

Johannesburg Stock Exchange

Trading and Information Solution

JSE Specification Document Volume 01E – Basic Native Trading Gateway

| | |
|------------------------|-----------------------------|
| Version | 4.05 |
| Release Date | 28 May 2024 |
| Number of Pages | 83 (Including Cover Page) |

1 DOCUMENT CONTROL

1.1 Table of Contents

| | | |
|----------|--|-----------|
| 1 | DOCUMENT CONTROL | 2 |
| 1.1 | Table of Contents | 2 |
| 1.2 | Document Information | 4 |
| 1.3 | Revision History..... | 4 |
| 1.4 | References | 7 |
| 1.5 | Contact Details | 8 |
| 1.6 | Definitions, Acronyms and Abbreviations..... | 9 |
| 2 | OVERVIEW | 11 |
| 2.1 | Types of Native Gateways | 11 |
| 2.2 | Order management, Quote Management and other services..... | 11 |
| 2.3 | Colocation and Client Access Network (CAN) services | 11 |
| 3 | SERVICE DESCRIPTION | 12 |
| 3.1 | System Architecture | 12 |
| 3.2 | Order Handling | 12 |
| 3.2.1 | Order Types | 13 |
| 3.2.2 | Order Time in Force (TIF) | 16 |
| 3.2.3 | Cross Orders..... | 17 |
| 3.2.4 | Pegged Orders (Hidden)..... | 18 |
| 3.2.5 | Order Capacity..... | 18 |
| 3.2.6 | Order Management..... | 18 |
| 3.2.7 | Unsolicited Order Updates..... | 20 |
| 3.2.8 | Self-Match Prevention | 20 |
| 3.2.9 | Trade Cancellations..... | 20 |
| 3.3 | Gateway Validations..... | 20 |
| 3.4 | Quotation Conventions..... | 21 |
| 3.5 | Market Operations | 21 |
| 3.5.1 | Order Management..... | 21 |
| 3.5.2 | Trade Cancellations and Corrections | 21 |
| 3.6 | Conditionally required fields | 22 |
| 3.7 | Time Stamps and Dates..... | 22 |
| 3.8 | Market Operations Announcements..... | 22 |
| 3.9 | Mapping Native Order ID to MITCH Order ID | 24 |
| 3.10 | Execution Report | 26 |
| 3.11 | Order Status | 28 |
| 3.12 | Orders and Execution Identifiers | 29 |
| 3.12.1 | Client Order IDs | 29 |
| 3.12.2 | Execution IDs..... | 29 |
| 3.12.3 | Order IDs | 29 |
| 3.12.4 | Public Order IDs..... | 29 |
| 3.12.5 | Instrument Identification..... | 29 |
| 3.13 | Party Identification | 29 |
| 3.14 | Field Value Validations..... | 30 |
| 3.15 | Validation of ASCII characters | 34 |
| 3.16 | Rejection Logic..... | 34 |
| 3.17 | Functional and Implementation Limitations..... | 34 |
| 4 | CONNECTIVITY | 37 |
| 4.1 | Interface User ID (ComplIDs)..... | 37 |
| 4.2 | Passwords | 37 |
| 4.3 | Production IP Addresses and Ports | 37 |
| 4.4 | Failover and Recovery | 37 |
| 4.5 | Establishing a Connection | 38 |

