HISTORICAL FULL BOOK - DERIVATIVES MARKET (NON-SOM) (BINARY FORMAT)

Update History

No.	Issue Date	Details
1	2014-12-01	First Issue
1.1	2016-07-04	Data Enrichment - Section 4.2 – Revised as Null Value will also be found in Trade Amendment (356) message - Section 4.2 – Clarify field values of Price and Quantity in Trade Amendment (356) message for cancelled trades already “given up”
2	2016-11-04	Revised Edition with the following updates: - Sections 5 – Include a new type of information - Derivatives Market Alert (MC177_All_YYYYMMDD) in the file - Sections 5.1 – Add new Market Alert (323) message
3	2018-04-16	Data Enrichment - Section 1.1 – Add a new field “CommodityID” - Section 1.2 – Add a new field “TickStepSize” and change the description of the field “PriceQuotationFactor” - Section 1.4 – Add new field “PriceQuotationFactor” and 8-byte filler

The Historical Full Book - Derivatives Market (Non-SOM) (Binary Format) includes 5 types of information – (1) Derivatives Reference data, (2) Derivatives Status data, (3) Derivatives Full Order Book data, (4) Derivatives Block Trade and Trade Amendment data and (5) Derivatives Market Alert data. Please refer to the below sub-sections for the details of the 5 types of information.

The following table lists out the data files to be found in each issue:

File Name	Contents
Non-SOM Products	
MC101_All_YYYYMMDD	Derivatives Static Reference (Partition 1)
MC201_All_YYYYMMDD	Derivatives Static Reference (Partition 2)
MC151_All_YYYYMMDD	Derivatives Static Reference (OAPI)
MC161_All_YYYYMMDD	Derivatives Status
MC121_All_YYYYMMDD	Derivatives Full Order Book file (Partition 1)
MC221_All_YYYYMMDD	Derivatives Full Order Book file (Partition 2)
MC167_All_YYYYMMDD	Derivatives Block Trades and Trade Amendments
MC177_All_YYYYMMDD	Derivatives Market Alert

1) YYYYMMDD is the date of file

2) If there is no record in the file, a dummy file with zero-length size will be provided.

1. Derivatives Reference

The Derivatives Reference file is in binary format and contains five types of messages – ***CommodityDefinition***, ***ClassDefinition***, ***SeriesDefinitionBase, SeriesDefinitionExtended*** and ***CombinationDefinition***. The filenames of the 3 Derivatives Reference files are as follows:

MC101_All_YYYYMMDD – Non-SOM Partition 1 (SeriesDefinitionBase,CombinationDefinition)
MC201_All_YYYYMMDD – Non-SOM Partition 2 (SeriesDefinitionBase,CombinationDefinition)
MC151_All_YYYYMMDD – Non-SOM (CommodityDefinition,ClassDefinition,SeriesDefinitionExtended)
where YYYYMMDD is the date of the Derivatives Reference Data file

The layout of the Derivatives Reference is as follows:

<RecordLength><PacketHeader><DerivativesReference>…<RecordLength><PacketHeader><DerivativesReference>…<RecordLength><PacketHeader><DerivativesReference>

Following is the message layout of the ***RecordLength***

Offset	Field	Format	Len	Description
0	RecLen	Uint16	2	Size of the record (including this field)
Total length			2	

Following is the message layout of the ***PacketHeader***

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 *January 1,* 1970*, 00:00:00 GMT*, precision is provided to the nearest millisecond.

Offset	Field	Format	Len	Description
Total length			16	

<DerivativesReference> contains different combinations of the four types of messages – ***CommodityDefinition***, ***ClassDefinition***, ***SeriesDefinitionBase, SeriesDefinitionExtended*** and ***CombinationDefinition.*** For example:

In MC151 data file, the order of messages can be as following:

<CommodityDefinition><CommodityDefinition><ClassDefinition><SeriesDefinitionExtended> or ***<CommodityDefinition><CommodityDefinition><ClassDefinition><SeriesDefinitionExtended><ClassDefinition><SeriesDefinitionExtended>***

In MC101/201 data files, the order of messages can be as following:

<SeriesDefinitionBase><CombinationDefinition><SeriesDefinitionBase><CombinationDefinition>

or

<SeriesDefinitionBase><SeriesDefinitionBase><CombinationDefinition><CombinationDefinition>

Followings are the message layouts of the ***CommodityDefinition***, ***ClassDefinition***, ***SeriesDefinitionBase, SeriesDefinitionExtended*** and ***CombinationDefinition***

1.1 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 of OMD-D Interface Specification 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 *

Offset	Field	Format	Len	Description	Values
					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*

1.2 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 8.4 of OMD-D Interface Specification 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 8.3 of OMD-D Interface Specification 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 bid and ask options in the same Series.	0–255
8	CommodityCode	Uint16	2	Numerical identifier of the	

Offset	Field	Format	Len	Description	Values
				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, Baits before Normal Orders, Time * 8 Inverted Price, Baits 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
29	PremiumUnit4Price	Uint8	1	The premium unit that describes the price unit in the order.	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 of OMD-D Interface Specification 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?	Y Yes N No
80	SettlementCurrencyID	String	32	Full descriptive name of the Settlement Currency. The representation of the	See Currency Values in section 3.1.2 of OMD-D Interface Specification for full details.

Offset	Field	Format	Len	Description	Values
				currency follows the S.W.I.F.T. handbook and ISO 3166 standard.	
112	Effective Tomorrow	Uint8	1	This declaration is for series to be traded the next day	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*

1.3 Series Definition Base (303)

Describes basic series information 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.	303 Series Definition Base
4	OrderbookID	Uint32	4	Numerical identifier of the order book This is the unique identifier for the series The Combination Definition is retrieved through this field which links it to the Series Definition Base	
8	Symbol	String	32	Short Name	
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	

Offset	Field	Format	Len	Description	Values
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 Class Definition (302) field "DecimalsInStrikePrice". Not applicable for Combo Series.
48	ExpirationDate	String	8	Expiry date of the series	YYYYMMDD
56	Filler	Uint16	2		
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*

1.4 Series Definition Extended (304)

Describes series static data 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.	304 Series Definition Extended
4	OrderBookID	UInt32	4	Orderbook ID	0 If Not Available
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 8.4 of OMD-D Interface Specification for a list of possible values
42	InstrumentGroup	Uint8	1	Instrument Group	See section 8.3 of OMD-D Interface Specification 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 bid and ask options in the same Series.	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	

Offset	Field	Format	Len	Description	Values
				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
73	EffectiveTomorrow	Uint8	1	This declaration is for next day series	0 False 1 True
74	PriceQuotationFactor	Int32	4	Implies the contracted value of the product / series	Decimal places determined from Class Definition field "DecimalInContractSize"
78	Filler	String	2		
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	Filler	String	8		

Total Length	104

1.5 Combination Definition (305)

Describes a combination orderbook 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.	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		

2. Derivatives Status

The Derivatives Status file is in binary format and contains two types of messages – ***MarketStatus***, ***SeriesStatus*** and ***CommodityStatus***. There is a Derivatives Status file with filename MC161_All_YYYYMMDD, where YYYYMMDD is the date of the Derivatives Status file.

The layout of the Securities Status is as follows:

<RecordLength><PacketHeader><DerivativesStatus>…<RecordLength><PacketHeader><DerivativesStatus>…<RecordLength><PacketHeader><DerivativesStatus>

Following is the message layout of the ***RecordLength***

Offset	Field	Format	Len	Description
0	RecLen	Uint16	2	Size of the record (including this field)
Total length			2	

Following is the message layout of the ***PacketHeader***

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.
Total length			16	

<DerivativesStatus> contains different combinations of the three types of messages – ***MarketStatus***, ***SeriesStatus*** and ***CommodityStatus.*** For example:

<MarketStatus><SeriesStatus><CommodityStatus><MarketStatus> or
<MarketStatus><SeriesStatus><MarketStatus><SeriesStatus>

Followings are the message layouts of the ***MarketStatus***, ***SeriesStatus*** and ***CommodityStatus***

2.1 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 8.4 of OMD-D Interface Specification for a list of possible values 0 If Not Available
7	Instrument	Uint8	1	Instrument Group Populated only if StateLevel = 2, 3	See section 8.3 of OMD-D Interface Specification 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

Offset	Field	Format	Len	Description	Values
46	State	Uint16	2	Numeric identification of the State Type.	See full list of states within section 8.1 of OMD-D Interface Specification 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.

