Circular no.: MCX/CTCL/266/2024 April 29, 2024

Trading Interface - MCX FIX API – v1.4

In terms of provisions of the Rules, Bye-Laws and Business Rules of the Exchange and in continuation to Exchange circular no. MCX/CTCL/443/2023 dated June 30, 2023, MCX/CTCL/478/2021 dated August 02, 2021, MCX/CTCL/507/2021 dated August 10, 2021 MCX/CTCL/291/2022 dated May 18, 2022, Circular no.: MCX/CTCL/364/2022 June 21, 2022 and Circular no.: MCX/CTCL/133/2024 dated March 11, 2024. Members of the Exchange are notified as under:

Trading Members and Vendors are requested to note that, the Exchange is in the process of implementing, Encryption/Decryption for interactive messages in Trading Interface - FIX API, details of which are provided in the revised Trading Interface - FIX API v1.4.

Trading members and empanelled vendors are requested to take note of the same.

For and on behalf of
Multi Commodity Exchange of India Ltd.

Abhay Angarkar
AVP - Technology
Encl.: As above

Kindly contact Customer Service Team on 022 – 6649 4040 or send an email at customersupport@mcxindia.com for any clarification.

Multi Commodity Exchange of India Limited

Trading Interface

FIX API

Version 1.4

29 April 2024

Confidential

Restriction on Use and Disclaimer of Information and Data

All The information contained in this document constitutes a trade secret and/or information that are commercial or financial and confidential or privileged. It is furnished in confidence with the understanding that it will not, without the prior written permission of MCX, be used or disclosed for other than allowed purposes.

The copyright in this work may be vested with MCX and / or its suppliers. No part of this document may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means whether, electronic, mechanical, or otherwise without the prior written permission of MCX.

The recipient acknowledges that MCX and its suppliers may have copyright in the work. The recipient further agrees that the work is confidential information and contains proprietary MCX information belonging to MCX and / or its suppliers. The recipient manifests, by its receipt of the work, its acknowledgment of MCX and / or its suppliers copyright in the work, its acceptance that the work is confidential information, and its compliance with the terms contained in this notice.

Although MCX has made every effort to provide accurate information at the date of publication, it does not give any representations or warranties as to the accuracy, reliability or completeness of the information in this document. Accordingly, MCX, its subsidiaries and their employees, officers and contractors and its suppliers shall not, to the extent permitted by law, be liable for any direct or indirect loss arising in any way (including by way of negligence) from or in connection with anything provided in or omitted from this document or from any action taken, or inaction, in reliance on this document.

MCX reserves the right to amend details in this document at any time and without notice.

Confidential

Copyright

All trademarks that appear in the document have been used for identification purposes only and belong to their respective companies.

Document details

Name	Version no.	Description
Trading Interface - MCX FIX API	V 1.4	API Documentation for FIX Trading Interface

Document Revision List

Revision No.	Revision Date	Revision Description
1	11-Mar-2024	**Addition:** • Addition to Section 6.2.11 Execution Report -Tag 39 OrdStatus • Addition of enums ○ 1 (PartiallyFilled) ○ 2 (Filled) ○ 8 (Rejected) • Section 3.10.9 ExecutionReport (8) NOTE added at the end of message • Section 6.2.11 Execution Report added enum 355 = Risk Reduction Timer Expired • Section 6.4.7 ExDEx notification added enum 355 = Risk Reduction Timer Expired • Addition of Section 5.13: Encryption/Decryption for Interactive API • Addition of New Tag 307574, 307575, in message Logon Response (Section 6.1.2) • Addition of Annexure for Encryption/Decryption with example code snippets in C++ and C# (Section 8) • Modified to Section 6.2.11 Execution Report Tag 527 SecondaryExecID Modified description. **Deletion:** • Deleted enum value P (Short Code Identifier) - Tag 447 PartyIDSource • Deleted all enums except 36 (Entering trader) - Tag 452 PartyRole
2	29-April-2024	**Modification:** • In section 5.13 step 2: session logon response description. • In Section 3.10.9 Note updated. • In section 6 Message formats updated the table. **Addition:** • In section 5.13 New point is added encryption/decryption changes for session logon response and other response messages.

Confidential

MCX Contact Details

The Exchange's Member & Vendor may contact Technology Division to seek clarification at:	
Multi Commodity Exchange of India Limited Exchange Square, Suren Road, Chakala, Andheri (East), Mumbai 400 093. www.mcxindia.com	Tel: +91 – 22 – 66494000 / 67318888 Fax: +91 – 22 – 66494151 Email – apisupport@mcxindia.com

Confidential

Restriction on Use and Disclaimer of Information and Data.

All The information contained in this document constitutes a trade secret and/or information that are commercial or financial and confidential or privileged. It is furnished in confidence with the understanding that it will not, without the prior written permission of MCX, be used or disclosed for other than allowed purposes.

The copyright in this work may be vested with MCX and / or its suppliers. No part of this document may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means whether electronic, mechanical, or otherwise without the prior written permission of MCX.

The recipient acknowledges that MCX and its suppliers may have copyright in the work. The recipient further agrees that the work is confidential information and contains proprietary MCX information belonging to MCX and / or its suppliers. The recipient manifests, by its receipt of the work, its acknowledgment of MCX and / or its suppliers copyright in the work, its acceptance that the work is confidential information, and its compliance with the terms contained in this notice.

Although MCX has made every effort to provide accurate information at the date of publication, it does not give any representations or warranties as to the accuracy, reliability or completeness of the information in this document. Accordingly, MCX, its subsidiaries and their employees, officers and contractors and its suppliers shall not, to the extent permitted by law, be liable for any direct or indirect loss arising in any way (including by way of negligence) from or in connection with anything provided in or omitted from this document or from any action taken, or inaction, in reliance on this document.

MCX reserves the right to amend details in this document at any time and without notice. Strictly for private circulation only.

This document must not be circulated to other users without prior permission of MCX.

Confidential

Contents

1.Introduction 12
2 Purpose 13
3 Service Description 13
3.1 FIX Session Concept 13
3.2 Party Identification 13
3.3 Security Identification 14
3.4 OrderID Policy 14
3.5 Uniqueness Of Identifiers 15
3.6 TimeStamps 16
3.7 Quantity Fields 16
3.8 Pending Responses “Request State Unknown” 17
3.9 TerminalInfo Rules 17
3.10 Order Management 17
3.10.1 Order Types/Trading,Execution and Validity Restrictions 18
3.10.2 Relevant FIX Fields for the identification of the Order Types 19
3.10.3 Notional Value And Quantity Checks 20
3.10.4 Cancellation 20
3.10.5 Modifications 20
3.10.6 Self Match Preventions 21
3.10.7 Text Fields 21
3.10.8 Order Status and Execution Report 22
3.10.9 ExecutionReport (8) “Request Status Unknown” 23
3.10.10 Order Book Restatement 24
3.10.11 Mass Cancellation Notification 24
3.10.12 Trading Session Events 25
3.11 Custom Tags 27
3.12 Trade Capture 29
3.12.1 Trade Capture via Trading Session 29
3.12.2 Trade Capture Report-Identification Criteria 29
3.12.2.1 Message Event Source 29
3.12.3 Trade Reconciliation (On-Book Trades) 29
3.12.4 Best Practices For Order Management 30
3.13 Mass Deletion Request 30
3.14 Drop Copy for Order Information 31
3.15 Strategy ID & STSN 31
4 Connectivity and Session Parameters 32
4.1 Session Identification and Authentication 32
4.1.1Session Identification and Logon Parameters 32
4.1.2 Network Authentication 33
4.1.3 Session Logon 33
4.1.4 Trader Logon 34

Confidential

4.1.5 IP Addresses and Ports 34
4.2 Failover 35
4.2.1 Network Failover 35
4.2.2 Application Failover 35
4.2.3 Best Practice 35
4.2.4 Message Throttling and Queuing 36
4.2.5 Mass Cancellation on Disconnect 36
5 Session Layer 36
5.1 Logon 36
5.2 Sequence Number 36
5.1 Heartbeat 37
5.2 Test Request 37
5.3 Resend Request 37
5.4 Reject 37
5.5 Sequence Reset 37
5.6 Gap Fill Mode 37
5.7 Reset Mode 38
5.8 Logout 38
5.9 Possible Resend 38
5.9.1 Messages from Client 38
5.9.2 Messages to Client 38
5.10 Recovery 39
5.10.1 Outage on the Client Side 39
5.10.2 Outage on MCX FIX Gateway Side 39
5.11. Data Types 40
5.12. Message Header and Trailer 41
5.12.1. Message Header 41
5.12.2 Message Trailer 43
5.13 Encryption/Decryption for Interactive API 44
6. Message Formats 45
6.1 Session Layer 46
6.1.1 Logon 46
6.1.2 Logon Response 48
6.1.3 User Request 49
6.1.4 User Response 49
6.1.5 User Password Change Request 50
6.1.6 User Password Change Response 50
6.1.7 Session Password Change Request 50
6.1.8 Session Password Change Response 50
6.1.9 Logout 51
6.1.10 Heartbeat 51
6.1.11 Test Request 51

Confidential

6.1.12 Resend Request....52
6.1.13 Sequence Reset....52
6.2 Order Handling....52
6.2.1 New Order Single....52
6.2.2 Order Cancel/Replace Request....59
6.2.3 Order Cancel Request....65
6.2.4 Reject....68
6.2.5 User Order Mass Action Request....70
6.2.6 User Order Mass Action Report....74
6.2.7 Trade Capture Report....78
6.2.8 Trading Session Status....87
6.2.9 User Notification....88
6.2.10 Session Details List....89
6.2.11 Execution Report....90
6.3 Multileg Order Handling....99
6.3.1 New Order Complex....99
6.3.2 Modify Order Complex Request....105
6.3.3 New Order Multileg....111
6.4 Ex/DEx....114
6.4.1 Ex/DEx Entry Request....114
6.4.2 Ex/DEx Entry Confirmation....115
6.4.3 Ex/DEx Cancellation Request....116
6.4.4 Ex/DEx Cancellation Confirmation....117
6.4.5 Ex/DEx Modification Request....118
6.4.6 Ex/DEx Modification Confirmation....119
6.4.7 ExDEx notification....120
6.5 Trade Related Handling....122
6.5.1 Trade Modification Request....122
6.5.2 Trade Modification Response....123
6.5.3 Trade Modification Notification....124
6.5.4 Trade Enhancement Notification....125
6.5.5 Resubmit Approval Request....126
6.5.6 Resubmit Approval Confirmation....127
6.6 Others....128
6.6.1 Market Wide OI Violation....128
6.6.2 Market Wide OI Alerts....130
6.6.3 Margin Change Notification....131
6.6.4 Market Status Notification....133
6.6.5 Auction Notification....134
6.6.6 News....135
6.6.7 System Information....135
6.6.8 Business Message Reject....136
6.6.9 User Party Risk Limits Request....138
6.6.10 User Party Risk Limits Report....141

Confidential

6.6.11 Institution Trade Acceptance 144
6.6.12 Institution Trade Rejection Report.....148
7.Occurences 151
7.1 Parties 152
7.2 Requesting Parties.....156
7.3 Target Parties.....158
8. Appendix.....161
9. Annexure for Encryption/Decryption.....167

Confidential

List Of Abbreviations

Abbreviation	Description
FIX	Financial Information eXchange
TCP/IP	Transmission Control Protocol / Internet Protocol
FPL	FIX Protocol Limited
ETI	Enhanced Trading Interface
MIC	Market Identifier Code
GUI	Graphical User Interface
XMIC	Exchange Market Identifier Code.

Confidential

1.Introduction

The target of this documents is to provide a way to access the MCX trading system via an interface using the FIX protocol. The interface includes basic trading functionality for MCX (MCX) Derivatives markets in a consolidated manner.

The MCX FIX Gateway provides the following trading functions:

- Order management
- Execution notifications

Additionally, the MCX FIX Gateway enables participants to subscribe to private trading data for each market type in broadcast form:

  - Trade notifications at session level.

Note: The MCX FIX Gateway does not provide any reference data. Participants are asked to retrieve reference data via the file interface provided on SFTP server.

Confidential

2 Purpose

The purpose of this document is to provide an overview of the MCX FIX Gateway for the trading system.

The focus of the description is to capture trading system specific behavior, highlight where it deviates from the recommended FIX Protocol standard and keep the amount of the FIX specification which needs to be included in this document to a minimum.

3 Service Description

3.1 FIX Session Concept

As per the FIX Protocol standard, a FIX session is defined as a bi-directional stream of ordered messages between two parties within a contiguous sequence number.

A FIX session will be initiated by the participant and maintained between the participant and the FIX Gateway over the course of a trading day.

Fix Gateway supports Trading session with order management, trade and general news events can be subscribed. Each session will receive information for all of its own orders.

3.2 Party Identification

The participant is an entity accessing the MCX - Trading System .

A participant may have several business units as independent entities taking part in trading at the exchange. Business units are identified by a business unit ID. A business unit belongs to a participant.

A user is a person, such as a trader or an exchange market supervisor that interacts with the Trading System. Users are identified by a user ID. A user belongs to one business unit. A user is a trader or administrator that logs on to the system to enter commands on the trading system.

Confidential

3.3 Security Identification

Instruments traded on MCX will be identified by the product identifier (Symbol (55)) and the instrument identifier (SecurityID (48)). Both fields must be provided on the FIX requests operating on instrument level. For messages operating on product level e.g. UserOrderMassActionRequest (UCA) only Symbol (55) will be provided.

For the identification of an instrument traded on MCX Derivatives only the marketplace-assigned identifier (Segment derivative) with SecurityIDSource (22) = "M" (Marketplace-assigned identifier) will be supported and must be pro- vided in the FIX request. Both single and multileg instruments are uniquely identified by the corresponding instrument ID. MCX Derivatives messages sent to the customers will also contain the marketplace- assigned identifier in the component <Instrument>.

FIX request must contain additionally the currency (Currency (15) / UCurrency (30015)) to identify the instrument uniquely.

- ISIN: SecurityID (48) with SecurityIDSource (22) = "4" (ISIN)

<Instrument>	Derivatives	Description in Derivatives - all Messages
Symbol (55)	Yes	Product identifier
SecurityID (48)	Yes	Instrument identifier (marketplace-assigned identifier)
SecurityIDSource (22)	Yes	"M" (Marketplace- assigned identifier)
ProductComplex (1227)	Yes	Instrument type

3.4 OrderID Policy

The standard FIX policy regarding usage of Client Order IDs is supported by the MCX FIX Gateway. Order related messages must include a unique customer defined identifier, the Client Order ID, in the ClOrdID (11) field.

ClOrdIDs with 20 characters or less are accepted. Characters in ASCII range 32-126 are allowed.

A ClOrdID (11) may only be used once per business day and trading session. But in order modification or cancellation exchange generated order id needs to be specified along with ClOrdID. Additionally the MCXFIX Gateway enforces the uniqueness of ClOrdID (11) values among currently live orders. It is not recommended to reuse ClOrdIDs from inactive orders.

The Client Order ID needs to change on every modification and cancellation request; the original scope is specified by the OrigClOrdID (41). In this way the customer is able to find and track individual requestsby their Client Order ID. This FIX concept is called message chaining and intended for order handling through a single interface and session.

Note: The FIX Gateway will ignore trailing spaces in the field ClOrdID when a client order ID is checkedfor uniqueness among currently live orders. A newly entered ClOrdID (11) is considered

Confidential

duplicate by the FIX Gateway, if it only differs in the number of trailing spaces from the ClOrdID (11) of a live order. In this case the FIX Gateway will send a Business Message Reject (j) message

denoted by Business- RejectReason (380) = 0 (Other) and Text (58) = "ClOrdID is not unique.".

Example: If a live order exists with the ClOrdID (11) = "Test", any request with ClOrdID (11) = "Test " willbe rejected. Note that this has no impact on the OrigClOrdID (41), which still must provide the correct number of trailing spaces to identify the corresponding order.

3.5 Uniqueness Of Identifiers

The following table documents the criteria required for uniqueness of IDs:

FIX Field	Description	Rule	Uniqueness
ClOrdID (11)	Unique customer defined order request identifier.	The uniqueness of ClOrdID (11) is checked at entry timeamong currently live orders for the same session, only if it provided. Duplicate ClOrdID (11) values for the same session will be rejected.	- Session
ExecID (17)	The field ExecID (17) in the Execution Report provides aunique identifier and can be used for the identification of duplicate order messages.	Is unique per session.	- Session
OrderID (37)	Exchange Order ID gener-ated by the MCX System; it remains constant over the lifetime of an order.	An exchange order ID is guaranteed to be unique among all orders of the same product.	- Product
TrdMatchID (880)	Unique identifier for each price level (match step) of amatch event; it is used for public trade reporting (on- book trades only).	Is unique per product and business day.	- Business Day - Product
SecondaryExecID (527) LegExecID (1893) SideTradeID (1506)	Private identifier of an order match step event, which canbe used for reconciliation between Order Executions and Trade Notifications.	Is unique per product and business day.	- Business Day - Product

Confidential

TradeReportID (571)	In messages sent via FIX , the field TradeReportID (571) in the Trade Capture Report pro-vides a unique trade identifier and can be used for the identification of duplicate trade confirmation messages.	Is unique per business day and business unit.	- Business Day - Business Unit
TradeID (1003) OrigTradeID (1126)	The TradeID (1003) field in the Trade Notification uniquely identifies all allocations referring to the same matching event, instrument and price. The OrigTradeID (1126) is delivered for trade reversals and contains the TradeID (1003) of the original trade.	Is unique per product and business day.	- Business Day - Product
TransactTime (60) MassActionReportID (1369)	Transaction timestamp which provides date and time in UTC, represented as nanoseconds past the UNIX epoch (00:00:00 UTC on 1 January 1970).	Is unique per product.	- Product

3.6 TimeStamps

All timestamps which are assigned by exchange provides date and time in UTC, represented as nanoseconds past the UNIX epoch (00:00:00 UTC on 1 January 1970).

3.7 Quantity Fields

Technically, the quantity fields are defined with a format that allows up to 15 digits before decimal point and at most 4 significant decimal places (Qty (15.4)).

Functionally, the usage and the allowed formats for the quantity fields are different depending on the market type and on the instrument setup:

For MCX Derivatives the user is not allowed to enter quantity fields with decimals (other than "0"). Entering a value with decimals or a value higher than the allowed maximum value leads to a rejection.

Confidential

3.8 Pending Responses "Request State Unknown"

For some FIX requests, a pending FIX response "request status unknown" will be generated by the MCX FIX Gateway when the status of the corresponding request is unknown. For example:

- if no response from the back end is received within a certain time (if the back end response is received by the MCX FIX Gateway afterwards an additional response message will be sent).

- if a MCX response does not contain a clearly defined processing result (e.g. ETI field SessionRejectReason (373) = "105" (Error converting response or broadcast)).

in some recovery situations (e.g. several modify order requests with PossDupFlag (43) = "Y" for the same order).

If a pending message indicating that the status of a FIX request is unknown is received, clients are requested to check the status of the request in an alternative way (e.g. via GUI).

Following table contains the information about the requests, for which pending responses "request status unknown" are generated, and the FIX fields used as identification criteria of the corresponding response messages:

Request	Response Message	Response Identification Criteria
User/TradeCaptureReport(UAE/AE)	User/TradeCaptureReportAck(UAR/AR)	TrdRptStatus (939) = 8 (Pending verification)

3.9 TerminalInfo Rules

Terminal Info Rules - for 15-digit unique no
Initial 12 digits

- CTCL - 6 - pin code of Terminal Location,
- 3 digits - Branch id of Terminal Location,
- 3 - Running serial number of the Terminal within the branch
- IBT - "111111111111"
- DMA - "222222222222"
- Wireless Technology - "333333333333"

13th Digit Not generated through program trading software - "0"

- Generated through program trading software - "1"
- Smart Order Routing Without Program trading software = "2"
- Smart Order Routing with Program trading software = "3"

14-15 digit -valid vendor code / In-house CTCL code for member

3.10 Order Management

A FIX session can only modify or cancel own orders (i.e. orders previously submitted successfully on the same FIX session).

Confidential

3.10.1 Order Types/Trading,Execution and Validity Restrictions

The following order types are supported via the FIX interface:

Order Type	Description	Relevant FIX Tags
Limit (L)	Limit orders include a specified price limit and may not be executed at a price worse than that limit.	- OrdType (40) = '2' - Price (44)
Stop Market (S)	Stop orders are orders that create market orders when the specified trigger price is reached. Stop orders are not visible in the order book for any market participant.	- OrdType (40) = '3' - StopPx (99)
Stop Limit (SL)	Stop limit orders create limit orders when the specified trigger price is reached. Stop limit orders are not visible in the order book for any market participant.	- OrdType (40) = '4' - Price (44) StopPx (99)
Market to Limit	The price at which the market order got converted into regular limit order.	OrderType (40) = '5' MPP (25001)
Immediate or Cancel (IOC)	An IOC order is to be filled immediately, either completely or to the extent possible; the portion that cannot be filled immediately is canceled. The execution restriction IOC is allowed for Market and Limit orders.	- TimeInForce (59) = '3'
Standard	The complete order history of an order having this flag set will be part of the listener broadcast and may be recovered via retransmission requests. Standard orders can be persistent or non-persistent.	ApplSeqIndicator (28703) = 1, (Recovery_Required)
Iceberg (Ice)	Limit Order that contains a peak quantity and an overall quantity. Once the displayed quantity has been completely executed, a new peak is entered into the book. Iceberg is not added as new order type.	MaxShow(210)
Lean	For Lean orders, the execution notifications Can be recovered (via retransmission requests on the session data channel).All data on lean orders are visible only to the session that submitted the order.Lean Orders are always non-persistent.	ApplSeqIndicator(28703) =0(No_ Recovery_Required)

Confidential

Good-for-day (Day)	All orders are assumed to be day orders unless otherwise specified. The validity of a day order ends at the close of that day's trading period.	- TimeInForce (59) = '0'
Good-till-date (GTD)	Order carries a specified date up to one year from entry on which the order is auto-matically canceled.	- TimeInForce (59) = '6' - ExpireDate (432)
Good-till-canceled (GTC)	Order remains valid until it is executed, canceled, or if the contract expires.	- TimeInForce (59) = '1'
End of Session(EOS)	Order remains valid until the end of session	-TimeinForce(59)=7
Persistent	A persistent order is an order that survives a trading interruption. A persistent order is an order that is reinstated at the beginning of the Business Day (if valid for that Business Day) or after a Market Reset.	ExecInst (18) = H (encoded as bit map with first bit being set)

3.10.2 Relevant FIX Fields for the identification of the Order Types

The following table shows the relevant FIX fields for identifying an order type, the usage of the relevanttags and components and the assignments of the tag values depending on the order type attribute.

