Commodity Code is a part of the Series definition. eg. 2005 (HKB). | |
| 6 | Suspended | String | 1 | Defines if the commodity is suspended or not. | <ul style="list-style-type: none"> Y Yes N No |
| 7 | Locked | UInt8 | 1 | Specifies if the underlying is locked for post-trade operations | <ul style="list-style-type: none"> 1 Yes 2 No |
| Total Length | | | 8 | | |

3.9 ORDER BOOK DATA

During auction session, Add Order (330) messages will not be transmitted until the completion of auction but Aggregate Order Book Update (353) will be disseminated to provide change in price depth as other trading sessions.

3.9.1 Add Order (330)

All outstanding orders of the VCM series will be transmitted through Add Order messages and the interval of this transmission is set at one second. The OrderbookID is unique per series but will not increment consecutively. Unique orders are identified by OrderBookID, Side and OrderID. The OrderBookPosition identifies the rank of an order when compared to other orders within the orderbook for each series.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|-------------------|--------|-----------|--|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 330 Add Order |
| 4 | OrderbookID | UInt32 | 4 | Uniquely identifies a series available for trading | |
| 8 | OrderID | UInt64 | 8 | Unique identifier per series and side for each order performed within the trading system | Values may not be consecutive |
| 16 | Price | Int32 | 4 | Price | Decimal places determined from Series Definition Base field 'NumberOfDecimalsPrice' |
| 20 | Quantity | UInt32 | 4 | Number of contracts | |
| 24 | Side | UInt8 | 1 | Side of the order | 0 Bid 1 Offer |
| 25 | LotType | UInt8 | 1 | Lot Type | Lot Type. Values: 0 Undefined * 1 Odd Lot * 2 Round Lot 3 Block Lot * 4 All or None Lot * |
| 26 | OrderType | UInt16 | 2 | Order Type Bitmap | Additional order attributes. Values: 0 Not applicable 1 Force * 2 Short Sell * 4 Market Bid * 8 Price Stabilization * 16 Override Crossing * 32 Undisclosed * 1024 Fill-and-kill immediately * 2048 Firm color disabled * 4096 Convert to aggressive (if locked market)* 8192 Bait/implied order <i>NOTE: Applicable types may be defined by the marketplace.</i> <i>NOTE 2: This field is a bit map. Multiple values may be set simultaneously.</i> |
| 28 | OrderBookPosition | UInt32 | 4 | Order rank information for the order position within the order book for each series | Integer |
| Total Length | | | 32 | | |

Remark *: denotes that the value is not currently in use for existing products. However, please note that HKEX may use the value anytime when introducing new products

3.9.2 Aggregate Order Book Update (353)

Aggregate Order Book Update (353) messages will be sent whenever there is a orderbook change within the top 5 price levels.

For series with pre-auction periods the best bid and ask may both be equal to the calculated opening price or may be equal to Null if the book is uncrossed but with market orders present. See Examples 5 and 6 in Section 5 for details.

Refer to Section 5 - Aggregate Order Book Management for details on the Aggregate Order Book Update message.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|-------------------|-------------------|--------|-----------------------|--|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 353 Aggregate Order Book Update |
| 4 | Orderbook ID | UInt32 | 4 | Uniquely identifies a series available for trading | |
| 8 | Filler | String | 3 | | |
| 11 | NoEntries | UInt8 | 1 | Number of book entries within the message | |
| 12 | AggregateQuantity | UInt64 | 8 | Aggregated number of shares. | |
| 20 | Price | Int32 | 4 | Price | Decimal places determined from Series Definition Base field 'NumberOfDecimalsPrice' Null when PriceLevel is 255 |
| 24 | NumberOfOrders | UInt32 | 4 | Number of orders | |
| 28 | Side | UInt8 | 1 | Side of the order | 0 Bid 1 Offer |
| 29 | Filler | | 1 | | |
| 30 | PriceLevel | UInt8 | 1 | Indicates the price level (within top 5) of the information carried in the message | 1 to 5 Level 2 Orderbook Liquidity |
| 31 | UpdateAction | UInt8 | 1 | Type of market data update action | 0 New 1 Change 2 Delete 74 Clear |
| 32 | Filler | String | 4 | | |
| Total Length..... | | | 12 + 24n _o | | |

(n_o = value of NoEntries)

3.9.3 Quote Request (336)

The Quote Request message is generated whenever market participants request a new quotation.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|-------------------|--------------|--------|-----|--|---------------------------------|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message. | 336 Quote Request |
| 4 | OrderbookID | UInt32 | 4 | Order book ID | |
| 8 | NumberOfLots | Int32 | 4 | Number of Lots | |
| 12 | BidAskFlag | UInt8 | 1 | Indicates if the quote request is for a Bid or Ask (or both) | 0 Bid 1 Ask 2 Bid and Ask |
| 13 | Filler | String | 3 | | |
| Total Length..... | | | 16 | | |

3.10 TRADE AND PRICE DATA

3.10.1 Trade (350)

The Trade message is generated each time a trade has been performed.

The executions may be marked as non-printable. If a participant is looking to use the data in trade tickers or volume calculations, it is recommended that participants ignore messages marked as non-printable to prevent double counting.

- NOTE 1: Combination orders on the book that execute will always be represented by this message.
- NOTE 2: Combination orders that execute will have the Printable flag set to "N". The trades that occur in the legs of the Combo Series will be printable. This avoids double counting of the combination order and its leg trades. Leg trades will be published with the Trade message.

Important messages: Clients who intend to calculate trade statistics based on Trade (350) and Trade Amendment (356) are advised NOT to compare their internally aggregated turnover with those carried in the Trade Statistics (360) as they may not be synchronized given the former trade messages are disseminated on streaming basis whilst the latter statistics are published on snapshot basis. Please also note that price information (Open/High/Low/Last Price) provided in Trade Statistics (360) cannot be derived from Trade (350) and Trade Amendment (356). For information only, the price information excludes (i) Reported Trade (DealType = 4 or above); and (ii) Leg Trades from Combo versus Combo Trades (where OrderID being zero and not "Occurred at Cross" (i.e. DealType not = 2 or 3). However, it is important to note that the stated behavior is not exclusive to Leg Trades from Combo versus Combo Trades and thus cannot be used to differentiate such Leg Trades).

Message Fields

| Offset | Field | Format | Len | Description | Values |
|-------------------|----------------|--------|-----|---|---|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message | 350 Trade |
| 4 | OrderbookID | UInt32 | 4 | Orderbook ID | |
| 8 | OrderID | UInt64 | 8 | Order ID | 0 If Not Available |
| 16 | Price | Int32 | 4 | Traded Price | Decimal places determined from Series Definition Base field 'NumberOfDecimalsPrice' Null means N/A |
| 20 | TradeID | UInt64 | 8 | Match ID | |
| 28 | ComboGroupID | UInt32 | 4 | Used to group combo and leg executions together | |
| 32 | Side | UInt8 | 1 | Side of Orderbook ID | 0 Not Available 1 Not Defined * 2 Buy Order 3 Sell Order |
| 33 | DealType | UInt8 | 1 | Deal Type Bitmap | 0 None 1 Printable (see note) 2 Occurred at Cross 4 Reported Trade |
| 34 | TradeCondition | UInt16 | 2 | The condition in which a trade was executed. Field is a Bitmap. | 0 None 1 Late Trade * 2 Internal Trade / Crossing 8 Buy Write * 16 Off Market * |
| 36 | DealInfo | UInt16 | 2 | Information Bitmap of a deal. | 0 None 1 Reported Trade |
| 38 | Filler | String | 2 | | |
| 40 | Quantity | UInt64 | 8 | The quantity being matched in this execution. | |
| 48 | TradeTime | UInt64 | 8 | Date and time of the last trade in UTC timestamp (nanoseconds since 1970) precision to the nearest 1/100 th second | UTC Timestamp |
| Total Length..... | | | 56 | | |

Remark *: denotes that the value is not currently in use for existing products. However, please note that HKEX may use the value anytime when introducing new products

Note : The Printable property does not apply to Reported Trade. For other trades, i.e. trades matched by the trading system, it will be OFF for trades on Combo Series to avoid double counting of the trade quantity to market turnover.

