Johannesburg Stock Exchange

Trading and Information Solution

JSE Specification Document

Volume 02 – FIX Trading Gateway (FIX 5.0 SP2)

Version	3.03
Release Date	4 August 2016
Number of Pages	66 (Including Cover Page)

1 DOCUMENT CONTROL

1.1 Table of Contents

1.2 Document Information

Drafted By	Trading and Market Services: TMS Trading
Status	Final
Version	3.03
Release Date	4 August 2016

1.3 Revision History

Date	Version	Description
08 July 2011	1.00	Initial Draft
30 November 2011	1.01	JSE Specification Updated
5 July 2013	2.00	Functionality Updates related to the 2013 product upgrade
4 November 2013	2.01	Minor corrections to Order Cancel/Replace Request msg
22 August 2014	2.02	Introducing the ability to submit Limit Orders with attribute Exclude Hidden Limit Orders
29 February 2016	3.00	Integrated Trading and Clearing Project changes. Equity Market Enhancements: • Hidden Order functionality enhanced • Introduction of EOD Volume Auction • Addition of SecondaryClientOrderID
4 April 2016	3.01	Updated to remove Pegged Hidden Order functionality
25 May 2016	3.02	Equity Market Enhancement functionality added to the FIX Trading Gateway
4 August 2016	3.03	Updated description of GDX TIF Cross Type (549) changed to a required field Updated Display Quantity description Updated Min Qty description Added enumerations for Exec Restatement Reason (378) Added clarity on populating of the party block for Order Cancel Reject message

1.4 References

FIXT 1.1 Specification
FIX 5.0 (Service Pack 2) Specification

1.5 Contact Details

JSE Limited Trading and Market Services Division One Exchange Square Gwen Lane, Sandown South Africa Tel: +27 11 520 7000 www.jse.co.za	**Trading and Market Services ITAC Queries** Email: ITACTradingAPI@jse.co.za

1.6 Definitions, Acronyms and Abbreviations

Client	A member firm connected to the trading 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.
JSE	Johannesburg Stock Exchange.
NSX	Namibian Stock Exchange
Server	The FIX trading gateway at the JSE for the JSE and NSX markets
Trader	Each order must be submitted under a JSE Trader ID.
Trader Group	Trader group within a Firm which the Trader belongs to.
Repeating Group	A set of related attributes which occur more than once within a field of a message
Open Order	Identifies an order which has a remaining quantity in the order book. An amendment or a cancellation can be done for an Open Order.
Parked Order	Identifies an order which is not yet been activated. GFA, GFX, ATC orders will be parked until the relevant auction call phase is started. Unelected Stop and Stop Limit orders will be parked until the stop price is reached.
Client Account	This is the Client Account as reflected in the JSE back office system for whom the order is submitted by the Firm.
Visible Order	Identifies an order that is visible to the market. The order has a Disclosed Quantity that is equal to Order Quantity.
Passive Order	An order residing in the order book.
MES	Minimum Execution Size is the minimum volume of the Hidden Pegged and Pegged Limit order which is permitted to execute.
MRS	Minimum Reserve Size (~~is the equivalent of the JSE's MOS – for Hidden Pegged and Pegged Limit orders~~) MRS is the minimum order volume for orders to qualify as Hidden Pegged and Pegged Limit orders. Applicable for Orders with TIF of GDX.
Trading Gateway	The interface at the JSE for the JSE and NSX Markets that allows member firms to submit and manage orders and trades.
CPP Session	Closing Price Publication is the session where the Closing Price is calculated and published to the Market.
CPX Session	Closing Price Cross is the session where automated trading can occur at the Closing Price calculated during the CPP session.
CPX TIF	Closing Price Cross is a Time In Force where respectively submitted orders are only executed during the CPX session at the Closing Price calculated in the Closing Price Publication session. CPX TIF Orders submitted before the CPX session are parked until the start of the CPX Session

EHL	Exclude Hidden Pegged and Pegged Limit order is an order ~~attribute to indicate~~ that ~~such limit orders~~ will only match with existing Visible Orders in the order book and will expire if the remainder is not filled within a period of time defined. At no point will these orders execute with Hidden Pegged and Pegged Limit orders.
EOD Volume Auction Uncrossing	A dark auction call which is triggered at end of the day after the CPX session. The uncrossing will happen at the closing price.
Pegged Order	A hidden order pegged to the mid-point of the best bid and offer price or pegged to the best bid(offer) for instrument
Pegged Limit Order	A pegged order with a stop price also known as a hard limit.
Cross Order Trade	A trade resulting from the submission of a Cross Order by market participants that results only in a trade and has no impact to orders
Hidden (Limit) Orders	Pegged and Pegged Limit orders are hidden orders on the central order book and not visible to the market. Hidden (Limit) Orders will be synonymous with Pegged and Pegged Limit Orders.

2 OVERVIEW

The System offers a trading gateway which will allow member firms to submit and manage orders. The interface enables clients to perform the activities outlined below.

Order Handling

(i) Submit an order

(ii) Cancel an order

(iii) Mass cancel orders

(iv) Amend an order

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.

3 SERVICE DESCRIPTION

3.1 System Architecture

Clients will connect to the server through a standard TCP IP connection.

The trading system can consist of a series of parallel partitions each of which services an exclusive set of instruments. Each application message transmitted by the server will include the identity of the partition that generated the message.

3.2 Order Handling

3.2.1 Order Types

Clients may submit the order attributes outlined below via the New Order message. The server will generally acknowledge or reject as per the Fix Protocol.

Clients may submit the order attributes outlined below via the New Order – Single message.

Order Type	Description	Relevant FIX Tags
Market	An order which will be executed at the best possible prices on the contra side. Market Orders entered during the Continuous Trading session will execute against each contra order in the order book until it is fully filled. If, after executing against all orders in the order book there is a remainder, it will expire. Market Orders which are submitted during an auction call session will reside in the order book until the uncrossing is performed at which point the remainder of unexecuted Market Orders will be expired.	OrderType (40) = 1
Limit	An order which will contain a limit price and will execute at prices equal to or better than its limit price. If after executing against all appropriately priced orders in the order book there is a remainder, it will be added to the order book or expired based on the time in force (TIF).	OrderType (40) = 2 Price (44)
Stop	A Stop Order is a Market Order that will remain unelected (without entering the order book) until the stop price is reached. Once elected, it will be treated similar to a regular new Market Order.	OrderType (40) = 3 StopPx (99)
Stop Limit	A Stop Limit Order is a Limit Order that will remain unelected (without entering the order book) until the stop price is reached. Once elected, a Stop Limit Order will be treated similar to a regular new Limit Order.	OrderType (40) = 4 StopPx (99) Price (44)

~~Hidden Pegged and Pegged Limit~~	~~Identifies an order that is not visible to the market. The limit order has an order quantity that is greater than or equal to MRS (Minimum Reserve Size), the Display Quantity is equal to zero and has a Minimum Execution Size specified~~	~~Order Type (40)=4~~ ~~DisplayQty (1138) = 0 AND DisplayMethod (1084) = 4~~ ~~Min Qty (110)~~
Cross Orders	The details of both sides of a trade are entered by a single member firm. The matching of the submitted sides of the cross order will happen upon submission if all relevant validations are successful. A submitted cross order will not execute against other orders in the order book.	OrdType (40) = 2 CrossType (549) = 5
Pegged Order	Large-in-size orders can be pegged to the mid-point of the visible BBO, pegged to the best Bid (if it is buy pegged order) and best Offer (if it is a sell pegged order). These orders will not be visible to the market but would be able to interact with other hidden pegged orders or aggressing visible orders	OrdType (40) = P PegPriceType (1094) = 2, 5 DisplayMethod (1084) = 4 DisplayQty = 0 MinQty (110)
Pegged Limit Order	A pegged order with an order time "pegged limit" and stop limit price will be considered as a pegged order with a hard limit. If the hard limit of a pegged order is breached, it will not trade.	OrdType (40) = R PegPriceType (1094) = 2, 5 DisplayMethod (1084) = 4 DisplayQty = 0 MinQty (110) StopPx (99) ExecInst(18) is optional

3.2.2 Order Time in Force (TIF)

Order Type	Description	Relevant FIX Tags
Day	Orders with the DAY time in force will be expired at Market End of the trading on the day they are submitted	TimeInForce (59) = 0
Immediate or Cancel (IOC)	Orders with the IOC time in force (except for Stop and Stop Limit orders) will be executed on receipt and the remainder, if any, will be immediately expired. An IOC order may be partially filled.	TimeInForce (59) = 3
Fill or Kill (FOK)	Orders with the FOK time in force (except for Stop and Stop Limit orders) will either be fully executed on receipt or immediately expired.	TimeInForce (59) = 4
At the Open (OPG)	An order that may only be entered and executed in the opening auction.	TimeInForce (59) = 2
At the Close (ATC)	An order that may only be executed in the closing auction.	TimeInForce (59) = 7

Good for Auction (GFA)	An order that may only be executed in the next auction (which may or may not be scheduled e.g. opening, closing, re-opening, volatility etc.	TimeInForce (59) = 9
Good For Intraday Auction (GFX)	An order that may only be executed in the Intraday auction.	TimeInForce (59) = 8
Good Till Time (GTT)	An order that will expire at a specified time during the current day or at market end.	TimeInForce (59) = 6 ExpireTime (126)
Good Till Date (GTD)	An order that will expire at Market End of the specified day. Maximum order duration is applicable for GTD orders and will be set at 90 days.	TimeInForce (59) = 6 ExpireDate (432)
Good Till Cancelled (GTC)	An order that will expire at Market Start on the trading day after the maximum order duration. Maximum order duration is applicable for GTC orders and will be set at 90 days.	TimeInForce (59) = 1
Closing Price Cross (CPX)	An order that may only be executed during the Closing Price Cross session.	TradingSessionID (336) = a
Exclude Hidden Pegged and Pegged Limit (EHL)	An order that will only match with existing Visible Orders in the order book and will expire if the remainder is not filled within a period of time defined. At no point will these orders execute with Hidden Pegged and Pegged Limit orders. (Refer section 3.2.3)	TimeInForce (59) = 6 ExpireTime (126) ExecInst(18) = v
Good for Volume Auction Uncross (GDX)	An order that will only participate in the End of day 'Volume Auction Uncross'	TimeInForce (59) = 0 TradingSessionID (336) = b ~~ExecInst(18) = u~~

3.2.2.1 Order Capacity

The server recognises two order capacities; agency and principal. Clients are responsible for indicating the capacity an order is submitted under. If a New Order – Single message does not contain the OrderCapacity (528) field, it will be rejected by the server.

