INTERFACE SPECIFICATIONS

HKEx Orion Market Data Platform
Securities Market & Index Datafeed Products
Binary Protocol

Version v1.1
31 July 2012

DOCUMENT HISTORY

Distribution Version

Version	Date of Issue	Comments
V1.0	20 April 2012	First Distribution Issue
V1.1	31 July 2012	Revised Edition with the following updates; - Add additional notes on Sections 3.4.2, 3.9.7 - Section 3.7.2 – refine encoded method used in SecurityNameGB & SecurityNameGCCS to be Unicode UTF-16LE & the values for Style & NoUnderlyingSecurities - Sections 3.9.1 – 3.9.6 – align to industry practice to use " 0 for bid, 1 for offer" instead of "1 for buy & 2 for sell" - Section 3.9.6 – add new UpdateAction "74 for Orderbook Clear" for the clients to clear their aggregate order books - Section 3.12.1 – provision for Chinese Exchange news - Section 4.1 – refine the diagram for gap detection mechanism - Section 5 – elaborate more on aggregate order book management with more typical examples for different book operations

CONTENTS

1. INTRODUCTION

1.1 PURPOSE

This document specifies the Binary interface of the HKEx Orion Market Data Platform ("OMD").

This document is the Transmission Specification(s) of the relevant Datafeed(s) under your Market Data Vendor Licence Agreement or the Market Data End-User Licence Agreement ("Licence Agreement"). Please refer to Section 1.2 and the summary table at Section 1.3 for the information applicable to the Datafeed(s) under your Licence 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 Licence 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

Appendix A: List of Indices under OMD Index

All chapters and appendices except Chapter 3 and Appendix A are applicable to all Datafeeds unless otherwise specified. In Chapter 3, there are indications* in individual sections/sub-sections for their applicability to individual Datafeeds, and Appendix A is applicable to OMD Index only. The information is also summarised in Section 1.3 Summary Table.

* Example

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.3	●	●	●	●

1.3 SUMMARY TABLE

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.1	●	●	●	●
3.2	●	●	●	●
3.3	●	●	●	●
3.4	●	●	●	●
3.5	●	●	●	●
3.6	●	●	●	●
3.7	●	●	●	
3.8	●	●	●	
3.9.1			●	
3.9.2			●	
3.9.3			●	
3.9.4	▲	▲	▲	
3.9.5	▲	▲	▲	
3.9.6	●	●		
3.9.7	●	▲		
3.10.1		●	●	
3.10.2		●	●	
3.10.3	●			
3.10.4	●	●		
3.10.5	●	●		
3.10.6	●	●	●	
3.11.1	●	●		
3.11.2	●	●		
3.11.3	●	●		
3.12	●	●		
3.13				●

● The information supplied in the corresponding sub-section applies to the Datafeed(s)

▲ Complimentary service to the Datafeed(s).

2.   SYSTEM OVERVIEW

2.1   SCOPE

Figure 1: Access to Market Data
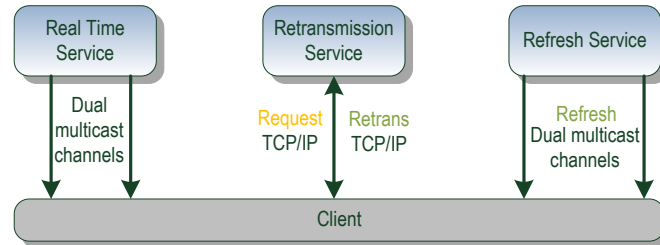
OMD provides market data represented in an efficient binary message format for all instruments listed on the Securities Market. It has been designed for high throughput and low latency.

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.

2.2.1 Start of Day

OMD will normally be brought up around 1:30am. This start up time, however, is not rigid and the Exchange has the right to adjust this time according to the different trading situations.

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 reference data for all markets, securities, liquidity providers and currency rates is published each day shortly after the start of day.

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 2 seconds on each channel when there is no activity.

2.2.3 End of Day

OMD will normally be shut down at 6:30pm. This shutdown time, however, is not rigid and the Exchange has the right to adjust this time according to the different trading situations.

At the end of the business day, the server will stop sending messages (including heartbeats) on each channel. This is normally at 6:30pm.

2.2.4 Error Recovery

2.2.4.1 System Component Failure

If a system component fails 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 site.

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

A Sequence Reset message will be sent on each channel when OMD is brought up. This will be followed by a snapshot of each channel. After the snapshot the market data feed will return to normal operations.

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.

2.4 RACE CONDITIONS

The information supplied in this section does not apply to OMD Index.

Due to the nature of the exchange matching system 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 Security Status (21) message is sent marking a security as suspended, however for a very short time after this message, the regular order and trade information for this security may continue to arrive. As a second example the Trading Session Status (20) messages and market activity are also decoupled; e.g. for a short time after a TradingSesStatus of "Halted" is reported realtime data for that same market may continue to arrive.

3.   MESSAGE FORMATS

3.1   DATA TYPES

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.1	●	●	●	●

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

Format	Description
String	ASCII characters which are left aligned and null padded.
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.

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 Int64 fields of the Index Data (71) message.

The Int64 null representation is 0x8000000000000000 (Hex 2's complement) or -9223372036854775808 (Decimal).

3.1.2 Currency Values

See the ISO-4217 Currency Codes for a full list of possible data values. Currently the system uses the following codes; 'HKD' – Hong Kong dollars,'USD' – US dollars, 'EUR' – Euro, 'JPY' – Japanese Yen, 'GBP' – United Kingdom Sterling, 'CAD' – Canadian Dollars, 'SGD' – Singapore Dollars, 'CNY' – Chinese Renminbi. HKEx may add or delete currency code(s), whenever applicable, in the future.

3.2   PACKET STRUCTURE

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.2	●	●	●	●

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.

Packet Header	Message 1	Message 2	…	Message n

The maximum length of a packet is 1500 bytes.

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: Sequence Reset (100) Logon (101) Logon Response (102) Retransmission Request (201) Retransmission Response (202) Refresh Complete (203) Market Definition (10) Security Definition (11) Liquidity Provider (13) Currency Rate (14) Trading Session Status (20) Security Status (21) Add Order (30) Modify Order (31) Delete Order (32) Add Odd Lot Oder (33) Delete Odd Lot Order (34) Aggregate Order Book Update (53) Broker Queue (54) Trade (50) Trade Cancel (51) Trade Ticker (52) Closing Price (62) Nominal Price (40) Indicative Equilibrium Price (41) Statistics (60) Market Turnover (61) Yield (44) News (22) Index Definition (70) Index Data (71)

3.3   PACKET HEADER

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.3	●	●	●	●

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

Offset	Field	Format	Len	Description
8	SendTime	Uint64	8	The number of nanoseconds since *January 1*, 1970, *00:00:00 GMT*, precision is provided to the nearest millisecond.
Packet length			16	

3.4   CONTROL MESSAGES

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.4	⬤	⬤	⬤	⬤

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 intraday in case of a disaster recovery.

