National Commodity and Derivatives Exchange

Drop Copy Gateway

FIX 5.0SP2 Specification

Version	4.01
Release Date	20th June 2017

1 DOCUMENT CONTROL

1.1 Table of Contents

1.2 Revision History

Date	Ver	Sections	Description
29 May 2015	1.00		Initial Draft
5 Jan 2015	2.00		Added datatype and field length
XX May 2016	3.00	7.6.4	CR message updated to reflect the increase of the Text (58) field to 200.
		8.1	List of Session Reject reasons updated.
		7.6.4	RiskReductionMode (32033) included in the CR message.
		3.6.3 & 7.6.1	Indicator for executions for multi-leg orders included in the Execution Report.
02 Jun 2017	4.00	7.6.1 7.6.3 7.6.4	Internal Reference flag added Changes related to mass cancellation based on segment Security type and internal reference flag added
20th June 2017	4.01	7.6.1	Added LastOptPx(32022) and Volatility(1188) to Execution Report message.

1.3 References

FIXT 1.1 Specification
FIX 5.0 (Service Pack 2) Specification
NCDEX – Trading Gateway (FIX 5.0) Specification

1.4 Exclusions

The given functionality is currently not supported by NCDEX

Sr. NO	Functionality	
1	Order Type	Market
		Market If Touched
		Trailing Stop
		Trailing Stop Limit
		Reserve
		Minimum Fill
		Named
		At the Open (OPG)
		Fill or Kill(FOK)
		At the Close (ATC)
		Good For Auction(GFA)
		Good Till Time (GTT)
		Closing Price Cross
2	Give-up from CTCL frond end	
3 4	Negotiated Trades (EFP) Spread Instrument	Stop Stop_Limit
5	Strategies	BF CD

1.5 Definitions, Acronyms and Abbreviations

Client A participant or service bureau connected to the drop copy gateway.

FIX Version 5.0 (Service Pack 2) of the Financial Information Exchange Protocol.

FIX Connection A bi-directional stream of ordered messages between the client and server within a particular login. A FIX connection ends when the client logs out or if the TCP/IP connection is terminated.

FIX Session A bi-directional stream of ordered messages between the client and server within a continuous sequence number series. A single FIX session can exist across multiple FIX connections.

FIXT Version 1.1 of the Financial Information Exchange Session Protocol.

NCDEX National Commodity and Derivatives Exchange

NNF ID This should be a fifteen digit ID with the following breakup.

a. First 3 digits will be the identity code for distinguishing orders i.e. whether the order is program generated (Automated Trading) or manually (CTCL Workstations) entered. Program generated (Automated trading) or manually entered should send the ATS/CTCLvendor code respectively.

E.g. If an order is generated through CTCL workstation, they should enter the CTCL vendor code and if an order is generated thought ATS (also a CTCL workstation), they should enter the ATS vendor code.

b. Next 6 digits will be unique approved person code. (first 3 digits will be branch code and last 3 digits will be CTCL terminal Id)

c. Next 6 digits will be the Pin code.

Order Executable interest which may be an order.

Order Book Each instrument is traded across multiple separate and distinct order books (e.g. regular, odd lot, etc.). Each Execution Report includes an indication of the instrument and order book to which it relates.

Trading Gateway The interface of NCDEX that allows participants and service bureaus to submit and manage their trading interest.

Server The drop copy gateway of NCDEX.

Trading Mnemonic Each order must include the trading mnemonic it is submitted under. Trading privileges are assigned at the level of trading mnemonics.

2 OVERVIEW

NCDEX offers a drop copy gateway that will enable participants to receive additional copies of the Execution Reports generated by the matching system. This interface may also be used by clients to download the current status of all their active orders in the event of a failure. The drop copy service cannot be used to submit orders or receive market data.

The Drop Copy Gateway will also be used to send the updates received on risk limit status updates/ warnings (via Party Risk Limits Update Report) and risk limit statistics (via Party Risk Limit Report) relating to entities.

The interface is a point-to-point service based on the technology and industry standards TCP/IP, FIXT and FIX. The session and application event models and messages are based on versions 1.1 and 5.0 (Service Pack 2) of the FIXT and FIX protocols respectively. Please refer to Section 7.2 for the instances where the server varies from the FIX protocol.

The encryption of messages between the client and server is not supported.

The exchange requires that the vendor/exchange's member undergo a conformance test upon completing development of the interface. The vendor/exchange's member must contact the exchange to schedule an appropriate period for testing.

The vendor/exchange's member may contact the CTCL division of the exchange to seek clarification

CTCL Team (Technology Department)

National Commodity & Derivatives Exchange Limited,
Akruti Corporate Park, 1st Floor, Near G.E.Garden, L.B.S. Marg,
Kanjurmarg (West), Mumbai - 400 078
Phone (022) 66406717/66406609-12
Fax (022) 66473201/66406899
Email: CTCL@ncdex.com

3 SERVICE DESCRIPTION

3.1 Services Supported by Trading Gateway

A description of the services (e.g. order types, notification of market operations actions, etc.) available via the Trading Gateway is provided in the FIX specification for this interface which vendors are encouraged to read together with this specification.

3.2 Connection Configuration

3.2.1 Real-Time Connections

A real-time client will receive a drop copy of each eligible Execution Report immediately after it is published.

A participant connection will be configured to receive a drop copy of all the Execution Report messages generated for the firm for the events outlined in Section 0. The connection of a service bureau will be configured to receive drop copies for all the firms it serves. If required, a firm or service bureau connection could be configured to only receive drop copies for selected trading mnemonics.

For the purpose of redundancy, the service supports the configuration of multiple drop copy connections to send the same information on the activity of the selected firms/mnemonics.

The identity of the CompID that transmitted the order a particular drop copy relates to will be specified in the header field OnBehalfOfCompID (115).

Please refer to Section 6.4 for a description of how the Execution Reports published during the time a real-time client is disconnected from the server may be recovered.

A real-time client may also use the open order download service (outlined in Section 3.4) to recover the status of all active orders in the event of a system failure.

3.2.2 Non-Real Time Connections

Execution Reports will not be streamed to non-real time clients. Such a client may only connect to the server to use the order download service outlined in Section 3.4.

3.3 Supported Events

Clients will receive drop copies of the Execution Reports generated for the following events:

(i) Order accepted
(ii) Order pending
(iii) Order rejected
(iv) Order executed
(v) Order expired
(vi) Order cancelled
(vii) Order cancel/replaced
(viii) Order cancel/replace pending
(ix) Trade cancellation
(x) Trade correction

3.4 Open Order Download

Any client may use the Mass Order Status Request message to download the current status of each active order for a specified trading mnemonic. The request may apply to all active orders for the trading mnemonic or be limited to only orders for a specified instrument or segment.

The total number of Mass Order Status Requests that a client may submit is limited to *<20>* each day. A client may request NCDEX to reset its request count. This feature is intended to help manage an emergency situation and should not be relied upon as a normal practice.

If a request is successful, the server will respond with an Execution Report for each active order side[1]for the specified trading mnemonic and instrument/segment. Each such message will include the MassStatusReqID (584) of the request, an ExecID (17) of "0" and an ExecType (150) of Order Status (I). The last Execution Report sent by each partition in response to the request will include a LastRptRequested (912) of Last Message (Y).