Following information has to be considered:

- "Y" or "N" will indicate if tags and components are mandatory or optional for the type of orderspecified.
- Other values will describe the tag values allowed/supplied for the type of order specified.
  Example:
  =1/=2 means that one of the valid values "1", "2" must be supplied.

3.10.1.1.1 Empty cells will indicate that a tag or component is not allowed for the type of order specified.

Tag	Field Name	M to L	L	S	SL	IOC
18	ExecInst	N	N	N	N	N
40	OrdType	=5	=2	=3	=4	=1/=2
44	Price		Y		Y	-/Y
59	TimeInForce	N	N	N	N	=3
99	StopPx			Y	Y	
	Derivatives	ü	ü	ü	ü	ü

Confidential

3.10.3 Notional Value And Quantity Checks

When entering or replacing an order, additional checks may be activated to prevent orders with a too large value (notional value check) or with a too large order size (quantity check) from entering the order book.

The notional value check functionality prevents orders with a too large order value from entering the order book. The validation verifies the order value against the maximum order value limit set by the participant for their traders.

The quantity check functionality prevents orders with a too large order size from entering the order book. The validation verifies the order quantity against the maximum order quantity limit set by the participant for their traders. (Iceberg orders will be validated with the full order quantity when entering the order book.)

The execution of these validations depends on the values entered in the component <ValueChecks- Grp>:

3.10.4 Cancellation

The FIX session may only cancel orders that have been entered previously via the same.

Cancelling an order will remove the remainder of a live order from the order book. The participant must use the order id(37) provided in order confirmation. The MCX FIX Gateway will respond with an ExecutionReport (8) or OrderCancelReject (9) message for confirmation or rejection respectively.

Participants can also submit a UserOrderMassActionRequest (UCA) in order to delete all active orders for the respective session in a given product. The UserOrderMassActionRequest (UCA) can be further restricted to a defined trader and/or a defined instrument. The user may delete only part of their orders for one instrument by entering the additional filter criteria side and price.

3.10.5 Modifications

The FIX session may only modify orders that have been entered previously via the same session.

The participant must use the order id(37) to identify the order to modify.

The MCX FIX Gateway will respond with an ExecutionReport (8) or OrderCancelReject (9) message for confirmation or rejection respectively.

The ExecutionReport (8) will contain ExecRestatementReason (378) = 181 (ownerschip changed) if the order ownership was changed. This will be the case if the submitter (Entering Trader) of the modify request is different from the original owner of the order.

Orders that have been completely filled may not be modified anymore.

Note: Modifications of the total order quantity to a quantity less than or equal to the cumulated executed order quantity will be interpreted as a cancel request.

Confidential

3.10.6 Self Match Preventions

Self-Match refers to trades where both buyer and seller clients are same. Such Self-Match does not server any economic purpose of Members/Clients instead Members have to incur additional expenses like Transaction charges, CTT, etc

Buy and Sell orders having the same PAN (belonging to member and/or UCC) are considered as self-match orders and will not be matched with each other.

Self-match prevention (SMPF) refers to a mechanism by which Exchange restricts execution of self-match trades whereby when an active order gets potentially matched with a passive order resulting in Self-Match Trade, System cancels the active or passive order as per SMPF order identifier received with order instead of executing the self-match trade.

3.10.7 Text Fields

The MCX trading system supports three free-format text fields for trader-specific comments to an order. The mapping of the MCX text fields to the FIX tags is as follows:

ETI Text Field	Relevant FIX Tags
Free Text Field 1	Party Role (Client Code)
Free Text Field 2	Party Role (CP Code)
Free Text Field 3	FreeText3 (25002)

Confidential

3.10.8 Order Status and Execution Report

The ExecutionReport (8) message is used to communicate events that affect an order.
The field ExecType (150) specifies the type of event. The field OrdStatus (39) specifies new status of the order.
The different scenarios and usage of the OrdStatus (39) and ExecType (150) are as follows:

Scenario	OrdStatus (39)	ExecType (150)
Order book replay: Transmission of all active orders	0 = New 1 = Partially filled	D = Restated
Successful submission of an order	0 = New	0 = New
Rejected submission of an order	4 = Cancelled	4 = Cancelled
Successful modification of an order	0 = New 1 = Partially filled 2 = Filled	5 = Replaced
Rejected modification of an order	4 = Cancelled	n/a
Successful cancellation of an order	4 = Cancelled	4 = Cancelled
Rejected cancellation of an order	4 = Cancelled	n/a
Partial fill	1 = Partially filled	F = Trade (in FIX 4.4)
Complete fill	2 = Filled	F = Trade (in FIX 4.4)
Triggered Stop Order	0 = New	L = Triggered by system
Modification of an order to Immediate or Cancel (IOC) triggered by the system	0 = New	5 = Replaced
Unsolicited modification triggered by third party	1= New 2= Partially filled 3 = Filled	5 = Replaced
Unsolicited cancellation triggered by third party	4 = Cancelled	4 = Cancelled
Unsolicited cancellation triggered by the trading system	4 = Cancelled	4 = Cancelled

Confidential

Cancellation of not (fully) executed Immediate or Cancel (IOC) Order	4 = Cancelled	4 = Cancelled
Suspended of order (RRM)	4 = Cancelled	4 = Cancelled
Cancellation due to Self-Match Prevention (SMP)	4 = Cancelled	4 = Cancelled

3.10.9 ExecutionReport (8) "Request Status Unknown"

As described in Pending Responses "Request Status Unknown**"** there are situations, in which the MCX FIX Gateway will generate "request status unknown" pending ExecutionReport (8) messages as response to the order requests.

The different scenarios for the order requests and their usage of the fields OrdStatus (39), ExecType (150) and ExecRestatementReason (378) in the ExecutionReport (8) messages sent by the MCX FIX Gateway are as follows:

Scenario	OrdStatus (39)	ExecType (150)	ExecRestatementReason (378)
Submission of an Order - Order status is un-known	A = Pending New	A = Pending New	100 = Unknown Order State
Cancellation of an Order - Order status is un-known	6 = Pending Cancel	6 = Pending Cancel	100 = Unknown Order State
Modification of an Order - Order status is un-known	E = Pending Replace	E = Pending Replace	100 = Unknown Order State

NOTE:

For incoming active order which get fully traded, User will receive order status tag 39 = 2 in 1*st* execution report. Also, in such case 1*st* execution report 35=8 indicates overall status of the order.

Individual trade details can be found in subsequent execution reports which will begin with

Confidential

39=1(partially filled) and ends with 39=2(fully filled).
In case of partial trade, user will receive order status (39) as 1(partially filled) always.

In case of full trade, which is executed partially, user will receive order status as 39=2 fully filled in execution report 35=8 which indicates overall status of the order. Then user will receive subsequent execution reports 35=8 with order status as 39=1 which will show how order is executed partially then at the end user will receive 35=8 with order status as 39=2 in which we will receive tags 31, 32, 527 ,880. In such case we receive 2 execution reports with 39=2 fully fill tag. 1st one will show overall status and last one will give us all the details with tags 31, 32, 527 ,880.

3.10.10 Order Book Restatement

During the start-of-day phase and after a market reset event (an exchange system failure), all active orders of a session will be transmitted to the market participant via the respective session.

During Order Book Restatement ExecutionReport (8) messages for each restated order of the corresponding session are provided and finally a TradingSessionStatus (h) message indicates the end of therestatement per product.

The reason for the restatement is communicated in field ExecRestatementReason (378) in message ExecutionReport (8).
Each end of restatement message initiates the start of trading for a product.
ExecRestatementReason (378) will have the value "1" (GT renewal / restatement).

3.10.11 Mass Cancellation Notification

Mass cancellation notification is not provided on a single order level. The owning session will be informed about the scope of the cancellation by a summary record. The summary record will also provide the entering party involved and the reason for the mass cancellation.
Unsolicited order mass cancellation is communicated by the MCX FIX Gateway via the

OrderMassAction- Report (UBZ) message.

The reason for the mass cancellation event is communicated in field MassActionReason (28721), the scope of the deleted orders in field ExecInst (18).

Orders that couldn't be canceled due to an incompatible instrument state are provided in the component <NotAffectedOrdersGrp>. For each order the Exchange Order ID (NotAffectedOrderID (1371)) is delivered. For messages generated via Trading session the component contains for each order additionally the corresponding FIX Client Order ID (NotAffOrigClOrdID (1372)). This field will be set to "[N/A]" for messages delivered.

The number of entries in the components <AffectedOrdersGrp> and <NotAffectedOrdersGrp> is lim ited. For this reason the information about a Mass Cancellation event can be split into several Order-MassActionReport (UBZ) messages. The message OrderMassActionReport (UBZ) contains the field LastFragment (893) to indicate if the message is the last message related to an event (LastFrag-ment (893) = Y (Last message)) or if additional messages will follow (LastFragment (893) = N (Not last message)).

The following unsolicited mass cancellation events may occur:

Confidential

Mass Cancellation Event	MassActionReason *(28721)*
Product Holiday	Product State Holiday *(106)*
Product Halt	Product State Halt *(105)*
Instrument Suspension	Instrument Suspension *(107)*
Product temporarily not tradeable	Product temporarily not tradeable *(111)*
Member disabled	Member has been disabled *(117)*

Additional events related to technical reasons are also possible. See complete list of values for the field MassActionReason (28721) in the description of message OrderMassActionReport (UBZ).

The information about the scope of the deleted orders as result of the event is delivered in the field ExecInst (18) in message OrderMassActionReport (UBZ):

ExecInst (18)
H
H Q

3.10.12 Trading Session Events

The TradingSessionStatus (h) message is used by the FIX Gateway for all session related events. Following are the list of Trading Session Events.
XMIC- Exchange Market Identifier Code.

Event	Level	TradSesEvent *(1368)*
Market Reset	XMIC	Market reset *(102)*
End of Restatement	Product	End of restatement *(103)*
Service Resumed	Product	Service resumed *(105)*
End of Service	XMIC	No more messages for this trading venue *(200)*
End of Service	FIX session	Message transmis-sion ended *(201)*
Session Disconnect	XMIC	Message processing suspended *(202)*
Session Connect	XMIC	Message processing resumed *(203)*

- **Market Reset**: informs the participant that the matching engine has been restarted; this event can affect only some products of the related exchange (XMIC).
- **End of Service**: informs the participant about the end of message transmission for an exchange (XMIC)within a FIX session or for the whole FIX session.

Confidential

- **Session Disconnect**: informs the participant about the disconnection of the FIX ETI session.
- **Session Connect**: informs the participant about the (re)connection of the FIX session.

Confidential

3.11 Custom Tags

Following are the list of custom tags used in MCX Trading Index FIX API. All custom tags are in the range of 20000 – 31000.

Tag No	Field
20122	StrategySeqNum
25001	MPP
25002	FreeText3
28721	MassActionReason
25023	Return Code
25024	Return Code Source
25025	SMPFOrderIdentifier
25026	StrategyID
25027	StrategySeqNum
25028	MultiLegOrdType
25029	ReturnCodeText
25030	TradeModReqID
25031	OldAccountType
25032	OldClientID
25033	OldCPCode
25034	NewAccountType
25035	NewClientID
25036	NewCPCode
25037	Accept/RejectionTime
25038	TrdModStatus

Confidential

25039	TradeRejReason
25040	ConfirmationTime
25041	ResubmitReqID
25043	GroupID
25044	ViolationStatus
25045	OIQuantityLimit
25046	OIPercentageLimit
25047	EffectiveOILimit
25048	ActualOIQuantity
25049	Last Update Date
25050	AlertLevel
25051	AlertPercentage
25052	UtilizedAlertPercentage
25053	AuctionEnquiryInfoReqID
25054	AuctionEnquiryStatus
25055	NoRecords
25056	AuctionEnquiryTradingStatus
25057	PartyIDClearingUnit
30649	StrategyID

Confidential

3.12 Trade Capture

3.12.1 Trade Capture via Trading Session

MCX Derivatives trading participants will receive trade confirmation on a separate session for their own user id.

3.12.2 Trade Capture Report-Identification Criteria

3.12.2.1 Message Event Source

The field MessageEventSource (1011) is used in User/TradeCaptureReport (UAE/AE) messages sent to the customers. The field describes the event that caused the creation of the message and can be used to distinguish the messages of the different message flows.

The User/TradeCaptureReport (UAE/AE) message is used to receive trade confirmations (on-book).

Trade Type	MessageEventSource (1011)
on-book	200 = On-Book Trade Notification

3.12.3 Trade Reconciliation (On-Book Trades)

There are several identifiers that can be used to associate an ExecutionReport (8) with User/Trade- CaptureReports (UAE/AE) and public trades on the market data interface.

Every **match event** with one or more executions (match steps) in a simple or complex instrument results in one ExecutionReport (8) message per matching step for each order. A User/TradeCaptureReport (UAE/AE) will then be sent to confirm each trade at each price level.

For complex instruments (only MCX Derivatives), there is a User/TradeCaptureReport (UAE/AE) for each leg execution of the instrument.

Every **match step** occurring in the exchange has an identifier that is provided in the field TrdMatchID (880) in the ExecutionReport (8) as well as in the User/TradeCaptureReport (UAE/AE). This identifier allows participants to link Trade Capture Reports and the corresponding Execution Report.

The TradeID (1003) field in the User/TradeCaptureReport (UAE/AE) uniquely identifies all allocations referring to the same matching event, instrument and price.

The field SideTradeID (1506), which is unique for a product and business day, in the User/TradeCapture-Report (UAE/AE) provides the private identifier of an order match event, which can easily be reconciled with the corresponding ExecutionReport (8) for orders in the following way:

- for order match events in simple instruments, the ExecutionReport (8) message provides a private execution identifier, SecondaryExecID (527).

Confidential

- for order match events in complex instruments (only MCX Derivatives) the ExecutionReport (8) message provides the order execution ID on each price level and additionally the order leg execution ID, LegExecID (1893).

Match Reporting	ExecutionReport (8)	User/TradeCapture-Report (UAE/AE)
Trade event on instrument level: public trade volume reporting.	TrdMatchID (880)	TrdMatchID (880)
Identifier for all allocations referring to the same instrument.		TradeID (1003), OrigTradeID (1126)
Private execution identifier in Order in a simple instrument.	SecondaryExecID (527)	SideTradeID (1506)
Private execution identifier for an order in a complex instrument. (only MCX Derivatives)	LegExecID (1893)	SideTradeID (1506)
System Order ID generated by the MCX System.	OrderID (37)	OrderID (37)
Unique identifier of the order. The Client Order ID of the MCX Enhanced Trading Interface (ETI) is provided.	SecondaryOrderID (198)	SecondaryOrderID (198)

3.12.4 Best Practices For Order Management

All order response information in the MCX FIX Gateway is sent out immediately after the order has been processed by the core matching process.

All order response information in the MCX FIX Gateway is preliminary; this includes ExecutionReports (8) sent out for persistent.

For these reasons, a participant application always needs to confirm the preliminary execution information with the corresponding legally binding User/TradeCaptureReport (UAE/AE).

User/TradeCaptureReports (UAE/AE) are not delivered via Trading FIX sessions. For the reception of the legally binding User/TradeCaptureReports (UAE/AE) is required.

3.13 Mass Deletion Request

The UserOrderMassActionRequest (UCA) will allow deletion of multiple orders. Orders may be filtered by Product identifier (Symbol) or Product identifier (Symbol) and Security identifier (SecurityID).
Note: The ProductComplex (1227) will not be allowed on this request as no filtering by

instrument type will be supported. It is not possible - for example - to restrict a mass

cancellation operation to "Standard Option Strategies".
The user may delete orders owned by a different trader. In this case the owning trader of the

Confidential

orders to be deleted must be provided in the party <target executing trader>.
Users may delete only part of their orders for one instrument by entering the additional filter

criteria side and price. For the buy side the orders will be deleted starting from the highest price until the price specified in the filter, for the sell side starting from the lowest price.

The request will be answered by one or more UserOrderMassActionResponse (UCAR) messages having MassActionResponse (1375) set to "2" (Completed), if successful.

A rejected request will be answered by a UserOrderMassActionResponse (UCAR) message having MassActionResponse (1375) set to "0" (Rejected) and providing an error code/explaining text in Return-Code (25023) / ReturnCodeText (25029) respectively.

Additional information in the response message UserOrderMassActionResponse (UCAR) for requests processed successfully:

- Orders that couldn't be canceled due to an incompatible instrument state are provided with their Exchange Order ID (NotAffectedOrderID (1371)) and with their FIX Client Order ID (NotAffOrig- ClOrdID (1372)) in the component <NotAffectedOrdersGrp>.

- The number of entries in the components <AffectedOrdersGrp> and <NotAffectedOrdersGrp> is limited. For this reason the response to a Mass Cancellation Request can be split into sev- eral UserOrderMassActionResponse (UCAR) messages. The message UserOrderMassAction- Response (UCAR) contains the field LastFragment (893) to indicate if the message is the lastresponse message related to a Mass Cancellation Request (LastFragment (893) = Y (Last message)) or if additional messages will follow (LastFragment (893) = N (Not last message)).

3.14 Drop Copy for Order Information

Drop copy functionality for standard orders of the current business day is provided by ETI Channel.

3.15 Strategy ID & STSN

The Strategy that a member wants to implement through Algorithm Trading shall be approved by the exchange. Post Approval Exchange will allocate particular Strategy ID to specific User ID(ATS User) for the approved strategy.

- All Order Entry/ Modification / Cancellation requests coming through third party application should contain the Strategy ID. If Order Entry/ Modification / Cancellation requests contains the Strategy ID other than assigned to the user the respective Order Entry/ Modification / Cancellation requests shall be rejected by the system.

- The users can be assigned product specific strategies with different parameters and definition. For Strategy approved by exchange, Strategy ID may be assigned as 'NNNNN'. All parameters of the user defined strategies should be stored by the third party applications.

- Apart from the strategy Id, the third party application is expected to send 'Strategy Trigger Sequence Number (STSN)' with each of the entry / modification / cancellation request.

- ddd denotes the Day of Year. In case if day of year is 1 then, '001' needs to be entered

Confidential

- NNNNN - Number stored by 3rd party applications for product specific strategy, Max range is 99999

- Small N's- sequence running number, should be incremented with every request

Thus, the first part dddNNNNN - represent user defined strategies for a particular day of year.

- Each time there is need of sending entry or modification or cancellation request to exchange, third party application is expected to generate new STSN.
- STSN should also be stored with relevant values of trigger / Opportunity

- 3rd party application should have provision for complete audit trail of values defined by user against a strategy. As part of exchange inspection / system audit, this data will be verified with actual orders and values sent.

4 Connectivity and Session Parameters

4.1 Session Identification and Authentication

4.1.1 Session Identification and Logon Parameters

For each FIX session, a unique identifier, the SenderCompID (49) and a Password is assigned by MCX on registration. A participant may have multiple FIX sessions (connections to the FIX Gateway).

For security reasons a Password must be specified on the Logon (A) message. The initial password assigned by MCX for each FIX session should be changed during the first logon by specifying New-Password in the Logon (A) message.
Participants needs to use same password during session login and user login request.

Password Policy

- It should be Minimum of 8 characters in length and maximum 12 characters.
- It must be composed of following three-character sets:

o Upper / Lower 26 letters of English Alphabet A-Za-z o Ten digits 0-9

o Special characters (32 characters) - ('~!@#$%%^&*()_+-={}|[]\:;'<>?,./)

o Password should be case sensitive I.e. ABC is not as abc.

o If wrong password is attempted 5 times then session will get locked. In Login page error will come "Session locked out"

o Space is not a valid password character.

o Reset password- last 5 passwords cannot be used.

- Password has to be encrypted as per RSA token.

Confidential

- When password expires , user can send any one password change message for renewal.
- Connnection Gateway Request will succeed with existing password even if password is expired.

In exceptional circumstances, a password may need to be reset. Participants are able to perform a password reset via the Member Section (MAT).

All messages sent to the FIX Gateway should contain the assigned unique identifier of the FIX session in the field SenderCompID (49) in the TargetCompID (56) field.

The FIX Gateway has a two-step logon procedure, with a Logon (A) message (Session Logon) followed by one User Request (UBE/BE) messages (Trader Logons) at an application level.

4.1.2 Network Authentication

The FIX Gateway will validate the subnet/IP Address from where the FIX session is initiated during session logon. The FIX session logon (Logon (A) message) will be rejected by the FIX Gateway if the subnet cannot be authenticated. Participants are allowed to initiate/resume their FIX sessions from alternate locations, e.g., a backup site or disaster recovery location, MCX permits the setup of up to five IP subnet addresses for FIX session IDs.

4.1.3 Session Logon

The Logon (A) message authenticates a FIX session and establishes a connection to the FIX Gateway. This message must be the first one sent by the client. The FIX Gateway will validate the SenderCompID (49) and EncryptedPassword, TargetCompID(56) A successful logon will initiate a FIX session.

The MCX FIX Gateway support encryption. EncryptMethod (98) must therefore be set to "1400" (Tripedes) .An encrypted password must be sent to the gateway.

As an additional safeguard measure, the TestMessageIndicator (464) is used to indicate whether a FIXsession to be initiated will be used for Simulation or Production purposes. The FIX Gateway will reject a Logon (A) message in the event that the TestMessageIndicator (464) value does not match the target environment.

In order to enhance operational support and error analysis on both the session and application level, information about the client's FIX engine (FIXEngineName (1600),

FIXEngineVersion (1601), FIXEngin- eVendor (1602) as well as the used FIX application (ApplicationSystemName (1603), ApplicationSys- temVersion (1604), ApplicationSystemVendor (1605)) must be provided by the client in the Logon (A) message.

Note: The Logon (A) message is not used to log on and authenticate a trader on the

MCX trading system.

Confidential

4.1.4 Trader Logon

The User Request (UBE/BE) message identifies and authenticates a qualified trader establishing access to the MCX trading system.

Trading Session

The participant must provide the corresponding MCX User ID of the trader in the Username (553) field, and the corresponding password in the EncryptedPassword field.

A successful trader logon will grant the trader access to the MCX trading system.

A trader logon requires an active connection to the MCX trading system (indicated by a TradingSession- Status (h) message with TradSesEvent (1368) = 203 (Message processing resumed) sent previously by the FIX Gateway). Order related messages will only be accepted by the trading system if a traderis logged on successfully. Otherwise these messages will be rejected (e.g. "User not logged in" in the message ExecutionReport (8)) and have to be sent again by the customer using a new MsgSeqNum (34) and a new ClOrdID

(11). It is strongly recommended that order related messages should only be sent if a previous trader logon was positively confirmed.

4.1.5 IP Addresses and Ports

The FIX connection between a member's infrastructure and the MCX FIX Gateway service is established via a TCP/IP connection. The service comprises of primary and secondary

gateways, operated in the MCX Simulation and Production environments. The respective gateways will use distinct target IP addresses and port numbers.