The client must ignore 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 subscribed channels. Whenever the Sequence Reset message is received, clients need to clear all cached data and subscribe to the refresh channels to receive the current state of the market thereupon the clients can resume processing real-time messages with sequence number greater than the LastSeqNum in the Refresh Complete message (Section 3.6.1)."

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		

3.5   RETRANSMISSION

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.5	⬤	⬤	⬤	⬤

Refer to Retransmission service for details on the retransmission messages.

3.5.1   Logon (101)

The Logon message enables client authentication. This is not required for multicast channels and is only used to 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	
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 **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 with 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	

Offset	Field	Format	Len	Description	Values
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

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.6	⬤	⬤	⬤	⬤

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.

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

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.7	⬤	⬤	⬤	

3.7.1 Market Definition (10)

The Market Definition message is generated at the start of the business day for each market segment.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	`10` Market Definition

4	MarketCode	String	4	Market code	`MAIN`
					`GEM`
					`NASD`
					`ETS`

8	MarketName	String	25	Market Name	Alphanumerical
33	CurrencyCode	String	3	Base currency code of the market.	See Currency Values in section 3.1.2 for full details.
36	NumberOfSecurities	Uint32	4	Number of securities within the market	
Total Length			40		

3.7.2 Security Definition (11)

This Security Definition message contains all the reference data for a security.

Security Definition messages may be received intraday – for example the 'FreeText' field may be updated during the day.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	`11` Security Definition
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values `1 – 99999`

8	MarketCode	String	4	Market code	`MAIN`
					`GEM`
					`NASD`
					`ETS`

12	ISINCode	String	12	ISIN code of the security.	
24	InstrumentType	String	4	Instrument type of the security.	`BOND` Bonds `BWRT` Basket Warrants `EQTY` Equities `TRST` Trusts `WRNT` Warrants & structured products (DW & CBBC)
28	SpreadTableCode	String	2	Spread table code of the security.	Spread table as per Second Schedule of Rules of the Exchange: `01` Part A `03` Part B
30	SecurityShortName	String	40	Security short name	
70	CurrencyCode	String	3	Security currency code of the market.	See Currency Values in section 3.1.2 for full details.
73	SecurityNameGCCS	Binary	60	Security name in Traditional Chinese using Unicode	**Unicode UTF-16LE encoded**
133	SecurityNameGB	Binary	60	Security name in Simplified Chinese using Unicode	**Unicode UTF-16LE encoded**
193	LotSize	Uint32	4	Board lot size for the security	
197	PreviousClosingPrice	Int32	4	Previous closing price of the security	3 implied decimal places
201	Filler	String	1		
202	ShortSellFlag	String	1	Indicator for short-sell authorization.	`Y` Short-sell allowed `N` Short-sell not allowed

Offset	Field	Format	Len	Description	Values
203	Filler	String	1		
204	CCASSFlag	String	1	Indicates whether or not the security is a CCASS security	Y CCASS security N Non CCASS security
205	DummySecurityFlag	String	1	Dummy Security Flag.	Y Dummy security N Normal security
206	TestSecurityFlag	String	1	Test Security Flag	Y Test security N Normal security
207	StampDutyFlag	String	1	Indicator for stamp duty requirement	Y Stamp duty required N Stamp duty not required
208	Filler	String	1		
209	ListingDate	Uint32	4	Date of security listing	The representation is YYYYMMDD Value is 19000101 for unknown listing date
213	DelistingDate	Uint32	4	Date of security delisting	The representation is YYYYMMDD. Value is 0 if no date exists.
217	FreeText	String	38	Free text associated to the security	Fixed length array of free text. When there is no free text, spaces will be present instead.
Bonds Specific Data					
255	EFNFlag	String	1	EFN Indicator	Y EFN N Non-EFN
256	AccruedInterest	Uint32	4	Accrued interest of the security.	3 implied decimal places
260	CouponRate	Uint32	4	Coupon rate of a bond security	3 implied decimal places
Warrants, Basket Warrants and Structured Product specific data					
264	ConversionRatio	Uint32	4	Conversion ratio for Structured Product with stock underlying only	3 implied decimal places
268	StrikePrice	Int32	4	Strike price of the security.	3 implied decimal places
272	MaturityDate	Uint32	4	Date of maturity of a warrant or structured security	The representation is YYYYMMDD
276	CallPutFlag	String	1	Indicator of whether the warrant or structured product is a call or put option	For Derivative Warrants/Basket Warrants: C Call P Put For ELI & CBBC: C Bull P Bear / Rang
277	Style	String	1	Style of the basket warrant	A American style E European style <blank> Other
278	NoUnderlyingSecurities	Uint16	2	Number of underlying security codes within this message	0 to 20 for Basket Warrants 0 to 1 for Warrants and Structured Product
280	UnderlyingSecurityCode	Uint32	4	5-digit code identifying the underlying security.	
284	UnderlyingSecurityWeight	Uint32	4	The weight of the underlying security code.	
Total Length		$280 + 8n_U$			

(n_U = value of NoUnderlyingSecurities)

Note: ClosingPrice may be set to 0, for example after an IPO (no existing previous closing price)

3.7.3 Liquidity Provider (13)

The Liquidity Provider message is generated at the start of the business day for securities that have at least one liquidity provider.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	`13` Liquidity Provider
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values `1 – 99999`
8	NoLiquidityProviders	Uint16	2	Number of liquidity providers within this message.	`1` to `50`
10	LPBrokerNumber	Uint16	2	Broker number of the liquidity provider	
Total Length		$10 + 2n_T$			

(n_T = value of NoLiquidityProviders)

3.7.4 Currency Rate (14)

The Currency Rate message provides the foreign exchange conversion rates between various foreign currencies and the Hong Kong dollar.

The Currency Factor and Currency Rate fields should be interpreted as below:

   For example if 1 Euro is valued 10.22 HKD
   - Currency Factor will be 0 (1 EUR)
   - Currency Rate will be 102200 *(4 decimals implied)*

   For example if 1000 Japanese Yen is worth 90.678 HKD
   - Currency Factor will be 3 *(1000 JPY)*
   - Currency Rate will be 906780 (*4 decimals implied*)

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	`14` Currency Rate
4	CurrencyCode	String	3	Currency code.	See Currency Values in section 3.1.2 for full details.
7	Filler	String	1		
8	CurrencyFactor	Uint16	2	Currency factor conversion.	A non-zero value n means all price fields for this security should be interpreted as a value equal to the price multiplied by 10^n.
10	Filler	String	2		
12	CurrencyRate	Uint32	4	Currency rate	Rate, expressed in HKD for one foreign currency unit. 4 decimals implied.
Total Length		16			

3.8 STATUS DATA

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.8	⬤	⬤	⬤	

3.8.1 Trading Session Status (20)

The Trading Session Status provides information on the status of a market segment. It is sent whenever there is change of trading session.