The server will transmit a single Execution Report if the request is rejected or if there are no active orders for the specified trading mnemonic and instrument/segment. Such a message will include the MassStatusReqID (584) of the request, an ExecID (17) of "0", an ExecType (150) of Order Status (I) and an OrdStatus (39) of Rejected (8).The message will not include fields that relate to order-specific information (e.g. ClOrdID (11), OrderID (37), OrderQty (38), LeavesQty (151), CumQty (14), AvgPx (6), OrdType (40), etc.), ApplID (1180)and TransactTime (60). The reason for the rejection will be specified in the field OrdRejReason (103).

A Business Message Reject will be sent to reject a Mass Order Status Request if the server is unable to process it in the unlikely event of a system outage. If the outage occurs before the server has sent all of the messages in response to a Mass Order Status Request, it will terminate the open order download. An Execution Report will be sent if the open order download is terminated. It will include the MassStatusReqID (584) of the request, an ExecID (17) of "0", an ExecType (150) of Order Status (I) and an OrdStatus (39) of Rejected (8).

If a client specifies an instrument as well as a segment in the Mass Order Status Request, results will be given according to the value specified for the MassStatusReqType (585) field. I.e.

- If MassStatusReqType (585) = Open orders for specified instrument and PartyID combination (1), statuses of the orders belonging to the specified instrument will be given.

[1] Active market interests include new/partially filled orders, market interests that are in the pending queue due to risk management controls, un-elected stop and stop limit orders and parked orders

- If MassStatusReqType (585) = Open orders for specified segment and PartyID combination (100), statuses of the orders belonging to the specified segment will be given.
- If MassStatusReqType (585) = Open orders for specified PartyID (8), statuses of all orders belonging to the specified user will be given.

3.5 Risk Management Related

3.5.1 Communicating Member Status Changes and Warnings

The Exchange monitors risk for members, users and CP code.

The Exchange communicates the status updates and warnings issued on the entities via the Party Risk Limits Update Report. The message will be routed to the corresponding parties via Drop Copy FIX Gateway.

(Note: However, user status updates will be received directly through FIX trading gateway by the user. Rest all status updates, such as member square off, CP code suspension etc. will come via drop copy gateway**)**

3.5.2 Communicating Member Statistics

The Exchange also maintains the total collateral, initial margin and MTM margin of trading members and users. Furthermore, it maintains the MTM PL for trading members. Such statistics can change on every trade executed for a member.

The Exchange communicates such statistics via the Party Risk Limit Report. It will be routed to the required entities via Drop Copy FIX Gateway.

3.6 Execution Reports

The Execution Report message is used to communicate many different events to clients. The events are differentiated by the value in the ExecType (150) field as outlined below.

Exec Type	Usage	Ord Status
0	**Order Accepted** Indicates that a new order has been accepted.	0
A	**Order Pending** Indicates that a new order has been forwarded to the risk management system for validation.	A
8	**Order Rejected** Indicates that an order has been rejected. The reason for the rejection is specified in the field OrdRejReason (103).	8
F	**Order Executed** Indicates that an order has been partially or fully filled. The execution details (e.g. price and quantity) are specified.	1, 2
C	**Order Expired** This may indicate one of the following: • An order has expired in terms of its time qualifier. • An order has expired due to an execution limit breach. • When any remaining orders (except GTC and GTD) are expired at market close. • When orders are expired based on the cancel on disconnect/log out feature. • Orders expired due to triggering of circuit breakers	C
4	**Order Cancelled** Indicates that an order cancel request has been accepted and successfully processed. This message is also sent if the order is cancelled by market operations. In such a case the message will include an ExecRestatementReason (378) of Market Option (8). It will not include an OrigClOrdID (41).	4
5	**Order Cancel/Replaced** Indicates that an order cancel/replace request has been accepted and successfully processed.	0, 1
L	**Triggered** Indicates that a parked ATC, GFA or stop order has been activated and is available for execution.	0, 1, A
9	**Suspended** Indicates that an order that was active and was available for execution has been parked and is no longer available for execution.	0, 1, A
D	**Order Cancel/Replace by Market Operations or has been impacted by an Automatic Event** Indicates that an order has been amended by market operations or automatic event in the system. The message will include an ExecRestatementReason (378) of Market Option (8), when amended by market operations. It will include an ExecRestatementReason (378) of OrderRe-Priced (3) when automatically re-priced by the system.	0, 1

E	**Order Cancel/Replace Pending** Indicates that an order cancel/replace request has been forwarded to the risk management system for validation.	E
H	**Trade Cancel** Indicates that an execution has been cancelled. An ExecRefID (19) to identify the execution being cancelled will be included.	0, 1, E

Exec Type	Usage	Ord Status
G	**Trade Correct** Indicates that an execution has been corrected. The message will include an ExecRefID (19) to identify the execution being corrected and the updated execution details (e.g. price and quantity).	1, 2, E
I	**Order Status Response** Indicates the current status of an order.	0, 1, A, E
I	**Order Status Reject** Indicates that an order mass status request has been rejected.	8

3.6.1 Order Status

As specified in the FIX protocol, the OrdStatus (39) field of an Execution Report is used to convey the current state of an order. If an order simultaneously exists in more than one order state, the value with highest precedence is reported as the OrdStatus (39). The relevant order statuses are given below from the highest to lowest precedence.

Value	Meaning
E	Pending Replace
2	Filled
9	Suspended
4	Cancelled
C	Expired
1	Partially Filled
0	New
8	Rejected
A	Pending New

3.6.2 Order and Execution Identifiers

3.6.2.1 Client Order IDs

In the case of orders, the ClOrdID (11) included in each Execution Report will be that specified when the order was submitted. An order's ClOrdID (11) will be updated each time an Order Cancel/Replace Request or an Order Cancel Request is accepted.

The ClOrdID(11) is always expected to be present in the Execution Report unless the order happens to be an Implied order that has an Owner ID assigned.

3.6.2.2 Order IDs

The server uses the OrderID (37) field to affix the order identification numbers of the matching system. Order IDs are unique across trading days.

In terms of the FIX protocol, unlike ClOrdID (11) which requires a chaining through Cancel/Replace Requests and Cancel Requests, the OrderID (37) of an order will remain constant throughout its life.

3.6.2.3 Execution IDs

The server uses the ExecID (17) field to affix the execution identification numbers of the matching system. Execution IDs are unique across trading days.

3.6.2.4 Trade IDs

The server uses the TrdMatchID (880) field to affix a unique identifier for each trade. This identifier is referenced in the Trade Capture Reports published by the post trade system and the trade messages of the FAST market data feed. Trade IDs are unique across trading days. An Execution Report published to notify a client of a trade cancellation or correction includes the TradeID of the trade.The field Secondary Trade ID (1040) is a numeric code generated by the system for each trade. . This identifier is referenced in the Trade Capture Reports published by the post trade system and the trade messages of the FAST market data feed. The Secondary Trade IDs will be unique only for the current trading day. An Execution Report published to notify a client of a trade cancellation or correction includes the Secondary TradeID of the original trade.

3.6.3 Strategies

The values specified in the fields Price (44), StopPx (99) and LastPx (31) for Execution Reports relating to multi-legged instruments may contain negative prices.

If an order for a strategy receives an execution, it will receive an Execution Report for the multi-legged instrument as well as separate Execution Reports for each of the associated leg instruments. The field MultiLegReportingType (442) should be used to determine whether a particular Execution Report relates to the multi-legged instrument or a leg instrument.

While the ClOrdID (11) of an Execution Report for a leg trade will be the same as the ClOrdID (11) of the order for the multi-legged instrument, the OrderID (37) will not. The SecondaryOrderID (198) for a leg trade will contain the OrderID (37) of the associated order for the multi-legged instrument.