For each FIX session, two individual IP addresses and port numbers are assigned and communicated by MCX. Primary IP address and port is for default usage. Secondary combination is reserved for emergency cases (e.g. line outage).

Confidential

4.2 Failover

The MCX FIX Gateway service features a redundant setup of all components to provide a high level of availability and fault tolerance, and to facilitate a client's implementation of failover in network- and application level failure scenarios. Its setup offers connectivity to both FIX Gateways and thus provides a client's application with the possibility to select which FIX Gateway it will connect to.

A FIX session may be initiated via all available connections, but every session may only be instantiated once. Each TCP/IP connection may only support one session instance.

Both participant ports on the primary and on the secondary FIX Gateway are open. Every FIX session may only be logged in once via one of the connections. Per default only the primary FIX Gateway is connected to the market back end. Therefore the first FIX session logon to the secondary FIX Gateway may take some seconds.

4.2.1 Network Failover

The minimal network configuration that enables a network failover comprises two connections via dedicated leased line and/or via the Internet. Each line is unchangeably assigned to one FIX Gateway, one to the primary, the other to the backup gateway.

After a successful FIX logon to the secondary FIX Gateway, the port of the primary FIX Gateway connection will remain open, but any further logon attempts to the primary FIX Gateway connection will lead to a disconnect of this session.

4.2.2 Application Failover

In the event of a FIX Gateway failure, active FIX sessions connected to this gateway will be disconnected and the corresponding port will be closed. There will be no automatic FIX session failover in case of a FIX Gateway failure.

4.2.3 Best Practice

In all failover scenarios described above, participants may resume a FIX session for the same Sender-CompID (49) via connection to the secondary FIX Gateway. Participants should therefore implement a failover mechanism in their application, in order to be able to establish a FIX session over the alternative connection.
If a connection or a session logon fails or is not responded to immediately, a second attempt should only be made after a few seconds (30 seconds recommended).
Note: A failover will not cause a reset of sequence numbers on the FIX Gateway side, neither is a reset of sequence numbers required in the participant's application. After re-establishment of the FIX session via the alternative connection, the regular retransmission process of missed messages starts.

Confidential

4.2.4 Message Throttling and Queuing

All messages will be processed by the FIX engine on the exchange side. Nevertheless participants should not send more than n application messages (as specified by the exchange) per second, trading market and FIX session in order to prevent the MCX FIX Gateway from queuing. In case of exceeding a rate of n messages per second, the FIX Gateway may queue the affected messages internally and forward them subsequently to the back end, maintaining the maximum back end throttle rate.

4.2.5 Mass Cancellation on Disconnect

The FIX Gateway does not cancel orders in the event of a FIX session disconnection.

5 Session Layer

5.1 Logon

The Logon (A) message is the first message the participant needs to send after the TCP connection has been established. No encryption is supported by the FIX Gateway.

As the first message for the day the participant should send a Logon (A) message with sequence number 1.

A FIX session is identified by the field SenderCompID (49) and TargetCompID (56) in the message header.

SenderCompID (49), EncryptedPassword and BeginString (8) are validated during the session logon. If validation fails, the FIX Gateway will send a Logout (5) message specifying the reason for the rejection followed by the termination of the TCP connection.

Note: If validation during session logon has failed, the sequence number will not be reset.

In the event of an intraday restart the Logon (A) response message may provide a sequence number higher than expected by the participant. This would indicate that messages were missed. The participant should send a ResendRequest (2) message to trigger retransmission of the missed messages (please refer to chapter 5.5 Resend Request for more details).

Logon requests with ResetSeqNumFlag (141) set to "Y" will trigger a reset of sequence numbers at the participant side only. The FIX Gateway's sequence numbering will remain unchanged. Thus the customer is able to access all messages disseminated by the FIX Gateway including the transmission of all active orders at start of the business day.

Note: If a FIX session is successfully logged on subsequent Logon (A) messages will be discarded.

5.2 Sequence Number

All FIX messages are identified by a unique sequence number. The FIX Gateway will process messages in sequence per tradeable instrument.

Sequence numbers are reset by the FIX Gateway during down time after the end of each trading day. The same behaviour is expected for the FIX engine on the client side.

Sequence numbers sent by the client which are behind sequence expected will trigger a logout and TCP connection drop by the FIX Gateway.
Sequence numbers ahead of sequence will trigger a message recovery by the FIX Gateway
Via the ResendRequest (2) message.

Confidential

5.1 Heartbeat

The HeartBtInt (108) has to be specified by the participant during the FIX session logon.

A Heartbeat (0) message should be sent by the participant if no other message has been processed during the defined HeartBtInt (108) interval.

5.2 Test Request

A TestRequest (1) message should be sent if no in sequence message has been received for more than the heartbeat interval. If no in sequence message is received after that for more than the heartbeat interval, the TCP connection should be dropped.

5.3 Resend Request

A ResendRequest (2) message initiates the retransmission of missed messages and can be used if a sequence number gap has been detected. A Resend Request (2) message needs to be processed even if it is ahead of sequence.

The PossDupFlag (43) field set to "Y" in the Message Header of a FIX message indicates that a FIX engine is repeating transmission of already sent content (including MsgSeqNum (34)). In this case a new value is set in the field SendingTime (52) and the sending time of the original message is delivered in field OrigSendingTime (122).

The MCX FIX Gateway supports open or closed sequence range in a Resend Request (2) message (an open range is indicated by sequence number zero as the EndSeqNo (16)).

Note: No Gap Fill messages should be sent by the participant during the resend series for application messages. Application messages should always be re-transmitted since the MCX FIX Gateway requires all missed application messages for the purpose of reconciliation with the MCX trading system.

For resend request the maximum limit of recoverable data is data of session.

5.4 Reject

Session level rejects are used by the MCX FIX Gateway to indicate violations of the session protocol, missing fields or invalid values.

5.5 Sequence Reset

Two types of SequenceReset (4) messages are supported: Gap Fill mode and Reset mode.

5.6 Gap Fill Mode

This type of SequenceReset (4) message is the response to a ResendRequest (2) message. Gap Fill mode is indicated by GapFillFlag (123) field = "Y".

All gap fill messages should have PossDupFlag (43) = "Y" in the Message Header.

Note: Gap Fill mode should only be used by the participant for administrative messages

Confidential

5.7 Reset Mode

The Reset Mode of the SequenceReset (4) message may be used by the participant in emergency scenarios where all means of automatic recovery are lost (e.g. in case of an unrecoverable application failure).
Reset Mode is indicated if GapFillFlag (123) = "N" or if the field is omitted.
After the Reset Mode has been triggered, the TestRequest (1) message should be used by the participant to verify that the requested reset has been accepted by the FIX Gateway.

5.8 Logout

The Logout (5) message is used by the participant to gracefully close the FIX session. Messages need to be processed normally by the participant until the FIX Gateway sends the logout confirmation.
The MCX FIX Gateway will send a TradingSessionStatus (h) message when all messages for a FIX session have been processed. The FIX Gateway will subsequently log out the FIX session.
Note: The FIX Gateway will also send a Logout (5) message if validation fails for a FIX session logon. The reason for the rejection is specified in SessionStatus (1409). The Logout (5) message is followed by a drop of the TCP connection.

5.9 Possible Resend

5.9.1 Messages from Client

The FIX Gateway has no specific functionality for FIX messages from client with PossResend (97) = "Y".
Order requests with PossResend (97) = "Y" :

- Requests will be rejected if the ClOrdID (11) contained in the message has been processed before.
- Requests will be processed if the ClOrdID (11) in the request message has not been processed before.

Other requests with PossResend (97) = "Y" :

- No special processing, FIX requests will be processed as usual, independently of the value of the field PossResend (97).

5.9.2 Messages to Client

The FIX Gateway will set PossResend (97) = "Y" to indicate that a message sent to the client may contain information that has been sent under another sequence number.
If the customer receives a message from FIX Gateway containing PossResend (97) = "Y", the customer must check if the information contained in the message has been received in a previous message and has been already processed. If this is the case the customer should discard the message to avoid the processing of duplicate data.

Confidential

This is especially relevant for messages containing trading information (order and trade messages). For these messages the FIX Gateway will deliver fields that can be used for the identification of duplicate messages without checking the whole content of the FIX messages.
Relevant messages and fields to be used for the identification of duplicate messages:

Message content	FIX Message	FIX field with unique identifier	Deriv - atives
Order information	ExecutionReport (8)	ExecID (17)	ü
Trade notifications via FIX Back-office sessions	User/TradeCaptureReport (UAE/AE)	TradeReportID (571)	ü

5.10 Recovery

When a participant reconnects after a FIX session disconnection during the same business day, two different scenarios can be identified as a reason for the outage: namely outage on the client side and outage on MCX FIX Gateway side.

5.10.1 Outage on the Client Side

After resuming the FIX session, the participant may have missed some messages from the FIX Gateway. In this case, the sequence number of the next message received from the FIX Gateway will be ahead of the last MsgSeqNum (34) stored on the participant side.

The participant should send a ResendRequest (2) message in order to trigger all missed messages during the outage.

The FIX Gateway will return all potentially missed messages with PossDupFlag (43) = "Y" to indicate that a message may have been previously transmitted with the same MsgSeqNum (34).

Note: Mass cancellation service on disconnect is not supported by the MCX FIX Gateway. All open orders remain in the order book during an outage including orders.

5.10.2 Outage on MCX FIX Gateway Side

In the unlikely event that the disconnection was due to an outage on the MCX side, the participant should consider the following recovery mechanisms:

After reconnection of the FIX session, the FIX Gateway may receive a sequence number higher than the one expected and sends a ResendRequest (2) message to the participant.

The participant should resend all potentially missed messages with PossDupFlag (43) = "Y", to indicate that a message may have been previously transmitted with the same MsgSeqNum (34). The FIX Gateway will send responses to already processed messages with PossResend (97) = "Y". After a forced failover pending order messages might be rejected. These messages can

Confidential

be submitted again by the participant using a new MsgSeqNum (34) and a new ClOrdID (11).

Note: No Gap Fill messages should be sent by the participant during the resend series for application messages.

Application messages should always be re-transmitted since the MCX FIX Gateway requires all missed application messages for the purpose of reconciliation with the MCX trading system.

If a participant sends Gap Fill messages during the resend series for application messages the related orders might not be accessible any more via the FIX Gateway and related order specific information will not be forwarded to the FIX session.

This also holds true in case of Logon (A) message with ResetSeqNumFlag (141) = "Y"

5.11. Data Types

FIX LF supports the following data types

Data Type	Description
STRING	Alphanumeric free format strings containing characters from the following set: \x20,\x23-\x25, \x28-\x2A, \x2C-\x3B, \x3F, \x41-\x5F, \x61-\x7B, \x7D,\x7E.
INT (based on STRING)	Sequence of digits without commas or decimals and optional leading sign character.
FLOAT (based on STRING)	Sequence of digits with optional decimal point and leading sign character.
CHAR (based on STRING)	Single character value, can include any alphanumeric character.
BOOLEAN (based on CHAR)	Char field containing either 'Y' or 'N'.
MULTIPLEVALUESSTRING (based on STRING)	String containing one or more space delimited char fields.
LOCALMKTDATE (based on STRING)	Date of Local Market in YYYYMMDD format.
CURRENCY (based on STRING)	ISO currency code (3 characters).
LENGTH (based on INT)	Length of the FIX LF message in bytes. Value must be positive.
SEQNUM (based on INT)	Message sequence number. Value must be positive.
NUMINGROUP (based on INT)	Number of entries in a repeating group. Value must be positive.
QTY (based on FLOAT)	FLOAT (15.4) field with 15 digits before decimal point and at most 4 significant decimal places.
PRICE (based on FLOAT)	FLOAT (11.8) field representing a price with 11 digits before decimal point and at most 8 significant decimal places.
AMT (based on FLOAT)	FLOAT (11.8) field representing a prices times a quantity with 11 digits before decimal point and at most 8 significant decimal places.

Confidential

Data Type	Description
UTCTIMESTAMP (based on STRING)	Time/date combination in YYYYMMDD-HH:MM:SS[.ssssssss] format. The milli/micro/nanosecond part is optional. The FIX LF gateway provides timestamps in seconds only.
PERCENTAGE (based on FLOAT)	FLOAT (3.7) field representing a percentage with 3 digits before decimal point and at most 7 significant decimal places.

5.12. Message Header and Trailer

5.12.1. Message Header

Tag	Field Name	Req'd	Data Type	Description
8	BeginString	Y	STRING	Identifies beginning of new message and protocol version. Value: FIX.4.4 — Description: VERSION 44
9	BodyLength	Y	LENGTH	Message length, in bytes, forward to the CheckSum field.
35	MsgType	Y	STRING (3)	Defines the message type. Valid values are listed after this table.
34	MsgSeqNum	Y	SEQNUM	Message sequence number
43	PossDupFlag	N	BOOLEAN	Indicates possible retransmission of message with this sequence number. True — Original transmission False — Possible duplicate
49	SenderCompID	Y	STRING	Assigned identifier of the party sending the message.
52	SendingTime	Y	UTCTIMESTAMP	Time of message transmission.
56	TargetCompID	Y	STRING	Assigned identifier of the party receiving the message.
97	PossResend	N	BOOLEAN	Indicates that message may contain information that has been sent under another sequence number. True — Original transmission False — Possible Resend

Confidential

122	OrigSendingTime	N	UTCTIME-STAMP	Required if PossDupFlag (43) = ”Y”.

Valid Values of MsgType (datatype STRING (3))

Valid Value	Description
0	Heartbeat
1	Test Request
2	Resend Request
3	Reject
4	Sequence Reset
5	Logout
8	Execution Report
A	Logon
j	Business Message Reject
AE	Trade Capture Report
h	Trading Session Status
UBZ	User Order Mass Action Report
U6	Session Details List
U79	System Information
A	Logon Response
D	New Order Single Request
F	Order Cancel Request
G	Order Cancel/Replace Request
AB	New Order Complex
AC	Modify Order Complex Request
AD	New Order Multileg
UCA	User Order Mass Action Request
UCL	User Party Risk Limits Request
UCM	User Party Risk Limits Report
BE	User Request
BF	User Response
UCB	User Notification
U74	ExDEx Entry Request
U75	ExDEx Entry Confirmation
U76	ExDEx Cancellation Request
U81	ExDEx Cancellation Confirmation

Confidential

U77	ExDEx Modification Request
U80	ExDEx Modification Confirmation
U78	ExDEx Notification
U88	Market Wide OI Violation
U89	Market Wide OI Alerts
U91	Margin Change Notification
U90	Market Status Notification
U3	Trade Modification Request
U4	Trade Modification Response
U94	Trade Modification Notification
U86	Resubmit Approval Request
U87	Resubmit Approval Confirmation
U83	User Password Change Request
U85	User Password Change Response
U84	Session Password Change
U82	Session Password Change Response
U44	Auction Notifications
U31	Trade Enhancement Status
B	News
U5	Institution Trade Acceptance
U9	Institution Trade Rejection Report

5.12.2 Message Trailer

Tag	Field Name	Req'd	Data Type	Description
10	CheckSum	Y	INT	Three byte, simple checksum.

Confidential

5.13 Encryption/Decryption for Interactive API

Background

Based on regulatory requirement, the exchange is introducing encryption of messages for ETI API, for encryption the exchange is introducing, the GCM encryption mode with 256 bytes will be utilized for all transaction messages. The Key and IV (Initialization Vector)are provided in the Session Logon Response Message. Connection use the 256 bytes GCM encryption mode with a standard 32-byte key and a 12-byte IV, without the use of an Authenticator.

Overview

Member applications will now establish connections to the Exchange Gateway server, and each message will undergo encryption/decryption using the same session key (symmetric cryptography AES 256 bits GCM mode) at both the member and Exchange ends. The symmetric session key remains valid for the duration of a session. While encryption of messages within member environment towards their client may need to be done by respective members.

Proposed Methodology

Exchange proposes AES-256 GCM - symmetric encryption approach. Following is an overview.

Step 1: Session Logon Request

In the new login process with encryption, users will submit a session logon request with an encrypted password using PKCS#1. In this first message, request is non encrypted with only encrypted password.

Step 2: Session Logon Response

At the application-level encryption, the encrypted password from the session logon request will be utilized, and Gateway will create a Key & IV specific to each session login. These two new fields, key & IV, will be transmitted by the trading platform in the session logon response. In Session Logon response only the Key and IV vector will be encrypted using the password (sent from session logon request) as the key & IV, with the remaining bytes being padded with '0'on the right for both key and IV.

Example: -

Password:- abc@12345 (Received in Session Logon Request)

Correct approach: -

IV: - abc@12345**000**

Key: - abc@12345**000000000000000000000000**

Incorrect approach: -

IV: - 000abc@12345

Key: - **000000000000000000000000**abc@12345

Step 3: User Login Request / Other Order requests

Confidential

For subsequent requests after receiving session logon response, user must encrypt the request (Body fields) using the AES 256 GCM encryption mode using Key and IV shared in session logon response.

Step 4: User Login Response / Other Order response

Gateway will decrypt the request and send an encrypted response. Users are required to utilize the defined fields (Key & IV received in session logon response) to decrypt the responses.

encryption/decryption changes for session logon response and other resp msgs

RAW DATA (Tag 96): It will be received in all message responses except session logon response. This tag will contain the entire body message in encrypted format, the end user needs to decrypt this tag using key and IV received in session logon response, and once decrypted the tag, they will get the entire body fields in decrypted format, which they need to map to the body fields of the respected response structure accordingly.

6. Message Formats

This chapter provides an overview of supported message types and details on the administrative, technical andapplication messages used by the MCX FIX interface.

The structure of the header and trailer as well as details on the components used in application messages areprovided.

Messages	Fields Encrpyted	Mode	Key and IV
Session Logon Request	Message	PKCS#1	Encrypted using PKCS #1 256 bytes gateway public key
Session Logon Response	Only the Key and IV vector will be encrypted	AES_GCM_256 bytes encryption without authenticator	Use Password as Key and IV Appended with '0's for rest of the bytes
All Request message after Session Logon	Entire Message body Encrypted except for headers	AES_GCM_256 bytes encryption without authenticator	Use Key and IV which is shared in Session Logon Response
All Response message after Session Logon	Entire Message body Encrypted except for headers	AES_GCM_256 bytes encryption without authenticator	Use Key and IV which is shared in Session Logon Response

Confidential

6.1 Session Layer

6.1.1 Logon

The Logon message allows the client to connect to the FIX Gateway. It is also used by the FIX Gateway to confirm the logon.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='A' />				
98	EncryptMethod	Y	INT	Method of encryption. Value: 0 – None/other
108	HeartBtInt	Y	INT	Heartbeat interval 0 = gateway will not take any action for skipped heartbeats (only allowed for non-production environments) not set:fix gateway will define interval.
141	ResetSeqNumFlag	N	BOOLEAN	Logon requests with ResetSeqNumFlag (141) set to Y will trigger a reset of Sequence numbers at the participant side only. The Sequence numbering for the messages generated from FIX Gateway will remain unchanged. Value: N – No; Y – Yes, reset Sequence numbers
1401	EncryptedPassword-Len	N	LENGTH	Length in octets
554	EncryptedPassword	Y	STRING(344)	Password.Refer Password policy, Password should be encrypted with RSA and provided.
1600	FIXEngineName	N	STRING (30)	Provides the name of the infrastructure component being used for session level communication. Normally this would be the FIX Engine or FIX Gateway product name. This field is required in the messages sent to the FIX Gateway.
1601	FIXEngineVersion	N	STRING (30)	Provides the version of the infrastructure component. This field is required in the messages sent to the FIX Gateway.
1602	FIXEngineVendor	N	STRING (30)	Provides the name of the vendor providing the infrastructure component. This field is required in the messages sent to the FIX Gateway.