This message may be sent on a separate multicast channel to order and trade data and therefore may not be synchronized.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**20** Trading Session Status
4	MarketCode	String	4	Market segment identifier	**MAIN** **GEM** **NASD** **ETS**
8	TradingSessionID	Uint8	1	Identifies the trading session.	**1** Day
9	TradingSessionSubID	Uint8	1	Trading session sub-identifier.	**0** Day Close (DC) **1** Pre-trading (Order Input OI) **2** Opening or Opening Auction (Matching MA) **3** Continuous trading (Continuous CT) **7** Quiescent (Blocking BL) **100** Not Yet Open (NO) **101** No Cancel/Modification (NC) **102** Exchange Intervention (EI) **103** Close (CL) **104** Order Cancel (OC)
10	TradingSesStatus	Uint8	1	Status of the current trading session.	**0** Unknown (for NO) **1** Halted (for BL, EI) **2** Open (for OI, NC, MA, CT, OC) **3** Closed (for CL) **100** Day Closed (for DC)
11	TradingSesControlFlag	String	1	Indicates how control of trading session and sub-session transitions are performed.	**'0'** Automatic (Default) **'1'** Manual (this invalidates the normal schedule for the day)
12	Filler	String	4		
16	StartDateTime	Uint64	8	Start time of the trading status	The data is provided as number of nanoseconds since Unix epoch Jan 1st 1970. Set to **0** if no time is available.
24	EndDateTime	Uint64	8	End time of the trading status	The data is provided as number of nanoseconds since Unix epoch Jan 1st 1970. Set to **0** if no time is available.
Total Length			32		

3.8.2 Security Status (21)

The Security Status message is generated

- At the start of the business day if the security is halted.
- Whenever a security state changes.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	

Offset	Field	Format	Len	Description	Values
2	MsgType	Uint16	2	Type of message.	21 Security Status
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	SecurityTradingStatus	Uint8	1	Indentifies the trading status of a security.	2 Trading Halt 3 Resume
9	Filler	String	3		
Total Length			12		

3.9 ORDER BOOK DATA

The full order book information is not available in auction session until the completion of auction.

3.9.1 Add Order (30)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.1			●	

The Add Order message is generated when a new order is inserted into the order book. The Price will be set to 0 for Market orders. The OrderId is unique per security but will not increment consecutively.

Note for Securities instruments the OrderBookPosition is always set to zero.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	30 Add Order
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	OrderId	Uint64	8	Unique identifier per security for each order performed within the trading system	Values may not be consecutive
16	Price	Int32	4	Price	3 implied decimal places
20	Quantity	Uint32	4	Number of shares	
24	Side	Uint16	2	Side of the order	0 Bid 1 Offer
26	OrderType	String	1	Order type	'1' Market '2' Limit
27	Filler	String	1		
28	OrderBookPosition	Int32	4	Order rank information for the order position within the order book for each security	Integer
Total Length			32		

3.9.2 Modify Order (31)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.2			●	

The Modify Order message is generated when an existing order identified by the OrderId is modified. The only attribute that can be modified is the quantity.

Note for Securities instruments the OrderBookPosition is always set to zero.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	31 Modify Order
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	OrderId	Uint64	8	Unique identifier per security for each order performed within the trading system	Values may not be consecutive
16	Quantity	Uint32	4	Number of shares	
20	Side	Uint16	2	Side of the order	0 Bid 1 Offer
22	Filler	String	2		
24	OrderBookPosition	Int32	4	Order rank information for the order position within the order book for each security	Integer
Total Length			28		

3.9.3 Delete Order (32)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.3			●	

The Delete Order message is generated when an existing order identified by the OrderId is deleted.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	32 Delete Order
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	OrderId	Uint64	8	Unique identifier per security for each order performed within the trading system	Values may not be consecutive
16	Side	Uint16	2	Side of the order	0 Bid 1 Offer
Total Length			18		

3.9.4 Add Odd Lot Order (33)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.4	▲ **(via complimentary odd lot order feed)**	▲ **(via complimentary odd lot order feed)**	▲ **(via complimentary odd lot order feed)**	

▲ Complimentary service to the Datafeed(s)

The Add Odd Lot Order message is generated when a new odd lot order is inserted into the order book..

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	33 Add Odd Lot Order
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	OrderId	Uint64	8	Unique identifier per security for each order performed within the trading system	Values may not be consecutive
16	Price	Int32	4	Price	3 implied decimal places
20	Quantity	Uint32	4	Number of shares	
24	BrokerID	Uint16	2	Integer identifier uniquely identifying the Broker	Integer
26	Side	Uint16	2	Side of the order	0 Bid 1 Offer
Total Length			28		

3.9.5 Delete Odd Lot Order (34)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.5	▲ (via complimentary odd lot order feed)	▲ (via complimentary odd lot order feed)	▲ (via complimentary odd lot order feed)	

▲ Complimentary service to the Datafeed(s)

The Delete Odd Lot Order message is generated when an existing odd lot order identified by the OrderId is deleted.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	34 Delete Odd Lot Order
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	OrderId	Uint64	8	Unique identifier per security for each order performed within the trading system	Values may not be consecutive
16	BrokerID	Uint16	2	Integer identifier uniquely identifying the Broker	Integer
18	Side	Uint16	2	Side of the order	0 Bid 1 Offer
Total Length			20		

3.9.6 Aggregate Order Book Update (53)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.6	⬤	⬤		

Refer to Section 5 - Aggregate Order Book Management for details on the Aggregate Order Book Update message. The Aggregate Order Book Update message only applies to Board Lots.

For an UpdateAction of '74 – Orderbook Clear' please refer to Example 6 within the Aggregate Order Book Management section 5.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	53 Aggregate Order Book Update
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
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	3 implied decimal places
24	NumberOfOrders	Uint32	4	Number of orders	
28	Side	Uint16	2	Side of the order	0 Bid 1 Offer
30	PriceLevel	Uint8	1	Price level	
31	UpdateAction	Uint8	1	Type of market data update action	0 New 1 Change 2 Delete 74 Orderbook Clear
32	Filler	String	4		

Total Length.. $12 + 24n_O$

(n_O = value of NoEntries)

3.9.7 Broker Queue (54)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.9.7	●	▲ **(via complimentary conflated broker queue feed)**		

▲ Complimentary service to the Datafeed(s)

The Broker Queue message contains the priority list of the (max) top 40 broker IDs for a given side, and is generated whenever any of the entries in the list are modified. Entries are ordered according to distance away from the best price with best price brokers being at the front of the queue. The queue will also include spread level entries between each price level, which are marked when the Type field is set to 'S'. When the Type field is set to 'S', there are two possibilities;