3.10.2 Trade Amendment (356)

Represents a trade amendment or cancellation.

Notes:

- If a traded quantity is reduced then a Trade Amendment (356) delete (“TradeState” = 3) will be sent followed by a rectification (“TradeState” = 2).
- Price and Quantity may not be available for cancelled trades which have already been given up (“TradeState” = 1) and in such cases the Price and Quantity will be NULL and 0 respectively. Please ignore these values.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|--------------|--------|-----|---|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message | 356 Trade Amendment |
| 4 | TradeID | UInt64 | 8 | Match ID | |
| 12 | ComboGroupID | UInt32 | 4 | Used to group combo and leg executions together | |
| 16 | Price | Int32 | 4 | Traded Price | Decimal places determined from Class Definition field 'DecimalInPremium' "NULL" means not available |
| 20 | Quantity | UInt64 | 8 | Defines number of contracts in a deal. | "0" means not available if Price is "NULL" |
| 28 | TradeTime | UInt64 | 8 | Date and time of the last trade in UTC timestamp (nanoseconds since 1970) precision to the nearest 1/100 th second | UTC Timestamp |
| 36 | TradeState | UInt8 | 1 | Trade State | 1 Deleted (with given up trade). The trade has been deleted. 2 Rectified. The trade has been rectified. 3 Deleted (Normal). The trade has been deleted. |
| 37 | Filler | | 3 | | |
| Total Length | | | 40 | | |

3.10.3 Trade Statistics (360)

Trade information for completed deals. The trade information or statistics information carried in this message type is provided on snapshot basis.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------|-------------|--------|-----|--------------------------------|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message | 360 Trade Statistics |
| 4 | OrderbookID | UInt32 | 4 | Uniquely identifies the series | |
| 8 | Price | Int32 | 4 | Last Traded Price | Decimal places determined from Class Definition field 'DecimalInPremium' |
| 12 | DealSource | UInt8 | 1 | Deal Source of the last trade | See full list of deal sources at the end of this document Deal Source in Trade Statistics is only applicable to cases where Turnover is higher than that in the previous Trade Statistics for the same series, and should be ignored otherwise. |

| Offset | Field | Format | Len | Description | Values |
|--------------------|-------------------|--------|-----|--|--|
| 13 | Session | UInt8 | 1 | Session indicator used to distinguish between the T and T+1 sessions | 0 Statistics for T Session 1 Statistics for T+1 Session |
| 14 | Filler | | 2 | | |
| 16 | AggregateQuantity | Int64 | 8 | Volume – total within the latest deal(s) | |
| 24 | Open | Int32 | 4 | Price of the first committed Trade in the series during the respective Session | Decimal places determined from Class Definition field 'DecimalInPremium' |
| 28 | High | Int32 | 4 | Highest price of normal trades in the session. This is calculated after all Trades, Trades Cancellations and Trade Corrections have been taken into account. | Decimal places determined from Class Definition field 'DecimalInPremium' |
| 32 | Low | Int32 | 4 | Lowest price of normal trades in the session. This is calculated after all Trades, Trades Cancellations and Trade Corrections have been taken into account. | Decimal places determined from Class Definition field 'DecimalInPremium' |
| 36 | Filler | | 4 | | |
| 40 | TradeReportVolume | UInt64 | 8 | Total volume of reported trades for the respective Session | |
| 48 | DealCount | UInt32 | 4 | Number of deals completed in the respective Session | |
| 52 | Turnover | UInt64 | 8 | Cumulative volume for the respective Session | |
| Total Length | | | 60 | | |

3.10.4 Calculated Opening Price (364)

The Calculated Opening Price (COP) message indicates an instrument's theoretical opening price during the pre-opening phases of the market (prior to an auction). A COP message is generated when the indicative matching price or volume varies. If the Price set to Null, the COP is no longer applicable.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|------------------------|--------|-----|-------------------------------------|---|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message | 364 Calculated Opening Price |
| 4 | OrderbookID | UInt32 | 4 | Order book ID | |
| 8 | CalculatedOpeningPrice | Int32 | 4 | Calculated Opening Price | Decimal places determined from Series Definition Base field 'NumberOfDecimalsPrice' |
| 12 | Filler | | 4 | | |
| 16 | Quantity | UInt64 | 8 | Shows the quantity available at COP | |
| Total Length | | | 24 | | |

3.10.5 Estimated Average Settlement Price (365)

The Estimated Average Settlement (EAS) Price message is generated for selected securities at regular intervals throughout the business day. Clients should ignore EAS message if the EAS value is NULL.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|----------------|--------|-----|--|---|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message | 365 EAS Message |
| 4 | EASType | String | 1 | Determines whether this is a Stock EAS or Index EAS message | E Stock EAS Other Values: Please refer to the Index Code tables in the appendix. |
| 5 | InstrumentCode | String | 20 | Corresponds to the 'UnderlyingCode' field within the Commodity Definition message. | Stock EAS: 5 digit security codes with possible values 1 – 99999 Other Values: The Index Provider's Index Code. Please refer to the Index Code tables in the appendix. |
| 25 | EAS | Int64 | 8 | Estimated Average Settlement Value | 2 implied decimal places |
| 33 | Filler | String | 3 | | |
| Total Length | | | 36 | | |

3.11 NEWS

3.11.1 Market Alert (323)

The Market Alert message is generated periodically to relay market announcements and alerts. The "AlertID" and "Source" fields provide a unique key for any given announcement. "Source" field indicates the origin of the alert message. For each "Source", "AlertID" is unique on every business day.

If the size of a single announcement is greater than the maximum supported packet size, then the alert will be sent as multiple messages, each with the same 'AlertID'. These Market Alert (323) messages are disseminated sequentially each with the "LastFragment" field set to "N" except the last message within the alert which has the 'LastFragment' field set to 'Y'. Note that in this case the Header field would be the same for all messages within this announcement.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------|--------------|--------|-----|--|--|
| 0 | MsgSize | UInt16 | 2 | Size of the message | |
| 2 | MsgType | UInt16 | 2 | Type of message | 323 Market Alert |
| 4 | AlertID | UInt16 | 2 | The reference ID for this alert, unique for any given day | |
| 6 | Source | String | 1 | Source ID for this alert message | H Market Alerts sent through the Trading System M Other Market Alerts |
| 7 | Filler | String | 1 | | |
| 8 | Header | Binary | 320 | Header. In the case of multiple fragments, this field is only populated in the first fragment. | Unicode UTF-16LE encoded If Header starts with [C], the Market Alert is in Chinese. Otherwise, it is in English. |
| 328 | LastFragment | String | 1 | Indicates whether this message is the last in a sequence of messages. | Y Complete N Not complete |
| 329 | InfoType | UInt8 | 1 | Information Type | 0 Not Specified 1 Company Announcement 2 Market Message 3 Static Line 4 Notice Received |
| 330 | Priority | UInt8 | 1 | Priority | 0 Not Specified 1 Low priority 2 Medium priority 3 High priority 4 Critical priority |
| 331 | NoLines | UInt8 | 1 | Maximum 3 lines | |

| Offset | Field | Format | Len | Description | Values |
|--------------------|---------|--------|-----------------|---|--|
| 332 | Content | Binary | 320 | Market Alert Content – number of occurrences according to 'NoLines' field | Unicode UTF-16LE encoded The language will be either English or Chinese |
| Total Length | | | 332 + 320 n_p | | |

(n_p = value of NoLines)

3.12 CLEARING INFORMATION

The trading system disseminates clearing related information.