Confidential

Tag	Field Name	Req'd	Data Type	Description
1603	ApplicationSystem-Name	Y	STRING (30)	Provides the name of the application system being used to generate FIX application messages. This will normally be a trading system, OMS, or EMS. This field is required in the messages sent to the FIX Gateway.
1604	ApplicationSystem-Version	Y	STRING (30)	Provides the version of the application system being used to initiate FIX application messages. This field is required in the messages sent to the FIX Gateway.
1605	ApplicationSystem-Vendor	Y	STRING (30)	Provides the vendor of the application system. This field is required in the messages sent to the FIX Gateway.
1408	DefaultCstmApplVer-ID	Y	STRING (30)	FIX LF version.
28763	DefaultCstmApplVer-SubID	N	STRING (5)	Interface subversion. Identifies interface changes within one MCX release. Response only. Value: D0001 – Description: Derivatives subversion
339	TradSesMode	N	INT	Trading Session Mode. Response only. Value: 2 – Description: Simulation/Test Value: 3 – Description: Production
<Trailer />				

Confidential

6.1.2 Logon Response

This is sent by the gateway in response to the logon request.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='A' />				
98	EncryptMethod	Y	INT	Method of encryption. Value: 0 – None/Other
108	HeartBtInt	Y	INT	Heartbeat interval 0 = gateway will not take any action for skipped heartbeats (only allowed for non-production environments) not set:fix gateway will define interval.
141	ResetSeqNumFlag	N	BOOLEAN	Logon request with ResetSeqNumFlag (141) Set to Y will trigger a reset of sequence numbers at the participant side only.The sequence numbering for the messages generated from FIX Gateway will remain unchanged. Value: N – No; Y – Yes, reset sequence numbers
1408	DefaultCstmApplVer-ID	Y	STRING (30)	FIX LF version.
28763	DefaultCstmApplVer-SubID	N	STRING (5)	Interface subversion. Identifies interface changes within one MCX release. Response only. Value: D0001 – Derivatives subversion
339	TradSesMode	N	INT	Trading Session Mode. Response only. Value: 2 – Simulation/Test; 3 – Production
307574	Key	Y	STRING (33)	Cryptographic key for both the encryption and decryption of all messages between member application and allocated Gateway Server.
307575	IVVector	Y	STRING (13)	Cryptographic IV (Initialization Vector) for both the encryption and decryption of all messages between member application and allocated Gateway Server.
<Trailer />				

Confidential

6.1.3 User Request

Each trader needs to logon/logoff to/from MCX via the User Request message.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='BE' />				
553	Username	Y	INT	User ID
554	EncryptedPassword	Y	STRING (344)	Password.Refer Password policy, Password should be encrypted with RSA and provided.
923	UserRequestID	Y	STRING (1)	Unique identifier for a User Request.
924	UserRequestType	Y	INT	Indicates the action required by a User Request Message. Value: 1 – Log on user; 2 – Log off user
<Trailer />				

6.1.4 User Response

The User Response message is used to confirm the trader logon/logoff.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='BF' />				
553	Username	Y	INT	User ID
923	UserRequestID	Y	STRING (1)	Unique identifier for a User Request.
926	UserStatus	Y	INT	Indicates the status of a user. Value: 1 – Logged in; 2 – Not logged in
<Trailer />				

Confidential

6.1.5 User Password Change Request

This message is used to change the password of the user.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U83' />				
553	Username	Y	INT	User ID
554	EncryptedPassword	Y	STRING (344)	Password.Refer Password policy, Password should be encrypted with RSA and provided.
735	EncryptedNewPasswo -rd	Y	STRING(344)	Password.Refer Password policy, Password should be encrypted with RSA and provided.
<Trailer />				

6.1.6 User Password Change Response

This message confirms the User Password Change request.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U85' />				
745	Reserve	N	STRING	Free format string, can include characters, num- bers, punctuation and many special characters
<Trailer />				

6.1.7 Session Password Change Request

This message is for Session Password Change.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U84' />				
20055	PartyIDSessionID	Y	INT	Session ID/Logon ID
554	EncryptedPassword	Y	STRING (344)	Password.Refer Password policy, Password should be encrypted with RSA and provided.
735	EncryptedNewPasswo -rd	Y	STRING(344)	Password.Refer Password policy, Password should be encrypted with RSA and provided.
<Trailer />				

6.1.8 Session Password Change Response

This Mesage confirms session password change.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U82' />				
745	Reserve	N	STRING	Free format string, can include characters, num- bers, punctuation and many special characters
<Trailer />				

Confidential

6.1.9 Logout

The Logout message initiates or confirms the termination of a FIX session.It is also used by the FIX Gateway to reject the FIX session logon.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='5' />				
58	Text	N	STRING (128)	Free format text string.
1409	SessionStatus	N	INT	Session status. Value: 0 – Session active; 4 – Session logout complete
<Trailer />				

6.1.10 Heartbeat

The Heartbeat message may be used by the client and the FIX Gateway to monitor the status of the communication link during periods of inactivity.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='0' />				
112	TestReqID	N	STRING	Identifier included in Test Request message to be returned in resulting Heartbeat.
<Trailer />				

6.1.11 Test Request

The Test Request message is used to trigger a heartbeat message from the opposing application.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='1' />				
112	TestReqID	Y	STRING	Identifier included in Test Request message to be returned in resulting Heartbeat.
<Trailer />				

Confidential

6.1.12 Resend Request

The Resend Request is used by the client and the FIX Gateway to initiate the retransmission of messages in a recovery scenario.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='2' />				
7	BeginSeqNo	Y	SEQNUM	Message Sequence number of first message in range to be resent.
16	EndSeqNo	Y	SEQNUM	Message Sequemce number of last message in range to be resent.
<Trailer />				

6.1.13 Sequence Reset

The Sequence Reset message has two modes: Gap Fill mode is used in response to a Resend Request when one or more messages must be skipped over. Reset mode specifies an arbitrarily higher new Sequence number after an uncoverable application failure.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='4' />				
123	GapFillFlag	N	BOOLEAN	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent. Value: N – Description: Sequence Reset, Ignore Msg Seq Num Value: Y – Description: Gap Fill Message, Msg Seq Num Field Valid
36	NewSeqNo	Y	SEQNUM	New Sequence number.
<Trailer />				

6.2 Order Handling

6.2.1 New Order Single

The New Order Single message is used by the client to submit an order for single leg securities.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='D' />				
<Parties>, cardinality: 1-11, record counter: NoPartyIDs, containing occurrences: beneficiary , client_id , entering_trader , location_id , order_origination_firm , position_account , takeup_firm , execution_identifier_human , execution_identifier_algo , investment_identifier_human , investment_identifier_algo				

Confidential

<table>
<tr><td>448</td><td>>PartyID</td><td>Y</td><td>STRING (35)</td><td>Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).</td></tr>
<tr><td>447</td><td>>PartyIDSource</td><td>Y</td><td>CHAR</td><td>Identifies class or source of the PartyID (448) value. Required if PartyID is specified.<table><tr><th>Value</th><th>Description</th></tr><tr><td>D</td><td>Proprietary custom code</td></tr></table></td></tr>
<tr><td>452</td><td>>PartyRole</td><td>Y</td><td>INT</td><td>Party Role.<table><tr><th>Value</th><th>Description</th></tr><tr><td>36</td><td>Entering trader</td></tr></table></td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Person
</Parties>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Marketplace-assigned identifier

Confidential

Tag	Field Name	Req'd	Data Type	Description
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. Value: 1 – Simple Instrument; 5 – Futures Spread
167	SecurityType	N	STRING (4)	Indicates the type of security. Value: MLEG – Multileg Instrument; 0 – Cumulative at AssetType level; 1 – Underlying Futures; 2 – Options; 3 – Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
<TrdgSesGrp>, cardinality: 0-1, record counter: NoTradingSessions				
336	>TradingSessionID	Y	INT	Not used.
625	>TradingSessionSub-ID	Y	INT	Not used.
</TrdgSesGrp>				
<MtchgInst>, cardinality: 1, record counter: NoMatchInst				
1625	MatchInst	N	INT	Value: 2 – Do not match
28744	MatchInstCrossID	N	INT(10)	Numeric identifier. Contains the Self Match Prevention ID.
</MtchgInst>				

Confidential

Tag	Field Name	Req'd	Data Type	Description
<OrderAttributeGrp>, cardinality: 0-2, record counter: NoOrderAttributes, containing occurrences: liquidity_provision_activity_order , risk_reduction_order				
2594	>OrderAttributeType	N	INT	Type of order attribute. Value: 2 – Liquidity Provision Activity Order Value: 3 – Risk Reduction Order
2595	>OrderAttributeValue	N	STRING	Value associated with the order attribute type specified in OrderAttributeType (2594). Value: Y – Attribute is set.
</OrderAttributeGrp>				
<ValueChecksGrp>, cardinality: 2, record counter: NoValueChecks, containing occurrences: price_check , notional_value_check (NOT USED)				
1869	ValueCheckType	Y	INT	Type of value to be checked. Value: 1 – Price check
1870	ValueCheckAction	Y	INT	Action to be taken for the ValueCheckType (1869). ValueCheckAction (1870) = '2' (Best effort) is only possible, if ValueCheckType (1869) = '1' (Price check). Value: 1 – Check Value: 2 – Best effort
</ValueChecksGrp>				
210	MaxShow	N	QTY	The quantity to be made visible in the market data
25030	SMPFOrderIdentifier	Y	CHAR	Free format string, can include characters, numbers, punctuation and many special characters 0-Passive, 1-Active
25028	MarketProtection-Percentage	N	PERCENTAGE	Price per unit of quantity (e.g. per share)
1	Account	N	STRING (2)	Not used , Must be sent as A1.

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>11</td><td>ClOrdID</td><td>Y</td><td>STRING (20)</td><td>Client Order ID: Unique participant defined order request identifier; used for client order id chaining.</td></tr>
<tr><td>18</td><td>ExecInst</td><td>N</td><td>MULTIPLE-VALUESTRING</td><td>Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Persistent Order(FIX value-H)</td></tr>
</table>
separated by space.</td></tr>
<tr><td>38</td><td>OrderQty</td><td>Y</td><td>QTY</td><td>Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.</td></tr>
<tr><td>40</td><td>OrdType</td><td>Y</td><td>CHAR</td><td>Order type.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>2</td><td>Limit</td></tr>
<tr><td>3</td><td>Stop</td></tr>
<tr><td>4</td><td>Stop limit</td></tr>
<tr><td>5</td><td>Market To Limit</td></tr>
<tr><td>6</td><td>Auction Buy IN</td></tr>
<tr><td>7</td><td>Auction_sell_Out</td></tr>
</table></td></tr>
<tr><td>44</td><td>Price</td><td>N</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>CHAR</td><td>Side
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Buy</td></tr>
<tr><td>2</td><td>Sell</td></tr>
</table></td></tr>
<tr><td>25007</td><td>FreeText1</td><td>N</td><td>STRING (12)</td><td>For own order-send OWN
For client order-send Client code.</td></tr>
<tr><td>59</td><td>TimeInForce</td><td>N</td><td>CHAR</td><td>Specifies how long the order remains in effect. Absence of this field is interpreted as DAY.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Day (DAY)</td></tr>
<tr><td>1</td><td>Good Till Cancelled (GTC) - Standard Orders only</td></tr>
<tr><td>3</td><td>Immediate or Cancel (IOC)</td></tr>
<tr><td>6</td><td>Good Till Date (GTD) - Standard Orders only</td></tr>
<tr><td>7</td><td>Session (EOS)</td></tr>
</table></td></tr>
<tr><td>77</td><td>PositionEffect</td><td>Y</td><td>CHAR</td><td>Must be set to C, Valid characters : \x01-\x7E</td></tr>
<tr><td>99</td><td>StopPx</td><td>N</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
432	ExpireDate	N	LOCALMKT-DATE	Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices.
1031	CustOrderHandling-Inst	N	CHAR	Not used.
1724	OrderOrigination	N	INT	Direct market access identifier. Value: 5 – Direct market access
1815	TradingCapacity	N	INT	Not used. ,
25026	StrategyID	N	INT	Strategy Approved by the exchange should be used.Range being from 0 to 99999.
25029	StrategySeqNum	N	SEQNUM	Strategy sequence numbers.
1300	MarketSegmentID	N	INT(10)	Product identifier.
708	SenderLocationID	N	STRING(20)	SenderLocationID
1090	MaxPriceProtection	N	INT	Price per unit of quantity.
2404	ComplianceText	N	STRING (20)	Identification of the actual trader behind a transaction, required by Market Surveillance.
25008	FreeText2	N	STRING (12)	CPCode needs to be send.
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the order.
<Trailer />				

Confidential

6.2.2 Order Cancel/Replace Request

The Order Cancel/Replace Request is used to modify an existing order.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='G' />				
<Parties>, cardinality: 1-11, record counter: NoPartyIDs, containing occurrences: beneficiary , client_id , entering_trader , execution_identifier_human , execution_identifier_algo , investment_identifier_human , investment_identifier_algo , location_id , order_origination_firm , position_account , takeup_firm				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Value: 36 – Description: Entering trader Party Role.

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Value: 2 – Person Type of PartySubID.
</Parties>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Marketplace-assigned identifier

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1227</td><td>ProductComplex</td><td>N</td><td>INT</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>5</td><td>Futures Spread</td></tr></table>Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc.</td></tr>
<tr><td>167</td><td>SecurityType</td><td>N</td><td>STRING (4)</td><td>Indicates the type of security.<table><tr><th>Value</th><th>Description</th></tr><tr><td>MLEG</td><td>Multileg Instrument</td></tr><tr><td>0</td><td>Cumulative at AssetType Level</td></tr><tr><td>1</td><td>Underlying Futures</td></tr><tr><td>2</td><td>Options</td></tr><tr><td>3</td><td>Futures</td></tr></table></td></tr>
<tr><td>803</td><td>SecuritySubType</td><td>N</td><td>INT(10)</td><td>Strategy type.</td></tr>
<tr><td colspan="5"><Instrument></td></tr>
<tr><td colspan="5"><TrdgSesGrp>, cardinality: 0-1, record counter: NoTradingSessions</td></tr>
<tr><td>336</td><td>TradingSessionID</td><td>Y</td><td>INT</td><td>Not used.</td></tr>
<tr><td>625</td><td>>TradingSessionSub-ID</td><td>Y</td><td>INT</td><td>Not used.</td></tr>
<tr><td colspan="5"></TrdgSesGrp></td></tr>
<tr><td colspan="5"><MtchgInst>, cardinality: 1, record counter: NoMatchInst</td></tr>
<tr><td>1625</td><td>MatchInst</td><td>N</td><td>INT</td><td>Matching Instruction for the order.<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Do not match</td></tr></table></td></tr>
<tr><td>28744</td><td>MatchInstCrossID</td><td>N</td><td>INT(10)</td><td>Numeric identifier. Contains the Self Match Prevention ID</td></tr>
<tr><td colspan="5"><MtchgInst></td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
<OrderAttributeGrp>, cardinality: 0-1, record counter: NoOrderAttributes, containing occurrences: liquidity_provision_activity_order				
2594	>OrderAttributeType	N	INT	Type of order attribute. Value: 2 – Liquidity Provision Activity Order Value: 3 – Risk Reduction Order
2595	>OrderAttributeValue	N	STRING	Value associated with the order attribute type specified in OrderAttributeType (2594). Value: Y – Attribute is set.
</OrderAttributeGrp>				
<ValueChecksGrp>, cardinality: 1-3, record counter: NoValueChecks (NOT USED)				
1869	>ValueCheckType	Y	INT	Identifier for trading session Value: 1 – Price check
1870	>ValueCheckAction	Y	INT	Action to be taken for the ValueCheckType (1869). ValueCheckAction (1870) = '2' (Best effort) is only possible, if ValueCheckType (1869) = '1' (Price check). Value: 1 – Check Value: 2 – Best effort
</ValueChecksGrp>				
210	MaxShow	N	QTY	The quantity to be made visible in the market data
1	Account	N	STRING (2)	Not used , Must be sent as A1.
11	ClOrdID	N	STRING (20)	Client Order ID: Unique participant defined order request identifier; used for client order id chaining.

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>37</td><td>OrderID</td><td>Y</td><td>INT(20)</td><td>Exchange order ID generated by the MCX system.</td></tr>
<tr><td>18</td><td>ExecInst</td><td>N</td><td>MULTIPLE-VALUESTRING</td><td>Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Persistent Order(FIX value-H)</td></tr>
</table>
separated by space.</td></tr>
<tr><td>38</td><td>OrderQty</td><td>Y</td><td>QTY</td><td>Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.</td></tr>
<tr><td>40</td><td>OrdType</td><td>Y</td><td>CHAR</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>2</td><td>Limit</td></tr>
<tr><td>3</td><td>Stop</td></tr>
<tr><td>4</td><td>Stop limit</td></tr>
<tr><td>5</td><td>Market to limit</td></tr>
<tr><td>6</td><td>Auction Buy In</td></tr>
<tr><td>7</td><td>Auction Sell Out</td></tr>
</table>
</td></tr>
<tr><td>41</td><td>OrigClOrdID</td><td>Y</td><td>STRING (20)</td><td>ClOrdID (11) of the last successfully processed task (request) referring to the specific order.</td></tr>
<tr><td>44</td><td>Price</td><td>N</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>CHAR</td><td>Side- 1-Buy , 2-Sell.</td></tr>
<tr><td>25007</td><td>FreeText1</td><td>N</td><td>STRING (12)</td><td>For own order-send OWN
For client order-send Client code.</td></tr>
<tr><td>59</td><td>TimeInForce</td><td>N</td><td>CHAR</td><td>Specifies how long the order remains in effect. Absence of this field is interpreted as DAY.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Day (DAY)</td></tr>
<tr><td>1</td><td>Good Till Cancelled (GTC) - Standard Orders only</td></tr>
<tr><td>3</td><td>Immediate or Cancel (IOC)</td></tr>
<tr><td>6</td><td>Good Till Date (GTD) - Standard Orders only</td></tr>
<tr><td>7</td><td>Session (EOS)</td></tr>
</table>
</td></tr>
<tr><td>77</td><td>PositionEffect</td><td>Y</td><td>CHAR</td><td>Must be set to C, Valid characters: \x01-\x7E</td></tr>
<tr><td>99</td><td>StopPx</td><td>N</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
432	ExpireDate	N	LOCALMKT-DATE	Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices.
1031	CustOrderHandling-Inst	N	CHAR	Not used.
1724	OrderOrigination	N	INT	Direct market access identifier. Value: 5 – Direct market access
1815	TradingCapacity	N	INT	Not used.
25026	StrategyID	N	INT	Strategy Approved by the exchange should be used.Range being from 0 to 99999.
25029	StrategySeqNum	N	SEQNUM	Strategy Sequence numbers.
708	SenderLocationID	N	STRING(20)	SenderLocationID
1090	MaxPriceProtection	N	INT	Price per unit of quantity.
2404	ComplianceText	N	STRING (20)	Identification of the actual trader behind a transaction, required by Market Surveillance.
25008	FreeText2	N	STRING (12)	CPCode needs to be send.
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the order.
<Trailer />				

Confidential

6.2.3 Order Cancel Request

The Order Cancel Request is used to delete an existing order.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='F' />				
<Parties>, cardinality: 1-5, record counter: NoPartyIDs, containing occurrences: entering_trader , execution_identifier_human , execution_identifier_algo , investment_identifier_human , investment_identifier_algo				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Person
</Parties>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Marketplace-assigned identifier

Confidential

Tag	Field Name	Req'd	Data Type	Description
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. Value: 1 – Simple Instrument; 5 – Futures Spread
167	SecurityType	N	STRING (4)	Indicates the type of security. Value: MLEG – Multileg Instrument; 0 – Cumulative at AssetType Level; 1 – Underlying Futures; 2 – Options; 3 – Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
11	ClOrdID	Y	STRING (20)	Client Order ID: Unique participant defined order request identifier; used for client order id chaining.
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
41	OrigClOrdID	N	STRING (20)	ClOrdID (11) of the last successfully processed task (request) referring to the specific order.
1724	OrderOrigination	N	INT	Direct market access identifier. Value: 5 – Direct market access
2404	ComplianceText	N	STRING (20)	Identification of the actual trader behind a transaction, required by Market Surveillance.
25026	StrategyID	N	INT	Strategy Approved by the exchange should be used.Range being from 0 to 99999.
25029	StrategySeqNum	N	SEQNUM	Strategy Sequence numbers
708	SenderLocationID	N	STRING(20)	SenderLocationID
1090	MaxPriceProtection	N	INT	Price per unit of quantity.
<Trailer />				

Confidential

6.2.4 Reject

The Reject message is used by the FIX Gateway when a message is received but cannot be properly processed due to a session-level rule violation.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='3' />				
45	RefSeqNum	Y	SEQNUM	Reference message Sequence number.
58	Text	N	STRING (128)	Free format text string.
371	RefTagID	N	INT	The tag number of the FIX field being refer- enced.
372	RefMsgType	Y	STRING	The MsgType (35) of the FIX message being referenced. Values are listed below.
373	SessionRejectReason	N	INT	Code to identify reason for a session-level Reject message.
25023	ReturnCode	N	INT	Return code.
<Trailer />				

Vallues For SessionRejectReason.

Value	Description
0	Invalid tag number
1	Required tag missing
2	Tag not defined for this message type.
3	Undefined tag
4	Tag specified without value
5	Value is incorrect for this tag
6	Incorrect data format for value.
7	Decryption problem
8	Signature problem
9	CompID Problem
10	Sending time accuracy problem
11	Invalid msgtype
12	XML Validation Error
13	Tag appears more than once
14	Tag specified out of required order
15	Repeating group fields out of order.

Confidential

16	Incorrect NumInGroup count for repeating group
17	Non 'data' value includes field delimiter
18	Invalid/Unsupported Application Version
99	Other

Confidential

6.2.5 User Order Mass Action Request

The UserOrderMassActionRequest (UCA) will allow the deletion of multiple orders based on different filter criteria.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='UCA' />				
<Parties>, cardinality: 1-5, record counter: NoPartyIDs, containing occurrences: entering_trader , execution_identifier_human , execution_identifier_algo , investment_identifier_human , investment_identifier_algo				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Person
</Parties>				
<TargetParties>, cardinality: 0-2, record counter: NoTargetPartyIDs, containing occurrences: target_executing_trader , target_session_ID				
1462	>TargetPartyID	Y	INT	PartyID value within an target party repeating group.
1463	>TargetPartyIDSource	Y	CHAR	PartyIDSource value within an target party repeating group. Value: D – Proprietary custom code
1464	>TargetPartyRole	Y	INT	PartyRole value within a target party repeating group. Value: 12 – Executing trader; 55 – Session ID
</TargetParties>				
<Instrument>				

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>55</td><td>Symbol</td><td>Y</td><td>STRING (10)</td><td>MCX product identifier. If no product identifier is provided Symbol (55) must contain '[N/A]'.</td></tr>
<tr><td>48</td><td>SecurityID</td><td>Y</td><td>INT(20)</td><td>MCX instrument identifier.</td></tr>
<tr><td>22</td><td>SecurityIDSource</td><td>Y</td><td>STRING (1)</td><td>Source of the securityID<table><tr><th>Value</th><th>Description</th></tr><tr><td>M</td><td>Marketplace-assigned identifier</td></tr></table></td></tr>
<tr><td>1227</td><td>ProductComplex</td><td>N</td><td>INT</td><td>Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>5</td><td>Futures Spread</td></tr></table></td></tr>
<tr><td>167</td><td>SecurityType</td><td>N</td><td>STRING (4)</td><td>Indicates the type of security.<table><tr><th>Value</th><th>Description</th></tr><tr><td>MLEG</td><td>Multileg Instrument</td></tr><tr><td>0</td><td>Cumulative at AssetType Level</td></tr><tr><td>1</td><td>Underlying Futures</td></tr><tr><td>2</td><td>Options</td></tr><tr><td>3</td><td>Futures</td></tr></table></td></tr>
<tr><td>762</td><td>SecuritySubType</td><td>N</td><td>INT(10)</td><td>Strategy type.</td></tr>
<tr><td colspan="5"></Instrument></td></tr>
<tr><td>11</td><td>ClOrdID</td><td>Y</td><td>STRING (20)</td><td>Client Order ID: Unique participant defined order request identifier; used for client order id chaining.</td></tr>
</table>

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1374</td><td>MassActionScope</td><td>Y</td><td>INT</td><td>Specifies the session scope of Order Mass Action Request.<table><tr><th>Value</th><th>Description</th></tr><tr><td>6</td><td>All orders for a specific session</td></tr><tr><td>7</td><td>All orders for the own session</td></tr><tr><td>100</td><td>All orders for all sessions</td></tr></table></td></tr>
<tr><td>44</td><td>Price</td><td>N</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
<tr><td>54</td><td>Side</td><td>N</td><td>CHAR</td><td>Side<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>1724</td><td>OrderOrigination</td><td>N</td><td>INT</td><td>Direct market access identifier.<table><tr><th>Value</th><th>Description</th></tr><tr><td>5</td><td>Direct market access</td></tr></table></td></tr>
<tr><td colspan="5"><Trailer /></td></tr>
</table>

Confidential

6.2.6 User Order Mass Action Report

This message informs about mass cancellation events.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='UBZ' />				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier. If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Description: Marketplace assigned identifier
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. 1 – Simple Instrument 5 – Futures Spread
167	SecurityType	N	STRING (4)	Indicates the type of security. MLEG – Multileg Instrument 0 – Cumulative at AssetType Level 1 – Underlying Futures 2 – Options 3 – Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
<Parties>, cardinality: 3-4, record counter: NoPartyIDs, containing occurrences: entering_trader , entering_firm , executing_trader , session_ID				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>447</td><td>>PartyIDSource</td><td>Y</td><td>CHAR</td><td>Identifies class or source of the PartyID (448) value. Required if PartyID is specified.<table><tr><th>Value</th><th>Description</th></tr><tr><td>D</td><td>Proprietary custom code</td></tr></table></td></tr>
<tr><td>452</td><td>>PartyRole</td><td>Y</td><td>INT</td><td>Party Role.<table><tr><th>Value</th><th>Description</th></tr><tr><td>36</td><td>Entering trader</td></tr></table></td></tr>
<tr><td>2376</td><td>>PartyRoleQualifier</td><td>N</td><td>INT</td><td>Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyID-Source (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request mes-sages.<table><tr><th>Value</th><th>Description</th></tr><tr><td>22</td><td>Algorithm</td></tr><tr><td>24</td><td>Natural person</td></tr></table></td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Description: Person
</Parties>				
<NotAffectedOrdersGrp>, cardinality: unrestricted, record counter: NoNotAffectedOrders				
1371	NotAffectedOrderID	N	INT(20)	Exchange Order ID of an order whose cancellation is pending.
</NotAffectedOrdersGrp>				
11	ClOrdID	N	STRING (20)	Client Order ID: Unique participant defined order request identifier; used for client order id chaining.
54	Side	N	CHAR	Side Value: 1 – Description: Buy Value: 2 – Description: Sell
44	Price	N	PRICE	Price per unit of quantity (e.g. per share).
30018	UExecInst	N	MULTIPLE-VALUESTRING	Value: 1 – Description: Persistent Order(FIX value-H)
1369	MassActionReportID	Y	INT(20)	Transaction timestamp.