- The Item is non-zero – indicating the number of spread levels away from the best price
- The Item is zero – indicating that there are no brokers with orders at the spread level indicated by the previous entry of Type set to 'S'

Example: if the active offers are as below, and assuming a spread level is 0.01:

Ask Price	Broker ID
20.28	2137
20.28	4138
20.29	2141
20.29	5123
20.31	3145

Then the resulting Ask side Broker Queue will be represented as below:

Entry	1	2	3	4	5	6	7	8	9
Item	2137	4138	1	2141	5123	2	0	3	3145
Type	B	B	S	B	B	S	S	S	B

The Conflated Broker Queue Feed ("CBQ") which is included in SS (OMD Securities Standard), is provided to the Licensee of SP (OMD Securities Premium) as a complimentary service. The service provides broker queue information in conflated mode whilst SP provides market data in streaming mode. The service level between CBQ and SP is therefore inherently different by nature and the information in these two products is not synchronized. Licensed vendors are therefore reminded that if they plan to provide the CBQ along with the market depth available from SP, appropriate disclaimers and warnings should be provided to subscribers highlighting the service level difference.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	54 Broker Queue
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	ItemCount	Uint8	1	This field contains the number of items in the message – repeating items are shown indented below.	0 to 40
9	Side	Uint16	2	Side of the order	1 Buy 2 Sell
11	BQMoreFlag	String	1	Flag indicating if there are more broker numbers in the queue	Y More broker numbers exist in the queue N No more exist
12	Item	Uint16	2	This field contains either the broker number or the number of spreads away from the best price.	
14	Type	String	1	Indicates the type of information contained in the item	B Broker number S Number of Spread
15	Filler	String	1		
Total Length..		12 + 4n_I			

(n_I = value of ItemCount)

3.10 TRADE AND PRICE DATA

3.10.1 Trade (50)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.10.1		●	●	

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

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	50 Trade

Offset	Field	Format	Len	Description	Values
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values `1 – 99999`
8	TradeID	Uint32	4	Unique identifier per security for each trade performed within the trading system. The ID is reset for each trading day.	Starting from `1`, incrementing by 1 for each trade
12	Price	Int32	4	Price	3 implied decimal places
16	Quantity	Uint32	4	Number of shares	
20	TrdType	Int16	2	Public trade type.	`0` Automatch normal (AMS \<space\>) `4` Late Trade (Off-exchange previous day) (AMS "P") `22` Non-direct Off-Exchange Trade (AMS "M") `100` Automatch internalized (AMS "Y") `101` Direct off-exchange Trade (AMS "X") `102` Odd-Lot Trade (AMS "D") `103` Auction Trade (AMS "U") `104` Overseas Trade
22	Filler	String	2		
24	TradeTime	Uint64	8	Time of trade	The number of nanoseconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970 TradeTime precision is currently provided to the nearest second.
Total Length..			32		

3.10.2 Trade Cancel (51)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.10.2		⬤	⬤	

The Trade Cancel message is generated when a trade has been cancelled.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	`51` Trade cancel
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values `1 – 99999`
8	TradeID	Uint32	4	Unique identifier per security for each trade performed within the trading system. The ID is reset for each trading day.	Starting from `1`, incrementing by 1 for each trade
Total Length..			12		

3.10.3 Trade Ticker (52)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.10.3	●			

The Trade Ticker is an aggregation of several trades into one message, combining quantities of subsequent trades made on a given instrument at a given fixed price.

When a trade is cancelled, a Trade Ticker message will be generated with the TickerID set to the ticker which contains the cancelled trade, and with the AggregateQuantity set to remaining quantity outstanding.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	52 Trade ticker
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	TickerID	Uint32	4	Unique identifier per security for each trade ticker generated within the trading system. The ID is unique per security for each trading day.	Starting from 1, incrementing by 1 for each trade ticker
12	Price	Int32	4	Price	3 implied decimal places
16	AggregateQuantity	Uint64	8	Aggregated number of shares.	
24	TradeTime	Uint64	8	Time of trade	The number of nanoseconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970 Tradetime is up to seconds
32	TrdType	Int16	2	Public trade type.	0 Automatch normal (AMS <space>) 4 Late Trade (Off-exchange previous day) (AMS "P") 22 Non-direct Off-Exchange Trade (AMS "M") 100 Automatch internalized (AMS "Y") 101 Direct off-exchange Trade (AMS "X") 102 Odd-Lot Trade (AMS "D") 103 Auction Trade (AMS "U") 104 Overseas Trade
34	TrdCancelFlag	String	1	Indicates that the quantity of the Ticker must be decremented due to a trade cancellation.	Y Cancelled N Not cancelled
35	Filler	String	1		
Total Length...			36		

3.10.4 Closing Price (62)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.10.4	●	●		

The Closing Price message is generated near the end of the business day for each security. If the closing price is not available, the field 'ClosingPrice' is set to 0. Note that the 'NumberOfTrades' field is not populated for SS (OMD Securities Standard) clients.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	62 Closing Price
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	ClosingPrice	Int32	4	Current Day Closing Price	3 implied decimal places
12	NumberOfTrades	Uint32	4	Total Number of Trades performed on the given instrument	
Total Length..			16		

3.10.5 Nominal Price (40)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.10.5	●	●		

The Nominal message may be generated when an order is added, deleted or modified in a book or when trade or trade cancel is performed.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	40 Nominal Price
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	NominalPrice	Int32	4	Nominal price of a security	3 implied decimal places
Total Length			12		

Note: Nominal Price may be 0 in specific cases (e.g. no reference price)

3.10.6 Indicative Equilibrium Price (41)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.10.6	●	●	●	

The Indicative Equilibrium Price (IEP) message indicates an instrument's theoretical opening price during the pre-opening phases of the market (prior to an auction). An IEP message is generated when the indicative matching price or volume varies. If the Price set to 0, the IEP 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.	41 Indicative Equilibrium Price
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values 1 – 99999
8	Price	Int32	4	Price	3 implied decimal places
12	AggregateQuantity	Uint64	8	Aggregated number of shares.	

Offset	Field	Format	Len	Description	Values
Total Length			20		

3.11 VALUE ADDED DATA

3.11.1 Statistics (60)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.11.1	⬤	⬤		

The Statistics message provides statistics including volume-weighted average price and turnover. It is generated after every trade or trade cancel.

Note that the 'VWAP' field is not populated for SS (OMD Securities Standard) clients.