3.2.3 EHL Orders

An EHL order can only be a Visible Limit Order. Furthermore, it can only be submitted with a TIF (Time In Force) of GTT (Good Till Time). Any other Order Type or TIF will not be allowed with this attribute. This attribute will only be present in new order submissions. The ability to amend this attribute or to amend an existing order to be an EHL order or vice versa will not be allowed. Cancellation or amendment of EHL orders by Traders or Market Operations Users on behalf of Traders is allowed as long as the EHL order has not expired.

EHL orders will carry an expiry time defined by JSE. The default expiry time configured is 2 seconds. JSE will notify market participants of any changes made to the expiry time. Any unexecuted EHL order submitted by a Trader or Market Operations User on behalf of a Trader will expire within the time period.

It is required for Traders or Market Operations Users on behalf of Traders to specify an expiry time in the order upon submission. Time should be specified in the same format when submitting a GTT order i.e. YYYYMMDD-HH:MM:SS. However it should be noted that the expiry time defined by a Trader or Market Operations User on behalf of a Trader in an EHL order can be overridden (Time will be replaced with Current Time + EHL EXPIRY TIME (S)).

The expiry time will be replaced by the time defined by JSE given that the difference between the order submission time and expiry time submitted is greater than the defined number of seconds, order submission time is considered as the time the order enters the system. The updated expiry time will be sent in the Execution Reports for the EHL order.

Submission of EHL orders to the System will not be allowed when an instrument is in any other session than Continuous Trading, Continuous Trading 1 or Continuous Trading 2. If an EHL order exists in the order book when the session is moved from Continuous Trading, Continuous Trading 1 or Continuous Trading 2 to any other session (due to scheduled transitions or manual session movements), it will be expired immediately upon moving to the given session.

3.2.4 Cross Orders

Cross orders will only be enabled for Equity Market instruments. The functionality will be disabled for the Derivatives and Bonds markets.

Cross orders enable participants to report pre-negotiated trades to the trading system as on-book trades through the central order book.

Internal cross orders are submitted when a trade is agreed within a single member firm. The details of both sides of the trade are entered by a single member firm. Privileged users can submit cross orders via the FIX Trading gateway.

The matching of the submitted cross orders will happen immediately, should all the order validations be successful. Hence, the internal cross orders submitted are not available for execution with any other orders in the regular order book.

Clients will receive an Execution Report per side once a Cross Order has been successfully submitted and accepted by the system. Each Execution Report generated by the system will be sent back to the originating Comp ID that submitted the Cross Order.

3.2.5 Pegged Orders (Hidden)

Pegged orders will only be enabled for Equity Market instruments. The functionality will be disabled for the Derivatives and Bonds Markets.

Large-in-size hidden orders can be pegged to the mid-point of the visible BBO, pegged to the best Bid (if it is buy pegged order) and best Offer (if it is a sell pegged order). These orders will not be visible to the market, but will be able to interact with other hidden pegged orders or aggressing visible orders within the central order book.

It will be mandatory to specify a Minimum Execution Size (MES) for all pegged orders submitted to the system. All new pegged orders and pegged order amendments will be validated against the Minimum Reserve Size of the instrument.

A pegged order with an order time “pegged limit” and limit price will be considered as a pegged order with a hard limit. If the hard limit of a pegged order is breached, it will not trade.

~~3.2.3~~3.2.6 Order Management

~~3.2.3.1~~3.2.6.1 Cancellation

The remainder of an Open or Parked order may be cancelled via the Order Cancel Request message. The server will respond with an Execution Report or Order Cancel Reject to confirm or reject the cancellation request respectively.

The client will identify the order being cancelled by either its OrigClOrdID (41) or OrderID (37). If an Order Cancel Request contains values for both OrigClOrdID (41) and OrderID (37), the server will only process the OrderID (37). If an order submitted under a different Interface User ID (SenderCompID (49)) is being cancelled, the Order Cancel Request will include its OrderID (37).

As per standard FIX on order entry the Client would specify a Client Order ID. On an amendment or cancellation the client would need to specify a new ClOrdID on the request message with reference to the original ClOrdID on the OrigClOrdID (41) field.

If the amendment or cancellation is accepted, the order will from that point onwards be referred to by the new ClOrdID as of the request.

~~3.2.3.2~~3.2.6.2 Mass Cancellation

A client may mass cancel Open and Parked orders via the Order Mass Cancel Request message. The server will respond with an Order Mass Cancel Report to indicate, via the Mass Cancel Response (531) field, whether the request is successful or not. If the mass cancel request is processed by multiple partitions, an Order Mass Cancel Report will be transmitted for each partition.

If the mass cancel request is accepted by a partition, it will then transmit Execution Reports for each order that is cancelled and Order Cancel Rejects for each order that could not be cancelled. The ClOrdID (11) of all such messages will be the ClOrdID (11) of the Order Mass Cancel Request.

If the mass cancel request is rejected by a partition, the reason will be specified in the MassCancelReject Reason (532) field of the Order Mass Cancel Report.

Clients may use the Order Mass Cancel Request to mass cancel all orders or only those for a particular instrument or segment. A mass cancel request may apply to all the orders of the trading firm or only to those of a particular trader and trader group combination. It is required to specify the trading party when an Order Mass Cancel Request is submitted. The FIX fields relevant to each of the supported mass cancel combinations are outlined below.

	Trader and Trader Group Combination	**Firm**
All Orders	MassCancelRequestType (530) = 7 TargetPartyRole (1464) = 53 TargetPartyID (1462) TargetPartyRole (1464) = 76 TargetPartyID (1462)	MassCancelRequestType (530) = 7 TargetPartyRole (1464) = 1 TargetPartyID (1462)
All Orders for an Instrument	MassCancelRequestType (530) = 1 Security ID(48) SecurityIDSource (22) = 8 TargetPartyRole (1464) = 53 TargetPartyID (1462) TargetPartyRole (1464) =76 TargetPartyID (1462)	MassCancelRequestType (530) = 1 Security ID (48) SecurityIDSource (22) = 8 TargetPartyRole (1464) = 1 TargetPartyID (1462)
All Orders for a Segment	MassCancelRequestType (530) = 9 MarketSegmentID (1300) TargetPartyRole (1464) = 53 TargetPartyID (1462) TargetPartyRole (1464)= 76 TargetPartyID (1462)	MassCancelRequestType (530) = 9 MarketSegmentID (1300) TargetPartyRole (1464) = 1 TargetPartyID (1462)

3.2.6.3 Cancellation by Market Operations

An unsolicited Execution Report will be sent to the client if an order is cancelled by market operations. The ExecRestatmentReason (378) of such a message will be Market (Exchange) option (8). It will not include an OrigClOrdID (41).

~~3.2.3.3~~3.2.6.4 Amending an Order

The following attributes of an Open or Parked order may be amended via the Order Cancel/Replace Request message:

(i) Order quantity

(ii) Minimum quantity (Minimum Execution Size)

(iii) Price

(iv) Stop price

(v) Expiration time (GTT orders)

(vi) Expiration date (GTD orders)

(vii) Client Account

An Order Cancel/Replace Request must include values for the fields that are being updated as well as the current values for those that are not being amended. The server will respond with an Execution Report or Order Cancel Reject message to confirm or reject the cancel/replace request respectively.

The client will identify the order being amended by either its OrigClOrdID (41) or OrderID (37). If an Order Cancel/Replace Request contains values for both OrigClOrdID (41) and OrderID (37), the server will only process the OrderID (37).

If an order submitted under a different Interface User (SenderCompID) (49) is being amended, the Order Cancel/Replace Request will include its OrderID (37). If the amendment is successful, the order will be treated as one submitted under the Interface User (SenderCompID) (49) that sent the Order Cancel/Replace Request.

An order will lose time priority if its order quantity is increased or if its limit price is amended. A reduction in order quantity of an order or the amendment of its expiration time, expiration date or Client Account will not cause it to lose time priority.

Clients may not amend orders that are fully filled.

~~3.2.4~~3.2.7 Unsolicited Order Updates

The Execution Report message is used to notify the client if an order is executed or expired. The Client Order ID of the message will be that of the last New Order or Order Cancel/Replace Request that successfully updated the order.

~~3.2.5~~3.2.8 Order Status

As specified in the FIX protocol, the OrdStatus (39) field on the 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
2	Filled
9	Suspended

4	Cancelled
C	Expired
1	Partially Filled
0	New
8	Rejected

Please refer to Section 10.1.1 for process flow diagrams on the various statuses that may apply to an order.