If an order considered as a multi-leg order receives executions, rejection or expiration on the two leg instruments, the Execution Report generated will include a Multi-LegReportingType (442) of Multi-leg order on leg instruments (9).

3.6.4 Instrument Identification

Instruments may be identified using either the Symbol (55) or SecurityID (48) field.

The instrument identification included in an Execution Report will be that specified in the order the message relates to.

3.6.5 Party Identification

ID	Description	Relevant FIX Tags
Trading Mnemonic	Identifier of the trading mnemonic the message is submitted under. Trading privileges are assigned at the level of trading mnemonics.	PartyRole (452) = 53 PartyID (448)
Executing Firm	Identifier of the trading firm the interest is submitted under.	PartyRole (452) = 1 PartyID (448)

Clearing Firm	Identifier of the firm through which the trade should clear.	PartyRole (452) = 4 PartyID (448)
Contra Firm	Identifier of the counterparty trading firm in the case of a EFP trade.	PartyRole (452) = 17 PartyID (448)
NNF ID	The NNF ID specified on a client initiated message. A reply to a client initiated message will contain the original NNF ID	PartyRole (452) = Order Entry Operator ID (44)
Investor Account	Identifier of the investor account on whose behalf the order is submitted.	Account (1), AccountType (581) = Client (1)
House Account	Identifier of the of the firm account on whose behalf the order was submitted.	AccountType (581) = House (3)
Alloc Account	Identifier of the sub-account mnemonic through which the CP will be submitted.	AllocAccount (79)

3.7 Timestamps and Dates

The timestamps SendingTime (52), OrigSendingTime (122) and TransactTime (60) are in UTC and in the YYYYMMDD-HH:MM:SS.sss format.

All dates (i.e. MaturityDate (541) and ExpireDate (432)) are in the YYYYMMDD format and specified in the local date for the server (i.e. not in UTC)).

3.8 Repeating Groups (Components/Component Block)

If a repeating group is used in a message, the NoXXX field (for example NoPartyIDs field in the trading party repeating group) should be specified first before the repeating group starts. This is applicable for both the messages generated by the client and the server.

The messages generated by the server will have the fields within a repeating group in order. The messages generated by a client should have the first field in a repeating group in order. If the first field in a repeating group is in order, a message generated by a client will be accepted; else the message will be rejected.

If the same FIX tag is repeated with different values in the client generated message outside of a repeating group, the server takes the value in the last tag. The server will not reject such messages.

However, if a client initiated Logon message contains repeated tags; the server may not acknowledge the login request and will not send any reply. If client initiated other administrative messages or application messages contain repeated tags outside component blocks, such requests will be rejected by the server.

4 CONNECTIVITY

4.1 CompIDs

The CompID of each client must be registered with NCDEX before FIX communications can begin. A single client may have multiple connections to the server (i.e. multiple FIX sessions, each with its own CompID).

The CompID of the server will be DC Gateway Comp ID. The messages sent to the server should contain the CompID assigned to the client in the field SenderCompID (49) and DC Gateway Comp ID in the field TargetCompID (56). The messages sent from the server to the client will contain DC Gateway Comp ID in the field SenderCompID (49) and the CompID assigned to the client in the field TargetCompID (56).

4.1.1 Passwords

Each new CompID will be assigned a password on registration. Clients are strongly encouraged to change the password to one of their choosing via the Logon message. The acceptance of a login request indicates that the new password has been accepted. The new password will, if accepted, be effective for subsequent logins.

In terms of the password policy of NCDEX, the password of each CompID should be changed at least every *<30>* days. If not, the password will expire and the client will be unable to login to the server. In such a case, the client should contact Exchange to have its password reset. The SessionStatus (1409) of the server's Logon message will be Password Due to Expire (2) for the last *<5>* days of a password's validity period.

4.1.2 NNF ID

Clients connecting to NCDEX trading systems via FIX will have connections from either a NEAT front end system or any other front end.

If the request message contains a PartyRole (452) of Order Entry Operator (44) then the reply back from the server, in Execution Report message, will include the NNF ID specified in Order Entry Operator (44)

4.2 Production IP Addresses and Ports

The IP address of each client must be registered with NCDEX before FIX communications can begin.

4.3 Failover and Recovery

The system has been designed with fault tolerance and disaster recovery technology that ensures that trading should continue in the unlikely event of a process or site outage.

If the client is unexpectedly disconnected from the server, it should attempt to re-connect to primary site within a few seconds. The client should only attempt to connect to the secondary IP address and port if so requested by NCDEX.

If a service interruption (e.g. due to Order Cache failing over to its mirror or both Order Cache processes fail) occurs in the Drop Copy Gateway while it is servicing an Order Mass Status Request, the gateway will send an unsolicited Execution Report with a 'Rejected' state. It would include the MassStatusReqID (584) of the request, an ExecID (17) of '0' an ExecType (150) of Order Status (I) an OrdStatus (39) of Rejected (8) and an OrdRejReason (103) of '10005' – Application Unavailable. When the client receives this, he is expected to try and re-request.

If a client requests an open order book download when the service is unavailable, (e.g. both Order Cache instances down) the request will be rejected with a business reject, with reason 4 – Application Unavailable.

In the unlikely event of a site outage disaster on the NCDEX system, all orders will be cancelled and all unicast and multicast connectivity will be unavailable until the secondary site is invoked.

5 FIX CONNECTIONS AND SESSIONS

5.1 Establishing a FIX Connection

FIX connections and sessions between the client and server are maintained as specified in the FIX protocol.

Each client will use the assigned IP address and port to establish a TCP/IP session with the server. The client will initiate a FIX session at the start of each trading day by sending the Logon message. The client will identify itself using the SenderCompID (49) field. The server will validate the CompID, password and IP address of the client.

The server will break the TCP/IP connection if messages are received before the exchange of Logons.

System can be configured in such a way that the test request at logon is either disabled or enabled[2]. A test request will not be sent along with the logon reply if the test request switch is set to its default 'Off' mode. Depending on the System Configuration, the client's logon message will be responded in two ways:

If during a logon of a SenderCompID, the server receives a second connection attempt via the same TCP/IP connection while a valid FIX session is already underway for that same SenderCompID, the server will immediately break the TCP/IP connection with the client without sending any messages. If the server receives another connection attempt from the same SenderCompID, while a session is already established, the connection attempt will be rejected via a Reject message without breaking the existing TCP/IP connection with the client. The server will increment the next inbound message sequence number expected from the client as well as its own outbound message sequence number.

5.1.1 Test Request at logon Enabled

Once the client is authenticated, the server will respond with a Logon message, followed by a Test Request. The Logon message will confirm the logon status and the Test Request's purpose is to sync the Sequence numbers before sending any Missed Messages if any. The SessionStatus (1409) of this message will be Session Active (0). If the client's Logon message included the field NewPassword (925) and the client is authenticated, the SessionStatus (1409) of the Logon sent by the server will be Session Active (0). The client must wait for the server's Logon before sending additional messages. If additional messages are received from the client before the exchange of Logon messages, the TCP/IP connection with the client will be disconnected.

A successful logon response will be followed by a Test Request message. If the client responds to the Test Request with a Heartbeat message containing the appropriate Test Request ID and message sequence number, the server can start transmitting the missed messages or new messages in the Gateway.