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>28721</td><td>MassActionReason</td><td>Y</td><td>INT</td><td>Reason for mass cancellation.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>No special reason</td></tr>
<tr><td>1</td><td>Trading was stopped</td></tr>
<tr><td>2</td><td>Emergency</td></tr>
<tr><td>4</td><td>Stop Button Activated</td></tr>
<tr><td>5</td><td>Business unit suspended</td></tr>
<tr><td>6</td><td>Session loss or logout</td></tr>
<tr><td>8</td><td>Clearing Risk Control</td></tr>
<tr><td>100</td><td>Internal connection loss</td></tr>
<tr><td>107</td><td>Instrument Suspended</td></tr>
<tr><td>108</td><td>Price Brand Shrink</td></tr>
<tr><td>119</td><td>RRM Market Wide Position Limit</td></tr>
<tr><td>3</td><td>Market Maker protection</td></tr>
<tr><td>7</td><td>Duplicate Session Login</td></tr>
<tr><td>105</td><td>Product State Halt</td></tr>
<tr><td>106</td><td>Product State Holiday</td></tr>
<tr><td>109</td><td>Complex instrument deletion</td></tr>
<tr><td>110</td><td>Volatility Interruption</td></tr>
<tr><td>111</td><td>Product temporarily not tadeable</td></tr>
<tr><td>117</td><td>Member disabled</td></tr>
<tr><td>120</td><td>Product state EOD</td></tr>
</table>
</td></tr>
<tr><td colspan="5"><Trailer /></td></tr>
</table>

Confidential

6.2.7 Trade Capture Report

The Trade Capture Report message is used to report trades and trade reversals.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='AE' />				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier. If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Description: Marketplace-assigned identifier
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. 1 – Simple Instrument 5 – Futures Spread
167	SecurityType	N	STRING (4)	MLEG – Multileg Instrument 0 – Cumulative at AssetType Level 1 – Underlying Futures 2 – Options 3 – Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
<RegulatoryTradeIDGrp>, cardinality: 1, record counter: NoRegulatoryTradeIDs				
1903	RegulatoryTradeID	N	STRING	Trading Venue Transaction Identifier (TVTIC).

Tag	Field Name	Req'd	Data Type	Description

Confidential

1906	RegulatoryTrade-IDType	N	INT	Specifies the type of trade identifier provided in RegulatoryTradeID (1903). Value: 5 – Description: Trading venue transaction identifier (TVTIC)
<RootParties>, cardinality: 0-1, record counter: NoRootPartyIDs, containing occurrences: root_execution_venue				
1117	>RootPartyID	Y	STRING (10)	Root party identifier.
1118	>RootPartyIDSource	Y	CHAR	Identifies class or source of the RootPartyID (1117) value. Value: D – Description: Proprietary custom code Value: G – Description: Market identifier code
1119	>RootPartyRole	Y	INT	Identifies the type role of the RootPartyID (1117) specified. Value: 1 – Description: Executing firm Value: 12 – Description: Executing trader Value: 36 – Description: Entering trader Value: 73 – Description: Execution venue
</RootParties>				
<TrdCapRptSideGrp>, cardinality: 1, record counter: NoSides, containing occurrences: Side1				
54	Side	Y	CHAR	Side of trade. Value: 1 – Description: Buy Value: 2 – Description: Sell
1009	SideLastQty	N	QTY	Used to indicate the quantity on one of a multi-sided Trade Capture Report.
1506	SideTradeID	N	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
<Parties>, cardinality: 1-23, record counter: NoPartyIDs				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).

Confidential

Tag	Field Name	Req'd	Data Type	Description
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Entering trader
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyID-Source (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request mes-sages. Value: 22 – Algorithm; 24 – Natural person

Tag	Field Name	Req'd	Data Type	Description

Confidential

802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Description: Person
</Parties>				
25027	PartyIDClearingUnit	N	STRING	Party ID Clearing Unit.
1	Account	N	STRING (2)	Not used , Must be sent as A1.
77	PositionEffect	N	CHAR	Must be set to C, Valid characters : \x01-\x7E
25007	FreeText1	N	STRING (12)	For own order-send OWN For client order-send Client code.
1115	OrderCategory	N	CHAR	Defines the type of interest behind a trade (fill or partial fill). Value: 1 – Description: Order Value: 3 – Description: Multileg Order
1444	SideLiquidityInd	N	INT	Indicates whether the order added or removed liquidity. Field is used only for on-book trades. Value: 1 – Description: Added Liquidity (passive) Value: 2 – Description: Removed Liquidity (aggressive) Value: 4 – Description: Auction (neither passive nor aggressive, includes VDO matching at midpoint)
1851	StrategyLinkID	N	INT(10)	Identifier that links all trades resulting from a match step of a strategy order.
37	OrderID	N	INT(20)	Exchange Order ID generated by the MCX system.

Confidential

Tag	Field Name	Req'd	Data Type	Description
40	OrdType	N	CHAR	Value: 2 Limit; 3 Stop; 4 Stop limit; 5 Market To Limit; 6 Auction Buy IN; 7 Auction_sell_Out
44	Price	N	PRICE	Price per unit of quantity (e.g. per share).
151	LeavesQty	N	QTY	Quantity open for further execution. If the Ord-Status (39) is Canceled, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty (38) - CumQty (14).
14	CumQty	N	QTY	Cumulated executed quantity of an order. Total quantity (e.g. number of shares) filled.
<OrderAttributeGrp>, cardinality: 1-2, record counter: NoOrderAttributes				
2594	>OrderAttributeType	N	INT	Type of order attribute. Value: 2 Liquidity Provision Activity Order; 3 Risk Reduction Order
2595	>OrderAttributeValue	N	STRING	Value associated with the order attribute type specified in OrderAttributeType (2594). Value: Y Attribute is set.
</OrderAttributeGrp>				
1031	CustOrderHandling-Inst	N	CHAR	Not used.
25008	FreeText2	N	STRING (12)	CPCode needs to be send.
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the or-der.

Confidential

Tag	Field Name	Req'd	Data Type	Description
28585	SideLastPx	N	PRICE	Fill price for the original Eurex strategy. Only provided in case of a multileg instrument.
</TrdCapRptSideGrp>				
30	LastMkt	Y	EXCHANGE	Operating MIC. Market Identifier Code of the trading market according to ISO 10383.
31	LastPx	Y	PRICE	Price of this (last) fill.
32	LastQty	Y	QTY	Quantity (e.g. shares) bought/sold on this (last) fill.
75	TradeDate	Y	LOCALMKT-DATE	Indicates date of trade referenced in this message in YYYYMMDD format.
442	MultiLegReporting-Type	N	CHAR	This field indicates if the Trade Capture Report results from a single leg or multileg order. Value: 1 – Single Order; 2 – Complex Order
571	TradeReportID	Y	STRING (80)	Unique identifier of the Trade Capture Report.
574	MatchType	N	INT	The point in the matching process at which this trade was matched. 3 – Confirmed Trade Report (reporting from recognized markets.) 4 – Auto match incoming order 11 – Auto match resting order
700	ReversalIndicator	N	BOOLEAN	Reversal Indicator. Value: N – No reversal; Y – Reversal
748	TotNumTradeReports	N	INT(10)	Number of leg executions of the original strategy order.

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>828</td><td>TrdType</td><td>Y</td><td>INT</td><td>Identifier of the trade type.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Regular Trade (on-book)</td></tr><tr><td>1</td><td>Block Trade</td></tr><tr><td>12</td><td>Exchange for Swap (EFS)</td></tr><tr><td>50</td><td>Portfolio Compression Trade</td></tr><tr><td>1000</td><td>Vola Trade</td></tr><tr><td>1001</td><td>Exchange for physical (EFP) Fin trade</td></tr><tr><td>1002</td><td>Exchange for physical (EFP) Index Future trade</td></tr><tr><td>1004</td><td>Trade at market</td></tr><tr><td>1006</td><td>MCX EnLight trig-gered trade</td></tr><tr><td>1007</td><td>QTPIP Block Trade</td></tr></table></td></tr>
<tr><td>830</td><td>TransferReason</td><td>Y</td><td>INT</td><td>Not used.</td></tr>
<tr><td>856</td><td>TradeReportType</td><td>Y</td><td>INT</td><td>Not used.</td></tr>
<tr><td>880</td><td>TrdMatchID</td><td>N</td><td>INT(10)</td><td>Identifier assigned to a trade by a matching sys-tem.</td></tr>
<tr><td>1003</td><td>TradeID</td><td>Y</td><td>INT(10)</td><td>The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.</td></tr>
<tr><td>1011</td><td>MessageEventSource</td><td>Y</td><td>INT</td><td>Receiver of a message.<table><tr><th>Value</th><th>Description</th></tr><tr><td>200</td><td>On-Book Trade Notification</td></tr></table></td></tr>
<tr><td>1126</td><td>OrigTradeID</td><td>N</td><td>INT(10)</td><td>Used to preserve original trade id when original trade is being referenced in a subsequent trade transaction such as a transfer.</td></tr>
</table>

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1390</td><td>TradePublishIndicator</td><td>N</td><td>INT</td><td>Indicates if a trade should be or has been published via a market publication service.<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Deferred Publication</td></tr><tr><td>3</td><td>Published</td></tr></table></td></tr>
<tr><td>1596</td><td>ClearingTradePrice</td><td>N</td><td>PRICE</td><td>Clearing price.</td></tr>
<tr><td>1649</td><td>RelatedSymbol</td><td>N</td><td>INT(10)</td><td>Product identifier of the originating Eurex strategy.</td></tr>
<tr><td>1650</td><td>RelatedSecurityID</td><td>N</td><td>INT(20)</td><td>Instrument identifier of the original Eurex strategy.</td></tr>
<tr><td>1724</td><td>OrderOrigination</td><td>N</td><td>INT</td><td>Direct market access identifier.<table><tr><th>Value</th><th>Description</th></tr><tr><td>5</td><td>Direct market access</td></tr></table></td></tr>
<tr><td>1815</td><td>TradingCapacity</td><td>N</td><td>INT</td><td>Not used.</td></tr>
<tr><td>28586</td><td>OrderSide</td><td>N</td><td>CHAR</td><td>Side of the order in the original MCX strategy.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
</table>

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>28587</td><td>RelatedProduct-Complex</td><td>N</td><td>INT</td><td>Instrument type in the original Eurex strategy.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>5</td><td>Futures Spread</td></tr></table></td></tr>
<tr><td>28610</td><td>MatchSubType</td><td>N</td><td>INT</td><td>Indicates the call auction type the trade originates from. Field is used only for on-book trades.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Opening auction</td></tr></table></td></tr>
<tr><td>28736</td><td>ClearingTradeQty</td><td>N</td><td>QTY</td><td>Quantity used for clearing.</td></tr>
<tr><td>28750</td><td>MultilegPriceModel</td><td>N</td><td>INT</td><td>Price decomposition method for legs of a complex instrument.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Standard (leg level prices provided by system)</td></tr><tr><td>1</td><td>User defined (leg level prices provided by user)</td></tr></table></td></tr>
<tr><td>28758</td><td>NegotiationID</td><td>N</td><td>INT(10)</td><td>ID of a MCXEnLight negotiation.</td></tr>
<tr><td>28899</td><td>BasketProfileID</td><td>N</td><td>INT(10)</td><td>Basket Profile ID.</td></tr>
<tr><td>30060</td><td>UTransactTime</td><td>Y</td><td>INT(20)</td><td>Transaction timestamp. Number of nanoseconds since Jan 1, 1970.</td></tr>
<tr><td colspan="5"><Trailer /></td></tr>
</table>

Confidential

6.2.8 Trading Session Status

This message is sent to provide information about the trading status at the exchange.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='h' />				
58	Text	N	STRING (128)	Free format text string.
336	TradingSessionID	Y	INT	Not used.
1368	TradSesEvent	Y	INT	Trading session event type. Value: Description 101: Start of Service 102: Market Reset 103: End of restatement 104: End of service
340	TradSesStatus	Y	INT	State of the trading session. Value: Description 0: Unknown 2: Open 3: Closed
1300	MarketSegmentID	N	INT(10)	Product identifier.
30060	UTransactTime	N	INT(20)	Transaction timestamp. Number of nanoseconds since Jan 1, 1970.
30075	UTradeDate	N	LOCALMKT-DATE	Date of trading session in YYYYMMDD format.
<Trailer />				

Confidential

6.2.9 User Notification

The User Notification message is used to send information of an unsolicited trader logoff or send information of legal notifications.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='UCB' />				
<UsernameGrp>, cardinality: unrestricted, record counter: NoUsernames				
553	Username	Y	INT	User ID
</UsernameGrp>				
58	Text	N	STRING (128)	Free format text string.
926	UserStatus	Y	INT	Indicates the status of a user. Value: 7 – Forced user logout by ex-change; 10 – User stopped; 11 – User released
<Trailer />				

Confidential

6.2.10 Session Details List

Session Details List provides a list of all trading sessions of the own business unit.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U6' />				
<SessionGrp>, cardinality: unrestricted, record counter: NoSessions				
28766	GatewaySessionID	Y	INT(10)	The gateway session ID.
28730	SessionMode	Y	INT	The session type. Value: 0 – ETI High Frequency session 1 – ETI Low Frequency session 2 – MCX GUI session 3 – FIX LF session
28735	SessionSubMode	Y	INT	The session subtype. Value: 0 – Regular Trading session 1 – FIX Trading session
</SessionGrp>				
30060	UTransactTime	Y	INT(20)	Transaction timestamp. Number of nanoseconds since Jan 1, 1970.
<Trailer />				

Confidential

6.2.11 Execution Report

The Execution Report message is used to confirm the receipt of an order, confirm changes to an existing order, transmit all active orders, relay fill information.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='8' />				
<Parties>, cardinality: 0-10, record counter: NoPartyIDs, containing occurrences: beneficiary , entering_firm , entering_trader , executing_trader , executing_unit , location_id , order_origination_firm , position_account , session_ID , takeup_firm				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Person
</Parties>				
<InstrmtLegExecGrp>, cardinality: unrestricted, record counter: NoLegs				
600	LegSymbol	N	STRING (10)	Multileg instrument's individual security's Symbol.
602	LegSecurityID	N	INT(20)	Multileg instrument's individual security's SecurityID.
2680	LegAccount	N	STRING	Account mnemonic as agreed between buy and sell sides.
564	LegPositionEffect	N	CHAR	Must be set to C, Valid characters : \x01-\x7E
637	LegLastPx	N	PRICE	Execution price assigned to a leg of a multileg instrument.
1418	LegLastQty	N	QTY	Fill quantity for the leg instrument.

Confidential

Tag	Field Name	Req'd	Data Type	Description
1893	LegExecID	N	INT	Private identifier of a leg match event, which can be reconciled with the field SideTradeID (1506) in the User/TradeCaptureReport (UAE/AE). Private identifier of a leg match event, which can be reconciled with the field SideTradeID (1506) in the UserTradeCaptur-eReport (UAE). Private identifier of a leg match event, which can be reconciled with the field SideTradeID (1506) in the TradeCaptureReport (AE).
</InstrmtLegExecGrp>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier.If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Description: Marketplace-assigned identi-fier
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. Value: 1 – Description: Simple Instrument Value: 5 – Description: Futures Spread

Confidential

Tag	Field Name	Req'd	Data Type	Description
167	SecurityType	N	STRING (4)	Indicates the type of security. Value: MLEG – Multileg Instrument; 0 – Cumulative at AssetType Level; 1 – Underlying Futures; 2 – Options; 3 – Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
<MtchgInst>, cardinality: 1, record counter: NoMatchInst				
1625	MatchInst	N	INT	Matching Instruction for the order. Value: 2 – Do not match
28744	MatchInstCrossID	N	INT(10)	Numeric identifier. Contains the Self Match Prevention ID.
</MtchgInst>				
1	Account	N	STRING (2)	Not used , Must be sent as A1.
11	ClOrdID	N	STRING (20)	Client Order ID: Unique participant defined order request identifier; used for client order id chaining.
14	CumQty	Y	QTY	Cumulated executed quantity of an order. Total quantity (e.g. number of shares) filled.
17	ExecID	Y	STRING (80)	The field provides a unique identifier and can be used for the identification of duplicate order messages.
18	ExecInst	N	MULTIPLE-VALUESTRING	Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. Value: 1 – Persistent Order(FIX value-H)
31	LastPx	N	PRICE	Price of this (last) fill.
32	LastQty	N	QTY	Quantity bought/sold on this Last fill.
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>39</td><td>OrdStatus</td><td>Y</td><td>CHAR</td><td>Identifies current status of order.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>New</td></tr><tr><td>1</td><td>PartiallyFilled</td></tr><tr><td>2</td><td>Filled</td></tr><tr><td>4</td><td>Cancelled</td></tr><tr><td>8</td><td>Rejected</td></tr><tr><td>9</td><td>Suspended</td></tr></table></td></tr>
<tr><td>40</td><td>OrdType</td><td>N</td><td>CHAR</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Limit</td></tr><tr><td>3</td><td>Stop</td></tr><tr><td>4</td><td>Stop limit</td></tr><tr><td>5</td><td>Market To Limit</td></tr><tr><td>6</td><td>Auction Buy IN</td></tr><tr><td>7</td><td>Auction_sell_Out</td></tr></table></td></tr>
<tr><td>41</td><td>OrigClOrdID</td><td>N</td><td>STRING (20)</td><td>Client Order ID: Unique participant defined order request identifier; used for client order id chaining.</td></tr>
<tr><td>44</td><td>Price</td><td>N</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>CHAR</td><td>Side of order.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>25007</td><td>FreeText1</td><td>N</td><td>STRING (12)</td><td>For own order-send OWN
For client order-send Client code.</td></tr>
<tr><td>59</td><td>TimeInForce</td><td>N</td><td>CHAR</td><td>Specifies how long the order remains in effect.
Absence of this field is interpreted as DAY.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Day (DAY)</td></tr><tr><td>1</td><td>Good Till Cancelled (GTC) - Standard Orders only</td></tr><tr><td>3</td><td>Immediate or Cancel (IOC)</td></tr><tr><td>6</td><td>Good Till Date (GTD) - Standard Orders only</td></tr><tr><td>7</td><td>Session (EOS)</td></tr></table></td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
77	PositionEffect	N	CHAR	Must be set to C, Valid characters : \x01-\x7E
99	StopPx	N	PRICE	Price per unit of quantity (e.g. per share).
150	ExecType	Y	CHAR	Describes the specific ExecutionRpt while OrdStatus (39) will always identify the current order status. Value – Description 0 – New 1 – Partial fill 2 – Fill 4 – Canceled 5 – Replace 6 – Pending cancel 8 – Rejected 9 – Suspended A – Pending new D – Restated E – Pending replace F – Trade L – Triggered by system
151	LeavesQty	Y	QTY	Quantity open for further execution. If the OrdStatus (39) is Canceled, Expired, Calculated, or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty (38) - CumQty (14).
336	TradingSessionID	N	INT	Not used
378	ExecRestatement-Reason	N	INT	Code to identify reason for an Execution Report message sent with ExecType=Restated or used when communicating an unsolicited cancel. Valid values are listed after this table.