~~3.2.6~~3.2.9 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	Order Status
0	**New** Indicates that a new order has been accepted. This message will also be sent unsolicited if an order was submitted by JSE Market Operations on behalf of the client. This message will also be sent when an unelected order is elected and added to the order book without receiving an execution.	0
8	**Rejected** Indicates that an order has been rejected. The reason for the rejection is specified in the field OrdRejReason (103).	8
F	**Trade** Indicates that an order has been partially or fully filled. The execution details (e.g. price and quantity) are specified. This message will also be sent when an unelected order is elected and receives executions on aggression. This message will also be sent when a parked order with time in force GFX/GFA/ATC/CPX is un-parked and receives executions on aggression.	1, 2
C	**Expired** Indicates that an order has expired in terms of its time qualifier or due to an execution limit. This message will also be sent when orders are expired upon entering the order book when the number of orders in the order book is at the maximum allowed level. The reason for the expiration is specified in the Text (58) field. This message will also be sent when a Market Order or a Stop Order is expired at the point of aggressing the order book during the Continuous Trading session, if a circuit breaker is breached during that aggression, The reason for the expiration is specified in the Text (58) field. Additionally this message will be sent when the remaining orders (except GTC and GTD) are expired at the market close and when the orders are expired based on the auto cancellation on disconnect/log out feature.	C
4	**Cancelled** Indicates that an order cancel request has been accepted and successfully processed. This message will also be sent unsolicited if the order was cancelled by JSE Market Operations. In such a scenario the Execution Report will include an ExecRestatementReason (378) of Market Option (8). It will not include an OrigClOrdID (41).	4

5	**Replaced** Indicates that an order cancel/replace request has been accepted and successfully processed.	0, 1
L	**Triggered** Indicates that a parked ATC, GFX, GFA, CPX, stop or ~~stop~~MIT order has been activated and moved to the main container. The order is available for execution.	0, 1
9	**Suspended** Indicates that a GFA or GFX order that was active has been parked and is no longer available for execution.	0, 1
D	**Restated (Order Cancel/Replace by Market Operations)** Indicates that an order has been amended by JSE Market Operations or due to TIF CPX orders being re-priced during the CPX session. The unsolicited message will include an ExecRestatementReason (378) of Market Option (8). It will not include an OrigClOrdID (41).	0, 1
H	**Trade Cancel** Indicates that an execution has been cancelled by Market Operations or by clients. An ExecRefID (19) to identify the execution being cancelled will be included.	0, 1, 4, C
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, 4, C

It should be noted that the Exchange will generally not amend orders or trades. These events are included in the above table for completeness.

~~3.2.7~~3.2.10 Order and Execution Identifiers

~~3.2.7.1~~3.2.10.1 Client Order IDs

Clients must specify a Client Order ID when submitting an application message (i.e. New Order, Order Cancel Request, Order Mass Cancel Request or Order Cancel/Replace Request).
The server validates each ClOrdID (11) for uniqueness. Clients must comply with the FIX protocol and ensure unique ClOrdIDs across all messages (e.g. New Order – Single, Order Cancel Request, etc.) sent under a particular Interface User ID (SenderCompID (49)). Given that the server supports GTD and GTC orders, clients must ensure that their ClOrdIDs are unique across trading days (e.g. embed the date within the ClOrdID). The Execution Report transmitted to reject an order due to a duplicate ClOrdID (11) will not include the fields ExecID (17), OrderID (37), LeavesQty (151), TransactTime (60) and CumQty (14).

Clients must, in terms of the FIX protocol, specify the ClOrdID (11) when submitting an Order Cancel Request, Order Mass Cancel Request or Order Cancel/Replace Request.

~~3.2.7.2~~3.2.10.2 Order IDs

The server uses the OrderID (37) field of the Execution Report to affix the order identification numbers of the trading engine. Order IDs are unique across trading days in perpetuity.

This is an 11 character base 62 string with an 'O' prefix. After removal of the prefix, when converted to an 8 byte binary format, it will match the corresponding MITCH Order ID. Thus, FIX OrderID (37), and MITCH OrderID are all representations of the same identifier [in base 62 (plus 'O' prefix), hexadecimal, and binary formats respectively (Refer section 3.11 for details).

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.

Clients have the option of specifying the OrderID (37) when submitting an Order Cancel Request or Order Cancel/Replace Request.

Unlike the Client Order ID, which is updated on each successful Order Cancel/Replace or Order Cancel Request, the Order ID of an order will remain constant throughout its life.

~~3.2.7.3~~3.2.10.3 Execution IDs

The server uses the ExecID (17) field to affix a unique identifier for each Execution Report. ExecIDs are unique across trading days in perpetuity. If an Execution Report message is used to notify a client of a trade cancellation or correction, the ExecRefID (19) will refer to the Execution ID generated for the original execution.

~~3.2.7.4~~3.2.10.4 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 and MITCH market data feeds. Trade IDs are unique across trading days in perpetuity.

An Execution Report published to notify a client of a trade cancellation or correction includes the TradeID of the trade.

3.3 Instrument Identification

Instruments will be identified by a unique Instrument ID populated in the SecurityID (48) field. It is required to specify SecurityID Source (22) field as well.

3.4 Party Identification

ID	Description	Relevant FIX Tags
Executing Firm (Firm ID)	Identifier of the trading firm the order is submitted under.	PartyRole (452) = 1 PartyID (448)
Trader Group	Identifier of the trader group the order is submitted under.	PartyRole (452) = 76 PartyID (448)
Trader	Identifier of the trader the order is submitted under. This is the JSE Trader ID.	PartyRole (452) = 53 PartyID (448)
Client Account	Client Account information applicable to an order.	Account (1)
Couterparty Firm	Identifier of the counterparty trading firm the cross order is submitted under.	PartyRole (452) = 17 PartyID (448)
Counterparty Trader Group	Identifier of the counter party trader group the cross order is submitted under.	PartyRole (452) = 100 PartyID (448)

Counterparty Trader	Identifier of the counter party trader the cross order is submitted under. This is the JSE Trader ID.	PartyRole (452) = 37 PartyID (448)

It will be mandatory to specify the Trader Group (Party Role (452) = 76) and the Trader (Party Role (452) = 53) in New Order – Single, Order Cancel, Order Cancel/Replace messages. For the New Order Single (D), Order Cancel Request (F), and Order Cancel/Replace Request (G) messages, the message will be rejected if the Trading Party Component does not include a Party ID (448) Tag without a corresponding Party Role (452) Tag equal to 76 (Trader Group) and 53 (Trader) within the same repeating group. . For the New Order Cross (s) message, the message will be rejected if the Trading Party Component does not include a Party ID (448) Tag corresponding to a Party Role (452) Tag equal to 100 (Counterparty Trader Group) and 37 (Counterparty Trader) within the same repeating group. It will be optional to specify an Executing Firm (Party Role (452) = 1) and a Counterparty Firm (Party Role (452) = 17) in these messages.

Any messages rejected will be acknowledged to the client with a Business Message Reject (j) message with the following tags specified:

Business Reject Reason (380) = '0'

Text (58) = Trader of Trader Group not specified

3.5 Market Operations

3.5.1 Order Management

JSE Market Operations is able to submit an order, order cancel request, order cancel/replace request, on behalf of a client.

The client will be notified via an Execution Report of the order, order cancel request or cancel/replace request submitted on its behalf if and when it is accepted. In such a scenario the Execution Report will include an ExecRestatementReason (378) of Market Option (8) to indicate that the order has been cancelled or cancelled/replaced by Market Operations.

When an order is cancelled or cancelled/replaced by Market Operations the Execution Report generated will not include an OrigClOrdID (41).

If an order is submitted by Market Operations the Execution Report will include an IsMarketOpsRequest (27000) of Yes (1).

The client will not be notified if the action is rejected or queued.

This feature is intended to help a client manage an emergency situation and will not be relied upon as a normal business practice.

3.5.2 Trade Cancellations and Corrections

JSE Market Operations may cancel or correct any trade on behalf of a Client. Additionally participants may cancel their own trades. The server will transmit Execution Reports to the relevant clients to notify them of a trade cancellation or correction. The trade being cancelled or corrected will be identified via the ExecRefID (19) field. This field will contain the ExecID (17) of the Execution Report that was originally transmitted to notify the client of the trade.

If an execution received by an order is cancelled, the cancelled quantity will be cancelled. If the quantity is cancelled, the order will be restated to reduce its order quantity by the cancelled quantity. The client will receive two notifications in such a scenario; one for the trade cancel and another for the restatement.

The notification of the trade cancellation notifies the reduction of the quantity that was executed. The notification of the restatement adjusts the order quantity of the order and the

remaining quantity of the order to indicate the quantity cancelled from the trade is not added back to the order.

Market operations may also correct the price of an execution. A trade will not be corrected to increase the executed quantity.

3.6 Timestamps and Dates

The timestamps SendingTime (52), OrigSendingTime (122) and TransactTime (60) will be in UTC and in the YYYYMMDD-HH:MM:SS.sss format. ExpireTime (126), (will be in UTC and in the YYYYMMDD-HH:MM:SS format.)

All dates (i.e. ExpireDate (432)) will be in the YYYYMMDD format and specified in the local date and time for the server (i.e. not in UTC)).

3.7 Market Operations Announcements

The Client will receive market announcements via the News message on the Trading Gateway. It will contain the market operations announcement headline, text, urgency, the time that it was generated and the list of instruments if any, to which the announcement relates to.

JSE Market Operations has the ability to send:

- Private Announcements to a specific Interface User (CompID) which can be disseminated to all traders who connect to the Trading Gateway via the specific Interface User (CompID).
- Private Announcements to a specific Firm, where the announcement will be disseminated to all Interface User IDs (CompIDs) within that Firm who have logged into the Trading Gateway. Therefore, all traders who connect via the Firm's Interface Users (CompIDs) can receive the market operations announcement.

Recovery of any missed messages through the recovery channel using the current mechanism of requesting missed messages via the Resend Request.

Clients who request the missed messages will receive all the messages including market operations announcements relevant to the Interface User or Firm, with a sequence number equal to or larger than the requested sequence number published from the particular partition.

3.8 Repeating Groups (Components/Component Block)

If a repeating group is used in a message, the number of IDs (for example NoPartyIDs field in the trading party repeating group) will 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 as specified in messages on this document.

The messages generated by a client should have the first field in a repeating group in order as specified in messages on this document. 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.

3.9 Generating Reject Messages

If a required tag or a conditionally required tag is missing in a message sent by a client, the server will send a session reject message for that.

The server will also send a session reject message if the same FIX tag has been repeated within the client request.

Furthermore, if an unsupported value is sent with a tag, an execution report or an order cancel reject is sent by the server.

Session level validations are performed first; then Business Rejects and then rejections via Execution Reports.

3.10 Functional and Implementation Limitations

3.10.1.1 As an exception to 3.8, a session reject message will also be generated for an unsupported value. This will happen if the FIX tag supports a different data type compared to the internal field (for example if an invalid capacity with an unsupported data type for the FIX field, is sent with a New Order message, a session reject is sent but if an invalid account type is sent within the supported data type, an Execution Report message is sent).