| | | |
|----------|--|-----------|
| 4.5.1 | Maintaining a Connection | 38 |
| 4.5.2 | Application Sequence Numbers | 38 |
| 4.5.3 | Heartbeats | 38 |
| 4.6 | Terminating a Connection | 39 |
| 4.7 | Message Rate Throttling | 39 |
| 4.8 | Mass Cancellation On Disconnect | 39 |
| 4.9 | Unavailability of Real-Time Channel | 40 |
| 4.10 | Backward Compatibility | 40 |
| 5 | RECOVERY | 41 |
| 5.1 | Establishing a Connection | 41 |
| 5.2 | Heartbeats | 41 |
| 5.3 | Requesting Missed Messages | 41 |
| 5.4 | Response to a Missed Message Request..... | 42 |
| 5.5 | Termination of the Recovery Connection | 43 |
| 5.6 | Unavailability of Recovery Channel..... | 43 |
| 6 | MESSAGE FORMATS..... | 45 |
| 6.1 | Data Types | 45 |
| 6.2 | Message Overview | 45 |
| 6.2.1 | Administrative Messages..... | 45 |
| 6.2.2 | Application Messages of Basic Gateway: Order Handling | 47 |
| 6.2.3 | Other Application Messages of the Basic Gateway..... | 48 |
| 6.3 | Message Header | 49 |
| 6.4 | Administrative Messages | 50 |
| 6.4.1 | Logon | 50 |
| 6.4.2 | Logon Response..... | 50 |
| 6.4.3 | Logout..... | 50 |
| 6.4.4 | Heartbeat | 50 |
| 6.4.5 | Reject..... | 51 |
| 6.4.6 | Missed Message Request | 51 |
| 6.4.7 | Missed Message Request Ack | 51 |
| 6.4.8 | Transmission Complete | 51 |
| 6.4.9 | System Status..... | 52 |
| 6.5 | Application Messages of the Basic Gateway: Client-Initiated | 53 |
| 6.5.1 | New Order – Protocol Version 2 | 53 |
| 6.5.2 | New Order – Protocol Version 3..... | 57 |
| 6.5.3 | Order Cancel Request | 61 |
| 6.5.4 | Order Mass Cancel Request | 62 |
| 6.5.5 | Order Cancel/ Replace Request..... | 63 |
| 6.5.6 | New Order Cross | 66 |
| 6.6 | Application Messages of the Basic Gateway: Server-Initiated..... | 69 |
| 6.6.1 | Execution Report - Protocol Version: 2 | 69 |
| 6.6.2 | Execution Report - Protocol Version: 3 | 74 |
| 6.6.3 | Order Cancel Reject | 79 |
| 6.6.4 | Order Mass Cancel Report | 79 |
| 6.7 | Other Application Messages of the Basic Gateway: Server-Initiated..... | 80 |
| 6.7.1 | News | 80 |
| 6.7.2 | Business Reject | 80 |
| 7 | REJECT CODES..... | 81 |
| 7.1 | Login Response..... | 81 |
| 7.2 | Reject | 81 |
| 7.3 | Business Reject..... | 81 |
| 8 | PROCESS FLOWS | 82 |
| 8.1 | Order Status Changes..... | 82 |
| 8.1.1 | General | 82 |
| 8.1.2 | Market Operations Actions | 83 |

1.2 Document Information

| | |
|---------------------|-----------------------------|
| Drafted By | JSE |
| Status | Final |
| Version | 4.05 |
| Release Date | 28 May 2024 |

1.3 Revision History

| Date | Version | Description |
|------------------|---------|--|
| 08 July 2011 | 1.00 | Initial Draft |
| 30 November 2011 | 1.01 | JSE Specification Updates |
| 20 June 2012 | 1.02 | JSE Specification clarifications and updates |
| 31 January 2013 | 1.03 | JSE Specification minor clarifications and updates |
| 5 July 2013 | 2.00 | Functionality updates related to the 2013 product upgrade |
| 4 November 2013 | 2.01 | Minor specification correction to Order Mass Cancel report message |
| 20 February 2014 | 2.02 | Functionality updates for the introduction of the JSE Colocation service |
| 22 August 2014 | 2.03 | Introducing the ability to submit Exclude Hidden Limit Orders |
| 9 February 2016 | 3.00 | Integrated Trading and Clearing Project changes. Equity Market Enhancements: <ul style="list-style-type: none"> • Hidden Order functionality enhanced • Introduction of On Book Cross Order Trade • Introduction of EOD Volume Auction |
| 26 April 2016 | 3.01 | Cross Orders updated with more details Mass Cancellation details expanded Execution Instruction Field changed to Int8 Order Sub Type field removed from Order Cancel Replace Request message |
| 4 August 2016 | 3.02 | Updated relevant fields for GDY Added Display Quantity of a GDY should be zero. Updated Uint32 to Int32 of Security ID and Order Quantity fields Contra Trader field removed from Order Cancel Request message Increased length of Contra Firm field to 11 Increased length of Market Maker Firm field to 11 |
| 19 October 2016 | 3.03 | Updated Description of Execution Instruction field |
| 31 January 2017 | 3.04 | Partition ID added to definitions section Definition for IRD updated Order Book field data type changed to UInt8 in Quote Request Reject and Quote Status Report messages Multi Leg Reporting Type field data type changed to UInt8 in Execution Report Descriptions added to RP and SR for Trade Sub Type field in Quote Request message Multi Leg Reporting Type field added to RFQ Execution Report Detail added to event 12 regarding the message sent out for a Quote Cancel event in Private RFQ Negotiation |