The client would not receive any responses to application messages sent until sequence numbers are synchronized by responding to the Test Request via Heartbeat or Resend Request message. However, these messages will be processed by the system.

If the client ignores the Test Request because the sequence number in the message is higher than the expected sequence number, the Client is expected to send a Resend Request asking for the missed messages. After responding to the Resend Request the FIX Gateway would send another Test Request to make sure both the client and server is in sync before sending out any missed or new application messages.

If the client sends a Resend Request before the FIX Gateway send a Test Request, then the FIX Gateway will serve the Resend Request first. After responding to the Resend Request the FIX Gateway would send a Test Request to make sure both the client and server are in sync before sending out any missed or new application messages.

[2] This is done via the process configuration CLIENT SESSION CONFIRMATION.

When the client sends a logon with a sequence number higher than expected by the FIX Gateway, the FIX gateway will send a Resend Request followed by the Test Request. The client is expected to serve the Resend Request and respond to the Server's Test Request via a Heartbeat message.

If a logon attempt fails because of an invalid SenderCompID, invalid TargetCompID or invalid IP address, invalid password or not having the appropriate privileges to login to the gateway the server will break the TCP/IP connection with the client without sending a Logout or Reject message. If the server receives a second connection attempt from the same TCP/IP while a valid FIX session is already underway for that same SenderCompID the system will reject the second attempt with a reject message while maintaining the original connection. If the server receives a second connection attempt from a different TCP/IP while a valid FIX session is already underway for that same SenderCompID, the server will break the TCP/IP connection for the second attempt without sending a Logout or Reject message. As the logon attempt failed, the server will not increment the next inbound message sequence number expected from the client.

If a logon attempt fails because of an expired password a locked CompID or if logins are not currently permitted, the server will send a Logout message and then break the TCP/IP connection with the client. The server will increment the next inbound message sequence number expected from the client as well as its own outbound message sequence number.

If a logon attempt fails because of a session level failure (e.g. due to invalid EncryptMethod or DefaultApplVerID…etc) the inbound sequence number and the outbound sequence number both will not be incremented**.** In this scenario the message sequence number 1 will be sent with the **Error! Reference source not found.** message.

However if a session level failure occurs due to a message sent by a client which contains a sequence number that is less than what is expected and the PossDupFlag (43) is not set to 'Y', then the server will send a Logout message and terminate the FIX connection. In this scenario the inbound sequence number will not be incremented but the outbound sequence number will be incremented.

If during a logon of a SenderCompID, the server receives a second connection attempt via the same TCP/IP connection while a valid FIX session is already underway for that same SenderCompID, the server will send a Reject message and then break the TCP/IP connection with the client. The server will increment the next inbound message sequence number expected from the client as well as its own outbound message sequence number

The outbound sequence number will be 1 only if ResetSeqNumFlag(141)=Y in the Logon message sent by the client. If the ResetSeqNumFlag=N, there is no assurance that the MsgSeqNum(34) of the reply message to Logon will be 1.

5.2 Maintaining a FIX Session

5.2.1 Message Sequence Numbers

As outlined in the FIXT protocol, the client and server will each maintain a separate and independent set of incoming and outgoing message sequence numbers. Sequence numbers should be initialized to 1 (one) at the start of the FIX session and be incremented throughout the session.

Monitoring sequence numbers will enable parties to identify and react to missed messages and to gracefully synchronize applications when reconnecting during a FIX session.

If any message sent by the client contains a sequence number that is less than what is expected and the PossDupFlag (43) is not set to "Y", the server will send a Logout message and terminate the FIX connection. The Logout will contain the next expected sequence number in the Text (58) field.

A FIX session will not continue to the next trading day. The server will initialize its sequence numbers at the start of each day. The client is expected to employ the same logic.

5.2.2 Heartbeats

The client and server will use the Heartbeat message to exercise the communication line during periods of inactivity and to verify that the interfaces at each end are available. The heartbeat interval will be the HeartBtInt (108) specified in the client's Logon message.

The server will send a Heartbeat anytime it has not transmitted a message for the heartbeat interval. The client is expected to employ the same logic.

If the server detects inactivity for a period longer than the heartbeat interval plus a reasonable transmission time, it will send a Test Request message to force a Heartbeat from the client. If a response to the Test Request is not received by a reasonable transmission time, the server will send a Logout and break the TCP/IP connection. The client is expected to employ similar logic if inactivity is detected on the part of the server.

5.2.3 Increasing Expected Sequence Number

The client or server may use the Sequence Reset message in Gap Fill mode if it wishes to increase the expected incoming sequence number of the other party.

The client or server may also use the Sequence Reset message in Sequence Reset mode if it wishes to increase the expected incoming sequence number of the other party. The MsgSeqNum (34) in the header of such a message will be ignored. The Sequence Reset mode should only be used to recover from an emergency situation. It should not be relied upon as a regular practice.

5.3 Terminating a FIX Connection

The client is expected to terminate each FIX connection at the end of each trading day before the server shuts down. The client will terminate a connection by sending the Logout message. The server will respond with a Logout to confirm the termination. The client will then break the TCP/IP connection with the server. As recommended in the FIXT protocol, clients are advised to transmit a Test Request, to force a Heartbeat from the server, before initiating the logout process.

All open TCP/IP connections will be terminated by the server when it shuts down (a Logout will not be sent). Under exceptional circumstances the server may initiate the termination of a connection during the trading day by sending the Logout message. The server will terminate the TCP/IP connection (a Logout will not be sent) if the number of messages that are buffered for a client exceeds *<1,000>*.

If, during the exchange of Logout messages, the client or sever detects a sequence gap, it should send a Resend Request.

5.4 Re-Establishing a FIX Session

If a FIX connection is terminated during the trading day it may be re-established via an exchange of Logon messages. Once the FIX session is re-established, the message sequence numbers will continue from the last message successfully transmitted prior to the termination.

5.4.1 Test Request at logon Enabled

Once the client is authenticated, the server will respond with a Logon message, followed by a Test Request. The Logon message will confirm the logon status and the Test Request's purpose is to sync the Sequence numbers before sending any Missed Messages if any. The SessionStatus (1409) of this message will be Session Active (0). If the client's Logon message included the field NewPassword (925) and the client is authenticated, the SessionStatus (1409) of the Logon sent by the server will be Session Active (0).

If the client responds to the Test Request with a Heartbeat message containing the appropriate Test Request ID and message sequence number, the server can start transmitting the missed messages or new messages in the Gateway. The client would not receive any responses to application messages sent until sequence numbers are synchronized by responding to the Test Request via Heartbeat or Resend Request message. However, these messages will be processed by the system. If the client does not respond to the Test Request during the heartbeat interval, the gateway will disconnect the client.

If the client ignores the Test Request because the sequence number in the message is higher than the expected sequence number, the client is expected to send a Resend Request asking for the missed messages. After responding to the Resend Request the FIX Gateway would send another Test Request to make sure both the client and server is in sync before sending out any missed or new application messages.

If the client sends a Resend Request before the FIX Gateway send a Test Request, then the FIX Gateway will serve the Resend Request first. After responding to the Resend Request the FIX Gateway would send a Test Request to make sure both the client and server are in sync before sending out any missed or new application messages.

When the client sends a logon with a sequence number higher than expected by the FIX Gateway, the FIX gateway will send a Resend Request followed by the Test Request. The client is expected to serve the Resend Request and respond to the Server's Test Request via a Heartbeat message.