3.12.1 Open Interest (366)

Typically issued at start of day to show the Previous Day settlement price and open interest (for products not tradable in After-Hours Trading session) information. It is also issued around the end of day with the Current Day information. In the event of corrections, this message will be resent.

Message Fields

| Offset | Field | Format | Len | Description | Values |
|--------------------|--------------|--------|-----|---|--|
| 0 | MsgSize | Uint16 | 2 | Size of the message | |
| 2 | MsgType | Uint16 | 2 | Type of message | 366 Open Interest |
| 4 | DayIndicator | Uint16 | 2 | Session indicator used to distinguish between the previous and current business days for Settlement, GrossOI and NetOI* | 0 Current Business Day 1 Previous Business Day |
| 6 | Filler | | 6 | | |
| 12 | OrderbookID | Uint32 | 4 | Uniquely identifies the series | |
| 16 | Settlement | int32 | 4 | If DayIndicator = 1, Settlement Price determined after the T session of the previous business day if DayIndicator = 0, Settlement Price determined after the T session of the current business day | Decimal places determined from Class Definition field 'DecimalInPremium' "NULL" means not available |
| 20 | DealCount | Uint32 | 4 | Deal Count of the current business day up to the close of the T session | |
| 24 | GrossOI | int32 | 4 | If DayIndicator = 1, Gross Open Interest right before the start of the current business day if DayIndicator = 0, Gross Open Interest up to the close of the T session of the current business day | "NULL" means not available |
| 28 | NetOI | int32 | 4 | If DayIndicator =1, Net Open Interest right before the start of the current business day If DayIndicator = 0, Net Open Interest up to the close of the T session of the current business day | "NULL" means not available |
| 32 | Turnover | Uint64 | 8 | Turnover of the current business day up to the close of the T session | |
| Total Length | | | 40 | | |

* A business day starts when the market opens in the morning and ends at the close of the after hour trading session of the day.

4. RECOVERY

OMD provides three different mechanisms for recovering missed data:

Line arbitration – using dual multicast channels (Line A and Line B)

Retransmission Server – recovery of a limited number of messages

Refresh Server – snapshot of current market state

These mechanisms should be used as described in the following table.

Table 2: Recovery Mechanisms

| Event | Action |
|---|--|
| Packet lost on one either Line A or Line B | Try to recover data from the other line with a configurable timeout (“arbitration mechanism”). |
| Dropped packet(s) on both Line A and Line B | Recover dropped message(s) from the Retransmission Server. |
| Late start up or extended intraday outage | Wait for a refresh of the current market state and then continue with real time messages. |

4.1 GAP DETECTION

Each packet provides the sequence number (SN) of the first message it contains. This sequence number starts at 1 and increases with each subsequent message.

The sequence numbers provided in every packet header is calculated by adding the previous sequence number and the message count, as shown in table below:

Table 3: Sequence Number Calculation

| Packet | Sequence Number | Message Count |
|----------|-----------------|---------------|
| Packet 1 | 1 | 4 |
| Packet 2 | 5 | 2 |
| Packet 3 | 7 | 1 |
| Packet 4 | 8 | 3 |
| Packet 5 | 11 | 1 |

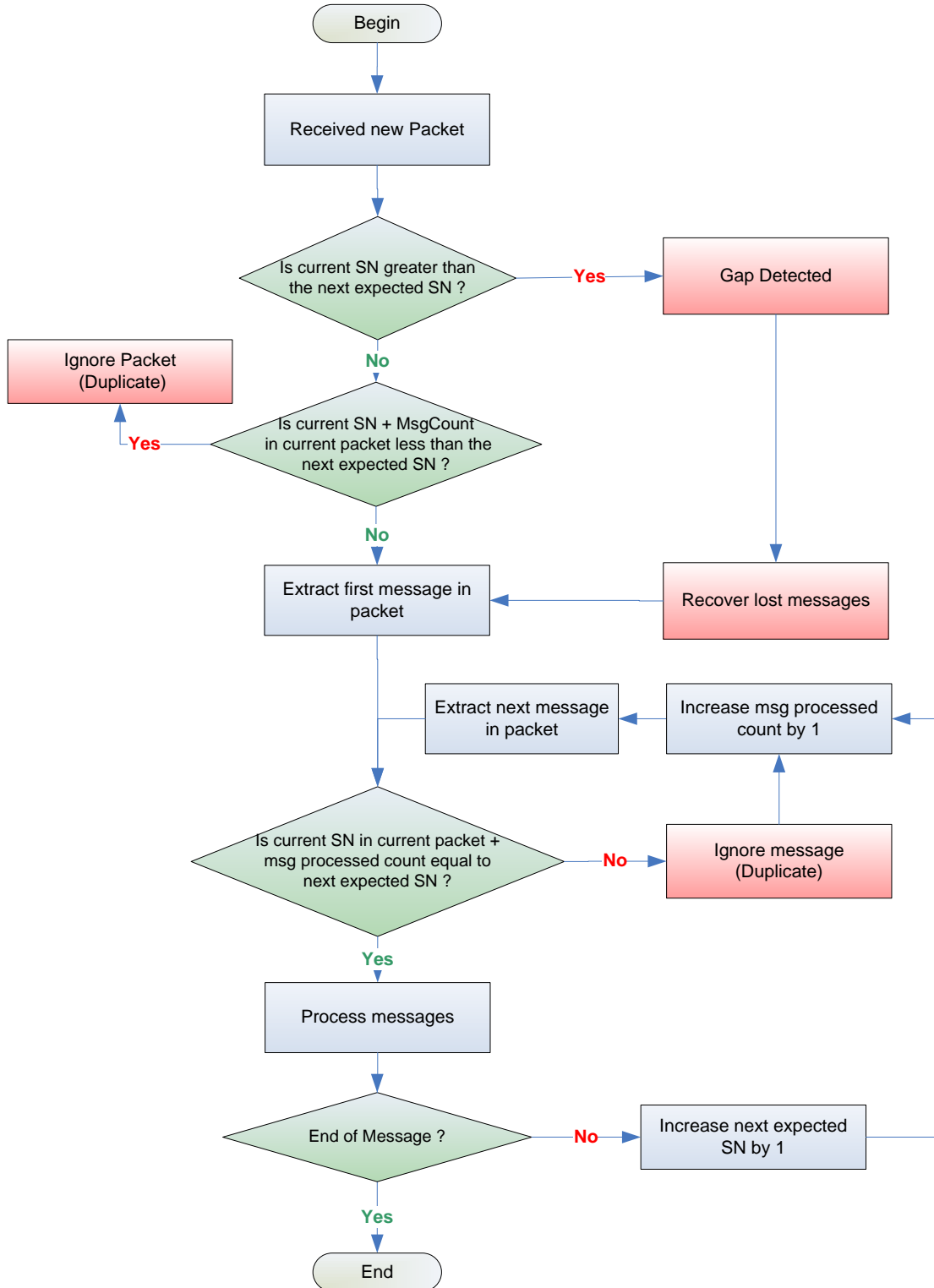
If the client drops the first five packets they would request a gap fill for messages 1-11.