| | | | |
|-----------------|------|--|--|
| 19 April 2017 | 3.05 | <p>6.7.10 Descriptions of Near Month Type and Far Month Type fields updated</p> <p>6.8.1 Descriptions of Multi Leg reporting Type field updated</p> <p>8.2.1 Detail added to event 12 regarding the message sent out for a Quote Cancel event</p> <p>6.6.1 Description of Execution Instruction field of Execution Report message of the Basic Native Gateway updated</p> <p>6.8.1 Description of LastOptPx and Volatility fields updated</p> <p>3.16.5 Details of additional clearing member instructions on Native Enhanced Gateway added</p> <p>6.5.1, 6.7.1 Description for 'Expire Time' field updated</p> <p>6.7.1 New field 'Secondary Trade Report ID' added to New Order message</p> <p>6.7.5 New field 'Secondary Trade Report ID' added to Order Cancel/Replace Request message</p> <p>6.7.10 Description for 'Strike Price' field updated</p> <p>6.8.1 New field 'Secondary Trade Report ID' added to Execution Report message</p> | |
| 19 July 2019 | 3.06 | <p>3.2.1 and 3.2.4 Details about Adjustable Cross Order added</p> <p>3.6.1 Details regarding the assigning of the Client Order ID of the base order to Implied Out and Implied match orders added.</p> <p>3.7.2 New section added related to configuring a time period in which user creation requests are accepted by the system</p> <p>6.5.5 New enum added to Cross Type field</p> | |
| 01 August 2019 | 3.07 | 1.4 | Definition of 'EHL' removed |
| | | 3.2 | EHL Orders removed |
| | | 3.2.2 | EHL reference removed |
| | | 3.16.4 | Details on 'Public Order IDs' added |
| | | 6.5.1 | 'Execution Instruction' field updated according to changes implemented in |
| | | 6.6.1 | 'Execution Instruction' field updated. New fields 'Display Quantity' and 'Public Order ID' added |
| 04 October 2019 | 4.00 | All | Functionality updates related to the 2020 Product Upgrade |
| 04 October 2019 | 4.00 | 6.4.1 | Changes to 'Password' and 'New Password' fields on the Logon message. |
| | | 6.6.1 | Introduction of 'Last Liquidity Indicator' field to Basic Native Gateway Execution Report. |
| | | 3.7.3 | Detail on identification of rejections of user-created instruments. |
| | | 6.7.10 | Introduction of 'Reference Instrument Identifier' field to 'Security Definition Request' message. |
| 03 March 2020 | 4.00 | 3.14 | Working indicator 1 was added to ExecType = New. Working indicator for ExecType = 5 changed to 0. Working indicator for ExecType = D changed to 0. |
| | | 3.21 | Working indicator sent in Ers after an Order Cancel / Replace request was changed to 'Unset (0)'. Working indicator for ExecType = D changed to 0. |
| | | 6.8.1 | Execution Instruction for subsequent Ers published for an order was changed to 0. |
| 24 April 2020 | 4.00 | 6.4.1 | Update to the description of the Protocol Version field. |

| | | | |
|-------------------------------|----------------------|---|---|
| 10 June 2020 | 4.00 | 6.6.1 | Added value 0 to Last Liquidity Indicator |
| | | 3.11 | Added a note to indicate that TransactTime will now have a degree of nanosecond accuracy. |
| | | 4.2 | New system behavior when client sends Logon message with 'New Password' that does not comply with the password policy has been added as a note. |
| 19 October 2020 | 4.01 | 4.5 | Section updated to reflect new behaviour for establishing user connections |
| | | 6.1 | Description of Price field updated divisor changed from 10 ⁴ to 10 ⁸ |
| | | 6.6.1 | Two versions of Native Basic Execution Report message introduced (for Native protocol 1 and Native protocol 2) |
| | | 6.6.1 | New field 'TypeofTrade' introduced to Execution Report of Native Protocol 2 |
| | | 6.7.1, 6.7.3, 6.7.4, 6.7.5, 6.8.1, 6.8.2, 6.8.3 | Introduction of new enumeration of FX Auction order book |
| 26 November 2020 | 4.02 | 3.11 | Example added to highlight the values disseminated in 'Transact Time' field post Upgrade 2020 |
| | | 4.10 | New section added to explain backward compatibility that was introduced with JSE Upgrade 2020 (i.e. Protocol Versioning) |
| 24 May 2022 | 4.03 | 6.4.1 | Updated default value for protocol versioning |
| 28 March 2024 | 4.04 | 3.2.3 | Updated the Central Order Book Cross functionality wording to include circuit breaker rule |
| | | 3.2.8, 6.5.2, 6.4.1, 6.6.2 | Introduced details on Self-Match Prevention functionality |
| | | Across sections | Removed sections on Enhanced Native Trading Gateway |
| 28 May 2024 | 4.05 | 4.10 6.4.1 | Updated the protocol version Added version 3 protocol to Login Message |