3.10.1.2 When a cancellation or amendment to a “Filled” or “Cancelled” or “Expired” order is rejected, order status is given as “Rejected” in the Order Cancel Reject message instead of the actual status of the order

3.10.1.3 When an amend request to change the side of an order is rejected, the order status is given as “Rejected” in the Order Cancel Reject message

3.10.1.4 Order Status (39) can be “Rejected” in the

3.10.1.5 **Execution Report**Execution Report and Order Cancel Reject if the order is unknown or the cancellation or amendment request cannot be processed anymore. There are some basic validations that need to be passed to process an order. If those validations fail, an order will not be processed anymore hence some of the actual information of the order will not be stamped in the messages generated by the server. For example, an amend request to change the Order Quantity (38) or Display Quantity (1138) to a negative value will be rejected via an Order Cancel Reject with the Order Status (39) as “Rejected”. Certain validations are done at the gateway where the request is rejected at the gateway and not passed through to the core matching system. Hence on the rejection message generated, certain details such as the current status of the order is not communicated back as this information is not available to the gateway. This is done to avoid irrational requests being processed by the system.

3.10.1.6 The order status will be communicated as "Expired" for an order which is getting cancelled in the case of Mass Cancel on Logout / Mass Cancel on Disconnect after the users logs out/disconnects if the appropriate setting is turned on.

3.10.1.7 Within a client generated message, if the same FIX tag has been repeated with different values, the server takes the value in the last tag. The server will not reject such messages

3.10.1.8 If the value Trader Group (76) has been repeated with the Party Role (452) in the Trading Party Component Block in a client generated message, the value in the last Party ID (448) (which is corresponding to the last Party Role (452)) is taken as the ID of the Trader Group. The server will not reject such messages.

3.10.1.9 Server does not keep track of filled/cancelled/expired/rejected orders (i.e. it does not keep track of orders whose life cycle is over). Also it is not possible to find whether there are active orders for a particular instrument or not; hence server does not reject a Mass Cancel request in total just because there are no orders for a particular user + instrument combination; the rest of the user + instrument combinations may have valid open orders.

3.10.1.10 An Order Mass Cancel Request should not be sent during Start of Trading session. If a request is sent, it will be rejected as expected. But thereafter in a subsequent session the client will not be able to mass cancel same orders again. But the client can individually cancel orders

3.10.1.11 At present, if an Order Mass Cancel Request is sent for instruments which are in multiple matching partitions, an Order Mass Cancel Report will be sent per matching partition with the confirmation/rejection of the cancellations of orders in that respective partition. This is because the system handles mass cancel requests per partition internally. The relevant partition will be stamped in the ApplID (1180) field in the Order Mass Cancel Report.

3.10.1.12 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;

Scenario 1>>

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").

Scenario 2>>

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".

Scenario 3>>

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".

3.11 Mapping FIX Order ID to MITCH Order ID

To convert FIX Order ID to MITCH Order ID:

Step 1 – Remove leading O (prefix)

Step 2 - Convert using the base 62 conversion table below.

Step 3 – Convert to binary

Order ID format (ASCII)

1 Byte	11 bytes
O	0-9, A-Z, a-z
Prefix	Base 62 encoded order id

E.g.

OrderID in FIX (ASCII base 62 characters)	O04Xj7Wu76ta
OrderID in MITCH gateway (Binary ID converted to decimal)	61512470073704470

Steps to follow

1. Remove the prefix from the ASCII order ID - "O" → 04Xj7Wu76ta
2. Convert using base 62 conversion in to decimal as depicted below
3. Note: Please refer to the base 62 conversion table attached below
4. Convert the decimal value to binary.

FIX Order ID (ASCII Character)	Corresponding decimal value	Base 62^x	value	Multiplied decimal value
a	36	62^0	1	36
t	55	62^1	62	3,410
6	6	62^2	3,844	23,064
7	7	62^3	238,328	1,668,296
u	56	62^4	14,776,336	827,474,816
W	32	62^5	916,132,832	29,316,250,624
7	7	62^6	56,800,235,584	397,601,649,088
j	45	62^7	3,521,614,606,208	158,472,657,279,360
X	33	62^8	218,340,105,584,896	7,205,223,484,301,568
4	4	62^9	13,537,086,546,263,552	54,148,346,185,054,208
0	0	62^10	839,299,365,868,340,224	-
OrderID in MITCH Gateway in Decimal				**61,512,470,073,704,470**

Note

1. Please use 64 bit integer data types for the calculation else integers will overflow
2. Excel also rounds the value since its using a 64 bit float data type for the calculation

The base 62 mapping table

0	0	20	K	40	e	60	y
1	1	21	L	41	f	61	z
2	2	22	M	42	g		
3	3	23	N	43	h		
4	4	24	O	44	i		
5	5	25	P	45	j		
6	6	26	Q	46	k		

7	7	27	R	47	l
8	8	28	S	48	m
9	9	29	T	49	n
10	A	30	U	50	o
11	B	31	V	51	p
12	C	32	W	52	q
13	D	33	X	53	r
14	E	34	Y	54	s
15	F	35	Z	55	t
16	G	36	a	56	u
17	H	37	b	57	v
18	I	38	c	58	w
19	J	39	d	59	x

3.12 Mapping Trade Match ID to MITCH Trade ID

To convert FIX Trade Match ID to MITCH Trade ID:

Step 1 - Remove leading (prefix)

Step 2 - Convert using base 62 using the base 62 conversion table.

Step 3 - Convert to binary

Trade ID format (ASCII)

1 Byte	9 bytes
T	0-9, A-Z, a-z
Prefix	Base 62 encoded Trade ID

E.g.:

ASCII trade id for FIX	T5DIF33YV0
Binary trade id (decimal) for MITCH	73120274710544

Steps to follow:

1. Remove the prefix from the ASCII order ID - "T" → 5DIF33YV0
2. Convert using base 62 conversion in to decimal as depicted below
3. Note: Please refer to the base 62 conversion table attached below
4. Convert the decimal value to binary.

FIX Trade ID	Corresponding decimal value	Base 62^x	value	Multiplied decimal value

(ASCII Character)				
0	0	62^0	1	0
V	31	62^1	62	1,922
Y	34	62^2	3,844	130,696
3	3	62^3	238,328	714,984
3	3	62^4	14,776,336	44,329,008
F	15	62^5	916,132,832	13,741,992,480
I	18	62^6	56,800,235,584	1,022,404,240,512
D	13	62^7	3,521,614,606,208	45,780,989,880,704
5	5	62^8	218,340,105,584,896	1,091,700,527,924,480
Trade ID in MITCH Gateway in Decimal				**1,138,517,709,214,786**

Note

1. Please use 64 bit integer data types for the calculation else integers will overflow
2. Excel also rounds the value since its using a 64 bit float data type for the calculation

4 CONNECTIVITY

4.1 Interface User ID (CompIDs)

The Interface User ID (CompID) of each client must be registered with the JSE and NSX markets before FIX communications can begin through the gateway. A single client may have multiple connections to the server (i.e. multiple FIX sessions, each with its own Interface User ID (CompID) if it has multiple valid Interface User IDs (CompIDs)).

The Interface User (CompID) of the server will be JSEFIXGW. The messages sent to the server will contain the Interface User (CompID) assigned to the client in the field SenderCompID (49)) and JSEFIXGW in the field TargetCompID (56). The messages sent from the server to the client will contain JSEFIXGW in the field SenderCompID (49) and the Interface User (CompID) assigned to the client in the field TargetCompID (56).

If the JSE enables password policies, each new Interface User ID (CompID) will be assigned a password on registration. Clients must change the password to one of their choosing via the Logon message. The acceptance of a login request that includes a password change request indicates that the new password has been accepted. The new password will, if accepted, be effective for subsequent logins. If a new password is rejected, the text of the Logout message will indicate why the Logon is rejected.

Depending on the password policy implemented by the JSE, the password of each Interface User ID (CompID) may need to be changed after a certain amount of days. If not, the password will automatically expire and the client will be unable to login to the server. In such a case, the client will contact JSE 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 if the JSE enables this feature.

4.2 Production IP Addresses and Ports

The IP address of each client must be registered with the System before FIX communications can begin. The IP addresses and ports of the production servers will be detailed in a consolidated JSE Production Market Facing Client document.

The JSE will assign each registered client to one of the primary IP addresses and ports and one of the secondary IP addresses and ports.

4.3 Failover and Recovery

The system has been designed with fault tolerance and disaster recovery technology that ensures that trading will 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 the JSE. Please refer to the separate Client Failover and Recovery Document.

4.4 Message Rate Throttling

The System has implemented a scheme for throttling message traffic where each Interface User ID (CompID) is only permitted to submit up to a specified number of messages per second. The maximum rate is yet to be decided by the JSE and will be confirmed at a later date.

Every message that exceeds the maximum rate of an Interface User ID (CompID) will be rejected via a Business Message Reject. Such a message will include a BusinessRejectReason (380) of Other (0) and an indication that the rejection was due to throttling in the Text (58) field.

An Interface User ID (CompID) will be disconnected by the server if its message rate exceeds its maximum rate more than MAX_THROTTLE_COUNT_PER_PERIOD <5> times in any THROTTLE_DISCONNECT_PERIOD <30> second duration. In such a case, the server will transmit a Logout message and immediately terminate the TCP/IP connection.

4.5 Mass Cancellation On Disconnect

At the request of the client, the server can be configured to automatically cancel all Open and Parked orders submitted under an Interface User ID (CompID) whenever it disconnects from the server.

This feature does not guarantee that all outstanding orders will be successfully cancelled as executions that occur very near the time of disconnect may not be reported to the client. During such a situation, the client will contact JSE Market Operations to verify that all orders have been cancelled and all Execution Reports have been received.

The configuration of the mass cancellation on disconnect feature cannot be updated during a FIX session.

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 FIXT 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 Interface User ID (SenderCompID (49)) field.

The server will validate the Interface User ID (CompID), password and IP address of the client. Once the client is authenticated, the server will respond with a Logon message. 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 server will respond with a Logon message. The SessionStatus (1409) of this message will be Session Active (0).

The client must wait for the server's Logon before sending additional messages. The server will break the TCP/IP connection if messages are received before the exchange of Logons.

Currently it is expected that the Client who connects to the server responds to the Test Request message sent by the server with a Heartbeat or Resend Request Message. If the client does not comply with this and sends Application Messages such as Order Mass Cancel Request before responding to the Test Request message, the server will not send a Reject message until the Client has responded to the Test Request message.