Once the FIX session is re-established successfully, the message sequence numbers will continue from the last message successfully transmitted prior to the termination.

5.4.2 Resetting Sequence Numbers: Starting a New FIX Session

5.4.2.1 Reset Initiated by the Client

If the client requires both parties to initialize (i.e. reset to 1) sequence numbers, it may use the ResetSeqNumFlag (141) field of the Logon message. The server will respond with a Logon with the ResetSeqNumFlag (141) field set to "Y" to confirm the initialization of sequence numbers. In such cases, if the MsgSeqNo (34) of the Logon message is not reset to 1, the server will break the TCP/IP connection after sending a Logout. Such a message will include a SessionStatus(1409) of Logout due to session level failure(101) and an indication of the same in the Text(58) field.

A client may also manually inform market operations that it would like the server to initialize its sequence numbers prior to the client's next login attempt.

These features are intended to help a client manage an emergency situation. Initializing sequence numbers on a re-login should not be relied upon as a regular practice.

5.4.2.2 Reset Initiated by the Server

The system has been designed with fault tolerance and disaster recovery technology that should ensure that the server retains its incoming and outgoing message sequence numbers for each client in the unlikely event of an outage.

However, clients are required to support a manual request by NCDEX to initialize sequence numbers prior to the next login attempt.

6 RECOVERY

6.1 Resend Requests

The client may use the Resend Request message to recover any lost messages. As outlined in the FIXT protocol, this message may be used in one of three modes:

(i) To request a single message. The BeginSeqNo (7) and EndSeqNo (16) should be the same.

(ii) To request a specific range of messages. The BeginSeqNo (7) should be the first message of the range and the EndSeqNo (16) should be the last of the range.

(iii) To request all messages after a particular message. The BeginSeqNo (7) should be the sequence number immediately after that of the last processed message and the EndSeqNo (16) should be zero (0).

The server caches the last *<1,000>* messages transmitted to each CompID. Clients are unable to use a Resend Request to recover messages not in the server's cache. If the client requests for a range of messages that have sequence numbers falling outside the cache size, a Sequence Reset message in Gap Fill mode will be sent for the missing messages and will send the available messages as per the request after that.

6.2 Possible Duplicates

The server handles possible duplicates according to the FIX protocol. The client and server will use the PossDupFlag (43) field to indicate that a message may have been previously transmitted with the same MsgSeqNum (34).

6.3 Possible Resends

The server does not handle possible resends for client-initiated messages and ignores the value in the PossResend (97) field of such messages.

The server may, in the circumstances outlined in Section 6.4 and, use the PossResend (97) field to indicate that an Execution Report may have already been sent under a different MsgSeqNum (34). The client should validate the ExecID (17) of such a message against those of Execution Reports already received during the current trading day.

If an Execution Report with same ExecID (17) had been processed, the resent message should be ignored. If the same ExecID (17) had not been processed, the Execution Report should be processed.

6.4 Transmission of Missed Messages

The Execution Reports generated during a period when a client is disconnected from the server will be sent to the client when it next reconnects. In the unlikely event the disconnection was due to an outage of the server, all such messages will include a PossResend (97) of "Y".

If a client is disconnected while an Open Order Download request submitted by the client is being processed by the server, requested Execution Reports will not be transmitted to the client when it next reconnects. In such situations the client is expected to send the Open Order Download request to the server once the connection is re-established.

7 MESSAGE FORMATS

This section provides details on the header and trailer, the seven administrative messages and six application messages utilized by the server. Client-initiated messages not included in this section are rejected by the server via a Reject or Business Message Reject.

7.1 Supported Message Types

7.1.1 Administrative Messages

All administrative messages may be initiated by either the client or the server.

Message	MsgType	Usage
Logon	A	Allows the client and server to establish a FIX session.
Logout	5	Allows the client and server to terminate a FIX session.
Heartbeat	0	Allows the client and server to exercise the communication line during periods of inactivity and verify that the interfaces at each end are available.
Test Request	1	Allows the client or server to request a response from the other party if inactivity is detected.
Resend Request	2	Allows for the recovery of messages lost during a malfunction of the communications layers.
Reject	3	Used to reject a message that does not comply with FIXT.
Sequence Reset	4	Allows the client or server to increase the expected incoming sequence number of the other party.

7.1.2 Application Messages

7.1.2.1 Client-Initiated

Message	MsgType	Usage
Order Mass Status Request	AF	Allows the client to request the status of all active orders for a particular trading mnemonic.

7.1.2.2 Server-Initiated

Message	MsgType	Usage
Execution Report	8	Indicates one of the following: (i) Order accepted (ii) Order pending (iii) Order rejected (iv) Order executed (v) Order expired (vi) Order cancelled (vii) Order cancel/replaced (viii) Order cancel/replace pending (ix) Trade cancellation or correction (x) Order status (xi) Order mass status request rejected

Message	MsgType	Usage
Order Mass Cancel Report	r	Indicates one of the following: (i) Mass order cancel request accepted. (ii) Mass order cancel request rejected.
Party Risk Limits Update Report	CR	Initiated by the server indicating a client of a Suspension, Square-off, Warning or Re-instate
Party Risk Limits Report	CM	Initiated by the server indicating Collateral, Margins and MTM related statistics
Business Message Reject	j	Indicates that an application message could not be processed.

7.2 Variations from the FIX Protocol

The server conforms to the FIX protocol except as follows:

(i) The Execution Report message includes the custom fields OrderBook (30001) and Order Source (30004). The data types of these fields are Int (i.e. integer) and String respectively.

(ii) The Logon (A) message includes the ConnectionType (22001) and WorkstationType (22001). The data types of the fields are ‘Enums’.

7.3 Message Header and Trailer

7.3.1 Message Header

Tag	Field Name	Req	Length	Description
8	BeginString	Y	N/A	FIXT.1.1
9	BodyLength	Y	N/A	Number of characters after this field up to and including the delimiter immediately preceding the CheckSum.
35	MsgType	Y	I32	Message type.
49	SenderCompID	Y	30	CompID of the party sending the message.
56	TargetCompID	Y	30	CompID of the party the message is sent to.
115	OnBehalfOfCompID	N	30	Required for server-initiated application messages. This will be the CompID of the connection that originated the order referenced in the message being drop copied.
34	MsgSeqNum	Y	I32	Sequence number of the message.
43	PossDupFlag	N	1	Whether the message was previously transmitted under the same MsgSeqNum (34). Absence of this field is interpreted as Original Transmission (N). **Value** **Meaning** Y Possible Duplicate N Original Transmission
97	PossResend	N	1	Whether the message was previously transmitted under a different MsgSeqNum (34). Absence of this field is interpreted as Original Transmission (N). **Value** **Meaning** Y Possible Resend N Original Transmission
52	SendingTime	Y	30	Time the message was transmitted.
122	OrigSendingTime	N	30	Time the message was originally transmitted. If the original time is not available, this should be the same value as SendingTime (52). Required if PossDupFlag (43) is Possible Duplicate (Y).
1128	ApplVerID	N	1	Version of FIX used in the message. Required if the message is generated by the server. **Value** **Meaning** 9 FIX50SP2