1.4 References

None

1.5 Contact Details

| | |
|---|--|
| <p>JSE Limited Trading and Market Services Division One Exchange Square Gwen Lane, Sandown South Africa Tel: +27 11 520 7000</p> <p>www.jse.co.za</p> | <p>Client Services Centre Email: CustomerSupport@jse.co.za</p> |
| <p>Disclaimer: All rights in this document vests in the JSE Limited ("JSE") and Millennium IT Software (Private) Limited ("Millennium IT"). Please note that this document contains confidential and sensitive information of the JSE and Millennium IT and as such should be treated as strictly confidential and proprietary and with the same degree of care with which you protect your own confidential information of like importance. This document must only be used by you for the purpose for which it is disclosed. Neither this document nor its contents may be disclosed to a third party, nor may it be copied, without the JSE's prior written consent. The JSE endeavours to ensure that the information in this document is correct and complete but do not, whether expressly, tacitly or implicitly, represent, warrant or in any way guarantee the accuracy or completeness of the information. The JSE, its officers and/or employees accept no liability for (or in respect of) any direct, indirect, incidental or consequential loss or damage of any kind or nature, howsoever arising, from the use of, or reliance on, this information.</p> | |

1.6 Definitions, Acronyms and Abbreviations

| | |
|--------------------------------------|---|
| Client | A member firm connected to the trading gateway. |
| Client Account | This is the Client Account as reflected in the JSE back-office system for whom the order is submitted by the Firm. |
| CAN | JSE Client Access Network |
| 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 and published during the Closing Price Publication session. CPX TIF Orders submitted before the CPX session are parked until the start of the CPX Session |
| FX | Foreign Currency Market |
| JSE | Johannesburg Stock Exchange. |
| MES | Minimum Execution Size is the minimum volume of the Pegged Hidden or Pegged Hidden Limit order which is permitted to execute. |
| MRS | Minimum Reserve Size is the minimum order volume needed for orders to qualify as Pegged Hidden and Pegged Hidden Limit orders. |
| Naked Option | An options instrument created on top of a future or a spot instrument |
| NSX | Namibian Stock Exchange. |
| 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. |
| Passive Order | An order residing in the order book. |
| Server | The native trading gateway at the JSE for the JSE and NSX markets. |
| Trader Mnemonic | Each order must be submitted under a particular trading mnemonic. On the Native Trading Gateway this will be the concatenated identifier of the JSE Trader Group and the JSE Trader ID e.g. GR1_001215. |
| Visible Order | Identifies an order that is visible to the market. The order has a Disclosed Quantity that is equal to Order Quantity. |
| 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. |
| EOD Volume Uncrossing Auction | A dark auction call which is triggered at end of the day after the CPX session. The uncrossing will happen at the closing price. |
| 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. |

Partition ID Identity of the matching partition. Partition 1 has been allocated for Equity instruments while Partition 2 has been allocated for Derivative instruments.

2 Overview

The System offers multiple low latency native trading gateways that allow member firms to submit and manage orders and quotes.

The interface is a point-to-point service based on the TCP/IP standard. It uses a series of fixed-length binary messages. The encryption of messages between the client and server is not supported.

2.1 Types of Native Gateways

The System offers a Basic and an Enhanced Native Trading Gateway.

[a\) Basic Native Trading Gateway – supports the trading of equity instruments of the JSE and NSX markets.](#)

[a\)b\) Enhanced Native Trading Gateway – supports the trading of derivatives and bond instruments of the Derivative and Bond Markets, and intra-day instrument creations.](#)

[This document will only focus on the Basic Native Trading Gateway.](#)

2.2 Order management, Quote Management and other services

Two Native Trading Gateways will support trading of equity instruments for the JSE and NSX Equity Markets.

The Basic Native Trading Gateway interface enables clients to perform the activities outlined below.

Order Management for Equities Markets

- (i) Submit an order
- (ii) Cancel an order
- (iii) Mass cancel orders
- (iv) Amend an order

Market announcements for Equities instruments

- (i) News

2.3 Colocation and Client Access Network (CAN) services

[The JSE offers a Basic Native Trading Gateway service to members of the equities markets from colocation and CAN. The availability of such services are as follows:](#)

| <u>Market</u> | <u>Colocation Gateway Instances</u> | <u>CAN Gateway Instances</u> |
|------------------------------|-------------------------------------|------------------------------|
| Equity – JSE | 2 | 2 |
| Equity – NSX | | |

3 Service Description

3.1 System Architecture

Two Native Trading servers will be available to Clients per Gateway instance. A colocation and an CAN Gateway instance. Each server consists of two channels: a Real-Time channel and a Recovery channel.