All messages conform to the message level sequencing. Each channel has its own sequence number. This allows recipients to detect gaps or duplicates in each message sequence number and, if appropriate, reconcile them (line arbitration) with the primary or secondary multicast groups or request retransmission of the missing / corrupted messages.

Users should use this sequence number to detect gaps in the transmission of messages.

The following diagram illustrates how the message sequence number should be used to detect gaps in the feed.

Figure 2: Gap Detection using the Sequence Number (SN)



4.2 LINE ARBITRATION

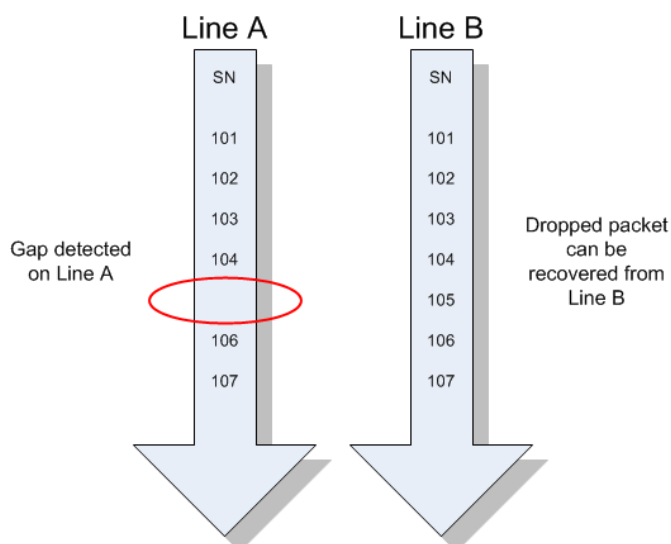
Client applications should check the sequence number (SN) and message count (MC) for every packet received. SNs are unique and increase monotonically for each service, the MC indicates the number of messages within each packet.

Line A and Line B are identical in terms of:

- SNs
- Messages that are sent
- Sequence in which messages are sent

However it is not guaranteed that a packet content between Line A and Line B will be the same. For example the third packet of the day from the Line A could contain SN 10 with MC 3, whereas the third packet of the day from Line B could contain SN 9 with MC 4. For this reason clients must arbitrate on SN (at the message level) rather than packet content. Client applications should listen to both Line A and Line B in real-time. Clients should look at packets coming from both lines and process the ones that arrive first, regardless of whether they came from Line A or Line B. It is advisable to apply the “first come – first served” rule.

Figure 3 – Detecting Missing Packets



Additional Notes;

- The above example of a dropped packet is a simplified example assuming 1 message per packet, in reality each packet is likely to contain multiple messages
- Whilst the order of individual messages between Line A and Line B will be identical, there is no guarantee that the packets will contain exactly the same messages.
- In the example below, three packets are sent on each line, but message 'OrderUpdate3' appears in one packet from Line A but in the subsequent packet on Line B.

Figure 4 – Normal Message Delivery

| Primary | | | Secondary | | |
|--|----|-----|-----------|----|--|
| Messages | MC | SN | SN | MC | Messages |
| OrderUpdate1 OrderUpdate2 OrderUpdate3 | 3 | 101 | 101 | 2 | OrderUpdate1 OrderUpdate2 |
| Trade1 OrderUpdate4 | 2 | 104 | 103 | 3 | OrderUpdate3 Trade1 OrderUpdate4 |
| Trade2 Statistics 1 | 2 | 106 | 106 | 2 | Trade2 Statistics 1 |

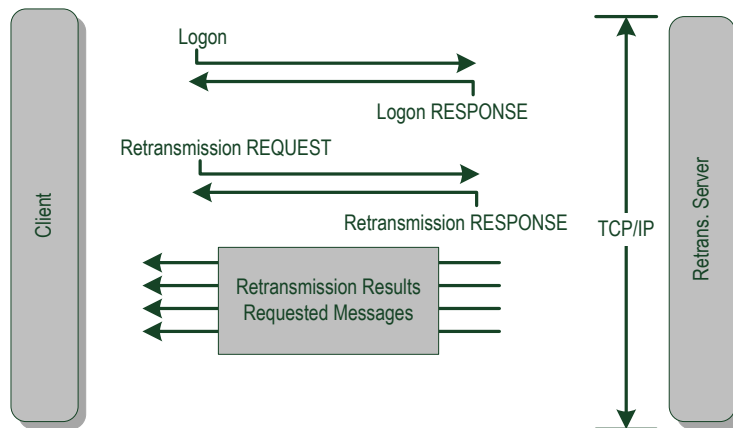
4.3 RETRANSMISSION SERVICE

The retransmission service is provided via the TCP/IP protocol and is designed to allow clients to recapture a small number of missed messages already published on the real time channels.

It is not intended that clients use the retransmission server to recover data after long outages or on late start up (in these situations, clients should use the Refresh service). To that end, it aims to support the retransmission of the data covering the market activities for the last 15-30 seconds only. This figure is indicative only and may be shorter than 15 seconds if a spike happens in the market. The sequence range of messages that a client can request and the number of retransmission requests permitted per day is also limited.

The following diagram illustrates the message flow during a retransmission session:

Figure 5: Retransmission Request



Logon

The client establishes a TCP/IP connection and initiates a session by sending the Logon message. Once the client is authenticated the server will respond immediately with the Logon Response message. If the client does not send a Logon message within the logon timeout interval, the server will close the connection.

Logons may be rejected for the following reasons:

- Invalid username
- User already connected

In all cases the server will close the connection after sending the Logon Response message.

Making a request

The client can make a retransmission request by sending the Retrans Request message. The server will respond with a Retrans Response message to indicate whether the request has been accepted or not.

In the case of a successful request the server will send the requested messages immediately after the Retrans Response message.

The sequence numbers will be the same as when they were first sent on the real time multicast channel. The framing of the retransmitted messages into a packet may differ from the original transmission.

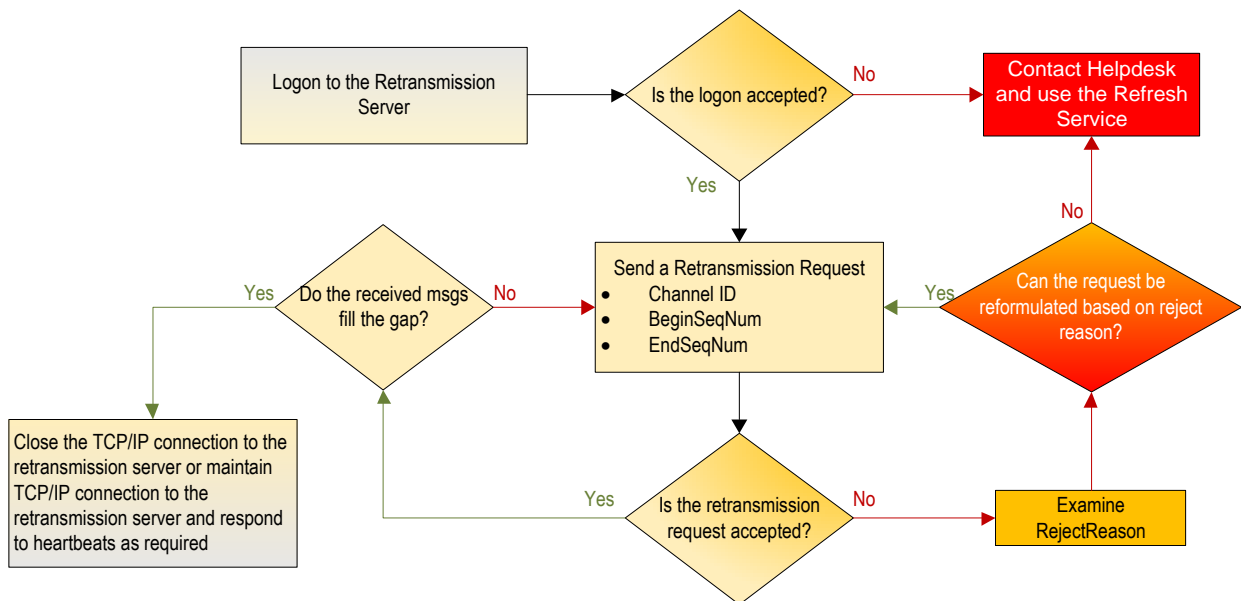
Retransmission requests may be rejected for the following reasons:

- Unknown channel ID or illegal (not authorized)
- Messages not available
- Exceeds maximum sequence range
- Exceeds maximum requests in a day

In the case where the client has exceeded the maximum number of requests allowed in a day, the server will close the connection after sending the Retrans Response message.

The following diagram is a guideline of the flow of logic when making a request:

Figure 6: Requesting Dropped Packets



Multiple requests and concurrent sessions

Clients can send multiple requests during a session and can keep the session open during idle periods by responding to heartbeats sent by the server. Concurrent sessions however will not be supported. Each user can only have one session open at a time.

If a client makes multiple requests, the server will process them serially. Clients are unable to cancel outstanding requests.

Heartbeats

To determine the health of the user connection on the TCP/IP channel, the Retransmission Server will send regular heartbeat packets to the user. The heartbeat frequency is 30 seconds. The client application must respond with a

“Heartbeat Response” packet. The time out for this heartbeat response packet is set at 5 seconds. If no response is received by the server within this timeframe, the TCP/IP session will be disconnected.

Figure 7: Retransmission Server Heartbeat Message



A “heartbeat response” packet consists in an exact copy of the incoming heartbeat packet.

Closing the session

Sessions should be terminated by gracefully closing the TCP/IP connection.

System limits

The system limits mentioned above are set as follows:

| System Limit | Value |
|--|--------|
| Maximum sequence range that can be requested | 10,000 |
| Maximum number of requests per day | 1,000 |
| Logon timeout (seconds) | 5 |
| Heartbeat interval (seconds) | 30 |
| Heartbeat response timeout (seconds) | 5 |

Please note that the maximum number of requests per day limit is across all channels.

High availability

For each site, two sets of IP address and port are provided for the retransmission service in order to facilitate high availability. Clients may connect to both retransmission servers at the start of the day and maintain the connection during the day by responding to heartbeats.

In the event that Retransmission Server A (RTS A) does not respond to a logon or retransmission request, Retransmission Server B (RTS B) should be used.

In the event of a failure of either RTS A or RTS B, there may be a short period of unavailability. This failure should be detected by clients through the loss of connection. In this case the other RTS should be used.

RTS B should not be used as a means of requesting from two sources at the same time.

Disaster recovery

Two sets of backup IP address and port are also provided for the disaster site’s retransmission service.

During normal conditions the retransmission service at the disaster site is not available. If clients attempt to connect, this will fail.

In the unlikely event of a disaster recovery situation, the retransmission service at the disaster site will be brought up and clients may connect via the backup IP addresses and ports.

4.4 REFRESH SERVICE

The refresh service is designed to allow clients to recover from a large scale data loss. This can happen after a late start or during a major outage.

Synchronization is on a per channel basis. For each real time multicast channel (besides those for Trade, Trade Amendment & Quote Request which are not recoverable via the Refresh service), there exists a corresponding refresh multicast channel on which snapshots of the market state are sent at regular intervals throughout the business day. No ordering should be assumed between the various different data types unless otherwise stated – this is due to the nature of using multiple different multicast channels for refresh.

Snapshot

A snapshot of the market state is described in the table below.

| Message | Snapshot description |
|-------------------|--|
| Series Definition | <p>A full list of all series, which includes any series modifications or series additions made intraday. The order is sent as:</p> <ul style="list-style-type: none"> • Commodity Definition (301) • Class Definition (302) • Series Definition Base (303) • Series Definition Extended (304) • Combination Definition (305) <p><i>NOTE: Next day series are not currently included in Refresh</i> <i>NOTE: Within the Refresh Service the Commodity Definition (301), Class Definition (302) and Series Definition Extended (304) are sent on a dedicated multicast channel and arrive in that order. Similarly the Series Definition Base (303) and Combination Definition (305) are sent on a dedicated multicast channel and arrive in that order.</i></p> |
| Market Status | The most recent Market Status message(s) – see section 3.8.1 for details about interpreting market state |
| Series Status | The most recent Series Status message(s) for each series |
| Commodity Status | The most recent Commodity Status message(s) for each commodity |
| Orders | The latest level 2 orderbook via Aggregate Order Book messages and full order book provided for VCM series via Add Order messages |
| Trade Statistics | The latest Trade Statistics messages for each series in the T and T+1 sessions <i>NOTE : T+1 may not be available for all series and in any event would not be available until after the T+1 session has begun</i> |
| EAS | The latest Estimated Average Settlement message for each underlying received (both for Cash and Index EAS) |
| COP | The latest Calculated Opening Price message for each series |
| Market Alert | Latest 400 Market Alert messages at the maximum since start of day |
| Open Interest | Potentially up to 2 latest Open Interest message for each series – one for the Current Trading Day, and one for the Previous Trading Day |

The ordering of refresh messages types within the multicast channels is detailed below;

| Channel | Refresh Sequence |
|--------------------|--|
| Static Base | Series Definition Base (303), Combination Definition (305) |
| Order | Calculated Opening Price (364), Aggregate Order Book Update (353), Add Order (330) |
| Static Extended | Commodity Definition (301), Class Definition (302), Series Definition Extended (304) |
| Series Status | Market Status (320), Commodity Status (322), Series Status (321) |
| Trade Statistics | Trade Statistics (360) |
| Market Alert | Market Alert (323) |
| Open Interest | Open Interest (366) |
| Implied Volatility | Implied Volatility (367) |
| EAS Price | EAS Price (365) |

Refresh complete

A Refresh Complete message is sent at the end of a snapshot indicating the sequence number with which the snapshot is synchronized.

Snapshot processing

Below is an overview of the steps to carry out in order to process a channel snapshot.

- Subscribe to the real time multicast channel and cache received messages.
- Subscribe to the corresponding refresh multicast channel and discard messages until the Refresh Complete message is received.
- Process received messages until the next Refresh Complete message is received.
- Store the LastSeqNum sequence number provided in the Refresh Complete.
- Unsubscribe to the refresh multicast channel.

- Discard the cached real time messages with sequence number less than or equal to LastSeqNum.
- Process the remaining cached real-time messages and resume normal processing.

Missed messages

The retransmission server does not support refresh channels. If a client misses messages, it must wait for the next snapshot. Similarly if a client starts listening during the middle of a snapshot, it must wait for the next snapshot.

4.4.1 Refresh of Market Status

Many of the messages within refresh are singular per series and the refresh server simply published the most recent value. However the refresh data will typically include a number of Market Status (320) messages will be sent together and correct processing users should refer the description given in Section 3.8.1 to determine the correct Trading and Instrument Session State.