7.3.2 Message Trailer

Tag	Field Name	Req	Length	Description
10	CheckSum	Y	30	

7.4 Administrative Messages

7.4.1 Logon

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	A = Logon
Message Body				
98	EncryptMethod	Y	1	Method of encryption. **Value** **Meaning** 0 None
108	HeartBtInt	Y	UI32	Indicates the heartbeat interval in seconds.
141	ResetSeqNum Flag	N	1	Indicates whether the client and server should reset sequence numbers. Absence of this field is interpreted as Do Not Reset Sequence Numbers (N). **Value** **Meaning** Y Reset Sequence Numbers N Do Not Reset Sequence Numbers
554	Password	N	30	Password assigned to the CompID. Required if the message is generated by the client.
925	NewPassword	N	30	New password for the CompID.
1409	SessionStatus	N	UI32	Status of the FIX session. Required if the message is generated by the server. **Value** **Meaning** 0 Session Active 2 Password Due to Expire
1137	DefaultApplVerID	Y	1	Default version of FIX messages used in this session. **Value** **Meaning** 9 FIX50SP2
Standard Trailer				

7.4.2 Logout

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	5 = Logout
Message Body				
1409	SessionStatus	N	UI32	Status of the FIX session. Required if the message is generated by the server. **Value** **Meaning** 4 Session logout complete 6 Account locked 7 Logons are not allowed at this time 8 Password expired 100 Other 101 Logout due to session level failure 102 Logout by market operations
58	Text	N	60	Text specifying reason for the logout. Required if SessionStatus (1409) is Other (100).
Standard Trailer				

7.4.3 Heartbeat

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	0 = Heartbeat
Message Body				
112	TestReqID	N	N/A	Required if the heartbeat is a response to a Test Request. The value in this field should echo the TestReqID (112) received in the Test Request.
Standard Trailer				

7.4.4 Test Request

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	1 = Test Request
Message Body				
112	TestReqID	Y	N/A	Identifier for the request.
Standard Trailer				

7.4.5 Resend Request

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	2 = Resend Request
Message Body				
7	BeginSeqNo	Y	I32	Sequence number of first message in range.
16	EndSeqNo	Y	I32	Sequence number of last message in range.
Standard Trailer				

7.4.6 Reject

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	3 = Reject
				Message Body
45	RefSeqNum	Y	I32	MsgSeqNum (34) of the rejected message.
372	RefMsgType	N	2	MsgType (35) of the rejected message.
371	RefTagID	N	I32	If a message is rejected due to an issue with a particular field its tag number will be indicated.
373	SessionReject Reason	N	I32	Code specifying the reason for the reject. Please refer to Section 8.1 for a list of reject codes.
58	Text	N	60	Text specifying the reason for the rejection.
Standard Trailer				

7.4.7 Sequence Reset

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	4 = Sequence Reset
Message Body				
36	NewSeqNo	Y	I32	Sequence number of the next message to be transmitted.
123	GapFillFlag	N	1	Mode in which the message is being used. Absence of this field is interpreted as Sequence Reset (N). **Value** **Meaning** Y Gap Fill N Sequence Reset
Standard Trailer				

7.5 Application Messages (Client-Initiated)

7.5.1 Order Mass Status Request

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	AF = Order Mass Status Request
Message Body				
584	MassStatusReqID	Y	25	Client specified identifier of the mass status request.
585	MassStatusReqType	Y	I32	Type of mass status request. **Value** **Meaning** 1 Open orders for specified instrument and PartyID combination 8 Open orders for specified PartyID 100 Open orders for specified segment and PartyID combination
453	NoPartyIDs	Y	I32	Number of party identifiers. The value in this field should always be “1”.
➡	448 PartyID	Y	30	Identifier of the trading mnemonic.
➡	447 PartyID Source	Y	1	**Value** **Meaning** D Proprietary/Custom Code
➡	452 Party Role	Y	I32	Role of the PartyID (448). **Value** **Meaning** 1 Executing Firm 53 Trading Mnemonic
55	Symbol	N	30	Identifier of the instrument. Required if MassStatusReq Type (585) is “1”.
1300	MarketSegmentID	N	30	Identifier of the segment. Required if MassStatusReq Type (585) is “100”.
Standard Trailer				

7.6 Application Messages (Server-Initiated)

7.6.1 Execution Report

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	8 = Execution Report
Message Body				
1180	ApplID	Y	I32	Identity of the partition that generated the message.
17	ExecID	Y	15	Server specified identifier of the message. Will be “0” if ExecType (150) is Order Status (I).
11	ClOrdID	Y	20	Client specified identifier of the order.
41	OrigClOrdID	N	25	OrigClOrdID (41), if any that was submitted with the order cancel or cancel/replace request.
37	OrderID	Y	12	Server specified identifier of the order.
584	MassStatusReqID	N	25	Client specified identifier of the mass status request. Required is the message in sent in response to such a request.
912	LastRpt Requested	N	1	Indicates the last message sent in response to a mass order status request. **Value** **Meaning** Y Last Message
442	MultiLegReportingType	N	I32	Type of trade. Absence of this field is interpreted as Trade of Single Instrument (1). **Value** **Meaning** 1 Trade of Single Instrument 2 Leg Trade of a Multi-Leg Instrument Trade 3 Trade of a Multi-Leg Instrument 9 Multi-leg order on leg instruments
198	Secondary OrderID	N	21	Server specified identifier of the order side for the multi-legged instrument. Required if MultiLegReportingType (442) is Leg Trade of a Multi-Leg Instrument Trade (2).

150	ExecType	Y	1	Reason the execution report was generated. **Value** **Meaning** 0 New 4 Cancelled 5 Replaced 8 Rejected 9 Suspended A Pending New C Expired D Restated E Pending Replace F Trade G Trade Correct H Trade Cancel I Order Status L Triggered
880	TrdMatchID	N	30	Identifier of the trade. This will be a 62 base encoded value in ASCII format. Required if ExecType (150) is Trade (F), Trade Correct (G) or Trade Cancel (H).
1040	Secondary Trade ID	Y	7	Numeric trade ID assigned for the trade.
19	ExecRefID	N	25	Reference to the execution being cancelled or corrected. Required if ExecType (150) is Trade Cancel (H) or Trade Correct (G).
378	Exec Restatement Reason	N	2	Reason the order was restated or cancelled by Market Operations. Required if ExecType (150) is Restated (D) or if the execution report is sent for an unsolicited cancellation. **Value** **Meaning** 3 Order Re-Priced 8 Market Option

39	OrdStatus	Y	1	Current status of the order. **Value** **Meaning** 0 New 1 Partially Filled 2 Filled 4 Cancelled 8 Rejected 9 Suspended A Pending New C Expired E Pending Replace
636	Working Indicator	N	1	Whether the order is currently being worked. **Value** **Meaning** N Order is Not in a Working State (Order has been accepted and is in an unelected state/parked queue.) Y Order is Being Worked (Order has been elected and has been added to the normal order book)
103	OrdRejReason	N	I32	Code specifying the reason for the reject. Please refer to Section 0 for a list of reject codes. Required if ExecType (150) is Rejected (8).
58	Text	N	60	Text specifying the reason for the rejection, cancellation or expiration
32	LastQty	N	20,8	Quantity executed in this fill. Required if ExecType (150) is Trade (F) or Trade Correct (G).
110	MinQty	N	30,8	Minimum quantity that must be filled.
31	LastPx	N	20,8	Price of this fill. Required if ExecType (150) is Trade (F) or Trade Correct (G).
151	LeavesQty	Y	30,8	Quantity available for further execution. Will be “0” if OrdStatus (39) is Filled (2), Cancelled (4), Rejected (8) or Expired (C).
14	CumQty	Y	30,8	Total cumulative quantity filled.
6	AvgPx	N	30,8	Average price of all fills for the order side.