2.2 Series Status (321)

The Series Status message is generated whenever a series is suspended or resumed, 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	Suspended	String	1	Defines if the series is suspended or not.	Y Yes N No
9	Filler	String	3		
Total Length			12		

2.3 Commodity Status (322)

The Commodity Status message is generated whenever a commodity state 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 series is suspended or not.	Y Yes N No
7	Filler	String	1		

Offset	Field	Format	Len	Description	Values
Total Length			8		

3. Derivatives Full Order Book

The Derivatives Full Order Book file is in binary format and contains six types of messages – ***AddOrder, ModifyOrder***, ***DeleteOrder, OrderBookClear, Trade*** and ***CalculatedOpeningPrice***. There are totally 2 files for the two partitions of the products. The filenames of the 2 Derivatives Full Order files are as follows:

MC121_All_YYYYMMDD – Non-SOM partition 1 full order book
MC221_All_YYYYMMDD – Non-SOM partition 2 full order book
where YYYYMMDD is the date of the Derivatives Full Order Book file

The layout of the Derivatives Full Order Book is as follows:

<RecordLength><PacketHeader><DerivativesFullOrderBook>…<RecordLength><PacketHeader><DerivativesFullOrderBook>…<RecordLength><PacketHeader><DerivativesFullOrderBook>

Following is the message layout of the ***RecordLength***

Offset	Field	Format	Len	Description
0	RecLen	Uint16	2	Size of the record (including this field)
Total length			2	

Following is the message layout of the ***PacketHeader***

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	The number of nanoseconds since *January 1*, 1970, *00:00:00 GMT*, precision is provided to the nearest millisecond.
Total length			16	

<DerivativesFullOrderBook> contains different combinations of the six types of messages – ***AddOrder, ModifyOrder***, ***DeleteOrder, OrderBookClear, Trade*** and ***CalculatedOpeningPrice*** For example:

<AddOrder><Trade><AddOrder><DeleteOrder><CalculatedOpeningPrice> or
<AddOrder><AddOrder><DeleteOrder><ModifyOrder><Trade><Trade>

Followings are the message layouts of the ***AddOrder, ModifyOrder***, ***DeleteOrder, OrderBookClear, Trade*** and ***CalculatedOpeningPrice***

3.1 Add Order (330)

The Add Order message is generated when a new order is inserted into the order book. 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.

Note that for series which have an auction session then DF subscribers will receive an **Orderbook Clear (335)** message followed by a large quantity of **Add Order (330)** messages representing the uncrossed book after the end of auction.

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 *NOTE: Applicable types may be defined by the marketplace.* *NOTE 2: This field is a bit map. Multiple values may be set simultaneously.*
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*

3.2 Modify Order (331)

The Modify Order message is generated when an existing order identified by the OrderID is modified. The attributes that can be modified are Position, Quantity, Price and OrderType.

In most cases, modification to an order will result in a Delete Order message followed by an Add Order message. Despite that, clients' application should have the ability to process Modify Order messages if received.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	331 Modify Order
4	Orderbook ID	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 – the new quantity of the order	
24	Side	Uint8	1	Side of the order	0 Bid 1 Offer
25	Filler		1		
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 NOTE: Applicable types may be defined by the marketplace. NOTE 2: This field is a bit map. Multiple values may be set simultaneously.
28	OrderBookPosition	UInt32	4	Order rank information for the order position within the order book for each series	Integer
Total Length			32		

3.3 Delete Order (332)

The Delete Order message is generated when an existing order identified by the OrderBookID, OrderID and Side 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.	332 Delete Order

Offset	Field	Format	Len	Description	Values
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	
16	Side	Uint8	1	Side of the order	0 Bid 1 Offer
17	Filler		1		
Total Length			18		

3.4 Orderbook Clear (335)

The Orderbook Clear message is used to inform subscribers that all existing orders should be removed from both the bid and ask sides of the specified orderbook. The message is typically used at the start and end of Auction;

*At the end of Auction, DF subscribers will receive an **Orderbook Clear (335)** message followed by a large quantity of **Add Order (330)** messages representing the uncrossed book.*

Subscribers should clear both bid and ask side orders for the specified orderbook.