Immediately after the Server sends the response to the Logon message, if the logon was successful, the server will send a Test Request with a specific TestRequestID (112), to identify if the Client is in sync with the server's outgoing sequence number prior to sending any (i.e. new or missed) Application Messages. If the Client replies to the Test Request with a Heartbeat message with the same TestRequestID (112) as in the Test Request, then that would imply that the client recognises the messages sent by the server and that the sequence numbers are in sync. The server would start sending any new or missed Application Messages to the Client only after receiving this Heartbeat message. Any Application Messages submitted before responding to the Test Request will be rejected with a Business Message Reject. Such a message will include a BusinessRejectReason (380) of Session not in sync (0) and an indication that the rejection was due to the client session not being in sync with the server in the Text (58) field. The client will not be disconnected from the Server; the Gateway will wait for the Test Request message from the client.

The client must send additional messages to the server only after responding to the Test Request. If the client sends any Application Messages before responding to the Test Request as mentioned above, those messages will be processed by the system but the Client would not receive any responses to them as the server has not established that the Client is in sync with the server's outgoing sequence numbers. When the Client sends a Logon message, and if the server receives a higher sequence number than expected, the server sends a Resend Request followed immediately by a Test Request. The client should respond to the Resend Request and afterwards respond to the Test Request to get back in sync with the server.

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 and once the response/s to the Resend Request is processed by the FIX Gateway, the FIX Gateway would send a Test Request to make sure both the client and server is in sync before sending out any missed or new application messages.

If a logon attempt fails because of an invalid Interface User ID (SenderCompID), invalid TargetCompID, invalid password or IP address or not having the appropriate privileges to login to the server gateway, the server will break the TCP/IP connection with the client without sending a Logout or Reject. If during a logon of a SenderCompID, the server receives a second connection attempt via different TCP/IP connection while a valid FIX session is already underway for that same SenderCompID, the server will break the TCP/IP connection with the second connection 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 the client tries to connect again with the same invalid information, the client will be disconnected again.

If a logon attempt fails because of an expired password, a locked Interface User ID (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. In these scenarios the next inbound sequence number expected from the client will be incremented but the outbound sequence number will not be incremented. In this scenario the message sequence number 1 will be sent with the Logout message.

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 Logout 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 impact of logon failures on sequence numbers is summarised in the table below:

Reason for Logon Failure	Session status (of logout)	Inbound Sequence Number	Outbound Sequence Number
Invalid or expired password	8 (password expired)	Incremented by 1	Does not increase (defaulted to 1)
Locked/suspended/inactivated CompID	6 (account locked)	Incremented by 1	Does not increase (defaulted to 1)
Logins are not currently permitted	7 (logins are not allowed)	Incremented by 1	Does not increase (defaulted to 1)
Session level failure (e.g. due to invalid EncryptMethod or DefaultApplVerID etc)	101 (logout session level failure)	Does not increase	Does not increase (defaulted to 1)
Login sequence number is less than the expected sequence number	101 (logout session level failure)	Does not increase	Incremented by 1

Second connection attempt via same TCP/IP connection while a valid FIX session is already underway for that same SenderCompID	No Session logout. Reject message with message type 3	Incremented by 1	Incremented by 1
Second connection attempt via different TCP/IP connection while a valid FIX session is already underway for that same SenderCompID	No logout or reject message	Does not increase	Does not increase

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 will 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.

If the server receives a message that cannot be processed (malformed message) it will not respond to that message and will not increment the sequence number maintained. In such a scenario, when the next readable message is received by the server it will detect a sequence gap between the client and server. The server will send a Resend Request to the client requesting for messages from the sequence number the server is maintaining. If the client does not correct the malformed message to a readable one, the above event model will be repeated until there is no sequence gap.

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 will only be used to recover from an emergency situation. It will 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~~ and then break the TCP/IP connection with the client. As recommended in the FIXT protocol, clients are advised to transmit a Test Request message 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 the JSE defined parameter for the JSE MAX_BUFFERED_COUNT <500> messages.

If, during the exchange of Logout messages, the client or sever detects a sequence gap, it will 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.

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. 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 and once the response/s to the Resend Request is processed by the FIX Gateway, the FIX Gateway would send a Test Request to make sure both the client and server is in sync before sending out any missed or new application messages.

5.4.1 Resetting Sequence Numbers: Starting a New FIX Session

5.4.1.1 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.

A client may also manually request JSE Market Operations to initialize the received sequence number 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 will not be relied upon as a regular practice.

5.4.1.2 **Reset Initiated by the Server**

The system has been designed with fault tolerance and disaster recovery technology that will 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 also required to support a manual request by the JSE Market Operations to initialize sequence numbers prior to the next login attempt.

5.5 Matching system failure

In event of matching system failure, following order entry, clients will receive a Business Reject Message with a BusinessRejectReason (380) of "4" indicating "Application not available".

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) will be the sequence number immediately after that of the last processed message and the EndSeqNo (16) will be zero (0).

The server caches a certain number of messages RESEND_CACHE_SIZE <2000> transmitted to each Interface User ID (CompID). Clients are unable to use a Resend Request to recover messages not in the server's cache.

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

6.3.1 Client-Initiated Messages

The server does not handle possible resends for client initiated messages (Example: New Order – Single.) and ignores the value in the PossResend (97) field of such messages.

6.3.2 Server-Initiated Messages

The server may, in the circumstances outlined in Sections 6.2, use the PossResend (97) field to indicate that an application message may have already been sent under a different MsgSeqNum (34). The client will validate the contents (e.g. ExecID) of such a message against those of messages already received during the current trading day to determine whether the new message should be ignored or processed.

6.4 Transmission of Missed Messages

The Execution Report, Order Cancel Reject, Order Mass Cancel Report, and Business Message Reject messages 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".

7 MESSAGE FORMATS

This section provides details on the header and trailer, the seven administrative messages and eight 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. All fields are encoded using printable ASCII. Any message not included in this section will be rejected by the server for this service.

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 or 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: Order Handling

7.1.2.1 Client-Initiated

Message	MsgType	Usage
New Order – Single	D	Allows the client to submit a new order.
Order Cancel Request	F	Allows the client to cancel an Open or Parked order.
Order Mass Cancel Request	q	Allows the client to mass cancel: (i) All Open and Parked orders. (ii) All Open and Parked orders for a particular instrument. (iii) All Open and Parked orders for a particular segment. The mass cancel may apply to the orders of a particular Interface User (CompID) or to all orders of the firm.
Order Cancel/Replace Request	G	Allows the client to amend/modify an Open or Parked order.
New Order Cross	s	Allows the client to submit a new Cross Order

7.1.2.2 **Server-Initiated**

Message	MsgType	Usage
Execution Report	8	Indicates one of the following: (i) Order accepted. (ii) Order rejected. (iii) Order executed. (iv) Order expired. (v) Order cancelled. (vi) Order cancelled/replaced. (vii) Trade cancelled. (viii) Trade corrected. (ix) Order Status
Order Cancel Reject	9	Indicates that an order cancel request or order cancel/ replace request has been rejected.
Order Mass Cancel Report	r	Indicates one of the following: (i) Order Mass cancel request accepted. (ii) Order Mass cancel request rejected.

7.1.3 Application Messages: Other

7.1.3.1 Server-Initiated

Message	MsgType	Usage
Business Message Reject	j	Indicates that an application message could not be processed and provides a description of the error.
News	B	Disseminates market operations announcements.

7.2 Variations from the FIX Protocol

The server conforms to the FIX protocol except as follows:

1. The Order Cancel Reject, Order Mass Cancel Report, messages include the field ApplID (1180).
2. The Order Cancel Reject message includes the NoPartyIDs (453) block which was introduced in Extension Pack 115.

7.3 Message Header and Trailer

7.3.1 Message Header

Tag	Field Name	Req	Description
8	BeginString	Y	FIXT.1.1
9	BodyLength	Y	Number of characters after this field up to and including the delimiter immediately preceding the CheckSum.
35	MsgType	Y	Message type.
49	SenderCompID	Y	Interface User ID (CompID) of the party sending the message. This identifies the Interface User.
56	TargetCompID	Y	Interface User ID (CompID) of the party the message is sent to.
34	MsgSeqNum	Y	Sequence number of the message.
43	PossDupFlag	N	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	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	N	Time the message was transmitted. Not required for incoming messages sent by the clients (even if sent by a client, no validation will be done). Required for outgoing messages sent by the server.
122	OrigSendingTime	N	Time the message was originally transmitted. If the original time is not available, this will be the same value as SendingTime (52). Required if PossDupFlag (43) is Possible Duplicate (Y).
1128	ApplVerID	N	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	Description
10	CheckSum	Y	

7.4 Administrative Messages

7.4.1 Logon

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	A = Logon
Message Body			
98	EncryptMethod	Y	Method of encryption. **Value** **Meaning** 0 None
108	HeartBtInt	Y	Indicates the heartbeat interval in seconds.
141	ResetSeqNum Flag	N	Indicates whether the client and server will 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	Password assigned to the Interface User (CompID). Required if the message is generated by the client. *This will be dependent on the password policy the JSE decides to enable.*
925	NewPassword	N	New password for the Interface User ID (CompID). *This will be dependent on the password policy the JSE decides to enable.*
1409	SessionStatus	N	Status of the FIX session. Required if the message is generated by the server. **Value** **Meaning** 0 Session Active 2 Password Due to Expire *This will be dependent on the password policy the JSE decides to enable.*
1137	DefaultApplVerID	Y	Default version of FIX messages used in this session which will be validated by the server. **Value** **Meaning** 9 FIX50SP2
Standard Trailer			

7.4.2 Logout

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	5 = Logout
Message Body			
1409	SessionStatus	N	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	This field will contain the next expected sequence number as well as the received sequence number if the server terminated the connection after receiving a sequence number that was less than what was expected. In other cases the field will contain the reason for the logout.
Standard Trailer			