Confidential

Tag	Field Name	Req'd	Data Type	Description
432	ExpireDate	N	LOCALMKT-DATE	Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices.
527	SecondaryExecID	N	INT(10)	Assigned by the party which accepts the order. Can be used to provide the ExecID used by an exchange or executing system.Refer Tag 10003 (Trade ID) from AE(Trade Capture Report) message for trade broadcast.
Trade	TradingSessionSubID	N	INT	Not used.
851	LastLiquidityInd	N	INT	Indicator to identify whether this fill was a result of a liquidity provider providing or liquidity taker taking the liquidity. Applicable only for OrdStatus of Partial or Filled. Value: Description 1: Add Liquidity 2: Removed Liquidity 4: Auction (neither passive nor aggressive, includes VDO matching at midpoint) 5: Triggered Stop Order 6: Triggered One-cancels-the-other Order 7: Triggered Market Order
880	TrdMatchID	N	INT(10)	Identifier assigned to a trade by a matching system.
1815	TradingCapacity	N	INT	Not used. ,
2523	CrossedIndicator	N	INT	Indicator will be delivered in case of deletion or modification due to Self Match Prevention. Value: Description 1: Cross rejected

Confidential

Tag	Field Name	Req'd	Data Type	Description
1823	Triggered	N	INT	Indicates if an order has been previously triggered. Value: 0 – Not triggered; 1 – Triggered; 2 – Triggered OCO
2404	ComplianceText	N	STRING (20)	Identification of the actual trader behind a transaction, required by Market Surveillance.
25008	FreeText2	N	STRING (12)	CPCode needs to be send.
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the order.
30060	UTransactTime	N	INT(20)	Transaction timestamp. Number of nanoseconds since Jan 1, 1970.
1031	CustOrderHandling-Inst	N	CHAR	Not used.
30737	mLReferenceNumber	N	INT	Integral Number
30738	SpreadPrice	N	PRICE	Decimal Number in floating point representation.
770	MultiLegType	Y	INT	Integral Number Value: 0 – Spread Order; 1 – Multi Leg Order
<Trailer />				

Confidential

Valid Values of ExecRestatementReason (datatype INT)

Valid Value	Description
1	GT renewal / restatement (no corporate action, order book restatement)
100	Unknown Order State
101	Order Added
102	Order Replaced
103	Order Canceled
105	Immediate or Cancel Order Canceled
108	Book Order executed
114	Order has been changed to IOC
122	Instrument State Change
135	Market Order Triggered
149	Closing Auction Only Order Activated
150	Closing Auction Only Order Inactivated
164	One-cancels-the-other Order Triggered
172	Stop Order Triggered
181	Ownership Changed
197	Order Cancellation Pending
199	Pending Cancellation Executed
212	Book Or Cancel Order Canceled
248	Active Order Deletion due to SMPF.
250	Active Order Modification due to SMPF.
252	Passive Order Deletion due to SMPF.
254	Passive Order Modification due to SMPF.
258	Book Order Restatement
261	Panic Cancel
270	ExDex Request Added
271	ExDex Request Modified
272	ExDex Request Cancelled
340	CLIP execution after improvement period
343	CLIP request canceled by arrangement time out
344	CLIP cancellation
355	Risk reduction timer expired

Confidential

6.3 Multileg Order Handling

6.3.1 New Order Complex

The New Order Complex message is provided to submit orders for securities that are made up of multiple securities, known as ”legs”.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='AB' />				
<Parties>, cardinality: 1-11, record counter: NoPartyIDs, containing occurrences: beneficiary , client_id , entering_trader , execution_identifier_human , execution_identifier_algo , investment_identifier_human , investment_identifier_algo , location_id , order_origination_firm , position_account , takeup_firm				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Person
</Parties>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier. If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Marketplace-assigned identifier

Confidential

Tag	Field Name	Req'd	Data Type	Description
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. Value: 1 – Simple Instrument; 5 – Futures Spread
167	SecurityType	N	STRING (4)	Value: MLEG – Multileg Instrument; 0 – Cumulative at AssetType Level; 1 – Underlying Futures; 2 – Options; 3 – Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
<LegOrdGrp>, cardinality: unrestricted, record counter: NoLegs				
2680	LegAccount	Y	STRING	Account mnemonic as agreed between buy and sell sides.
564	LegPositionEffect	Y	CHAR	Must be set to C, Valid characters : \x01-\x7E
</LegOrdGrp>				
<MtchgInst>, cardinality: 1, record counter: NoMatchInst				
1625	MatchInst	N	INT	Matching Instruction for the order. Value: 2 – Do not match
28744	MatchInstCrossID	N	INT(10)	Numeric identifier. Contains the Self Match Prevention ID.
</MtchgInst>				

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td colspan="5"><OrderAttributeGrp>, cardinality: 0-2, record counter: NoOrderAttributes, containing occurrences: liquidity_provision_activity_order , risk_reduction_order</td></tr>
<tr><td>2594</td><td>>OrderAttributeType</td><td>N</td><td>INT</td><td>Type of order attribute.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Liquidity Provision Activity Order</td></tr><tr><td>3</td><td>Risk Reduction Order</td></tr></table></td></tr>
<tr><td>2595</td><td>>OrderAttributeValue</td><td>N</td><td>STRING</td><td>Value associated with the order attribute type specified in OrderAttributeType (2594).
Y- Attribute is set.</td></tr>
<tr><td colspan="5"></OrderAttributeGrp></td></tr>
<tr><td colspan="5"><ValueChecksGrp>, cardinality: 2, record counter: NoValueChecks, containing occurrences: price_check, notional_value_check (NOT USED)</td></tr>
<tr><td>1869</td><td>ValueCheckType</td><td>Y</td><td>INT</td><td>Type of value to be checked.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Price Check</td></tr></table></td></tr>
<tr><td>1870</td><td>ValueCheckAction</td><td>Y</td><td>INT</td><td>Action to be taken for the ValueCheckType (1869). ValueCheckAction (1870) = '2' (Best effort) is only possible, if ValueCheckType (1869) = '1' (Price check).
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Check</td></tr><tr><td>2</td><td>Best effort</td></tr></table></td></tr>
<tr><td colspan="5"></ValueChecksGrp></td></tr>
<tr><td>210</td><td>MaxShow</td><td>N</td><td>QTY</td><td>The quantity to be made visible in the market Data</td></tr>
<tr><td>25030</td><td>SMPFOrderIdentifier</td><td>Y</td><td>CHAR</td><td>Free format string, can include characters, numbers, punctuation and many special characters
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Passive</td></tr><tr><td>1</td><td>Active</td></tr></table></td></tr>
<tr><td>11</td><td>ClOrdID</td><td>Y</td><td>STRING (20)</td><td>Client Order ID: Unique participant defined order request identifier; used for client order id chaining.</td></tr>
</table>

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>18</td><td>ExecInst</td><td>N</td><td>MULTIPLE-VALUESTRING</td><td>Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Persistent Order(FIX value-H)</td></tr></table></td></tr>
<tr><td>38</td><td>OrderQty</td><td>Y</td><td>QTY</td><td>Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.</td></tr>
<tr><td>40</td><td>OrdType</td><td>Y</td><td>CHAR</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Limit</td></tr><tr><td>3</td><td>Stop</td></tr><tr><td>4</td><td>Stop limit</td></tr><tr><td>5</td><td>Market To Limit</td></tr><tr><td>6</td><td>Auction Buy IN</td></tr><tr><td>7</td><td>Auction_sell_Out</td></tr></table></td></tr>
<tr><td>44</td><td>Price</td><td>Y</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>CHAR</td><td>Side
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>25007</td><td>FreeText1</td><td>N</td><td>STRING (12)</td><td>For own order-send OWN
For client order-send Client code.</td></tr>
<tr><td>59</td><td>TimeInForce</td><td>N</td><td>CHAR</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Day (DAY)</td></tr><tr><td>1</td><td>Good Till Cancelled (GTC) - Standard Orders only</td></tr><tr><td>3</td><td>Immediate or Cancel (IOC)</td></tr><tr><td>6</td><td>Good Till Date (GTD) - Standard Orders only</td></tr><tr><td>7</td><td>Session (EOS)</td></tr></table></td></tr>
<tr><td>432</td><td>ExpireDate</td><td>N</td><td>LOCALMKT-DATE</td><td>Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices.</td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
1031	CustOrderHandling-Inst	N	CHAR	Not used.
1724	OrderOrigination	N	INT	Direct market access identifier. Value: 5 – Description: Direct market access
1815	TradingCapacity	N	INT	Not used. ,
2404	ComplianceText	N	STRING (20)	Identification of the actual trader behind a transaction, required by Market Surveillance.
25008	FreeText2	N	STRING (12)	CPCode needs to be send.
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the order.
1300	MarketSegmentID	N	INT(10)	Product identifier.
708	SenderLocationID	N	STRING(20)	SenderLocationID
1090	MaxPriceProtection	N	INT	Price per unit of quantity.
<Trailer />				

Confidential

6.3.2 Modify Order Complex Request

The Modify Order Complex Request is used to modify a complex order (previously submitted using the New Order Complex messsage).

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='AC' />				
<Parties>, cardinality: 1-11, record counter: NoPartyIDs, containing occurrences: beneficiary , client_id , entering_trader , execution_identifier_human , execution_identifier_algo , investment_identifier_human , investment_identifier_algo , location_id , order_origination_firm , position_account , takeup_firm				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Value: 2 – Person Type of PartySubID.
</Parties>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier.If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Marketplace-assigned identifier

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>1227</td><td>ProductComplex</td><td>N</td><td>INT</td><td>Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Simple Instrument</td></tr><tr><td>5</td><td>Futures Spread</td></tr></table></td></tr>
<tr><td>167</td><td>SecurityType</td><td>N</td><td>STRING (4)</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>MLEG</td><td>Multileg Instrument</td></tr><tr><td>0</td><td>Cumulative at AssetType Level</td></tr><tr><td>1</td><td>Underlying Futures</td></tr><tr><td>2</td><td>Options</td></tr><tr><td>3</td><td>Futures</td></tr></table></td></tr>
<tr><td>762</td><td>SecuritySubType</td><td>N</td><td>INT(10)</td><td>Strategy type.</td></tr>
<tr><td colspan="5"></Instrument></td></tr>
<tr><td colspan="5"><LegOrdGrp>, cardinality: unrestricted, record counter: NoLegs</td></tr>
<tr><td>2680</td><td>LegAccount</td><td>Y</td><td>STRING</td><td>Account mnemonic as agreed between buy and sell sides.</td></tr>
<tr><td>564</td><td>LegPositionEffect</td><td>Y</td><td>CHAR</td><td>Must be set to C, Valid characters : \x01-\x7E</td></tr>
<tr><td colspan="5"></LegOrdGrp></td></tr>
<tr><td colspan="5"><MtchgInst>, cardinality: 1, record counter: NoMatchInst</td></tr>
<tr><td>1625</td><td>MatchInst</td><td>N</td><td>INT</td><td>Matching Instruction for the order.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Do not match</td></tr></table></td></tr>
<tr><td>28744</td><td>MatchInstCrossID</td><td>N</td><td>INT(10)</td><td>Numeric identifier. Contains the Self Match Prevention ID.</td></tr>
<tr><td colspan="5"></MtchgInst></td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
<OrderAttributeGrp>, cardinality: 0-1, record counter: NoOrderAttributes, containing occurrences: liquidity_provision_activity_order				
2594	>OrderAttributeType	N	INT	Type of order attribute. Value: 2 – Liquidity Provision Activity Order Value: 3 – Risk Reduction Order
2595	>OrderAttributeValue	N	STRING (4)	Value associated with the order attribute type specified in OrderAttributeType (2594). Value: Y – Attribute is set.
</OrderAttributeGrp>				
<ValueChecksGrp>, cardinality: 1-3, record counter: NoValueChecks (NOT USED)				
1869	>ValueCheckType	Y	INT	Value: 1 – Price check
1870	ValueCheckAction	Y	INT	Action to be taken for the ValueCheckType (1869). ValueCheckAction (1870) = '2' (Best effort) is only possible, if ValueCheckType (1869) = '1' (Price check). Value: 1 – Check Value: 2 – Best effort
</ValueChecksGrp>				
210	MaxShow	N	QTY	The quantity to be made visible in the market Data
11	ClOrdID	Y	STRING (20)	Client Order ID: Unique participant defined order request identifier; used for client order id chaining.
18	ExecInst	N	MULTIPLE-VALUESTRING	Instructions for order handling on exchange trading floor. If more than one instruction is applicable to an order, this field can contain multiple instructions separated by space. Value: 1 – Persistent Order(FIX value-H)

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>38</td><td>OrderQty</td><td>Y</td><td>QTY</td><td>Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.</td></tr>
<tr><td>40</td><td>OrdType</td><td>Y</td><td>CHAR</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>2</td><td>Limit</td></tr><tr><td>3</td><td>Stop</td></tr><tr><td>4</td><td>Stop limit</td></tr><tr><td>5</td><td>Market To Limit</td></tr><tr><td>6</td><td>Auction Buy IN</td></tr><tr><td>7</td><td>Auction_sell_Out</td></tr></table></td></tr>
<tr><td>41</td><td>OrigClOrdID</td><td>Y</td><td>STRING (20)</td><td>Client Order ID: Unique participant defined order request identifier; used for client order id chaining.</td></tr>
<tr><td>44</td><td>Price</td><td>Y</td><td>PRICE</td><td>Price per unit of quantity (e.g. per share).</td></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>CHAR</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>25007</td><td>FreeText1</td><td>N</td><td>STRING (12)</td><td>For own order-send OWN
For client order-send Client code.</td></tr>
<tr><td>59</td><td>TimeInForce</td><td>N</td><td>CHAR</td><td>Specifies how long the order remains in effect. Absence of this field is interpreted as DAY.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Day (DAY)</td></tr><tr><td>1</td><td>Good Till Cancelled (GTC) - Standard Orders only</td></tr><tr><td>3</td><td>Immediate or Cancel (IOC)</td></tr><tr><td>6</td><td>Good Till Date (GTD) - Standard Orders only</td></tr><tr><td>7</td><td>Session (EOS)</td></tr></table></td></tr>
<tr><td>432</td><td>ExpireDate</td><td>N</td><td>LOCALMKT-DATE</td><td>Date of order expiration (last day the order can trade), always expressed in terms of the local market date. The time at which the order expires is determined by the local market's business practices.</td></tr>
<tr><td>1031</td><td>CustOrderHandling-Inst</td><td>N</td><td>CHAR</td><td>Not used.</td></tr>
</table>

Confidential

Tag	Field Name	Req'd	Data Type	Description
1724	OrderOrigination	N	INT	Direct market access identifier. Value: 5 – Direct market access
1815	TradingCapacity	N	INT	Not used. ,
2404	ComplianceText	N	STRING (20)	Identification of the actual trader behind a transaction, required by Market Surveillance.
25008	FreeText2	N	STRING (12)	CPCode needs to be send.
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the order.
1300	MarketSegmentID	N	INT(10)	Product identifier.
708	SenderLocationID	N	STRING(20)	Sender Location ID
1090	MaxPriceProtection	N	INT	Price per unit of quantity.
<Trailer />				

Confidential

6.3.3 New Order Multileg

The New Order Multileg message is used by the participant to submit an order for multiple securities.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='AD' />				
<MultiLegOrdGrp>				
743	InstrumentIdentifier	Y	INT	The U/L Asset / producer Identifieris specified by Exchange for permitted U/L Asset / products for trading
44	Price	N	PRICE	Price per unit of quantity (e.g. per share).
731	MaxPricePercentage	N	PRICE	Decimal number in floating point representation
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
1300	MarketSegmentID	N	INT(10)	Product identifier.
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. Value: 1 – Simple Instrument; 5 – Futures Spread
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
40	OrdType	Y	CHAR	Value: 2 – Limit; 5 – Market

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td colspan="5"></MultiLegOrdGrp></td></tr>
<tr><td>708</td><td>TerminalInfo</td><td>Y</td><td>INT</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>11111111111</td><td>IBT</td></tr>
<tr><td>22222222222</td><td>DMA</td></tr>
<tr><td>33333333333</td><td>Wireless Technology</td></tr>
<tr><td>0</td><td>Order not generated through Program trading software</td></tr>
<tr><td>1</td><td>Order generated through Program trading software</td></tr>
<tr><td>2</td><td>Smart Order Rout-ing without Program trading software</td></tr>
<tr><td>3</td><td>Smart Order Routing with program trading software</td></tr>
</table>
Total 15 Characters
For 1st- 12 characters value(11111111111-33333333333),
For 13th character value(0-3) 14th - 15th digit should be valid
vendor code / in House CTCL code for member.
</td></tr>
<tr><td>11</td><td>ClOrdID</td><td>N</td><td>STRING (20)</td><td>Client Order ID: Unique participant defined order request identifier; used for client order id chaining.</td></tr>
<tr><td>25026</td><td>StrategyID</td><td>Y</td><td>INT</td><td>Strategy Approved by the exchange should be used.Range being from 0 to 99999.</td></tr>
<tr><td>25029</td><td>StrategySeqNum</td><td>Y</td><td>SEQNUM</td><td>Strategy Sequence numbers</td></tr>
<tr><td>581</td><td>AccountType</td><td>Y</td><td>STRING</td><td>Free format string, can include characters, num-bers, punctuation and many special characters
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Own</td></tr>
<tr><td>3</td><td>Client</td></tr>
<tr><td>5</td><td>Institution</td></tr>
</table>
</td></tr>
<tr><td>769</td><td>NoOfMultiLeg</td><td>Y</td><td>INT</td><td>Number of MultiLegOrdGrp repeating group instances.</td></tr>
</table>

Confidential

25030	SMPFOrderIdentifier	Y	CHAR	Free format string, can include characters, numbers, punctuation and many special characters Value: 0 – Passive; 1 – Active
770	MultiLegType	Y	INT	Integral number. Value: 0 – Spread Order; 1 – Multi Leg Order
768	AllOrNoneFlag	N	CHAR	Free format string, can include characters, numbers, punctuation and many special characters
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
25009	FreeText3	N	STRING (12)	Assigned by the party which originates the order.
<Trailer />				

Confidential

6.4 Ex/DEx

6.4.1 Ex/DEx Entry Request

This message informs about ExDex entry request.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U74' />				
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Confidential

6.4.2 Ex/DEx Entry Confirmation

This message informs about Exdex confirmation.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U75' />				
814	ExDExOrderNo	Y	INT	Unique Number assigned by Exchange
60	TransactTime	N	UTCTIME-STAMP	Transaction timestamp.
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Confidential

6.4.3 Ex/DEx Cancellation Request

This message informs about ExDEx cancellation events.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U76' />				
814	ExDExOrderNo	Y	INT	Unique Number assigned by Exchange
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Confidential

6.4.4 Ex/DEx Cancellation Confirmation

This message informs about ExDEx cancellation events.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U81' />				
814	ExDExOrderNo	Y	INT	Unique Number assigned by Exchange
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Confidential

6.4.5 Ex/DEx Modification Request

This message informs about ExDEx modification events.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U77' />				
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
814	ExDExOrderNo	Y	INT	Unique Number assigned by Exchange
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Confidential

6.4.6 Ex/DEx Modification Confirmation

This message confirms the successful processing of the referenced message.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U80' />				
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
814	ExDExOrderNo	Y	INT	Unique Number assigned by Exchange
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Confidential

6.4.7 ExDEx notification.

This message informs about ExDEx notification.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U78' />				
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
814	ExDExOrderNo	Y	INT	Unique Number assigned by Exchange
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
48	SecurityID	Y	INT(20)	MCX instrument identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
20655	TargetPartyIDSession-ID	N	INT	Session ID.
378	ExecRestatement-Reason	Y	INT	Code to identify reason for an Execution Report message sent with ExecType=Restated or used when communicating an unsolicited cancel. Valid values are listed after this table.
25087	ExDexStatus	Y	INT	Integral number. Value: 1 – Accepted; 2 – Cancelled
25086	ExDexFlag	Y	INT	Integral number. Value: 1 – Exercise Instruction; 2 – Don't Exercise Instruction
581	AccountType	Y	STRING	Free format string, can include characters, num-bers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25007	FreeText1	Y	STRING (12)	For own order-send OWN For client order-send Client code.

Confidential

Tag	Field Name	Req'd	Data Type	Description
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
718	UserReferenceText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

Valid Values of ExecRestatementReason (datatype INT)

Valid Value	Description
1	GT renewal / restatement (no corporate action, order book restatement)
100	Unknown Order State
101	Order Added
102	Order Replaced
103	Order Canceled
105	Immediate or Cancel Order Canceled
108	Book Order executed
114	Order has been changed to IOC
122	Instrument State Change
135	Market Order Triggered
149	Closing Auction Only Order Activated
150	Closing Auction Only Order Inactivated
164	One-cancels-the-other Order Triggered
172	Stop Order Triggered
181	Ownership Changed
197	Order Cancellation Pending
199	Pending Cancellation Executed
212	Book Or Cancel Order Canceled
248	Active Order Deletion due to SMPF.
250	Active Order Modification due to SMPF.
252	Passive Order Deletion due to SMPF.
254	Passive Order Modification due to SMPF.
258	Book Order Restatement
261	Panic Cancel
270	ExDex Request Added
271	ExDex Request Modified
272	ExDex Request Cancelled
340	CLIP execution after improvement period
343	CLIP request canceled by arrangement time out
344	CLIP cancellation
355	Risk reduction timer expired

Confidential

6.5 Trade Related Handling

6.5.1 Trade Modification Request.

The trade modify instruction request message provides the ability to request the change in account information for an already executed trade.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U3' />				
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
30048	SimpleSecurityID	Y	INT	Instrument Identifier.
20412	RootParty-IDExecutingTrader	Y	INT	Owning User ID.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
1003	TradeID	Y	INT(10)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.
1300	MarketSegmentID	Y	INT(10)	Product identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
751	OldAccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
756	OldUCCCode	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters
753	OldCPCode	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters
754	NewAccountType	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
755	NewUCCCode	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters
759	NewCPCode	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters
58	Text	N	STRING (128)	Free format text string.
<Trailer />				

Confidential

6.5.2 Trade Modification Response

The trade modification respone message provides the response of trade modification request.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U4' />				
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
30048	SimpleSecurityID	Y	INT	Instrument Identifier.
711	Echo	Y	INT	Vendors can use this as a reference in response from exchange.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
1003	TradeID	Y	INT(10)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.
20412	RootParty-IDExecutingTrader	Y	INT	Owning User ID.
751	OldAccountType	Y	STRING	Free format string, can include characters, num-bers, punctuation and many special characters Value: 1 – Description: Own Value: 3 – Description: Client Value: 5 – Description: Institution
756	OldUCCCode	Y	STRING	Previous Account information
753	OldCPCode	Y	STRING	Previous Participant Code
754	NewAccountType	Y	STRING	Free format string, can include characters, num-bers, punctuation and many special characters Value: 1 – Description: Own Value: 3 – Description: Client Value: 5 – Description: Institution
755	NewUCCCode	Y	STRING	New UCC Code
759	NewCPCode	Y	STRING	New Participant code
58	Text	N	STRING (128)	Latest User reference Text for the order.
<Trailer />				

Confidential

6.5.3 Trade Modification Notification

This message informs about the acceptance/rejection of a trade by the custodian.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U94' />				
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
30048	SimpleSecurityID	Y	INT	Instrument Identifier.
20412	RootParty-IDExecutingTrader	Y	INT	Owning User ID.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
1003	TradeID	N	INT(10)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.
766	RootPartyIDSubmitter	Y	INT	Root Party ID Submitter
711	Echo	N	INT	Vendors can use this as a reference in response from exchange.
761	OldCustomerorFirm	Y	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: Description 1: Own 3: Client 5: Institution
995	OldAccount	N	STRING	Previous Account information
764	OldClearingAccount	N	STRING	Previous Participant Code
767	NewCustomerorFirm	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: Description 1: Own 3: Client 5: Institution
763	NewAccount	N	STRING	New UCC Code
765	NewClearingAccount	N	STRING	New Participant code.
58	Text	N	STRING (128)	Latest User reference Text for the order.
<Trailer />				

Confidential

6.5.4 Trade Enhancement Notification

It provides a list of all trading sessions of the own business unit.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U31' />				
1596	ClearingTradePrice	N	PRICE	Clearing price.
60	TransactTime	Y	UTCTIME-STAMP	Transaction timestamp.
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
28736	ClearingTradeQty	N	QTY	Quantity used for clearing.
30048	SimpleSecurityID	Y	INT	Instrument identifier.
1003	TradeID	Y	INT(10)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.
20455	RootPartyIDSessionID	Y	INT	Session ID.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
1300	MarketSegmentID	N	INT(10)	Product identifier.
28582	MatchDate	N	LOCALMKT-DATE	Integral number.
581	AccountType	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	N	CHAR	Side Value: 1 – Buy; 2 – Sell
773	GiveUpStatus	N	INT	Integral number. Value: 1 – Accepted; 2 – Rejected; 3 – Pending
717	CPCode	N	STRING	The Participant code. Valid characters:\x00, \x21, \x23 \x24, \x28, \x29, \x2A, \x2B, \x2D, \x2E, \x2F, \x30-\x39, \x3A, \x3B, \x3D, \x3E, \x3F, \x40, \x41-\x5A \x5C, \x5D, \x5F, \x60, \x7B, \x7D
25007	FreeText1	N	STRING (12)	For own order-send OWN For client order-send Client code.
<Trailer />				

Confidential

6.5.5 Resubmit Approval Request

Resubmit Approval Request message is sent to the service "Order and Quote Management".