The ShortSellSharesTraded and ShortSellTurnover fields are only updated twice each day – at the end of the morning session and the end of the afternoon session.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	`60` Statistics
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values `1 – 99999`
8	SharesTraded	Uint64	8	Number of shares traded for a security	
16	Turnover	Int64	8	Current turnover	3 implied decimal places
24	HighPrice	Int32	4	Highest trade price currently performed for a security.	3 implied decimal places
28	LowPrice	Int32	4	Lowest trade price currently performed for a security	3 implied decimal places
32	LastPrice	Int32	4	Last trade price for a security.	3 implied decimal places
36	VWAP	Int32	4	Volume-Weighted Average Price.	3 implied decimal places
40	ShortSellSharesTraded	Uint32	4	Number of short-sell shares traded for a security.	
44	ShortSellTurnover	Int64	8	Current short-sell turnover for a security.	3 implied decimal places
Total Length..			52		

3.11.2 Market Turnover (61)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.11.2	⬤	⬤		

The Market Turnover message is generated at regular intervals throughout the day and contains the total turnover for all securities on a given market segment for a given trading currency.

When the CurrencyCode is blank, the turnover represents the total turnover traded on the given market segment for all trading currencies, expressed in HKD.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**61** Market Turnover
4	MarketCode	String	4	Market code	**MAIN** **GEM** **NASD** **ETS**
8	CurrencyCode	String	3	Currency code of all securities of which the market turnover is derived.	See Currency Values in section 3.1.2 for full details.
11	Filler	String	1		
12	Turnover	Int64	8	Total Traded Turnover of the stocks traded on the market segment in the respective currency	3 implied decimal places
Total Length..			20		

3.11.3 Yield (44)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.11.3	●	●		

The Yield message is generated for bond securities when their yield percentage changes.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**44** Yield
4	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values **1 – 99999**
8	Yield	Int32	4	Current yield of the bond security based on its coupon rate and nominal price	3 implied decimal places **0** means N/A
Total Length..			12		

3.12 NEWS

3.12.1 News (22)

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [●]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.12	●	●		

The News message is generated whenever a news update occurs. The message indicates which markets and/or securities the news is applied to. If NoMarketCode and NoSecurityCodes are both set to zero, the news applies to all markets.

The news may be fragmented across multiple consecutive messages. The LastFragment field will be set to 'Y' in the message that contains the last fragment.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	**22** News
4	NewsType	String	3	Type of Exchange news	**EXN** Exchange news **EXC** Chinese Exchange news
7	NewsID	String	3	Unique number for the news page within each NewsType.	
10	Headline	String / Binary	320	News headline	If **NewsType** is EXN **the Headline is ASCII encoded** If **NewsType** is EXC **the Headline is Unicode UTF-16LE encoded**
330	CancelFlag	String	1	Indicator of whether previously released exchange news (identified by NewsType and NewsID) has been cancelled.	**Y** Cancelled **N** Not cancelled
331	LastFragment	String	1	Indicates whether this message is the last in a sequence of messages.	**Y** Complete **N** Not complete
332	Filler	String	4		
336	ReleaseTime	Uint64	8	Release time of the news.	The number of nanoseconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970 ReleaseTime precision is currently provided to the nearest second.
344	Filler	String	2		
346	NoMarketCodes	Uint16	2	Number of market codes within this message.	**0** to **4**
348	MarketCode	String	4	Market code	**MAIN** **GEM** **NASD** **ETS**
$348 + 4n_M$	Filler	String	2		
$350 + 4n_M$	NoSecurityCodes	Uint16	2	Number of security codes within this message.	**0** to **200**
$352 + 4n_M$	SecurityCode	Uint32	4	Uniquely identifies a security available for trading	5 digit security codes with possible values **1 – 99999**
$352 + 4n_M + 4n_S$	Filler	String	2		
$354 + 4n_M + 4n_S$	NoNewsLines	Uint16	2	Number of news lines.	Maximum of 10 lines per "news page" is currently supported
$356 + 4n_M + 4n_S$	NewsLine	String / Binary	160	News line	If **NewsType** is EXN the NewsLine is ASCII encoded If **NewsType** is EXC the NewsLine is Unicode UTF-16LE encoded
Total Length		$356 + 4n_M + 4n_S + 160n_p$			

(n_M = value of NoMarketCodes)

(n_S = value of NoSecurityCodes)

(n_p = value of NoNewsLines)

3.13 INDEX DATA

The information supplied in this section and its sub-sections applies to the Datafeed(s) marked with [⬤]

Section	OMD Securities Standard (SS)	OMD Securities Premium (SP)	OMD Securities FullTick (SF)	OMD Index (Index)
3.13				●

The indices supplied under the OMD Index are described in more detail in Appendix A, as it may be amended from time to time.

3.13.1 Index Definition (70)

The Index Definition message contains the static referential data for the given index and is generated at the start of the 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.	70 Index Definition
4	IndexCode	String	11	Upstream source's index code.	
15	IndexSource	String	1	Index source.	C CSI H HSI S S&P
16	CurrencyCode	String	3	Currency code of Index Turnover.	See Currency Values in section 3.1.2 for full details. CurrencyCode can be blank if not defined by third party index compilers
19	Filler	String	1		
Total Length..			20		

3.13.2 Index Data (71)

The Index Data message contains all the real-time data for a given index. Fields within this message may be populated with null values to identify when an update is not provided. See section 3.1.1 (Null Values) for more information about null values.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	71 Index Data
4	IndexCode	String	11	Upstream source's index code.	
15	IndexStatus	String	1	Index status.	C Closing value I Indicative O Opening index P Last close value (prev. ses.) R Preliminary close S Stop loss index T Real-time index value IndexStatus can be blank if not defined by third party index compilers
16	IndexTime	Int64	8	Publisher timestamp.	The number of nanoseconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970 IndexTime precision is currently

Offset	Field	Format	Len	Description	Values
					provided to the nearest second.
24	IndexValue	Int64	8	Current value of the index.	4 implied decimal places
32	NetChgPrevDay	Int64	8	Net change in value from previous day's closing value versus last index value	4 implied decimal places
40	HighValue	Int64	8	Highest value for an index.	4 implied decimal places
48	LowValue	Int64	8	Lowest value for an index	4 implied decimal places
56	EASValue	Int64	8	Estimated Average Settlement Value	2 implied decimal places
64	IndexTurnover	Int64	8	Current turnover of underlying constituents	4 implied decimal places
72	OpeningValue	Int64	8	First value for an index.	4 implied decimal places
80	ClosingValue	Int64	8	Last value for an index	4 implied decimal places
88	PreviousSesClose	Int64	8	Previous session closing value (previous day's closing value for CSI and S&P, previous session for HSI)	4 implied decimal places
96	IndexVolume	Int64	8	Index volume of underlying constituents. Only applicable for CSI.	
104	NetChgPrevDayPct	Int32	4	Net change in percentage from previous day's closing value versus last value	4 implied decimal places
108	Exception	String	1	Exception indicator	#　Index with HSIL defined exceptional rule applied ' '　Normal index (empty string)
109	Filler	String	3		
Total Length..			112		