5. AGGREGATE ORDER BOOK MANAGEMENT

Book Identification

A book is uniquely identified by either OrderbookID or Symbol and there is a 1-to-1 map between these two identifiers. OrderbookID is an integer representation of 4 bytes, Symbol is a longer String representation (32 bytes) that gives the short name. Whilst Symbol is 'readable' to human users, the OrderbookID is more efficient when sending market data updates, and therefore OrderbookID is the identifier that appears on every series related message.

Partial Price Depth

The price level within the Aggregate Order Book Update message determines the number of price levels the order price is away from the best price for a given order book. An order with price level 1 means the order's price is the best price, a price level of 2 will be used for orders at the next best price, etc.

OMD provides a view of multiple price depths of aggregate order book for the Derivatives Market. This view can be visualized as a number of rows in a table for each of the bid and ask sides. On each side there are a number of rows showing the aggregate quantity available at a number of price levels. Below is using 10 price depths of aggregate order book as the example.

| Bid Side | | | Ask Side | | |
|------------|-------------------|-------|----------|-------------------|------------|
| PriceLevel | AggregateQuantity | Price | Price | AggregateQuantity | PriceLevel |
| 1 | 700 | 9730 | 9760 | 500 | 1 |
| 2 | 350 | 9720 | 9770 | 300 | 2 |
| 3 | 150 | 9710 | 9780 | 100 | 3 |
| 4 | 250 | 9700 | 9790 | 150 | 4 |
| 5 | 100 | 9690 | | | |

Book Updates

Book update messages are generated by OMD as delta messages defined in section 5 (**Aggregate Order Book Update (353)**). Each message may contain any combination of new, changed, deleted or orderbook clear entries for a book. The nature of an entry is defined by its UpdateAction.

New, to create/insert a new price level

Delete, to remove a price level

Change, to update aggregate quantity at a price level

Orderbook Clear, to inform users that all price levels should be cleared

Example 1 – Quantity Reduction and Explicit Addition

For example suppose the Ask order at price level 9770 is reduced in quantity and at the same time a new order is added at price level 9850, then the following message is sent;

| Offset | Field Name | Value |
|--------|-------------------|-----------|
| 0 | MsgSize | 60 |
| 2 | MsgType | 53 |
| 4 | OrderbookID | 1234 |
| 8 | Filler | NULL |
| 11 | NoEntries | 2 |
| 12 | AggregateQuantity | 200 |
| 20 | Price | 9770 |
| 24 | NumberOfOrders | 1 |
| 28 | Side | 1 (Offer) |
| 30 | PriceLevel | 2 |

| | | |
|----|-------------------|-----------|
| 31 | UpdateAction | 1 |
| 32 | Filler | NULL |
| 36 | AggregateQuantity | 300 |
| 44 | Price | 9850 |
| 48 | NumberOfOrders | 1 |
| 52 | Side | 1 (Offer) |
| 54 | PriceLevel | 5 |
| 55 | UpdateAction | 0 |
| 56 | Filler | NULL |

The resulting book should now be as follows:

| Bid Side | | | Ask Side | | |
|------------|-------------------|-------|----------|-------------------|------------|
| PriceLevel | AggregateQuantity | Price | Price | AggregateQuantity | PriceLevel |
| 1 | 700 | 9730 | 9760 | 500 | 1 |
| 2 | 350 | 9720 | 9770 | 200 | 2 |
| 3 | 150 | 9710 | 9780 | 100 | 3 |
| 4 | 250 | 9700 | 9790 | 150 | 4 |
| 5 | 100 | 9690 | 9850 | 300 | 5 |

Example 2 – Implicit Level Adjustments

The client must adjust the price level of entries below deleted or inserted entries. Potential level adjustments must be carried out after each single entry in Aggregate Order Book message.

For example, if a bid order with price 9740 and quantity 50 is added to the order book above, it will cause the following message to be sent:

| Offset | Field Name | Value |
|--------|-------------------|---------|
| 0 | MsgSize | 36 |
| 2 | MsgType | 53 |
| 4 | OrderbookID | 1234 |
| 8 | Filler | NULL |
| 11 | NoEntries | 1 |
| 12 | AggregateQuantity | 50 |
| 20 | Price | 9740 |
| 24 | NumberOfOrders | 1 |
| 28 | Side | 0 (Bid) |
| 30 | PriceLevel | 1 |
| 31 | UpdateAction | 0 |
| 32 | Filler | NULL |

After processing this message, the client's book should look as follows:

| Bid Side | | | Ask Side | | |
|------------|-------------------|-------|----------|-------------------|------------|
| PriceLevel | AggregateQuantity | Price | Price | AggregateQuantity | PriceLevel |
| 1 | 50 | 9740 | 9760 | 500 | 1 |
| 2 | 700 | 9730 | 9770 | 200 | 2 |
| 3 | 350 | 9720 | 9780 | 100 | 3 |

| | | | | | |
|---|-----|------|------|-----|---|
| 4 | 150 | 9710 | 9790 | 150 | 4 |
| 5 | 250 | 9700 | 9850 | 300 | 5 |

Price levels of the other 4 Bid orders must all be incremented although there will not be Aggregate Order Book Update messages sent for the increment.

Example 3 – Implicit Deletions

If a new book entry causes the bottom entry of a book to be shifted out of the book (i.e. more than 4 price levels away from the best price), the client must delete the excess entry. If the book shrinks again, OMD resends the entries that have temporarily fallen out.

For example, if a bid order with price 9750 and quantity 250 is added to the book above, it will cause the following message to be sent:

| Offset | Field Name | Value |
|--------|-------------------|---------|
| 0 | MsgSize | 60 |
| 2 | MsgType | 53 |
| 4 | OrderbookID | 1234 |
| 8 | Filler | NULL |
| 11 | NoEntries | 1 |
| 12 | AggregateQuantity | 250 |
| 20 | Price | 9750 |
| 24 | NumberOfOrders | 1 |
| 28 | Side | 0 (Bid) |
| 30 | PriceLevel | 1 |
| 31 | UpdateAction | 0 |
| 32 | Filler | NULL |

After processing this message, the client’s book should look as follows:

| Bid Side | | | Ask Side | | |
|------------|-------------------|-------|----------|-------------------|------------|
| PriceLevel | AggregateQuantity | Price | Price | AggregateQuantity | PriceLevel |
| 1 | 250 | 9750 | 9760 | 500 | 1 |
| 2 | 50 | 9740 | 9770 | 200 | 2 |
| 3 | 700 | 9730 | 9780 | 100 | 3 |
| 4 | 350 | 9720 | 9790 | 150 | 4 |
| 5 | 150 | 9710 | 9850 | 300 | 5 |

Price 9750 and quantity 250 is added according to the message.

Price 9700 and quantity 250 must be deleted by the client.

Example 4 – Explicit Additions

If orders are removed so that there are now less than 5 levels visible then the server will also automatically send the additional level(s) that are now revealed.