55	Symbol		Y	30	Identifier of the instrument.
453	NoPartyIDs		N		Number of party identifiers.
➡	448	PartyID	N	30	Identifier of the party.
➡	447	PartyID Source	N	1	Required if PartyID (448) is specified. **Value** **Meaning** D Proprietary/Custom Code
➡	452	Party Role	N	I32	Role of the PartyID (448). Required if PartyID (448) is specified. **Value** **Meaning** 1 Executing Firm 4 Clearing Firm 17 Contra Firm 44 Order Entry Operator ID 53 Trading Mnemonic
1	Account		N	32	Identifier of the investor account on whose behalf the interest is submitted.
581	AccountType		N	I32	Type of the investor account. **Value** **Meaning** 1 Client 3 House
79	AllocAccount		N	30	Sub-account mnemonic. Contains the CP Code of the Client ID.
40	OrdType		Y	1	Type of the order. **Value** **Meaning** 1 Market 2 Limit 3 Stop 4 Stop Limit K Market to Limit J Market If Touched
59	TimeInForce		N	I32	Time qualifier of the order. Absence of this field is interpreted as Day (0). **Value** **Meaning** 0 Day 1 Good Till Cancel (GTC) 3 Immediate or Cancel (IOC) 6 Good Till Date (GTD)
432	ExpireDate		N	8	Date the order expires. Required if TimeInForce (59) is GTD (6).

336	TradingSessionID	N	30	Session the order is valid for **Value** **Meaning** a Closing Price Cross
18	ExecInst	N	15	Execution instructions submitted with the order. Space separated field. **Value** **Meaning** n Do Not Cancel on Disconnect/L
54	Side	Y	1	Side of the order that was executed. **Value** **Meaning** 1 Buy 2 Sell
38	OrderQty	Y	30,8	Total order quantity. Order Quantity = Leaves Quantity + Cumulative Executed Quantity
1138	DisplayQty	N	30,8	Quantity currently displayed in the order book.
44	Price	N	30,8	Limit price. Required if OrderType (40) is Limit (2) or Stop Limit (4).
99	StopPx	N	30,8	Stop price/Trigger price. Required if OrderType (40) is Stop (3), Stop Limit (4) or Market If Touched (J).
583	ClOrdLinkID	N	20	Deal identifier of an EFP trade.
30004	OrderSource	N	24	Value submitted with the order, cancel request or amend request.
60	TransactTime	Y	21	Time the transaction represented by the Execution Report occurred.
30001	OrderBook	Y	I32	**Value** **Meaning** 1 Regular 4 EFP Trade
22009	InternalRefFlag	N	15	Internal Reference Flag
32022	LastOptPx	N	30,8	Converted price of the executed volatility of the options instrument.
1188	Volatility	N	30,8	Converted volatility of the executed price of the options instrument.
Standard Trailer				

7.6.2 Business Message Reject

Tag	Field Name	Req	Length	Description
Standard Header				
35	MsgType	Y	I32	j = Business Message Reject
Message Body				
45	RefSeqNum	Y	I32	MsgSeqNum (34) of the rejected message.
371	RefTagID	N	I32	If a Order Mass Status Request is rejected due to an issue with a particular field its tag number will be indicated.
372	RefMsgType	Y	2	MsgType (35) of the rejected message.
380	BusinessReject Reason	Y	I32	Code specifying the reason for the rejection. Please refer to Section 8.3 for a list of reject codes.
58	Text	N	60	Text specifying the reason for the rejection.
Standard Trailer				

7.6.3 Order Mass Cancel Report

Tag	Field Name	Req	Description	
Standard Header				
35	MsgType	Y	I32	r = Order Mass Cancel Report
Message Body				
1180	ApplID	Y	I32	Identity of the partition.
1369	MassActionReportID	Y	N/A	Server specified identifier of the message.
11	ClOrdID	Y	20	When mass cancellation is performed due to a partition failover, this will be the same value as MassActionReportID (1369).
530	MassCancelRequestType	Y	1	This field will be ‘z’ to indicate ‘Partition Suspended’

531	MassCancel Response	Y	1	Action taken by server. **Value** **Meaning** 0 Mass Cancel Request Rejected 1 Cancelled All Orders for Instrument 2 Cancelled All Orders for Underlying 7 Cancelled All Orders 9 Cancelled All Orders for Segment z Cancelled All Orders for Partition
532	MassCancelReject Reason	N	1	Code specifying the reason for the rejection. Required if MassCancelResponse (531) is Mass Cancel Request Rejected (0).
30001	OrderBook	N	I32	Identifier of the order book. Absence of this field is interpreted as Regular (1).OrderBook (30001) is not sent when mass cancelled due to a partition failover **Value** **Meaning** 1 Regular 4 EFP Trade
Standard Trailer				

7.6.4 Party Risk Limits Update Report

Tag	Field Name			Req	Len gth	Description
Standard Header						
35	MsgType			Y	I32	CR = Party Risk Limits Update Report
Message Body						
1667	RiskLimitReportID			Y	20	The unique identifier of the Party Risk Limits Update Report message.
79	AllocAccount			N	30	Sub-account mnemonic. The CP Code of the Client ID will be mentioned here.
<Party Risk Limits Group>						
1677	NoPartyRiskLimits			N	I32	The number of party risk limits. If specified, the value should always be 1.
➡	1324	ListUpdateAction		N	2	The source of the PartyID value. **Value Meaning** S Snapshot Required if NoPartyRiskLimits (1677) > 0.
	<Parties Group>					
➡	1671	NoPartyDetails		N	I32	Number of Party IDs. When PartyDetailRole (1693) is Client ID (3) with Executing Firm (1), value should be 2, in all other scenarios the value should be 1.
➡	➡	1691	PartyDetailID	N	30	The identifier of the party. Required if NoPartyDetails (1671) > 0.
➡	➡	1692	PartyDetailIDSource	N	1	The source of the PartyID value. Value Meaning D Proprietary/Custom Required if NoPartyDetails (1671) > 0.

➡	➡	1693	PartyDetailRole		N	I32	**Value** **Meaning** 1 Executing Firm 3 Client ID 4 Clearing Firm 53 Trader Mnemonic Required if NoPartyDetails (1671) > 0.
	<Risk Limits Group>						
➡	1669	NoRiskLimits			N	1	The number of risk limits. If specified, the value should always be 1.
➡	➡	1529	NoRiskLimitTypes		N	1	The number of risk limit types. If specified, the value should always be 1. Required if NoRiskLimits (1669) > 0.
➡	➡	➡	1530	RiskLimitType	N	I32	The type of risk limit. **Value** **Meaning** 8 Total margin 1000 Position Required if NoRiskLimitTypes (1592) > 0.
➡	➡	➡	1767	RiskLimitAction	N	I32	The action to be taken due to the update. **Value** **Meaning** 4 Warning 5 Square-Off 10 Suspend 100 Re-instate Required if NoRiskLimitTypes (1529) > 0.
➡	➡	➡	32033	RiskReductionMode	N	I32	Communication of whether the RRM applies or not. **Value** **Meaning** 0 Off 1 On
➡	➡	➡	1766	RiskLimitUtilizationAmount	N	30,8	Absolute amount of utilization of a party's RiskLimitType (1530) specified.