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)

```
Begin
  │
  ▼
Received new Packet
  │
  ▼
Is current SN greater than          Yes      Gap Detected
the next expected SN ?        ──────────►         │
  │                                                │
  │ No                                             │
  ▼                                                │
Ignore Packet          Yes    Is current SN + MsgCount      │
(Duplicate)        ◄───────    in current packet less than  │
                               the next expected SN ?        │
                                   │                         │
                                   │ No                      │
                                   ▼                         ▼
                               Extract first message  ◄── Recover lost messages
                               in packet
                                   │
                                   ▼
                     Extract next message  ◄──  Increase msg processed
                     in packet                  count by 1
                                   │                  ▲
                                   ▼                  │
                     Is current SN in current   No    Ignore message
                     packet + msg processed   ─────►  (Duplicate)
                     count equal to
                     next expected SN ?
                                   │
                                   │ Yes
                                   ▼
                     Process messages
                                   │
                                   ▼
                     End of Packet ?    No     Increase next expected
                                    ─────►      SN by 1
                                   │
                                   │ Yes
                                   ▼
                                  End
```

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, where as 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
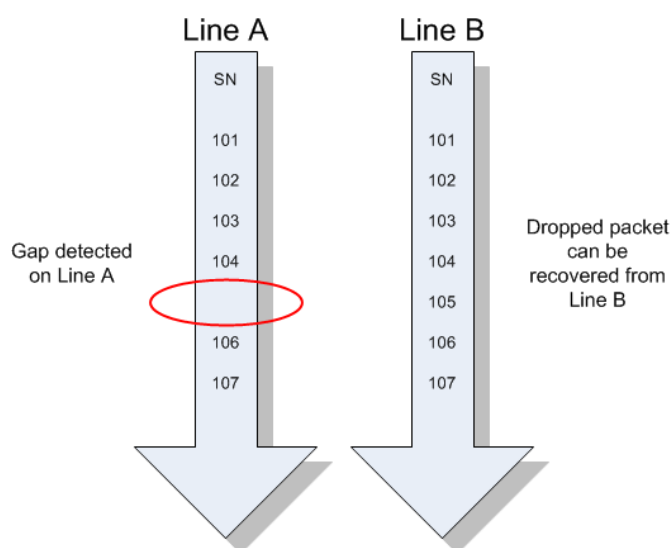
Additional Notes;

- 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 Statistics1	2	106		106	2	Trade2 Statistics1

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 supports the retransmission of the last 50,000 messages per multicast channel only. 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
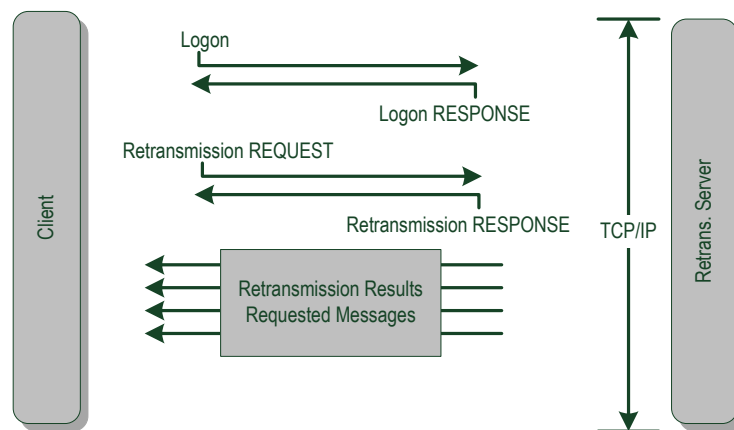
Logon

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
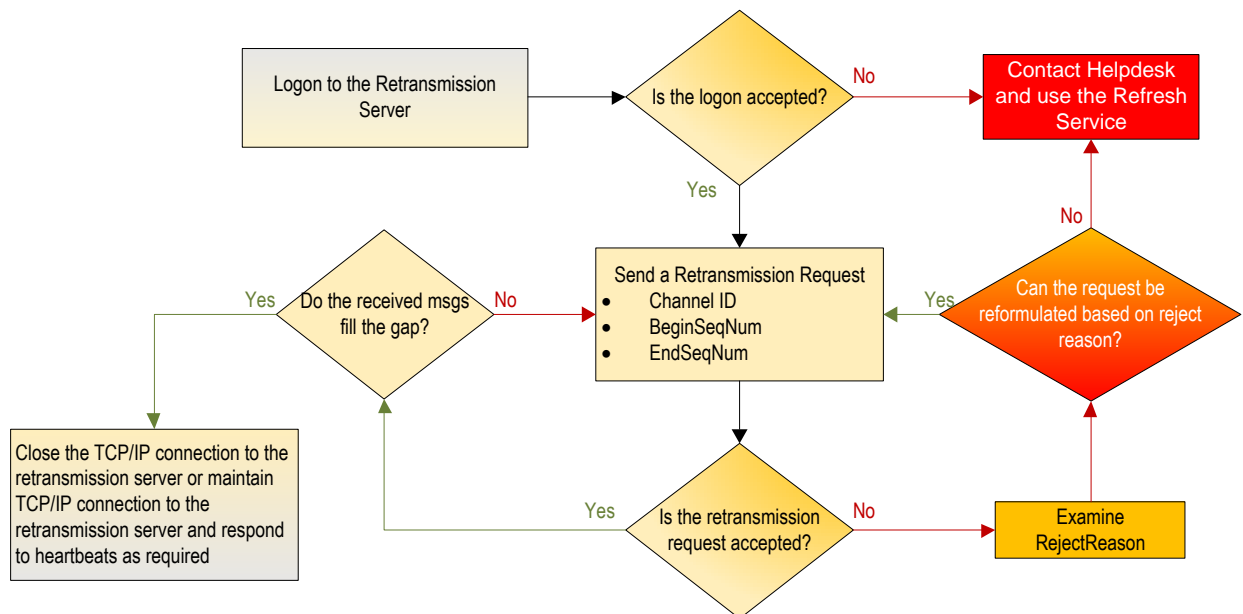
Multiple requests and concurrent sessions

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
Last number of messages available per channel ID	50,000
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, there exists a corresponding refresh multicast channel on which snapshots of the market state are sent at regular intervals throughout the business day.

Market state

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

Message	Snapshot description
Market Definition	Latest market static message for each market.
Security Definition	Latest security static message for each security.
Liquidity Provider	Latest liquidity provider message for each security.
Trading Session Status	Latest trading session status message for each market.
Security Status	Security status message for halted securities.