7.4.3 Heartbeat

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

7.4.4 Test Request

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

7.4.5 Resend Request

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

7.4.6 Reject

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	3 = Reject
Message Body			
45	RefSeqNum	Y	MsgSeqNum (34) of the rejected message.
372	RefMsgType	N	MsgType (35) of the rejected message.
371	RefTagID	N	If a message is rejected due to an issue with a particular field its tag number will be indicated.
373	SessionReject Reason	N	Code specifying the reason for the reject. Please refer to Volume 10 – Reject Codes and Reasons
58	Text	N	Text specifying the reason for the rejection.
Standard Trailer			

7.4.7 Sequence Reset

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	4 = Sequence Reset
Message Body			
36	NewSeqNo	Y	Sequence number of the next message to be transmitted.
123	GapFillFlag	N	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: Order Handling

7.5.1 New Order – Single

	Field Name	Req	Description
Standard Header			
35	MsgType	Y	D = New Order - Single
Message Body			
11	ClOrdID	Y	Client specified identifier of the order.
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
Component Block <Trading Party>		Y	Identifier of the trading party.
1	Account	Y	Client Account information. This is the Client Account of the firm who is sending the order. Mandatory 8 Digit value for the JSE Markets. Ensure it is populated and is numeric.
48	SecurityID	Y	Identifier of the instrument. (Instrument ID).
22	SecurityIDSource	Y	Identifier of the source of the SecurityID (48) value. **Value** **Meaning** 8 Exchange Symbol
40	OrdType	Y	Type of the order. **Value** **Meaning** 1 Market 2 Limit 3 Stop 4 Stop Limit P Pegged Order R Pegged Limit Order

1094	PegPriceType	N	Required if the Order Type is Pegged (P) or Pegged Limit (R). 2 = Mid-Price Peg 5 = Primary Peg (Buy at Bid, Sell at Offer) (Primary Peg value will be considered Pegged to Bid when side is Buy and Pegged to Offer when side is Sell)

<table>
<tr><td>59</td><td colspan="2">TimeInForce</td><td>N</td><td>Time qualifier of the order. Absence of this field is interpreted as DAY (0).
Value Meaning
0 DAY
1 Good Till Cancel (GTC)
2 At the Open (OPG)
3 Immediate or Cancel (IOC)
4 Fill or Kill (FOK)
6 Good Till Date (GTD)
7 At the Close (ATC)
8 Good for Intra-Day Auction (GFX)
9 Good for Auction (GFA)
(GTT is specified using 6 in this field and specifying an Expiry Time.)</td></tr>
<tr><td>126</td><td colspan="2">ExpireTime</td><td>N</td><td>Time the order expires which must be a time during the current trading day. Required if TimeInForce (59) is GTD (6) and ExpireDate (432) is not specified. This field will only be populated if the order is a GTT order.
If both the ExpireTime (126) and ExpireDate (432) are specified, it will be considered as a GTT order by the server. If a GTT order is sent with an already elapsed ExpireTime but with a future Date component, the order will be rejected.
Format: YYYYMMDD-HH:MM:SS</td></tr>
<tr><td>432</td><td colspan="2">ExpireDate</td><td>N</td><td>Date the order expires. Required if TimeInForce (59) is GTD (6) and ExpireTime (126) is not specified. This field will only be populated if the order is a GTD order
Format: YYYYMMDD</td></tr>
<tr><td>386</td><td colspan="2">NoTrading Sessions</td><td>N</td><td>Number of sessions the order is valid for. If specified, the value in this field should always be “1”.</td></tr>
<tr><td>➡</td><td>336</td><td>Trading Session ID</td><td>N</td><td>Session the order is valid for.
Value Meaning
a Closing Price Cross
b Good for Volume Auction Uncross (GDX)</td></tr>
<tr><td>54</td><td colspan="2">Side</td><td>Y</td><td>Side of the order taken by the submitting firm. Valid values are 1 and 2.
Value Meaning
1 Buy
2 Sell</td></tr>
</table>

30001	OrderBook	Y	Identifier of the order book. **Value** **Meaning** 1 Normal
38	OrderQty	Y	Total order quantity.
110	MinQty	N	Minimum Execution Size that needs to be specified for a Hidden Pegged and Pegged Limit Order which must be greater than or equal to Minimum Reserve Size. If the field DisplayMethod (1084) has a Value of 4, and the Order Type is Pegged Order (P) or Pegged Limit Order (R), it is mandatory to specify MinQty.
1138	DisplayQty	Y	This is the Visible Size of the order. It will be zero for a Hidden Pegged and Pegged Limit and GDX order and equal to order quantity displayed to the market for a normal order. If the field DisplayMethod (1084) has a value of 4, DisplayQty will be zero regardless of the value specified.
1084	DisplayMethod	N	This field is conditionally required for a Hidden Pegged and Pegged Limit order or hidden (GDX) order. **Value** **Meaning** 4 Undisclosed ~~(Hidden Pegged and Pegged Limit Order)~~ Any other value in this field will be rejected.
44	Price	N	Limit price. Required if OrderType (40) is Limit (2) or Stop Limit (4).
99	StopPx	N	Stop price. Required if OrderType (40) is Stop (3), ~~or~~ Stop Limit (4) or Pegged Limit (R).
528	OrderCapacity	Y	Capacity of the order. **Value** **Meaning** A Agency P Principal
60	TransactTime	Y	Time the order was created.

18	ExecInst	N	**Value** **Meaning** v Exclude Hidden Orders u Include in Volume Auction Uncross Required if OrdType (40) field Limit (2) and the TimeInForce (59) field is GTT (6) and the ExpireTime (126) field is specified. User should be able to submit value (u) if OrdType (40) field Limit (2) or Market (1) and the TimeInForce (59) field is Day (0) and user intends the order to participate in EOD Volume Auction uncross session. User should be able to submit value for this field if OrdType (40) is Pegged (P) or Pegged Limit (R). However, any Pegged orders with OrdType (40) Pegged (P) will expire at the start of the Closing Auction Call regardless of the ExecInst (18).
Standard Trailer			

7.5.2 Order Cancel Request

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	F = Order Cancel Request
Message Body			
11	ClOrdID	Y	Client specified identifier of the cancel request.
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
41	OrigClOrdID	N	ClOrdID (11) of the order being cancelled. Required if OrderID (37) is not specified.
37	OrderID	N	Server specified identifier of the order being cancelled. Required if OrigClOrdID (41) is not specified.
48	SecurityID	Y	Identifier of the instrument.
22	SecurityIDSource	Y	Identifier of the source of the SecurityID (48) value. **Value** **Meaning** 8 Exchange Symbol
Component Block <Trading Party>		Y	Identifier of the trading party.
54	Side	Y	Must match the value in the order.
30001	OrderBook	N	Identifier of the order book. Absence of this field is interpreted as Normal (1). **Value** **Meaning** 1 Normal
60	TransactTime	Y	Time the order cancel request was created.

Standard Trailer

7.5.3 Order Mass Cancel Request

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	q = Order Mass Cancel Request
Message Body			
11	ClOrdID	Y	Client specified identifier of the mass cancel request.
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
530	MassCancel RequestType	Y	Scope of the mass cancel request. **Value** **Meaning** 1 Cancel All Firm Orders for Instrument 7 Cancel All Firm Orders 9 Cancel All Firm Orders for Segment
48	SecurityID	N	Identifier of the instrument.(Instrument ID) Required if the scope of mass cancel is for an instrument.
30001	OrderBook	N	Identifier of the order book. Absence of this field is interpreted as Normal (1). **Value** **Meaning** 1 Normal
22	SecurityIDSource	N	Identifier of the source of the SecurityID (48) value. Required if SecurityID (48) is specified. **Value** **Meaning** 8 Exchange Symbol
1461	NoTargetPartyIDs	Y	Number of parties the mass cancel relates to for the JSE and NSX Markets that requires 3 blocks to represent the following: 1. Trader 2. Trader Group 3. Firm The Trader and Trader Group will be concatenated in the System to uniquely identify the Trader. If specified, the value in this field will be: 1 - Executing Firm 2 - Trader and Trader group combinations
➡	1462 TargetPartyID	Y	Identifier of the party the mass cancel relates to. Required if NoTargetPartyIDs (1461) is specified.

➡	1463	TargetParty IDSource	Y	Required if NoTargetPartyIDs (1461) is specified. **Value** **Meaning** D Proprietary/Custom Code
➡	1464	TargetParty Role	Y	Role of the TargetPartyID (1462). Required if NoTargetPartyIDs (1461) is specified. **Value** **Meaning** 1 Executing Firm 53 Trader 76 Trader Group
1300	MarketSegmentID		N	Identifier of the segment the order mass cancel request relates to. Please refer to Section 8 for the valid segments. Required if MassCancelRequestType (530) is Cancel All for Segment (9).
60	TransactTime		Y	Time the mass cancel request was created.
Standard Trailer				

7.5.4 Order Cancel/Replace Request

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	G = Order Cancel/Replace Request
Message Body			
11	ClOrdID	Y	Client specified identifier of the cancel/replace request.
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
41	OrigClOrdID	N	ClOrdID (11) of the order being amended. Required if OrderID (37) is not specified.
37	OrderID	N	Server specified identifier of the order being amended. Required if OrigClOrdID (41) is not specified.
Component Block <Trading Party>		Y	Identifier of the trading party.
1	Account	Y	Client Account information. This is the Client Account of the firm who is requesting the order cancel/replace Mandatory 8 Digit value for the JSE Markets. Ensure it is populated and is numeric.
48	SecurityID	Y	Identifier of the instrument. (Instrument ID).
22	SecurityIDSource	Y	Identifier of the source of the SecurityID (48) value. **Value** **Meaning** 8 Exchange Symbol
40	OrdType	Y	Must match the value in the order that is being cancelled/replaced.
59	TimeInForce	Y	Time qualifier of the order. **Value** **Meaning** 0 Day 1 Good Till Cancel (GTC) 2 At the Open (OPG) 3 Immediate or Cancel (IOC) 4 Fill or Kill (FOK) 6 Good Till Date (GTD) 7 At the Close (ATC) 8 Good for Intra-Day Auction (GFX) 9 Good for Auction (GFA) (GTT is specified using 6 in this field and specifying an Expire Time.)
126	ExpireTime	N	Time the order expires which must be a time during the current trading day. Required if TimeInForce (59) is GTD (6) and ExpireDate (423) is not specified.
432	ExpireDate	N	Date the order expires. Required if TimeInForce (59) is GTD (6) and ExpireTime (126) is not specified.