➡	➡	➡	1765	RiskLimitUtilizationPercent	N	30,8	Percentage of utilization of a party's RiskLimitType (1530) specified.
		<Instrument Scope Group>					
➡	➡	1534	NoRiskInstrumentScopes		N	1	The number of instrument scopes. The value will always be 1 if specified. Required if RiskLimitType (1530) = Position (1000).
➡	➡	➡	1536	InstrumentScopeSymbol	N	30	The symbol of the futures instrument the position limit applies to.
➡	➡	➡	1544	InstrumentScopeProductComplex	N	30	The base underlying the position limit applies to.
➡	➡	➡	1545	InstrumentScopeSecurityGroup	N	30	The base underlying to which the near month position limit applies to
➡	➡	➡	167	SecurityType	N	30	Valid Values; **Value** **Meaning** FUT Futures only OOF Options on Futures ALL Both Futures and O Required if InstrumentScopeSecurityGroup (1545) is specified and RiskLimitAction (1766) is set to Square off (5).
➡	➡	➡	711	NoUnderlyings	N	1	Number of underlyings. If present, the value in this field should always be "1".
➡	➡	➡	➡	UnderlyingSymbol (311)	N	30	Symbol of the underlying. Required if NoUnderlyings (711) is specified.
58	Text				N	200	Free format text string

32024	ServiceDeskUserID	N	30	User ID of the market operation user performing the manual status update Required if MemberUpdateSource is Service Desk User ID (1)
32025	MemberUpdateSource	Y	I32	Identifies the source of RiskLimitAction (1767). **Value** **Meaning** 0 RMS 1 Service Desk User
22009	InternalRefFlag	N	15	Internal Reference Flag
Standard Trailer				

7.6.5 Party Risk Limit Report

Tag	Field Name			Req	Length	Description
Standard Header						
35	MsgType			Y	I32	CM = Party Risk Limits Report
Message Body						
1667	RiskLimitReportID			Y	16	The unique identifier of the Party Risk Limits Report message.
<Party Risk Limits Group>						
1677	NoPartyRiskLimits			N	I32	The number of party risk limits. If specified, the value should always be 1.
➡	1324	ListUpdateAction		N	2	The source of the PartyID value. **Value** **Meaning** S Snapshot Required if NoPartyRiskLimits (1677) > 0.
	<Parties Group>					
➡	1671	NoPartyDetails		N	1	Number of Party IDs. If specified the value should always be 1.
➡	➡	1691	PartyDetailID	N	30	The identifier of the party. Required if NoPartyDetails (1671) > 0.

➡	➡	1692	PartyDetailIDSource		N	1	The source of the PartyID value. Value Meaning D Proprietary/Custom C Required if NoPartyDetails (1671) > 0.
➡	➡	1693	PartyDetailRole		N	I32	Value Meaning 1 Executing Firm 53 Trader Mnemonic Required if NoPartyDetails (1671) > 0.
	<Risk Limits Group>						
➡	1669	NoRiskLimits			N	1	The number of risk limits. If specified, the value should always be 1.
➡	➡	1529	NoRiskLimitTypes		N	1	The number of risk limit types. If specified, the value should always be 1. Required if NoRiskLimits (1669) > 0.
➡	➡	➡	1530	RiskLimitType	N	I32	The type of risk limit. Value Meaning 0 Total Collateral 2 MTM Margin 9 Initial Margin 99 MTM PL Required if NoRiskLimitTypes (1592) > 0.
➡	➡	➡	1766	RiskLimitUtilizationAmount	N	30,8	Absolute amount of utilization of a party's RiskLimitType (1530) specified.
➡	➡	➡	1765	RiskLimitUtilizationPercent	N	30,8	Percentage of utilization of a party's RiskLimitType (1530) specified.
58	Text				N	60	Free format text string
Standard Trailer							

8 REJECT CODES

8.1 Reject

Session Reject Reason	Meaning
1	Required tag missing[3]
2	Tag not defined for this message type[4]
4	Tag specified without a value
5	Value is incorrect (out of range) for this tag
6	Incorrect data format for value
10	SendingTime accuracy problem
9	CompID problem
10	SendingTime accuracy problem
11	Invalid MsgType[5]
13	Tag appears more than once
14	Tag specified out of required order
15	Repeating group fields out of order
16	Incorrect NumInGroup count for repeating group
18	Invalid or unsupported application version
99	Other

[3] This reject reason is sent when all the required tags for the message are not present in a message that is recognized by the gateway

[4] Delete this reject reason if the configuration to reject unknown fields in application messages is disabled.

[5] This reject reason is sent when a message that is not defined in the FIX data dictionary is received by the gateway

8.2 Execution Report

OrdRej Reason	Meaning
1	Unknown instrument
2	Exchange closed
3	Order exceeds limit (i.e. rejected by risk management system)
5	Unknown order
6	Duplicate order (i.e. duplicate ClOrdID)
16	Price exceeds current price band
10000	No open orders for specified Party ID
10001	Request limit for day reached
10003	Order download not permitted for specified Party ID
10004	Not authorised to request an open order download
10005	Open order download not permitted at this time
10006	Unknown Party ID
10007	Instrument not specified
10008	No open orders for specified instrument and Party ID combination
10009	Segment not specified
10010	Unknown segment
10011	No open orders for specified segment and Party ID combination

Please refer to the *NCDEX - Reject Codes and Reasons_v1.00.doc* for the list of reject codes and meanings specific to NCDEX.

8.3 Business Message Reject

Business Reject Reason	Meaning
0	Other
2	Unknown instrument
3	Unsupported message type[6]
4	Application not available
5	Conditionally required field missing
30	Session not in sync

[6] This reject reason is sent when the received message is not defined as a valid message for the Drop Copy Gateway

8.4 Functional and Implementation Limitations

8.4.1.1 All the FIX gateways (FIX Trading, FIX Drop Copy and FIX Post Trade) currently use a common library. The system hence accepts all FIX messages defined for all three gateways, and cannot distinguish between them per gateway.

It will validate the incoming messages in the following sequence:

1. The system initially does a FIX library level validation
2. The system does a validation for required fields
3. The system finally does the Gateway level validation

Hence;
If a message is sent which does not comply with the specific gateway being used (but is defined in a different FIX gateway), it will validate the required fields. If any of the required tags are missing, it will give out a session reject with message "Required tag missing").

If a message is sent which does not comply with the specific gateway being used (but is defined in a different FIX gateway), it will validate the required fields. If all required fields are available, a gateway validation gives out a business reject message "Unsupported Message Type".

If a message is sent which does not comply with any of the FIX gateways used it will then give out a session reject message "Invalid Msg Type”.

8.4.1.2 When an Order Mass Status Request is rejected at its entirety, an Execution Report is generated but it does not carry a client order id as the rejection is not related to a specific order. Hence there is an exception to the fact that tag 11 is required in the Execution Report.

8.4.1.3 If an undefined tag is sent along with any of the Administrative messages, then the system will ignore the undefined tags and process the rest of the message. This is a limitation exists in the FIX library**.**

8.4.1.4 The maximum length of the PartyID (448) field is 11. The value will be truncated to length 11 prior to the “user” validation. The system will accept the order if there is a corresponding trader group for the value after the truncation. Otherwise, the order will be rejected with reject reason "Unknown user (OwnerID)".

Example: Order with PartyID FT05TR011123 is entered to the system. (PartyID FT05TR011123 has more than 11 characters and not a configured user in the system). However, the system will truncate the trader group to 11 characters; FT05TR01112. If there is a corresponding trader for FT05TR01112 the system will accept the order. Otherwise, the system will reject the order stating “Unknown User”.