For example, if the bid order with price 9750 and quantity 250 is now removed from the book above and this reveals an 6th level which needs to be disseminated then it will cause the following message to be sent:

| Offset | Field Name | Value |
|--------|------------|-------|
| 0 | MsgSize | 60 |

| | | |
|----|-------------------|---------|
| 2 | MsgType | 53 |
| 4 | OrderbookID | 1234 |
| 8 | Filler | NULL |
| 11 | NoEntries | 2 |
| 12 | AggregateQuantity | 250 |
| 20 | Price | 9750 |
| 24 | NumberOfOrders | 1 |
| 28 | Side | 0 (Bid) |
| 30 | PriceLevel | 1 |
| 31 | UpdateAction | 2 |
| 32 | Filler | NULL |
| 36 | AggregateQuantity | 250 |
| 44 | Price | 9700 |
| 48 | NumberOfOrders | 1 |
| 52 | Side | 0 (Bid) |
| 54 | PriceLevel | 5 |
| 55 | UpdateAction | 0 |
| 56 | Filler | NULL |

The resulting order book should now be;

| Bid Side | | | Ask Side | | |
|------------|-------------------|-------|----------|-------------------|------------|
| PriceLevel | AggregateQuantity | Price | Price | AggregateQuantity | PriceLevel |
| 1 | 50 | 9740 | 9760 | 500 | 1 |
| 2 | 700 | 9730 | 9770 | 200 | 2 |
| 3 | 350 | 9720 | 9780 | 100 | 3 |
| 4 | 150 | 9710 | 9790 | 150 | 4 |
| 5 | 250 | 9700 | 9850 | 300 | 5 |

Example 5 – Market Orders

Market Orders may arrive during the Pre-Open period. If a Calculated Opening Price (COP) is available then the COP Merge will be applied – see example 7 below. However, if a COP is not available (i.e. the order book is not crossed or not locked), then the following table demonstrates how market orders are displayed.

- One or more market orders are present on the Bid side (since Price Level 1 has a Price which is Null)
- There are no market orders present on the Ask side (since Price Level 1 has a Price which is not Null)
- The best bid limit order is at price 9710 and the best ask limit order is at 9720

| Bid Side | | | Ask Side | | |
|------------|--------------------|-------|----------|--------------------|------------|
| PriceLevel | Aggregate Quantity | Price | Price | Aggregate Quantity | PriceLevel |
| 1 | 7900 | Null | 9720 | 8200 | 1 |
| 2 | 7700 | 9710 | 9730 | 2000 | 2 |
| 3 | 6800 | 9700 | 9740 | 1000 | 3 |
| 4 | 2000 | 9690 | 9750 | 1500 | 4 |
| 5 | 200 | 9680 | 9860 | 8000 | 5 |

Example 6 – COP Merge

Certain series involve a Pre-Open period during which the order book may be merged to the calculated opening price. Taking the previous example as a starting point, imagine the following two events occur;

- A new bid limit order arrives at price 9720, quantity 1000
- A COP message arrives for price 9720

These two events mean that the order book will be modified as shown below. Note the following points;

- Best bid and best ask are now the same price (this is the COP price)
- Any limit orders at or better than the COP are aggregated into Price Level 1 and this includes any market orders
- Remaining limit orders which are worse than the COP are shown at Price Level 2 and below

| Bid Side | | | Ask Side | | |
|------------|--------------------|-------|----------|--------------------|------------|
| PriceLevel | Aggregate Quantity | Price | Price | Aggregate Quantity | PriceLevel |
| 1 | 8900 | 9720 | 9720 | 8200 | 1 |
| 2 | 7700 | 9710 | 9730 | 2000 | 2 |
| 3 | 6800 | 9700 | 9740 | 1000 | 3 |
| 4 | 2000 | 9690 | 9750 | 1500 | 4 |
| 5 | 200 | 9680 | 9860 | 8000 | 5 |

Example 7 – Orderbook Clear

In certain failure scenarios the system may send an 'Orderbook Clear' message at which point clients should clear both Bid and Ask side orderbooks for the specified series. An example message is shown below.

Following an 'Orderbook Clear' message any existing orders for the series will be resent as normal to rebuild the current image.

| Field | Value |
|-------------------|--------|
| MsgSize | 36 |
| MsgType | 353 |
| Orderbook ID | 123456 |
| Filler | — |
| NoEntries | 1 |
| AggregateQuantity | 0 |
| Price | 0 |
| NumberOfOrders | 0 |
| Side | 0 |
| Filler | — |
| PriceLevel | 0 |
| UpdateAction | 74 |
| Filler | — |

6. AUCTION PERIOD SPECIAL HANDLING

During the pre-opening period the level 2 orderbook messages that are disseminated are adjusted so that clients do not receive a crossed orderbook. Note that this special handling only applies to instruments which have an opening auction.

During auction session before the matching takes place, a Calculated Opening Price (364) message will be disseminated in the event of crossed book. In such case, Aggregated Order Book Update (353) messages will be sent to update the top price level of both sides with Price equalling the COP, Number of Orders and Aggregate Quantity including those of market orders and orders at prices at or better than the COP. See Example 6 in the previous section for details. If the book is not crossed then a Calculated Opening Price (364) will not be available, and any market orders will be displayed at Price Level 1. See Example 6 in the previous section for details.

In this way:

- D-Lite clients will receive orderbook messages that are adjusted using 'COP Merge' (details below)

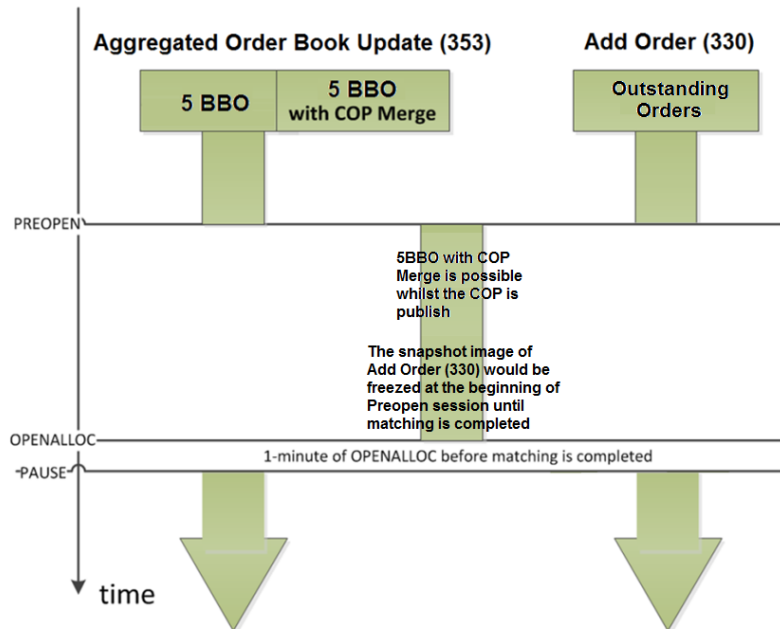


Figure 1 - Auction Period

COP Merge

Bids with a price that is the same or greater than the COP will be shown at the COP price.

Offers with a price that is the same or less than the COP will be shown at the COP price.