54	Side		Y	Must match the value in the order.
30001	OrderBook		N	Identifier of the order book. Absence of this field is interpreted as Normal (1). **Value** **Meaning** 1 Normal
386	NoTrading Sessions		N	Number of sessions the order is valid for. If specified, the value in this field should always be "1".
➡	336	Trading Session ID	N	Session the order is valid for. **Value** **Meaning** a Closing Price Cross b Good for Volume Auction Uncross (GDX)
38	OrderQty		Y	Total order quantity.
1138	Display Qty		Y	This is the Visible Size of the order. It will be zero for a Hidden Pegged, ~~and~~ Pegged Limit and GDX order~~.~~ and equal to order quantity displayed to the market for a normal order. If the field DisplayMethod (1084) has a value of 4, DisplayQty will be zero regardless of the value specified.
1084	DisplayMethod		N	Whether the order is a Hidden Pegged and Pegged Limit order. Required for hidden orders. (GDX orders) **Value** **Meaning** 4 Undisclosed ~~(Hidden Pegged and Pegged Limit Order)~~
110	MinQty		N	Minimum Execution Size that needs to be specified for a Hidden Pegged and Pegged Limit Order which must be greater than or equal to Minimum Reserve Size.
44	Price		N	Limit price. Required if OrderType (40) is Limit (2) or Stop Limit (4).
99	StopPx		N	Stop price. Required if OrderType (40) is Stop (3), ~~or~~ Stop Limit (4) or Pegged Limit (R).
60	TransactTime		Y	Time the cancel/replace request was created.
Standard Trailer				

7.5.5 Execution Report

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	8 = Execution Report
Message Body			
1180	ApplID	Y	Identity of the partition.

17	ExecID	Y	Server specified identifier of the message.
11	ClOrdID	Y	Client specified identifier of the order. If the execution report is generated as a response to an order cancel or order mass cancel request, this will be the client order ID specified in the order cancel or order mass cancel request. If a client order ID is not specified in the order cancel or order mass cancel request, this will be the original client order ID of the order being cancelled.[1]
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
41	OrigClOrdID	N	~~OrigClOrdID (41), if any that was submitted with the order cancel or cancel or replace request.~~ClOrdID (11), of the order which has been amended or cancelled. Stamped only in the immediate ER generated to convey an amendment/cancellation.[2]
37	OrderID	Y	Server specified identifier of the order. This will be a 62 base encoded value in ASCII format. By converting this to binary, this can be mapped with ITCH Order ID (refer to Section 3.10).
150	ExecType	Y	Reason the execution report was generated. **Value** **Meaning** 0 New 4 Cancelled 5 Replaced 8 Rejected C Expired D Restated F Trade G Trade Correct H Trade Cancel 9 Suspended L Triggered
19	ExecRefID	N	Reference to the execution being cancelled or corrected. Required if ExecType (150) is Trade Cancel (H) or Trade Correct (G).

[1] In an amendment/cancellation the ClOrdID(11) submitted with the order cancel or cancel/replace request will be stamped here
[2] Any subsequent ERs sent regarding to any executions, expirations etc. of the order will not be stamped the OrigClOrdID (41).

<table>
<tr><td>378</td><td>Exec Restatement Reason</td><td>N</td><td>This is used to indicate if an order was Cancelled or Cancel/Replaced by JSE Market Operations. Required if ExecType (150) is Cancelled (4) or Replaced (5).
Value Meaning
3 Order Re-priced
8 Market Option
99 Other</td></tr>
<tr><td>39</td><td>OrdStatus</td><td>Y</td><td>Current status of the order.
Value Meaning
0 New
1 Partially Filled
2 Filled
4 Cancelled
8 Rejected
C Expired
9 Suspended</td></tr>
</table>

Tag	Field Name	Req	Description
636	Working Indicator	N	~~This is used to send notification upon Stop/Stop Limit order and a CPX TIF order election.~~ Whether the order is currently being worked (elected) or not. **Value** **Meaning** N — Order is Not in a Working State (Order is accepted but in an unelected state) Y — Order is Being Worked (Order is accepted and elected).
30001	OrderBook	Y	**Value** **Meaning** 1 — Normal
103	OrdRejReason	N	Code specifying the reason for the reject. Please refer to Volume 10 – Reject Codes and Reasons for a list of reject codes. Required if ExecType (150) is Rejected (8).
58	Text	N	Text specifying the reason for the rejection or expiration
32	LastQty	N	Quantity executed in this fill. Required if ExecType (150) is Trade (F) or Trade Correct (G).
31	LastPx	N	Price of this fill in ZAC. Required if ExecType (150) is Trade (F) or Trade Correct (G).
151	LeavesQty	Y	Quantity available for further execution. It is the remaining quantity of the order. Will be “0” if OrdStatus (39) is Filled (2), Cancelled (4), Rejected (8) or Expired (C).
14	CumQty	Y	Total cumulative quantity filled.
48	SecurityID	Y	Identifier of the instrument. This will be the Instrument ID.
22	SecurityIDSource	Y	Identifier of the source of the SecurityID (48) value. **Value** **Meaning** 8 — Exchange Symbol
Component Block <Trading Party>		Y	Values specified in the order.
1	Account	Y	Client Account information. This is the Client Account of the firm who is sending the order.
40	OrdType	Y	Value submitted with the order.

59	TimeInForce	N	Value submitted with the order.
126	ExpireTime	N	Value submitted with the order.
432	ExpireDate	N	Value submitted with the order.
54	Side	Y	Value submitted with the order.
38	OrderQty	Y	Value submitted with the order.
1138	DisplayQty	N	Quantity currently displayed in the order book. This is the Visible Size. It is equal to zero for Hidden Pegged, ~~and~~ Pegged Limit and GDX orders and equal to the order quantity for normal orders. If the field DisplayMethod (1084) has a value of 4, DisplayQty will be zero regardless of the value specified.
1084	DisplayMethod	N	~~Whether the order is a Hidden Pegged and Pegged Limit order.~~ Required for hidden orders. **Value** **Meaning** 4 Undisclosed ~~(Hidden Pegged and Pegged Limit Order)~~
110	Min Qty	N	Minimum Execution Size that needs to be specified for a Hidden Pegged and Pegged Limit Order and needs to be greater than or equal to Minimum Reserve Size.
44	Price	N	Value submitted with the order.
99	StopPx	N	Value submitted with the order.
528	OrderCapacity	Y	Capacity of the order. **Value** **Meaning** A Agency P Principal
60	TransactTime	N	Time the transaction represented by the Execution Report occurred. Not populated if the ExecType (150) = 8 (Rejected)
880	TrdMatchID	N	Identifier of the trade. Required if ExecType (150) is Trade (F), Trade Correct (G) or Trade Cancel (H).
27000	IsMarketOpsRequest	N	Identifies whether an order was submitted on behalf of a client by Market Operations. **Value** **Meaning** 1 Yes

18	ExecInst	N	**Value** **Meaning** v Exclude Hidden Orders u Include in Volume Auction Uncross
548	Cross ID	N	The unique ID of the cross order. Only populated for execution report messages generated for cross orders. The value submitted with the New Order Cross Message will be populated.
549	Cross Type	N	The type of the cross order. Only populated for execution report messages generated for cross orders. The value submitted with the New Order Cross Message will be populated.
1094	PegPriceType	N	Required if the Order Type is Pegged or Pegged Limit. 2 = Mid-Price Peg 5 = Primary Peg (Buy at Bid, Sell at Offer)
Standard Trailer			

7.5.6 Order Cancel Reject

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	9 = Order Cancel Reject
Message Body			
11	ClOrdID	Y	ClOrdID (11) that was submitted with the order cancel or cancel/replace request being rejected.
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
1180	ApplID	Y	Identity of the partition.
41	OrigClOrdID	N	OrigClOrdID (41), if any, that was submitted with the order cancel or cancel/replace request being rejected.
37	OrderID	Y	Server specified identifier of the order for which the cancel or cancel/replace was submitted. Will be “NONE” if the order is unknown.
Component Block <Trading Party>		Y	Values specified in the order cancel or cancel/replace request.

39	OrdStatus	Y	Current status of the order. Will be Rejected (8) if the order is unknown. **Value** **Meaning** 0 New 1 Partially Filled 2 Filled 4 Cancelled 8 Rejected C Expired 9 Suspended
434	CxlRej ResponseTo	Y	Type of request being rejected. **Value** **Meaning** 1 Order Cancel Request 2 Order Cancel/Replace Request
30001	OrderBook	Y	**Value** **Meaning** 1 Normal
102	CxlRejReason	Y	Code specifying the reason for the rejection. Please refer to Volume 10 – Reject Codes and Reasons for a list of reject codes.
58	Text	N	Text specifying the reason for the rejection.
60	TransactTime	Y	Time the reject was generated by the System.
Standard Trailer			

7.5.7 Order Mass Cancel Report

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	r = Order Mass Cancel Report
Message Body			
1180	ApplID	Y	Identity of the partition.
1369	MassActionReportID	Y	Server specified identifier of the message.
11	ClOrdID	Y	Client specified identifier of mass cancel request.
526	SecondaryClOrdID	N	Assigned by the party which originates the order. Can be used to include an additional unique identifier.
30001	OrderBook	Y	**Value** **Meaning** 1 Normal
530	MassCancel RequestType	Y	Value specified in the mass cancel request.