Message	Snapshot description
Add Order	Snapshot for all non-empty books.
Add Odd Lot Order	Snapshot for all non-empty books.
Aggregate Order Book Update	Snapshot for all non-empty books.
Broker Queue	Snapshot for all non-empty books.
Trade	Latest non-cancel trade message for each security.
Trade Ticker	Latest non-cancel trade ticker message for each security.
Closing Price	Closing price message if available for each security.
IEP	Latest IEP message for each security if during auction phase.
Nominal Price	Latest nominal price message for each security.
Statistics	Latest statistics message for each security.
Market Turnover	Latest market turnover message for each market.
Currency Rate	Latest rates for each currency.
News	All news messages.
Index Definition	Latest index definition message for each index.
Index Data	Latest index data message for each index.
Yield	Latest yield message generated for bond securities when their yield percentage last changed

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.

5.   AGGREGATE ORDER BOOK MANAGEMENT

Book Identification

For each security there exists an odd lot book and a board lot book in the trading system. A book is therefore uniquely identified by SecurityCode.

Partial Price Depth

Securities shall be traded in accordance with the scale of spreads set out in the Second Schedule of the Rules of the Exchange respective to the Spread Table Code specified in their Securities Definition message. The tick level provides information on how many spreads from the best price for an order price whereas a price level is assigned to each price existing in the OMD order book. An order price with tick level 1 means the order price is the best price, a tick level of 2 means the order price is one spread from the best price, etc. The Aggregate Order Book Update message sends out the price level but not the tick level.

The concept of tick and price levels is illustrated in the table below, assuming that the best bid price of a security is 9800 and the spread is 10 for this price range. In the table there are orders in 5 bid prices so the number of price levels is 5 (contiguous price levels from 1 to 5); these orders are distributed over 10 spreads (tick levels) so the tick levels are from 1 to 10. Taking orders with bid price 9710 as example, it is the 5th price in the book so the price level will be 5 and it is 9 spreads from the best bid price so the tick level will be 10.

Bid Side			
Tick	PriceLevel	AggregateQuantity	Price
1	1	700	9800
2	2	350	9790
3			
4			
5	3	150	9760
6			
7			
8			
9	4	250	9720
10	5	100	9710

OMD provides a view of 10 tick depths of the aggregate order book for securities market and does not provide updates on price levels which are more than 9 spreads away from the best price. 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 and tick levels.

Bid Side				Ask Side			
Tick	PriceLevel	AggregateQuantity	Price	Price	AggregateQuantity	PriceLevel	Tick
1	1	700	9730	9760	500	1	1
2	2	350	9720	9770	300	2	2
3	3	150	9710	9780	100	3	3
4	4	250	9700	9790	150	4	4
5	5	100	9690				5
6	6	150	9680				6
7	7	50	9670				7
8	8	200	9660				8
9	9	100	9650				9
10							10

OMD only sends updates within the 10 tick levels in the aggregate order book except for Explicit Deletion (please refer Example 5 illustrated below for details)

Book Updates

Book update messages are generated by OMD as delta messages defined in section **3.9.6** (**Aggregate Order Book Update (53)**). Each message may contain any combination of new, changed or deleted 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	SecurityCode	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			
Tick	PriceLevel	AggregateQuantity	Price	Price	AggregateQuantity	PriceLevel	Tick
1	1	700	9730	9760	500	1	1
2	2	350	9720	9770	200	2	2
3	3	150	9710	9780	100	3	3
4	4	250	9700	9790	150	4	4
5	5	100	9690				5
6	6	150	9680				6
7	7	50	9670				7
8	8	200	9660				8
9	9	100	9650				9
10				9850	300	5	10

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	SecurityCode	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			
Tick	PriceLevel	AggregateQuantity	Price	Price	AggregateQuantity	PriceLevel	Tick
1	1	50	9740	9760	500	1	1
2	2	700	9730	9770	200	2	2
3	3	350	9720	9780	100	3	3
4	4	150	9710	9790	150	4	4
5	5	250	9700				5
6	6	100	9690				6
7	7	150	9680				7
8	8	50	9670				8
9	9	200	9660				9
10	10	100	9650	9850	300	5	10

Price levels of the other 9 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 9 spreads away from the best price), the client must delete the excess entry. If the book shrinks again, the server 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, and the bid quantity at price 9660 is reduced from 200 to 150, it will cause the following message to be sent:

Offset	Field Name	Value
0	MsgSize	60
2	MsgType	53

	4	SecurityCode	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	0
	32	Filler	NULL
	36	AggregateQuantity	150
	44	Price	9660
	48	NumberOfOrders	1
	52	Side	0 (Bid)
	54	PriceLevel	10
	55	UpdateAction	1
	56	Filler	NULL

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

Bid Side				Ask Side			
Tick	PriceLevel	AggregateQuantity	Price	Price	AggregateQuantity	PriceLevel	Tick
1	1	250	9750	9760	500	1	1
2	2	50	9740	9770	200	2	2
3	3	700	9730	9780	100	3	3
4	4	350	9720	9790	150	4	4
5	5	150	9710				5
6	6	250	9700				6
7	7	100	9690				7
8	8	150	9680				8
9	9	50	9670				9
10	10	150	9660	9850	300	5	10

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

Price 9650 and quantity 100 must be deleted by the client.

Price 9660 quantity must be reduced to 150 – PriceLevel 10 is used in the incoming message to reflect the new price level of the price 9660 after the addition of the price 9750.

Example 4 – Explicit Additions

If a match causes an order to be removed so that there are now less than 10 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 11th 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	SecurityCode	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	100
44	Price	9650
48	NumberOfOrders	1
52	Side	0 (Bid)
54	PriceLevel	10
55	UpdateAction	0
56	Filler	NULL

The resulting order book should now be;

	Bid Side			Ask Side			
Tick	PriceLevel	AggregateQuantity	Price	Price	AggregateQuantity	PriceLevel	Tick
1	1	50	9740	9760	500	1	1
2	2	700	9730	9770	200	2	2
3	3	350	9720	9780	100	3	3
4	4	150	9710	9790	150	4	4
5	5	250	9700				5
6	6	100	9690				6
7	7	150	9680				7
8	8	50	9670				8
9	9	200	9660				9
10	10	100	9650	9850	300	5	10

Example 5 – Explicit Deletions

Suppose a new book entry causes the last tick entry (Tick 10 in the previous order book in Example 4) to be shifted out of the book, if the shifted out entry is within 10 price level, OMD will send an explicit deletion for the entry. If the shifted out entry is outside the 10 price level, OMD will not send further updates on that price and the client must delete the excess entry (please refer to Example 3 above for details) to ensure their order book will not keep out-dated information.