7. HKEX DERIVATIVES MARKET

7.1 MARKET STATES

The market states and types of trading activity can take place in each state are explained below:

“AHT” means After Hour Trading Session.

| Status Code | Symbol | Description | Applicability | Session |
|-------------|-------------------|--|--|---------|
| 1 | OPENALLOC | OPEN ALLOCATION SESSION | Markets with Pre-market Opening Period | Regular |
| 2 | CLOSE | MARKET CLOSED | Markets not tradable in T+1 Session | Regular |
| 3 | OPEN | MARKET OPEN | All markets | Regular |
| 4 | PREOPEN | PREOPEN SESSION | Markets with Pre-market Opening Period | Regular |
| 5 | PREOPENALLOC | PREOPEN ALLOCATION SESSION | Markets with Pre-market Opening Period | Regular |
| 6 | PAUSE | MARKET PAUSE | All markets | Regular |
| 7 | PRE_MKT_ACT | PRE-MARKET ACTIVITIES | Markets without Pre-market Opening Period | Regular |
| 8 | CL_START | CLEARING SESSION START | Markets tradable in T+1 Session | All |
| 9 | CL_CLOSE | CLEARING SESSION CLOSED | All markets | All |
| 10 | AHT_CLOSE | AHFT MARKET CLOSED | Markets tradable in T+1 Session | AHT |
| 11 | AHT_CLR_INFO | AHFT RESET PRICE INFORMATION | Markets tradable in T+1 Session | AHT |
| 12 | AHT_INACT_T_ORDER | AHFT INACTIVE NON T+1 ORDER | Markets tradable in T+1 Session | AHT |
| 13 | AHT_NEXT_DAY | AHFT RESET PRICE INFORMATION FOR NEXT BUSINESS DAY | Markets tradable in T+1 Session | AHT |
| 14 | AHT_OPEN | AHFT MARKET OPEN | Markets tradable in T+1 Session | AHT |
| 15 | AHT_OPEN_PL | AHFT MARKET OPEN | Markets tradable in T+1 Session with price limit control enabled | AHT |
| 16 | AHT_PRE_MKT_ACT | AHFT PRE-MARKET ACTIVITIES | Markets tradable in T+1 Session | AHT |
| 17 | OPEN_PL | MARKET OPEN | Markets enabled with price limit control | Regular |
| 18 | CLOSE_TODAY | MARKET CLOSED FOR TODAY TRADING | Markets tradable in T+1 Session | Regular |
| 19 | OPEN_DPL | MARKET OPEN | Markets enabled with dynamic price banding mechanism | Regular |
| 20 | FAILOVER | SITE FAILOVER | All markets | All |
| 21 | CLOSE_TODAY_E | MARKET CLOSED FOR TODAY TRADING | Markets tradable in T+1 Session For London Metal Mini Futures Market (Market Code 120) only | Regular |
| 22 | AHT_CLOSE_E | AHFT MARKET CLOSED | Markets tradable in T+1 Session For London Metal Mini Futures Market (Market Code 120) only | AHT |

| | | | | |
|----|--------------------|--------------|---|---------|
| 23 | OPEN_DPL_VCM | MARKET OPEN | Markets enabled with dynamic price banding mechanism and VCM control | Regular |
| 24 | OPEN_VCM | MARKET OPEN | Markets enabled with VCM control | Regular |
| 25 | VCM_COOL_OFF_DPL * | VCM COOL OFF | Instrument Series State, VCM cool off status with dynamic price banding mechanism | Regular |
| 26 | VCM_COOL_OFF * | VCM COOL OFF | Instrument Series State, VCM cool off status | Regular |
| 27 | RESET_VCM | RESET VCM | Reset counter for VCM trigger time | Regular |
| 28 | HALT | HALT | Trading Halt | AHT |

* Applicable to Instrument Series (i.e. StateLevel 4) only

7.2 DEAL SOURCES

The deal source codes are explained below:

| Deal Source Code | Description |
|------------------|---|
| 1 | Matched by system, automatically. |
| 2 | Matched by system, manually. |
| 3 | Matched outside exchange, different brokers. |
| 4 | Matched outside exchange, different brokers, reg. by exchange. |
| 5 | Matched outside exchange, one broker. |
| 6 | Matched outside exchange, one broker, reg. by exchange. |
| 7 | Combination order matched against another combination order when matched by the Exchange, electronically. |
| 20 | Deal made at the end of an auction. |
| 36 | Tailor-made combination |
| 43 | Combo versus Outright |

7.3 LIST OF INSTRUMENT CODE

| Instrument Code | Description |
|-----------------|--|
| 4 | Futures |
| 6 | Call (American style) |
| 7 | Put (American style) |
| 22 | Call (European style) |
| 23 | Put (European style) |
| 170 | Options Straddle |
| 171 | Options Strangle |
| 172 | Standard Combo Series for Stock Options Market (SOM) – Synthetic Futures |
| 201 | Time Spread (level = 01) |
| 202 | Time Spread (level = 02) |
| 203 | Time Spread (level = 03) |
| 204 | Time Spread (level = 04) |
| 205 | Time Spread (level = 05) |
| 206 | Time Spread (level = 06) |
| 207 | Time Spread (level = 07) |
| 208 | Time Spread (level = 08) |
| 209 | Time Spread (level = 09) |
| 210 | Time Spread (level = 10) |
| 211 | Time Spread (level = 11) |
| 212 | Time Spread (level = 12) |
| 213 | Time Spread (level = 13) |
| 214 | Time Spread (level = 14) |
| 215 | Time Spread (level = 15) |
| 216 | Time Spread (level = 16) |
| 217 | Time Spread (level = 17) |
| 218 | Time Spread (level = 18) |
| 219 | Time Spread (level = 19) |
| 220 | Time Spread (level = 20) |
| 221 | Time Spread (level = 21) |
| 222 | Time Spread (level = 22) |
| 223 | Time Spread (level = 23) |
| 250 | Tailor-made combination |
| 254 | Exchange Rate |
| 255 | Payment Currency |

Note:

Instrument codes 4, 6, 7, 22 and 23 are for normal series whereas the remaining are for Combo Series.

7.4 LIST OF MARKET ID

| Market ID | Description | Market Type |
|-----------|---|-------------|
| 1 | CESC Index Futures / Options | Non-SOM |
| 2 | Stock Futures | Non-SOM |
| 3 | Three-Year Exchange Fund Note Futures | Non-SOM |
| 8 | Gold Futures | Non-SOM |
| 16 | Mini Hang Seng Index Futures / Options | Non-SOM |
| 20 | Stock Options | SOM |
| 24 | HIBOR | Non-SOM |
| 27 | Dividend Futures | Non-SOM |
| 34 | Hang Seng Index Futures / Options | Non-SOM |
| 35 | Flexible Hang Seng Index Options | Non-SOM |
| 37 | Flexible Hang Seng China Enterprises Index Options | Non-SOM |
| 38 | Hang Seng China Enterprises Index Futures / Options | Non-SOM |
| 51 | HSI Volatility Index Futures | Non-SOM |
| 60 | Sector Index Futures | Non-SOM |
| 70 | Renminbi Currency Futures / Options | Non-SOM |
| 93 | IBOVESPA Index Futures | Non-SOM |
| 96 | S&P BSE Sensex Index Futures | Non-SOM |
| 99 | FTSE/JSE Top 40 Index Futures | Non-SOM |
| 102 | MICEX Index Futures | Non-SOM |
| 108 | MSCI AxJ Futures | Non-SOM |
| 115 | Physically Settled CNH Gold Futures | Non-SOM |
| 116 | Physically Settled USD Gold Futures | Non-SOM |
| 117 | MOF T-Bond Futures | Non-SOM |
| 118 | Iron Ore Futures | Non-SOM |
| 120 | London Metal Mini Futures | Non-SOM |
| 122 | Cash-Settled RMB Currency Futures | Non-SOM |
| 125 | Cash-Settled CNHUSD Futures | Non-SOM |

7.5 LIST OF INDEX CODE FOR INDEX EAS

Index Provider's Index Code for Index EAS

| Index Code | Name of the Index | EAS Type |
|------------|---------------------------------------|----------|
| 0001400 | Hang Seng China Enterprises Index | H |
| 0000100 | Hang Seng Index | H |
| 0000101 | HSI Sub Indices – Finance | H |
| 0000102 | HSI Sub Indices – Utilities | H |
| 0000103 | HSI Sub Indices – Property | H |
| 0000104 | HSI Sub Indices – Commerce & Industry | H |