531	MassCancel Response	Y	Action taken by server. **Value** **Meaning** 0 Mass Cancel Request Rejected 1 Cancelled All Orders for Instrument 7 Cancelled All Orders 9 Cancelled All Orders for Segment
532	MassCancelReject Reason	N	Code specifying the reason for the rejection. Please refer to Volume 10 – Reject Codes and Reasons for a list of reject codes. Required if MassCancelResponse (531) is Mass Cancel Request Rejected (0).
Standard Trailer			

7.5.8 New Order Cross

	Field Name		Req	Description
Standard Header				
35	MsgType		Y	s = New Order Cross
Message Body				
548	Cross ID		Y	An identifier of the cross order. This will be unique across the trading day.
549	Cross Type		~~N~~Y	The type of the cross order. **Value** **Meaning** 5 Internal Cross
550	Cross Prioritization		Y	Indicates if one side or the other of a cross order should be prioritized. The value always has to be 0 (None). Any other value will be rejected via a Reject message.
552	NoSides		Y	Number of party identifiers. The value in this field should be “2” for Cross Type (549) = 5 Any other combination should be rejected via a Reject message.
➡	54	Side	Y	Side of the Order **Value** **Meaning** 1 Buy 2 Sell Any other value will be rejected via a Reject message.
➡	11	ClOrdID	Y	Client specified identifier of the order
➡	526	SecondaryClOrdID	N	Can be used to include an additional unique identifier.
➡	1	Account	Y	Client Account information. This is the Client Account of the firm Mandatory 8 Digit numeric value for the JSE Markets.
➡	528	Order Capacity	Y	Capacity of the Order **Value** **Meaning** A Agency P Principal Any other value will be rejected via a Reject message.

➡	453		NoPartyIDs	Y	Number of party identifiers. JSE Market requires 3 blocks to represent the following: 4. Trader 5. Trader Group 6. Firm The Trader and Trader Group will be concatenated in the System to uniquely identify the Trader ID. The value in this field will be 2 or 3. 2- Trader and Trader group combination. 3-Executing Firm, Trader and Trader group combination. It will be 2 for all client initiated messages as it is not required to enter Firm ID. It will be 3 for all server initiated Execution Reports as the Firm will be in the returned message. Any other value should be rejected via a Reject message.
➡	➡	448	PartyID	Y	Identifier of the Firm
➡	➡	447	PartyID Source	Y	Required if PartyID (448) is specified **Value** **Meaning** D Proprietary/Custom Code
➡	➡	452	Party Role	Y	Role of the specified PartyID (448) **Value** **Meaning** 1 Executing Firm 76 Trader Group 53 Trader Any other value will be rejected via a Reject message.
➡	54		Side	Y	Side of the Order **Value** **Meaning** 1 Buy 2 Sell Any other value will be rejected via a Reject message.
➡	11		ClOrdID	Y	Client specified identifier of the order
➡	526		SecondaryClOrdID	N	Can be used to include an additional unique identifier.
➡	1		Account	Y	Client Account information. This is the Client Account of the firm Mandatory 8 Digit value for the JSE Markets. Ensure it is populated and is numeric.
➡	528		Order Capacity	Y	Capacity of the Order **Value** **Meaning** A Agency P Principal Any other value will be rejected via a Reject message.

➡	453		NoPartyIDs	Y	Number of party identifiers. JSE Market requires 3 blocks to represent the following: 1. Counterparty Trader 2. Counterparty Trader Group 3. Counterparty Firm The Trader and Trader Group will be concatenated in the System to uniquely identify the Trader ID. The value in this field will be 2 or 3. 2- Trader and Trader group combination. 3-Executing Firm, Trader and Trader group combination. It will be 2 for all client initiated messages as it is not required to enter Firm ID. It will be 3 for all server initiated Execution Reports as the Firm will be in the returned message. Any other value should be rejected via a Reject message.
➡	➡	448	PartyID	Y	Identifier of the Firm
➡	➡	447	PartyID Source	Y	Required if PartyID (448) is specified **Value** **Meaning** D Proprietary/Custom Code
➡	➡	452	Party Role	Y	Role of the specified PartyID (448) **Value** **Meaning** 17 Counterparty Firm 100 Counterparty Trader Group 37 Counterparty Trader Any other value will be rejected via a Reject message.
48	SecurityID			Y	Identifier of the instrument. (Instrument ID).
22	SecurityIDSource			Y	Identifier of the source of the SecurityID (48) value. **Value** **Meaning** 8 Exchange Symbol
60	TransactTime			Y	Time the order was created.
44	Price			Y	Price of the cross order
38	OrderQty			Y	Total order quantity
40	OrdType			Y	Type of the Order **Value** **Meaning** 2 Limit Order Any other value will be rejected via a Session Reject message
59	TimeInForce			N	Time qualifier of the order **Value** **Meaning** 0 Day Any other value will be rejected via a Session Reject message
Standard Trailer					

7.6 Application Messages: Others

7.6.1 News

Tag	Field Name		Req	Description
Standard Header				
35	MsgType		Y	B = Market Operations Announcement
Message Body				
1180	ApplID		Y	Identity of the partition.
42	OrigTime		Y	Time the announcement was published which will be specified in UTC and in the YYYYMMDD-HH:MM:SS format.
61	Urgency		Y	Level of urgency of the announcement. **Value** **Meaning** 0 Normal 1 Flash (High Priority) 2 Background (Low Priority)
148	Headline		Y	Headline or subject of the announcement.
33	NoLinesOfText		Y	Number of lines of text. The value in this field will always be “1”.
➡	58	Text	Y	Text of the announcement.
146	NoRelatedSym		N	Number of related instruments.
➡	48	Security ID	N	Unique identifier of the instrument. Required if NoRelatedSym (146) is specified.
➡	22	SecurityIDSource	Y	Identifier of the source of the SecurityID (48) value. **Value** **Meaning** 8 Exchange Symbol
Standard Trailer				

7.6.2 Business Message Reject

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	j = Business Message Reject
Message Body			
379	BusinessReject RefID	N	Client specified identifier (e.g. ClOrdID) of the rejected message if it is available.
45	RefSeqNum	Y	MsgSeqNum (34) of the rejected message.
372	RefMsgType	Y	MsgType (35) of the rejected message.
371	RefTagID	N	If a message is rejected due to an issue with a particular field its tag number will be indicated.
380	BusinessReject Reason	Y	Code specifying the reason for the rejection. Please refer to Volume 10 – Reject Codes and Reasons for a list of reject codes.
58	Text	N	Text specifying the reason for the rejection (not including TagID. See RefTagID for this information).
Standard Trailer			

7.7 Components of Application Messages

7.7.1 Trading Party

Tag	Field Name		Req	Description
453	NoPartyIDs		Y	Number of party identifiers. JSE Market requires 3 blocks to represent the following: 1. Trader 2. Trader Group 3. Firm The Trader and Trader Group will be concatenated in the System to uniquely identify the Trader ID. The value in this field will be 2 or 3. 2- Trader and Trader group combination. 3-Executing Firm, Trader and Trader group combination. It will be 2 for all client initiated messages as it is not required to enter Firm ID. It will be 3 for all server initiated Execution Reports as the Firm will be in the returned message. It will be 2 (Trader, Trader Group) for server initiated Order Cancel Reject messages.
➡	448	PartyID	Y	Identifier of the party.
➡	447	PartyID Source	Y	Required if PartyID (448) is specified. **Value** **Meaning** D Proprietary/Custom Code
➡	452	Party Role	Y	Party Role of the specified PartyID (448). It will be mandatory to have Party Role Trader Group (76) and Trader (53) for the New Order – Single, Order Cancel Request, Order Cancel/Replace Request messages. **Value** **Meaning** 53 Trader 76 Trader Group 1 Executing Firm

8 SEGMENTS

The following segments will be available for the JSE and NSX Markets.

Segment	
ZA01	JSE Top Companies
ZA02	JSE Medium Liquid
ZA03	JSE Less Liquid
ZA04	JSE Specialist Products
ZA06	JSE Exchange Traded Products
ZA11	NSX Local Listed Companies
ZA12	NSX Dual Listed Companies

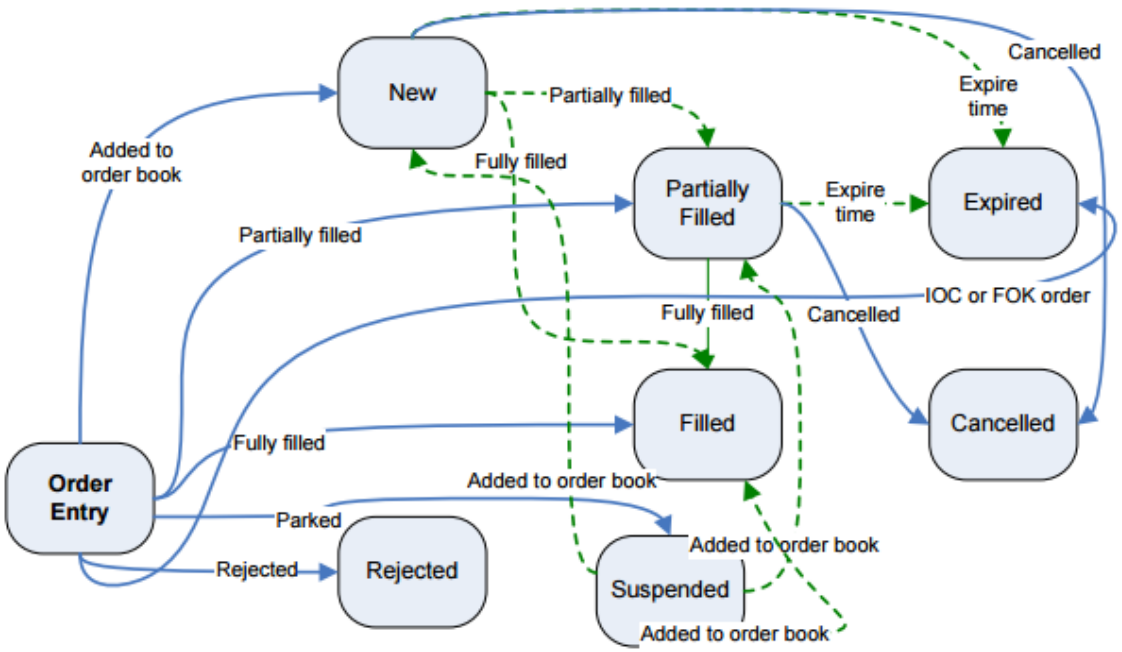
9 REJECT CODES

Please refer to Volume 10 - JSE Reject Codes Specification for the full list of reject codes and meanings specific to the ~~JSE and NSX~~System applicable across markets.

10 PROCESS FLOWS

10.1 Order Handling

10.1.1 Order Status Changes



10.1.3 **JSE Market Operations Actions**