For example, if an ask order with price 9750 and quantity 300 is added to the order book above, it will cause the following message to be sent:

Offset	Field Name	Value
0	MsgSize	60
2	MsgType	53
4	SecurityCode	1234
8	Filler	NULL
11	NoEntries	1
12	AggregateQuantity	300
20	Price	9750
24	NumberOfOrders	1
28	Side	1 (Offer)

30	PriceLevel	1	
31	UpdateAction	0	
32	Filler	NULL	
36	AggregateQuantity	300	
44	Price	9850	
48	NumberOfOrders	1	
52	Side	1 (Offer)	
54	PriceLevel	6	
55	UpdateAction	2	
56	Filler	NULL	

The resulting order book should now be;

Bid Side				Ask Side			
Tick	PriceLevel	AggregateQuantity	Price	Price	AggregateQuantity	PriceLevel	Tick
1	1	50	9740	9750	300	1	1
2	2	700	9730	9760	500	2	2
3	3	350	9720	9770	200	3	3
4	4	150	9710	9780	100	4	4
5	5	250	9700	9790	150	5	5
6	6	100	9690				6
7	7	150	9680				7
8	8	50	9670				8
9	9	200	9660				9
10	10	100	9650				10

Explicit Deletions versus Implicit Deletions

Suppose initially bid orders are queued in 8 price levels in the aggregate order book and assume there is no order inputted at price 9770 & 9730. The aggregate order book will be as follows.

Bid Side			
Tick	PriceLevel	AggregateQuantity	Price
1	1	700	9800
2	2	350	9790
3	3	150	9780
4			
5	4	250	9760
6	5	100	9750
7			
8	6	400	9730
9	7	200	9720
10	8	300	9710

When new bid orders at 3 different prices (9860, 9850 & 9840) arrived, the resulting book will be changed as follows:

Bid Side				
Tick	PriceLevel	AggregateQuantity	Price	
1	1	450	9860	← new order, Explicit Addition
2	2	550	9850	← new order, Explicit Addition
3	3	650	9840	← new order, Explicit Addition

4				
5				
6				
7	4	700	9800	← previous best bid, now at PriceLevel 4 (Tick 7), Implicit Level Adjustment
8	5	350	9790	← previous 2nd best bid, now at PriceLevel 5 (Tick 8), Implicit Level Adjustment
9	6	150	9780	← previous 3rd best bid, now at PriceLevel 6 (Tick 9), Implcit Level Adjustment
10				
11	7	250	9760	← orders exceed 10 Tick but within 10 PriceLevel, Explicit Deletion
12	8	100	9750	← orders exceed 10 Tick but within 10 PriceLevel, Explicit Deletion
13				
14	9	400	9730	← orders exceed 10 Tick but within 10 PriceLevel, Explicit Deletion
15	10	200	9720	← orders exceed 10 Tick but within 10 PriceLevel, Explicit Deletion
16	11	300	9710	← orders exceed 10 Tick & exceed 10 PriceLevel, Implicit Deletion

Orders in shaded area which were originally within the 10 tick levels offered in OMD now fall outside the 10 tick levels. OMD will send Explicit Delete for orders which fall outside 10 tick levels but are within 10 price levels (i.e. entries highlighted in blue). However OMD will not send Explicit Delete for orders which are outside 10 price levels (i.e. entries highlighted in pink) and the client must delete the excess entries (i.e. Implicit Delete by the client).

The following message will be sent:

Offset	Field Name	Value
0	MsgSize	180
2	MsgType	53
4	SecurityCode	1234
8	Filler	NULL
11	NoEntries	7
12	AggregateQuantity	450
20	Price	9860
24	NumberOfOrders	1
28	Side	0 (Bid)
30	PriceLevel	1
31	UpdateAction	0
32	Filler	NULL
36	AggregateQuantity	550
44	Price	9850
48	NumberOfOrders	1
52	Side	0 (Bid)
54	PriceLevel	2
55	UpdateAction	0
56	Filler	NULL
60	AggregateQuantity	650
68	Price	9840
72	NumberOfOrders	1
76	Side	0 (Bid)
78	PriceLevel	3
79	UpdateAction	0
80	Filler	NULL
84	AggregateQuantity	250
92	Price	9760
96	NumberOfOrders	1
100	Side	0 (Bid)

102	PriceLevel	7	
103	UpdateAction	2	
104	Filler	NULL	
108	AggregateQuantity	100	
116	Price	9750	
120	NumberOfOrders	1	
124	Side	0 (Bid)	
126	PriceLevel	8	
127	UpdateAction	2	
128	Filler	NULL	
132	AggregateQuantity	400	
140	Price	9730	
144	NumberOfOrders	1	
148	Side	0 (Bid)	
150	PriceLevel	9	
151	UpdateAction	2	
152	Filler	NULL	
156	AggregateQuantity	200	
164	Price	9720	
168	NumberOfOrders	1	
172	Side	0 (Bid)	
174	PriceLevel	10	
175	UpdateAction	2	
176	Filler	NULL	

Example 6 – 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 security. An example message is shown below.

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

Offset	Field Name	Value	
0	MsgSize	36	
2	MsgType	53	
4	SecurityCode	1234	
8	Filler	NULL	
11	NoEntries	1	
12	AggregateQuantity	0	
20	Price	0	
24	NumberOfOrders	0	
28	Side	0	
30	PriceLevel	0	
31	UpdateAction	74	
32	Filler	NULL	

APPENDIX A – LIST OF INDICES UNDER OMD INDEX

The information supplied in this appendix applies to OMD Index only.

The indices supplied under the OMD Index are described in the table below, as it may be amended from time to time by HKEx-IS pursuant to clause 2.2 of the Licence Agreement.   The mark [●] specifies if an index disseminated under the OMD Index is Third Party Content under the Licence Agreement.

Name of the Index disseminated under the OMD Index	Third Party Content under the Licence Agreement
CSI 300 Index	●
CSI China Mainland Consumer Index	●
CSI Cross-Straits 500 Index	●
CSI HK Mainland Enterprises Index	●
CSI Hong Kong 100 Index	●
CSI Hong Kong Dividend Index	●
CSI Hong Kong Listed Tradable Mainland Consumption Index	●
CSI Hong Kong Listed Tradable Mainland Real Estate Index	●
CSI Hong Kong Middle Cap Select Index	●
CSI Hong Kong Private-owned Mainland Enterprises Index	●
CSI Hong Kong State-owned Mainland Enterprises Index	●
CSI Overseas Mainland Enterprises Index (HKD)	●
CSI RAFI Hong Kong 50 Index	●
Hang Seng China Affiliated Corporations Index (HSCCI)	●
Hang Seng China Enterprises Index (HSCEI)	●
Hang Seng Index (HSI)	●
HSI Sub Indices (HSI-Finance, HSI-Utilities, HSI-Property, HSI-Commerce & Industry)	●
HSI Volatility Index (VHSI)	●
SSE 50 Index	●
SSE 180 Governance Index	●
SSE 180 Index	●
SSE 380 Index	●
SSE Commodity Equity Index	●
SSE Composite Index	●
SSE Dividend Index	●
SSE Mega-cap Index	●
SSE Mid Cap Index	●
SSE Industry Top Index	●
S&P/HKEx LargeCap Index	
S&P/HKEx GEM Index	