Message Fields

Offset	Field	Format	Len	Description	Values
0	MsgSize	Uint16	2	Size of the message	
2	MsgType	Uint16	2	Type of message.	335 Orderbook Clear
4	Orderbook ID	Uint32	4	Uniquely identifies a series available for trading	
Total Length			8		

3.5 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.

For DF subscribers; if the OrderID within the Trade (350) message is non-zero then users must reduce the resting order identified by the ‘Quantity’ within the Trade (350) message. If the outstanding quantity is zero the order must be deleted.

Important messages: Subscribers 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) and the Series Statistics (363) 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) and Series Statistics (363) 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 1/100th 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.

3.6 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	

Offset	Field	Format	Len	Description	Values
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		

4. Block Trade and Trade Amendment

The Derivative Block Trade and Trade Amendment file is in binary format and contains two types of messages – ***Trade*** and ***TradeAmendment***. There is a Block Trade and Trade Amendment file with filename MC167_All_YYYYMMDD, where YYYYMMDD is the date of the Derivatives Block Trade and Trade Amendment file.

The layout of the Derivatives Block Trade and Trade Amendment is as follows:

<RecordLength><PacketHeader><DerivativeBlockTrade&TradeAmendment>…<RecordLength><PacketHeader><DerivativeBlockTrade&TradeAmendment>…<RecordLength><PacketHeader><DerivativeBlockTrade&TradeAmendment>

Following is the message layout of the ***RecordLength***

Offset	Field	Format	Len	Description
0	RecLen	Uint16	2	Size of the record (including this field)
Total length			2	

Following is the message layout of the ***PacketHeader***

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	The number of nanoseconds since *January 1,* 1970*, 00:00:00 GMT*, precision is provided to the nearest millisecond.
Total length			16	

<DerivativesBlockTrade&TradeAmendment> contains different combinations of the two types of messages – ***Trade*** and ***TradeAmendment.*** For example:

<Trade><Trade><TradeAmendment><TradeAmendment> or
<Trade><TradeAmendment><Trade><TradeAmendment>

Followings are the message layouts of the ***Trade*** and ***TradeAmendment***

4.1 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.

For DF subscribers; if the OrderID within the Trade (350) message is non-zero then users must reduce the resting order identified by the 'Quantity' within the Trade (350) message. If the outstanding quantity is zero the order must be deleted.

Important messages: Subscribers 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) and the Series Statistics (363) 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) and Series Statistics (363) 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.	

Offset	Field	Format	Len	Description	Values
48	TradeTime	Uint64	8	Date and time of the last trade in UTC timestamp (nanoseconds since 1970) precision to the nearest 1/100th 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.

4.2 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 1/100th 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. The trade has been deleted.
37	Filler		3		
Total Length.....			40		

5. Derivatives Market Alert

The Derivatives Market Alert file is in binary format and contains one type of message – ***MarketAlert***. There is one Derivatives Market Alert file with filename MC177_All_YYYYMMDD, where YYYYMMDD is the date of the Derivatives Market Alert file.

The layout of the Market Alert is as follows:

<RecordLength><PacketHeader><DerivativesMarketAlert>…<RecordLength><PacketHeader><DerivativesMarketAlert>

Following is the message layout of the ***RecordLength***

Offset	Field	Format	Len	Description
0	RecLen	Uint16	2	Size of the record (including this field)
Total length			2	

Following is the message layout of the ***PacketHeader***

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	The number of nanoseconds since *January 1*, 1970, *00:00:00 GMT*, precision is provided to the nearest millisecond.
Total length			16	

<DerivativesMarketAlert> contains one type of message – ***MarketAlert*** only.

Following is the message layout of the ***MarketAlert***

5.1 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	`Y` Complete

Offset	Field	Format	Len	Description	Values
				the last in a sequence of messages.	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	
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)