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U86' />				
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
30048	SimpleSecurityID	Y	INT	Instrument identifier.
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
<Trailer />				

Confidential

6.5.6 Resubmit Approval Confirmation

Resubmit Approval Confirmation message is sent to the service "Order and Quote Management".

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U87' />				
779	LastUpdateTime	N	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
30048	SimpleSecurityID	Y	INT	Instrument Identifier.
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
773	GiveUpStatus	Y	INT	Integral number. Value: 1 – Accepted; 2 – Rejected; 3 – Pending
<Trailer />				

Confidential

6.6 Others

6.6.1 Market Wide OI Violation

The Market Wide OI Violation message is a general message used by the FIX Gateway.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U88' />				
42	OrigTime	Y	UTCTIME-STAMP	Creation timestamp of the news message.
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
148	Headline	Y	STRING	The headline of a news message.
58	Text	Y	STRING (128)	Free format text string.
1940	AssetType	Y	INT	Integral number. Value: 0 – Description: Asset Class Value: 1 – Description: Asset Value: 2 – Description: Underlying Value: 3 – Description: Product
48	SecurityID	N	INT(20)	MCX instrument identifier.
25044	ViolationStatus	Y	INT	Integral number. Value: 0 – Description: Active indicates violation status is Active Value: 1 – Description: This means received message is for change of status from active status to violated.
25045	OIQuantityLimit	Y	INT	This defines the fixed quantity limit of % market wide open interest for particular Asset Type and Instrument identifier.
25046	OIPercentageLimit	Y	INT	This defines OI% Limit in terms of % Market wide open interest for particular asset type and instrument identifier,
25047	EffectiveOILimit	Y	INT	This defines actual OI Quantity for Particular Asset Type and Instrument Identifier for which Violation has been generated
25048	ActualOIQuantity	Y	INT	Number of used bytes for field VarText (30355).

Confidential

<table>
<tr><td>167</td><td>SecurityType</td><td>N</td><td>STRING (4)</td><td>Indicates the type of security.<table><tr><td>Value</td><td>Description</td></tr><tr><td>MLEG</td><td>Multileg Instrument</td></tr><tr><td>0</td><td>Cumulative at AssetType Level</td></tr><tr><td>1</td><td>Underlying Futures</td></tr><tr><td>2</td><td>Options</td></tr><tr><td>3</td><td>Futures</td></tr></table></td></tr>
<tr><td>541</td><td>MaturityDate</td><td>N</td><td>LOCALMKT-DATE</td><td>Maturity date of instrument in terms of seconds from 01-01-1970 00:00:00 hours.When asset type is 2 -underlying and security type is option(3),this will suggest violation at option expiry level.for other cases it can be 0.</td></tr>
<tr><td colspan="5"><Trailer /></td></tr>
</table>

Confidential

6.6.2 Market Wide OI Alerts

The Market Wide OI Alerts message is a general message used by the FIX Gateway.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U89' />				
42	OrigTime	Y	UTCTIME-STAMP	Creation timestamp of the news message.
148	Headline	Y	STRING	The headline of a news message.
58	Text	Y	STRING (128)	Free format text string.
1940	AssetType	Y	INT	Integral number. Value: Description 0: Asset Class 1: Asset 2: Underlying 3: Product
48	SecurityID	Y	INT(20)	MCX instrument identifier.
25050	AlertLevel	Y	INT	Integral number. Value: Description 1: 1st Lowest of 3 alerts set for OI 2: 2nd of 3 alert set for OI 3: 3rd High of 3 alerts set for OI 4: Alert for100% breach of OI
25051	AlertPercentage	Y	PERCENTAGE	Indicates % of each alert level
25052	UtilizedAlert-Percentage	Y	PERCENTAGE	Utilized % for particular Asset Type and Instrument Identifier
167	SecurityType	N	STRING (4)	Indicates the type of security. Value: Description MLEG: Multileg Instrument 0: Cumulative at AssetType Level 1: Underlying Futures 2: Options 3: Futures
807	LastModifyDate	Y	UTCTIME-STAMP	This contains the latest time for which message is received
541	MaturityDate	N	LOCALMKT-DATE	Maturity date of instrument in terms of seconds from 01-01-1970 00:00:00 hours.When asset type is 2 -underlying and security type is option(3),this will suggest violation at option expiry level.for other cases it can be 0.
<Trailer />				

Confidential

6.6.3 Margin Change Notification

The Margin Change Notification message.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U91' />				
775	NumOfRecords	Y	INT	Number of records.
743	InstrumentIdentifier	Y	INT	The U/L Asset / producer Identifieris specified by Exchange for permitted U/L Asset / products for trading
777	OldInitialBuyMargin-Rate	N	PERCENTAGE	Initial Margin on Buy position with implicit 2 decimal
778	OldInitialSellMargin-Rate	N	PERCENTAGE	Initiall Margin on Sell position withimplicit 2 decimal
781	OldSpecialBuyMargin-Rate	N	PERCENTAGE	Special Margin on Buy position withimplicit 2 decimal
782	OldSpecialSellMargin-Rate	N	PERCENTAGE	Special Margin on Sell position withimplicit 2 decimal
784	OldExtremeLossBuy-MarginRate	N	PERCENTAGE	Extreme Loss Margin on Buy positionwith implicit 2 decimal
785	OldExtremeLossSell-MarginRate	N	PERCENTAGE	Extreme Loss Margin on Sell positionwith implicit 2 decimal
787	NewInitialBuyMargin-Rate	N	PERCENTAGE	Initial Margin on Buy position withimplicit 2 decimal
788	NewInitialSellMargin-Rate	N	PERCENTAGE	Initiall Margin on Sell position withimplicit 2 decimal
791	NewSpecialBuy-MarginRate	N	PERCENTAGE	Special Margin on Buy position withimplicit2 decimal
792	NewSpecialSell-MarginRate	N	PERCENTAGE	Special Margin on Sell position withimplicit 2 decimal
794	NewExtremeLossBuy-MarginRate	N	PERCENTAGE	Extreme Loss Margin on Buy positionwith implicit 2 decimal
795	NewExtremeLossSell-MarginRate	N	PERCENTAGE	Extreme Loss Margin on Sell positionwith implicit 2 decimal
776	OldInitMarginFlatPerc	N	INT	Integral number. Value: 1 – Intial Margin in Percentage; 2 – Initial Margin in Absolute
901	OldInitMarginSpread-BenefitFlag	N	INT	Integral number. Value: 0 – Off; 1 – On

Confidential

Tag	Field Name	Req'd	Data Type	Description
780	OldSpecialMarginFlat-Perc	N	INT	Integral number. Value: 1 – Intial Margin in Percentage; 2 – Initial Margin in Absolute
783	OldConfigurationfor-ExtremeLossMargin	N	INT	1 – Extreme loss margin in percentagewith 2 decimalsc
786	NewInitMarginFlat-Perc	N	INT	Integral number. Value: 0 – Off; 1 – On
810	NewInitMarginSpread-BenefitFlag	N	INT	Integral number. Value: 0 – Off; 1 – On
790	NewSpecialMargin-FlatPerc	N	INT	Integral number. Value: 1 – Intial Margin in Percentage; 2 – Initial Margin in Absolute
793	NewConfigurationfor-ExtremeLossMargin	N	INT	1 – Extreme loss margin in percentagewith 2 decimals
<Trailer />				

Confidential

6.6.4 Market Status Notification

The Market Status Notification message

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U90' />				
1300	MarketSegmentID	N	INT(10)	Product identifier.
817	GroupID	Y	INT	0 – Indicates applicable for all groups Nonzero indicates for specific product Group.
818	Status	Y	INT	Integral number. Value: 1 – Open; 2 – Closed; 3 – Halt
336	TradingSessionID	Y	INT	Not used.
625	TradingSessionSubID	Y	INT	Not used.
<Trailer />				

Confidential

6.6.5 Auction Notification

This message confirms the succesful processing of the referenced message.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U44' />				
779	LastUpdateTime	Y	UTCTIME-STAMP	Time at which request is confirmed by the exchange in terms of seconds from 01-01-1970 00:00:00 hours
38	OrderQty	Y	QTY	Quantity ordered. This represents the number of shares for equities or par, face or nominal value for FI instruments.
44	Price	N	PRICE	Price per unit of quantity (e.g. per share).
14	CumQty	Y	QTY	Cumulated executed quantity of an order. Total quantity (e.g. number of shares) filled.
1300	MarketSegmentID	Y	INT(10)	Product identifier.
30048	SimpleSecurityID	Y	INT	Instrument Identifier.
336	TradingSessionID	Y	INT	Not used.
25056	AuctionEnquiry-TradingStatus	Y	INT	Integral number. Value: 0 – Not Defined; 1 – Open; 2 – Closed
<Trailer />				

Confidential

6.6.6 News

The News message provides public information from the MCX market supervision.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='B' />				
42	OrigTime	Y	UTCTIME-STAMP	Creation timestamp of the news message.
30354	VarTextLen	Y	INT	Number of used bytes for field VarText (30355).
148	Headline	Y	STRING	The headline of a news message.
30355	VarText	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters
<Trailer />				

6.6.7 System Information

System Information provides a list of all trading sessions of the own business unit.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U79' />				
336	TradingSessionID	Y	INT	Not used.
516	OrderPercent	Y	PERCENTAGE	Order Percent.
25031	TradeModTimeInd	Y	UTCTIME-STAMP	0-In case trade modification acceptance time has not elapsed. 1- In case trade modification acceptance time has elapsed
25032	DeliveryInstruction	Y	UTCTIME-STAMP	0-In case acceptance time has not elapsed. 1- In case acceptance time has elapsed
818	Status	Y	INT	Integral number. Value: 1 – Open; 2 – Closed; 3 – Halt
<Trailer />				

Confidential

6.6.8 Business Message Reject

The Business Message Reject message is sent to reject an application-level message which fulfills session-level rules and cannot be rejected via any other means.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='j' />				
45	RefSeqNum	Y	SEQNUM	Reference message Sequence number.
58	Text	N	STRING (128)	Free format text string.
372	RefMsgType	Y	STRING	The MsgType (35) of the FIX message being referenced.
380	BusinessRejectReason	Y	INT	Code to identify reason for a Business Message Reject message. Valid values are listed after this table.
379	BusinessRejectRefID	N	STRING (20)	The value of the business-level ID field on the message being referenced.
<Trailer />				

Valid Values of BusinessRejectReason (datatype INT)

Valid Value	Description
0	Other
1	Unknown ID
3	Unsupported message type
4	Application not available
5	Conditionally required field missing
6	Not authorized
8	Throttle limit exceeded
102	Service temporarily not available
103	Service not available
105	Error converting response or broadcast
200	Internal technical error
210	Validation Error
211	User already logged in
217	Session login limit (per time interval) reached
223	User Entitlement Data time out
225	User login tries limit (per time interval) reached
226	Limit of outstanding session/user logins reached (per Business Unit)
227	Limit of outstanding session/user logins reached (per Session)
10000	Order not found
10001	Price not reasonable
10002	Duplicate Order (ClOrdID)
10003	another quote activation request is pending for the same Product / Instrument Type / SessionID
10006	Stop buy price not reasonable

Confidential

10007	Stop sell price not reasonable
10008	GFD order is not executable on current business day
10010	Throttle limit for creation of complex instrument exceeded
10011	Order maintenance not allowed in current state

Confidential

6.6.9 User Party Risk Limits Request

The User Party Risk Limits Request is used to inquire pre-trade risk limits. These limits can be defined for on-book trading on product and user risk group level.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='UCL' />				
<Parties>, cardinality: 1-2, record counter: NoPartyIDs, containing occurrences: entering_trader , executing_unit				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 = Algorithm; 24 = Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 = Person
</Parties> *<RiskLimitTypesGrp>, cardinality: 1-64, record counter: NoRiskLimitTypes*				
1530	>RiskLimitType	Y	INT	Specifies the type of risk limit. Value: 4 = Long limit; 5 = Short limit
1533	>RiskLimitPlatform	Y	INT	The area to which risk limit is applicable. Scope of Pre-Trade risk limits. Value: 0 = On-book; 1 = Off-book
28776	RiskLimit-RequestingPartyRole	N	INT	Requesting party role for a pre-trade risk limit. Value: 4 = Clearing firm; 22 = Exchange; 59 = Executing unit

Confidential

Tag	Field Name	Req'd	Data Type	Description
28778	>RiskLimitViolation-Indicator	N	INT	Indicator for a pre-trade risk limit violation. Value: 0 = No; 1 = Yes
28775	>RiskLimitGroup	N	STRING (1)	User Pre-Trade risk group
28777	>RiskLimitQty	N	QTY	Risk limit quantity.
25205	>NettingCoefficient	N	FLOAT	Weighting factor used in the Extended PTRL consumption calculation. Default value is 1.0000 for trading members and 0.0000 for clearing members.
28779	>RiskLimitOpenQty	N	QTY	Open quantity for orders .
28780	>RiskLimitNet-PositionQty	N	QTY	Net position quantity for a requested pre-trade risk limit scope.
</RiskLimitTypesGrp>				
1666	RiskLimitRequestID	Y	STRING (1)	Unique identifier for the User Party Risk Limits Request.
1300	MarketSegmentID	Y	INT(10)	Product identifier.
1533	RiskLimitPlatform	Y	INT	The area to which risk limit is applicable. Scope of Pre-Trade risk limits. Value: 0 = On-book; 1 = Off-book
460	Product	Y	CHAR	Free format string, can include characters, numbers, punctuation and many special characters Value: O = Option; F = Future
<Trailer />				

Confidential

6.6.10 User Party Risk Limits Report

The User Party Risk Limits Report message is used as the application level response to a User Party Risk Limits Request.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='UCM' />				
<Parties>, cardinality: 2, record counter: NoPartyIDs, containing occurrences: entering_trader , executing_trader				
448	PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering Trader

Tag	Field Name	Req'd	Data Type	Description

Confidential

2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyID-Source (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 Algorithm; 24 Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 Person
</Parties>				
<RiskLimitTypesGrp>, cardinality: 1-64, record counter: NoRiskLimitTypes				
1530	>RiskLimitType	Y	INT	Specifies the type of risk limit. Value: 4 Long limit; 5 Short limit
1533	>RiskLimitPlatform	Y	INT	The area to which risk limit is applicable. Scope of Pre-Trade risk limits. Value: 0 On-book; 1 Off-book
28776	RiskLimit-RequestingPartyRole	N	INT	Requesting party role for a pre-trade risk limit. Value: 4 Clearing firm; 22 Exchange; 59 Executing unit

Confidential

Tag	Field Name	Req'd	Data Type	Description
28778	>RiskLimitViolation-Indicator	N	INT	Indicator for a pre-trade risk limit violation. Value: 0 – No; 1 – Yes
28775	>RiskLimitGroup	N	STRING (1)	User Pre-Trade risk group
28777	>RiskLimitQty	N	QTY	Risk limit quantity.
25205	>NettingCoefficient	N	FLOAT	Weighting factor used in the Extended PTRL consumption calculation. Default value is 1.0000 for trading members and 0.0000 for clearing members.
28779	>RiskLimitOpenQty	N	QTY	Open quantity for orders.
28780	>RiskLimitNet-PositionQty	N	QTY	Net position quantity for a requested pre-trade risk limit scope.
</RiskLimitTypesGrp>				
1666	RiskLimitRequestID	Y	STRING (1)	Unique identifier for the User Party Risk Limits Request.
1300	MarketSegmentID	Y	INT(10)	Product identifier.
1533	RiskLimitPlatform	Y	INT	The area to which risk limit is applicable. Scope of Pre-Trade risk limits. Value: 0 – On-book; 1 – Off-book
1880	PartyDetailDefinition-Result	Y	INT	Integral number. Value: 0 – Active; 1 – Suspend
460	Product	Y	CHAR	Free format string, can include characters, numbers, punctuation and many special characters Value: O – Option; F – Future
<Trailer />				

Confidential

6.6.11 Institution Trade Acceptance

The Institutional Trade Acceptance Report message provides the Acceptance response of an Institutional Trade. This message is unsolicited and only applicable to Institutional Trades..

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U5' />				
<Parties>, cardinality: 0-3, record counter: NoPartyIDs, containing occurrences: client_id , entering_trader , executing_trader				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 – Algorithm; 24 – Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 – Person
</Parties>				
<Instrument>				
55	Symbol	Y	STRING (10)	MCX product identifier. If no product identifier is provided Symbol (55) must contain '[N/A]'.
48	SecurityID	Y	INT(20)	MCX instrument identifier.
22	SecurityIDSource	Y	STRING (1)	Source of the securityID Value: M – Marketplace-assigned identifier

Confidential

Tag	Field Name	Req'd	Data Type	Description
1227	ProductComplex	N	INT	Identifies an entire suite of products for a given market. In Futures this may be 'interest rates', 'agricultural', 'equity indexes', etc. Value: Description 1: Simple Instrument 5: Futures Spread
167	SecurityType	N	STRING (4)	Indicates the type of security. Value: Description MLEG: Multileg Instrument 0: Cumulative at AssetType Level 1: Underlying Futures 2: Options 3: Futures
762	SecuritySubType	N	INT(10)	Strategy type.
</Instrument>				
37	OrderID	Y	INT(20)	Exchange Order ID generated by the MCX system.
30060	UTransactTime	N	INT(20)	Transaction timestamp. Number of nanoseconds since Jan 1, 1970.
31	LastPx	N	PRICE	Price of this (last) fill.
1003	TradeID	Y	INT(10)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
1815	TradingCapacity	N	INT	Not used.

Confidential

Tag	Field Name	Req'd	Data Type	Description
581	AccountType	N	STRING	Free format string, can include characters, numbers, punctuation and many special characters Value: 1 – Own; 3 – Client; 5 – Institution
54	Side	Y	CHAR	Side Value: 1 – Buy; 2 – Sell
25042	AutoAcceptIndicator	N	INT	Integral number. Value: 1 – Accepted; 2 – Rejected; 3 – Pending
9019	ConfirmationTime	N	UTCTIME-STAMP	Confirmation Time
32	LastQty	N	QTY	Quantity (e.g. shares) bought/sold on this (last) fill.
58	Text	N	STRING (128)	Free format text string.
76	ExecBroker	N	INT	Integral number.
<Trailer />				

Confidential

6.6.12 Institution Trade Rejection Report

When clearing member of specific CP rejects,then that is notified with Institution Trade Rejection Report.

Tag	Field Name	Req'd	Data Type	Description
<Header with MsgType='U9' />				
<Parties>, cardinality: 0-3, record counter: NoPartyIDs, containing occurrences: client_id , entering_trader , executing_trader				
448	>PartyID	Y	STRING (35)	Party identifier/code. Party identifier/code. See PartyIDSource (447) and PartyRole (452).
447	>PartyIDSource	Y	CHAR	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Value: D – Description: Proprietary custom code
452	>PartyRole	Y	INT	Party Role. Value: 36 – Description: Entering trader

Confidential

Tag	Field Name	Req'd	Data Type	Description
2376	>PartyRoleQualifier	N	INT	Indicates the nature of value submitted in the field PartyID (448). PartyRoleQualifier (2376) = 21 (Specialist) and 22 (Algorithm) are only possible together with PartyIDSource (447) = 'D'. PartyRoleQualifier (2376) = 24 (Natural person) is only possible together with PartyIDSource (447) = 'P'. PartyRoleQualifier (2376) = 21 (Specialist) is not allowed in request messages. Value: 22 = Algorithm; 24 = Natural person
802	>NoPartySubIDs	N	NUMINGROUP	Number of PartySubIDs. Only one entry. Only in combination with PartyRole (452) = 7 (Entering firm), 12 (Executing trader) and 36 (Entering trader) possible.
523	>PartySubID	N	STRING (6)	Person (Short name).
803	>PartySubIDType	N	INT	Type of PartySubID. Value: 2 = Person
</Parties>				
1003	TradeID	Y	INT(10)	The unique ID assigned to the trade entity once it is received or matched by the exchange or central counterparty.
1506	SideTradeID	Y	INT(10)	Used to represent the trade ID for each side of the trade assigned by an intermediary.
30060	UTransactTime	N	INT(20)	Transaction timestamp. Number of nanoseconds since Jan 1, 1970.
31	LastPx	N	PRICE	Price of this (last) fill.
1815	TradingCapacity	N	INT	Not used.
75	TradeDate	N	LOCALMKT-DATE	Indicates date of trade referenced in this message in YYYYMMDD format.
32	LastQty	N	QTY	Quantity (e.g. shares) bought/sold on this (last) fill.

Confidential

<table>
<tr><th>Tag</th><th>Field Name</th><th>Req'd</th><th>Data Type</th><th>Description</th></tr>
<tr><td>54</td><td>Side</td><td>Y</td><td>CHAR</td><td>Side<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Buy</td></tr><tr><td>2</td><td>Sell</td></tr></table></td></tr>
<tr><td>25042</td><td>AutoAcceptIndicator</td><td>N</td><td>INT</td><td>Integral number.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Accepted</td></tr><tr><td>2</td><td>Rejected</td></tr><tr><td>3</td><td>Pending</td></tr></table></td></tr>
<tr><td>58</td><td>Text</td><td>N</td><td>STRING (128)</td><td>Free format text string.</td></tr>
<tr><td>76</td><td>ExecBroker</td><td>N</td><td>INT</td><td>Integral number.</td></tr>
<tr><td colspan="5"><Trailer /></td></tr>
</table>

Confidential

7.Occurences

OrderAttributeGrp

This section contains all occurrences of structure "OrderAttributeGrp".

liquidity_provision_activity_order

- OrderAttributeType = LIQUIDITY_PROVISION_ACTIVITY_ORDER (mandatory)
- OrderAttributeValue (mandatory)

risk_reduction_order

- OrderAttributeType = RISK_REDUCTION_ORDER (mandatory)
- OrderAttributeValue (mandatory)

Confidential

7.1 Parties

This section contains all occurrences of structure "Parties".

beneficiary

- PartyID (mandatory)
- PartyRole = BENEFICIARY (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

clearing_firm

- PartyID (mandatory)
- PartyRole = CLEARING_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

clearing_organization

- PartyID (mandatory)
- PartyRole = CLEARING_ORGANIZATION (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

clearing_unit

- PartyID (mandatory)
- PartyRole = CLEARING_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

client_id

- PartyID (mandatory)
- PartyRole = CLIENT_ID (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

entering_firm

- PartyID (mandatory)
- PartyRole = ENTERING_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

Confidential

entering_trader

- PartyID (mandatory)
- PartyRole = ENTERING_TRADER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

executing_firm

- PartyID (mandatory)
- PartyRole = EXECUTING_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

executing_system

- PartyID (mandatory)
- PartyRole = EXECUTING_SYSTEM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

executing_trader

- PartyID (mandatory)
- PartyRole = EXECUTING_TRADER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- PartySubIDType = PERSON (mandatory)

executing_unit

- PartyID (mandatory)
- PartyRole = EXECUTING_UNIT (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

execution_identifier

- PartyID (mandatory)
- PartyRole = EXECUTING_TRADER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

execution_identifier_algo

- PartyID (mandatory)

Confidential

- PartyRole = EXECUTING_TRADER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- PartyRoleQualifier = ALGORITHM (mandatory)

execution_identifier_human

- PartyID (mandatory)
- PartyRole = EXECUTING_TRADER (mandatory)
- PartyIDSource = SHORT_CODE_IDENTIFIER (mandatory)
- PartyRoleQualifier = NATURAL_PERSON (mandatory)

investment_identifier

- PartyID (mandatory)
- PartyRole = INVESTMENT_DECISION_MAKER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

investment_identifier_algo

- PartyID (mandatory)
- PartyRole = INVESTMENT_DECISION_MAKER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- PartyRoleQualifier = ALGORITHM (mandatory)

investment_identifier_human

- PartyID (mandatory)
- PartyRole = INVESTMENT_DECISION_MAKER (mandatory)
- PartyIDSource = SHORT_CODE_IDENTIFIER (mandatory)
- PartyRoleQualifier = NATURAL_PERSON (mandatory)

location_id

- PartyID (mandatory)
- PartyRole = LOCATION_ID (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

order_origination_firm

- PartyID (mandatory)

Confidential

- PartyRole = ORDER_ORIGINATION_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

position_account

- PartyID (mandatory)
- PartyRole = FLEXIBLE_ACCOUNT_IDENTIFIER (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

session_ID

- PartyID (mandatory)
- PartyRole = SESSION_ID (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

settlement_account

- PartyID (mandatory)
- PartyRole = SETTLEMENT_ACCOUNT (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

settlement_firm

- PartyID (mandatory)
- PartyRole = SETTLEMENT_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

settlement_location

- PartyID (mandatory)
- PartyRole = SETTLEMENT_LOCATION (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

settlement_unit

- PartyID (mandatory)

- PartyRole = SETTLEMENT_FIRM (mandatory)

Confidential

- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

takeup_firm

- PartyID (mandatory)
- PartyRole = TAKEUP_FIRM (mandatory)
- PartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

7.2 Requesting Parties

This section contains all occurrences of structure "RequestingParties".

requesting_entering_firm

- RequestingPartyID (mandatory)
- RequestingPartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- RequestingPartyRole = ENTERING_FIRM (mandatory)

requesting_executing_system

- RequestingPartyID (mandatory)
- RequestingPartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- RequestingPartyRole = EXECUTING_SYSTEM (mandatory)

requesting_executing_trader

- RequestingPartyID (mandatory)
- RequestingPartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- RequestingPartyRole = EXECUTING_TRADER (mandatory)

Confidential

RootParties

This section contains all occurrences of structure "RootParties".

root_entering_trader

- RootPartyID (mandatory)
- RootPartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)
- RootPartyRole = ENTERING_TRADER (mandatory)

root_execution_venue

- RootPartyID (mandatory)
- RootPartyIDSource = Market_identifier_code (mandatory)
- RootPartyRole = Execution_venue (mandatory)

Confidential

7.3 Target Parties

This section contains all occurrences of structure "TargetParties".

target_executing_trader

- TargetPartyID (mandatory)
- TargetPartyRole = EXECUTING_TRADER (mandatory)
- TargetPartyIDSource = PROPRIETARYCUSTOMCODE (mandatory)

target_session_ID

- TargetPartyID (mandatory)
- TargetPartyRole = SESSION_ID (mandatory)
- TargetPartyIDSource = PROPRIETARYCUSTOMCODE (mandatory

TrdCapRptSideGrp

This section contains all occurrences of structure "TrdCapRptSideGrp".

Side1

- Side (mandatory)
- SideLastQty (mandatory)
- SideTradeID (mandatory)
- Parties
  - clearing_firm
  - clearing_organization
  - executing_trader
  - executing_firm
  - executing_unit
  - session_ID
  - order_origination_firm
  - beneficiary
  - position_account
  - takeup_firm
  - settlement_location
  - settlement_account

Confidential

  - settlement_firm
  - client_id
  - investment_identifier
  - execution_identifier
  - clearing_firm
  - settlement_firm
  - client_id
  - execution_identifier_human
  - execution_identifier_algo
  - investment_identifier_human
  - investment_identifier_algo

- PartyIDClearingUnit (mandatory)
- Account (mandatory)

- PositionEffect (mandatory)
- FreeText1 (mandatory)
- OrderCategory (mandatory)
- SideLiquidityInd (mandatory)
- StrategyLinkID (mandatory)
- OrderID (mandatory)
- OrdType (mandatory)
- Price (mandatory)
- LeavesQty (mandatory)
- CumQty (mandatory)
- OrderAttributeGrp
- CustOrderHandlingInst (mandatory)
- FreeText2 (mandatory)
- FreeText3 (mandatory)
- SideLastPx (mandatory)

Confidential

ValueChecksGrp

This section contains all occurrences of structure "ValueChecksGrp".

notional_value_check

- ValueCheckType = NOTIONAL_VALUE_CHECK (mandatory)
- ValueCheckAction (mandatory)

price_check

- ValueCheckType = PRICE_CHECK (mandatory)
- ValueCheckAction (mandatory)

Confidential

8. Appendix

Code Snippet - c++

Password has to be encrypted as per RSA token.

/** The sample code is provided *AS-IS* basis as an reference. The code is not be used in your application or system without* fully testing and validating the readiness of your application or system.*/

```
#include <openssl/pem.h>
#include <openssl/ssl.h>
#include <openssl/rsa.h>
#include <openssl/evp.h>
#include <openssl/bio.h>
#include <openssl/err.h>
#include <stdio.h>
#include <math.h>
#include <iostream>
#include <fstream>

using namespace std;

int padding = RSA_PKCS1_PADDING;

int Base64Encode(const char* message, char** buffer, int length) { //Encodes
a string to base64

BIO *bio, *b64;

FILE* stream;

int encodedSize = 4*ceil((double)length/3);

*buffer = (char *)malloc(encodedSize+1);

stream = fmemopen(*buffer, encodedSize+1, "w");

b64 = BIO_new(BIO_f_base64());

bio = BIO_new_fp(stream, BIO_NOCLOSE);

bio = BIO_push(b64, bio);

BIO_set_flags(bio, BIO_FLAGS_BASE64_NO_NL); //Ignore newlines - write
everything in one line

BIO_write(bio, message, length);

BIO_flush(bio);

BIO_free_all(bio);

fclose(stream);

return (0); //success

}
```


Confidential

```
int calcDecodeLength(const char* b64input) { //Calculates the length of a
decoded base64 string

int len = strlen(b64input);

int padding = 0;

if (b64input[len-1] == '=' && b64input[len-2] == '=') //last two chars are =
padding = 2;

else if (b64input[len-1] == '=') //last char is =

padding = 1;

return (int)len*0.75 - padding;

}

int Base64Decode(char* b64message, char** buffer) { //Decodes a base64
encoded string BIO *bio, *b64;

int decodeLen = calcDecodeLength(b64message),

len = 0;

*buffer = (char*)malloc(decodeLen+1);

FILE* stream = fmemopen(b64message, strlen(b64message), "r"); b64 =
BIO_new(BIO_f_base64());

bio = BIO_new_fp(stream, BIO_NOCLOSE);


bio = BIO_push(b64, bio);

BIO_set_flags(bio, BIO_FLAGS_BASE64_NO_NL); //Do not use newlines to flush
buffer len = BIO_read(bio, *buffer, strlen(b64message));

//Can test here if len == decodeLen - if not, then return an error

(*buffer)[len] = '\0';

BIO_free_all(bio);

fclose(stream);

return (0); //success

}

RSA * createRSAWithFilename(char * filename,int spublic)

{

FILE * fp = fopen(filename,"rb");

if(fp == NULL)
```


Confidential

```
{

printf("Unable to open file %s \n",filename);

return NULL;

}

RSA *rsa= RSA_new() ;

if(spublic)

{

rsa = PEM_read_RSA_PUBKEY(fp, &rsa,NULL, NULL);

}

else

{

rsa = PEM_read_RSAPrivateKey(fp, &rsa,NULL, NULL);

}

return rsa;

}

//Function encrypts the value returns length of encrypted text

int public_encrypt(unsigned char * data,int data_len, unsigned char
*encrypted)

{

RSA * rsa =
createRSAWithFilename((char*)"/home/a1783558/public.pem",1);//Path of public
key int result = RSA_public_encrypt(data_len,data,encrypted,rsa,padding);
return result;

}

void printLastError(char *msg)

{

char * err;

ERR_load_crypto_strings();

ERR_error_string(ERR_get_error(), err);

printf("%s ERROR: %s\n",msg, err);

free(err);

}
```


Confidential

```
int main(){

//Text to be encrypted

char plainText[2048/8] = "Hello this is Ravi"; //key length : 2048 unsigned
char encrypted[4098]={};

int encrypted_length= public_encrypt((unsigned
char*)plainText,strlen(plainText),encrypted);

if(encrypted_length == -1){

printLastError("Public Encrypt failed ");

exit(0);

}

printf("Encrypted length =%d\n",encrypted_length);

//Copy char array to pointer

char * enc = (char*)encrypted;

//Base64 Encoding

Base64Encode(enc ,&enc, encrypted_length);

cout<<enc<<"\n";

}
```

Code Snippet - java

Password has to be encrypted as per RSA token.

/** The sample code is provided *AS-IS* basis as an reference. The code is not be used in your application or system without* fully testing and validating the readiness of your application or system.*/

```
package com.mcx;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.nio.charset.StandardCharsets;
import java.security.*;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;
import java.util.Base64;
public class SampleCodeJava {

public static void main(String[] args) throws NoSuchAlgorithmException,
NoSuchPaddingException, InvalidKeySpecException, //Ciphper encoding
Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");

String clearTextPwd = "MyPassword";
```


Confidential

```
PublicKey publicKey = loadPublicKey();

cipher.init(Cipher.ENCRYPT_MODE, publicKey);

byte[] encrypted =
cipher.doFinal(clearTextPwd.getBytes(StandardCharsets.UTF_8)); String
encryptedPwd = Base64.getEncoder().encodeToString(encrypted)

System.out.println("ClearText Password: " + clearTextPwd);

System.out.println("Encrypted Password: " + encryptedPwd);

}

private static PublicKey       loadPublicKey() throws
InvalidKeySpecException, NoSuchAlgorithmException {

//      The below working sample public key provided as inline key. However
in production region, PEM encoded Public Key of MCX to be downloaded from
MCX FTP server.

String publicKeyPEM =
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0gzs2VUrYlrnDPDrsSJf\n" +
"G2VGjos7/WuF5LWn28O0ET/RWkq4R1PiQCG8Dur15u+KuBt3vIQTE4n7S/Q5FtlU\n" +
"p2/QWjQchKhnnIAeblpBlIXTdGdIzZLakSSFN6+H66lWFPkYL8d+F8Koizt2KBC/\n" +
"4Dl5RR8hHeIKneeSa23S8YsA06blqmFw5uL95LNtL9dtXNzw+sIt72iYcmMl4cC2\n" +
"e8tNGcG0GWYqV4N4MGAdX0gHaZtsEDM1UU4D6fM1tU5ddfLtGPEqAcYaY13znSVW\n" +
"XU2jeORt9zdfJ2+48dylWc6slfJ0O3sQBVWytM1L09TIXZ/v+DK4kAeIQlxI3BaU\n" +
"bQIDAQAB";

publicKeyPEM = publicKeyPEM.replaceAll("\\s", "");

// decode to get the binary DER representation

byte[] publicKeyDER = Base64.getDecoder().decode(publicKeyPEM); KeyFactory
keyFactory = KeyFactory.getInstance("RSA");

PublicKey publicKey = keyFactory.generatePublic(new
X509EncodedKeySpec(publicKeyDER)); return publicKey;

}
}
```

Code Snippet - CSharp

Password has to be encrypted as per RSA token.

/** The sample code is provided *AS-IS* basis as an reference. The code is not be used in your application or system without* fully testing and validating the readiness of your application or system.*/

```
using System;
using System.Text;
using Org.BouncyCastle.Crypto;
using Org.BouncyCastle.OpenSsl;
using Org.BouncyCastle.Crypto.Encodings;
using Org.BouncyCastle.Crypto.Engines;
```


Confidential

```
using System.IO;

namespace SampleCodeCSharp

{

class Program

{

public static string publicKey = "-----BEGIN PUBLIC KEY-----MIIBIjANB" +

"gkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0gzs2VUrYlrnDPDrsSJfG2VGjos7/WuF" +
"5LWn28O0ET/RWkq4R1PiQCG8Dur15u+KuBt3vIQTE4n7S/Q5FtlUp2/QWjQchKhnnIAebl" +
"pBlIXTdGdIzZLakSSFN6+H66lWFPkYL8d+F8Koizt2KBC/4Dl5RR8hHeIKneeSa23S8" +
"YsA06blqmFw5uL95LNtL9dtXNzw+sIt72iYcmMl4cC2e8tNGcG0GWYqV4N4MGAdX0gHaZt" +
"sEDM1UU4D6fM1tU5ddfLtGPEqAcYaY13znSVWXU2jeORt9zdfJ2+48dylWc6slfJ0O3sQB" +
"VWytM1L09TIXZ/v+DK4kAeIQlxI3BaUbQIDAQAB-----END PUBLIC KEY-----";

public static string PasswordPlainText = "MyPassword";

static void Main(string[] args)

{

Program prg = new Program();

string EncryptedText = prg.RsaEncryptWithPublic(PasswordPlainText,
publicKey);

Console.WriteLine(EncryptedText);

Console.ReadLine();

}

public string RsaEncryptWithPublic(string clearText, string publicKey)

{

try

{

var bytesToEncrypt = Encoding.UTF8.GetBytes(clearText);

var encryptEngine = new Pkcs1Encoding(new RsaEngine());

using (var txtreader = new StringReader(publicKey))

{

var keyParameter = (AsymmetricKeyParameter)new
PemReader(txtreader).ReadObject(); encryptEngine.Init(true, keyParameter);

}
```


Confidential

```
var encrypted =
Convert.ToBase64String(encryptEngine.ProcessBlock(bytesToEncrypt, 0,
bytesToEncrypt.Length)) return encrypted;

}
catch (Exception ex)

{

throw ex;
}
}
}
}
```

9. Annexure for Encryption/Decryption

For symmetric encryption/decryption methodology –

A. Encryption:

Initialization→

```
void encrypt_EVP_aes_256_gcm_init(EVP_CIPHER_CTX **ctx, unsigned char
*key, unsigned char *iv)
{
if(!(*ctx = EVP_CIPHER_CTX_new()))
handleErrors();
if(1 != EVP_EncryptInit_ex(*ctx, EVP_aes_256_gcm(), NULL, key, iv))
handleErrors();
}
```

Encryption→

```
void encrypt(EVP_CIPHER_CTX *ctx, unsigned char *plaintext, int
plaintext_len, unsigned char *ciphertext, int *ciphertext_len)
{
int len;
if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
handleErrors();
*ciphertext_len = len;
}
```

B. Decryption:

Initialization→

```
void decrypt_EVP_aes_256_gcm_init(EVP_CIPHER_CTX **ctx, unsigned char
*key, unsigned char *iv)
{
if(!(*ctx = EVP_CIPHER_CTX_new()))
handleErrors();
if(1 != EVP_DecryptInit_ex(*ctx, EVP_aes_256_gcm(), NULL, key, iv))
handleErrors();
}
```


Confidential

Decryption→

```
int decrypt(EVP_CIPHER_CTX *ctx, unsigned char *ciphertext, int
ciphertext_len, unsigned char *plaintext, int *plaintext_len)
{
int len;
if(1 != EVP_DecryptUpdate(ctx, plaintext, &len, ciphertext,
ciphertext_len))
handleErrors();
*plaintext_len = len;
}
```

Example: Encryption using GCM mode

Code Snippet - C++

```
int gcm_encrypt(unsigned char *plaintext, int plaintext_len,
                unsigned char *aad, int aad_len,
                unsigned char *key,
                unsigned char *iv, int iv_len,
                unsigned char *ciphertext,
                unsigned char *tag)
{
    EVP_CIPHER_CTX *ctx;

    int len;

    int ciphertext_len;


    /* Create and initialise the context */
    if(!(ctx = EVP_CIPHER_CTX_new()))
        handleErrors();

    /* Initialise the encryption operation. */
    if(1 != EVP_EncryptInit_ex(ctx, EVP_aes_256_gcm(), NULL, NULL, NULL))
        handleErrors();

    /*
     * Set IV length if default 12 bytes (96 bits) is not appropriate
     */
    if(1 != EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_GCM_SET_IVLEN, iv_len, NULL))
        handleErrors();

    /* Initialise key and IV */
    if(1 != EVP_EncryptInit_ex(ctx, NULL, NULL, key, iv))
        handleErrors();

    /*
     * Provide any AAD data. This can be called zero or more times as
     * required
```


Confidential

```
     */
    if(1 != EVP_EncryptUpdate(ctx, NULL, &len, aad, aad_len))
        handleErrors();

    /*
     * Provide the message to be encrypted, and obtain the encrypted output.
     * EVP_EncryptUpdate can be called multiple times if necessary
     */
    if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
        handleErrors();
    ciphertext_len = len;

    /*
     * Finalise the encryption. Normally ciphertext bytes may be written at
     * this stage, but this does not occur in GCM mode
     */
    if(1 != EVP_EncryptFinal_ex(ctx, ciphertext + len, &len))
        handleErrors();
    ciphertext_len += len;

    /* Get the tag */
    if(1 != EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_GCM_GET_TAG, 16, tag))
        handleErrors();

    /* Clean up */
    EVP_CIPHER_CTX_free(ctx);

    return ciphertext_len;
}
```

Code Snippet – C#

```
public byte[] AESGCMEncryptionNew(byte[] dataToEncrypt)
        {
            byte[] resultWithNoTag;
            GcmBlockCipher cipher = new GcmBlockCipher(new AesFastEngine());


            cipher.Init(true, new AeadParameters(new
KeyParameter(Encoding.UTF8.GetBytes("ABCGFEDTHGYJSTFGABCGFEDTHGYJSTFG")), 128,
Encoding.UTF8.GetBytes("dsfrehbndrft")));
            byte[] encryptedResult = new
byte[cipher.GetOutputSize(dataToEncrypt.Length)];
            int lenRes = cipher.ProcessBytes(dataToEncrypt, 0,
dataToEncrypt.Length, encryptedResult, 0);
            cipher.DoFinal(encryptedResult, lenRes);
            resultWithNoTag = new byte[encryptedResult.Length - 16];
            Array.Copy(encryptedResult, resultWithNoTag,
resultWithNoTag.Length);
            return resultWithNoTag;
        }
```


Confidential

Example: Decryption using GCM mode

Code Snippet - C++

```
int gcm_decrypt(unsigned char *ciphertext, int ciphertext_len,
                unsigned char *aad, int aad_len,
                unsigned char *tag,
                unsigned char *key,
                unsigned char *iv, int iv_len,
                unsigned char *plaintext)
{
    EVP_CIPHER_CTX *ctx;
    int len;
    int plaintext_len;
    int ret;

    /* Create and initialise the context */
    if(!(ctx = EVP_CIPHER_CTX_new()))
        handleErrors();

    /* Initialise the decryption operation. */
    if(!EVP_DecryptInit_ex(ctx, EVP_aes_256_gcm(), NULL, NULL, NULL))
        handleErrors();

    /* Set IV length. Not necessary if this is 12 bytes (96 bits) */
    if(!EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_GCM_SET_IVLEN, iv_len, NULL))
        handleErrors();

    /* Initialise key and IV */
    if(!EVP_DecryptInit_ex(ctx, NULL, NULL, key, iv))
        handleErrors();

    /*
     * Provide any AAD data. This can be called zero or more times as
     * required
     */
    if(!EVP_DecryptUpdate(ctx, NULL, &len, aad, aad_len))
        handleErrors();

    /*
     * Provide the message to be decrypted, and obtain the plaintext output.
     * EVP_DecryptUpdate can be called multiple times if necessary
     */
    if(!EVP_DecryptUpdate(ctx, plaintext, &len, ciphertext, ciphertext_len))
        handleErrors();
    plaintext_len = len;

    /* Set expected tag value. Works in OpenSSL 1.0.1d and later */
    if(!EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_GCM_SET_TAG, 16, tag))
        handleErrors();

    /*
     * Finalise the decryption. A positive return value indicates success,
```


Confidential

```
     * anything else is a failure - the plaintext is not trustworthy.
     */
    ret = EVP_DecryptFinal_ex(ctx, plaintext + len, &len);

    /* Clean up */
    EVP_CIPHER_CTX_free(ctx);

    if(ret > 0) {
        /* Success */
        plaintext_len += len;
        return plaintext_len;
    } else {
        /* Verify failed */
        return -1;
    }
}
```

Code Snippet – C#

```
public byte[] AESGCMDecryptionNew(byte[] dataToDecrypt)
        {
            byte[] resultWithNoTag;
            GcmBlockCipher cipher = new GcmBlockCipher(new AesFastEngine());
            cipher.Init(true, new AeadParameters(new
KeyParameter(Encoding.UTF8.GetBytes("ABCGFEDTHGYJSTFGABCGFEDTHGYJSTFG")), 128,
Encoding.UTF8.GetBytes("dsfrehbndrft")));
            byte[] decryptedResult = new
byte[cipher.GetOutputSize(dataToDecrypt.Length)];
            int lenRes = cipher.ProcessBytes(dataToDecrypt, 0,
dataToDecrypt.Length, decryptedResult, 0);
            cipher.DoFinal(decryptedResult, lenRes);
            resultWithNoTag = new byte[decryptedResult.Length - 16];
            Array.Copy(decryptedResult, resultWithNoTag,
resultWithNoTag.Length);
            return resultWithNoTag;
        }
```


Confidential