Clients will use the Real-Time channel to submit orders, cancellation requests and cancel/replace requests. Real-time updates to orders (e.g. acknowledgement, reject, fill, etc.) will be transmitted on this channel.

Clients will use the Recovery channel to request for messages missed during periods where there was a disconnect from the Real-Time channel.

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. [The JSE Equity Market utilises partition 1.](#)

[This market will also make use of the **Basic Native Trading Gateway**, which is identical in all aspects to the Enhanced Native Trading Gateway, with the exception of certain message structures, to enable market-specific business requirements to have minimal impacts on each other.](#)

3.2 Order Handling

[The Basic Native Trading Gateways will facilitate the clients to manage orders of equity instruments. An order submitted for a derivative instrument will be rejected if the request is sent to the Basic Native Trading Gateway.](#)

[The Clients will use the Basic Native Trading Gateway New Order Message when trading equities instruments. Similarly, the system will use a Basic Execution Report message when acknowledging or rejecting the new order requests of the Clients.](#)

[Please refer to Section 3.5 for instances where a Basic New Order message will be rejected via another message.](#)

3.2.1 Order Types

[Given below is a list of all the order types supported by the system for the Equity market.](#)

| Order Type | Description | Relevant Fields |
|--------------------------------------|---|--|
| Market | <p>An order which will be executed at the best possible prices on the contra side.</p> <p>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.</p> <p>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.</p> | Order Type = 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). | Order Type = 2 Limit Price |
| Stop | A Stop Order is a Market Order that will remain unelected (without entering the order book) until the stop price is reached. This is used to exit from a loss-making position. Once elected, it will be treated like a regular new Market Order. | Order Type = 3 Stop Price |
| 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 like a regular new Limit Order. | Order Type = 4 Stop Price Limit Price |
| 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. | Cross Type = 5 Order Type = 2 Time In Force = 0 |
| Cross Orders (With Adjustable Price) | This order type is used to ensure a cross order can execute successfully, even if not submitted with a price that falls within the BBO. Should the price of the Cross Order not fall within the BBO, execution will occur at the prevailing mid-price. | Cross Type = 50 Order Type = 2 Time In Force = 0 |

| | | |
|----------------------------------|--|---|
| <p>Pegged Hidden Orders</p> | <p>Large-in-size orders can be pegged to the mid-point of the visible BBO, pegged to the best Bid (if it is a 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</p> <p>To be eligible as a hidden order the quantity of the order must be large in size and satisfy the Minimum Reserve Size (MRS) specific to each instrument.</p> | <p>Order Type = 50 Order Sub Type Display Quantity = 0 Minimum Quantity >= MRS</p> |
| <p>Pegged Limit Hidden Order</p> | <p>A pegged order with an order type “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.</p> <p>To be eligible as a hidden order the quantity of the order must be large in size and satisfy the Minimum Reserve Size (MRS) specific to each instrument.</p> | <p>Order Type = 51 Order Sub Type Limit Price Display Quantity = 0 Minimum Quantity >= MRS</p> |

3.2.2 Order Time in Force (TIF)

| Order Type | Description | Relevant Fields |
|---------------------------------------|--|----------------------------------|
| Day (DAY) | Orders with the DAY time in force will be expired at Market End of the trading on the day they are submitted | Time In Force = 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. | Time In Force = 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. | Time In Force = 4 |
| At the Open (OPG) | An order that may only be entered and executed in the opening auction. | Time In Force = 5 |
| At the Close (ATC) | An order that may only be executed in the closing auction. | Time In Force = 10 |
| 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.). | Time In Force = 9 |
| Good For Intraday Auction (GFX) | An order that may only be executed in the Intraday auction. | Time In Force = 51 |
| Good Till Time (GTT) | An order that will expire at a specified time during the current day or at Market End. | Time In Force = 8 Expire Time |
| 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 calendar days. | Time In Force = 6 Expire Time |
| Good Till Cancelled (GTC) | An order that will expire at Market Start on the next trading day after the maximum order duration. Maximum order duration is applicable for GTC orders and will be set at 90 days. | Time In Force = 1 |
| Closing Price Cross (CPX) | Orders with this time qualifier can only participate in the Closing Price Cross session. They may be entered during other sessions that accept orders but they stay parked in the parked queue until the Closing Price Cross session starts. At the start of the Closing Price Cross session, parked CPX orders are injected into the order book. CPX orders expire at the end of the CPX session. | Time In Force = 12 |
| Good for Volume Auction Uncross (GDX) | An order that will only participate in the End of Day 'Volume Auction Uncross' | Time In Force = 50 |

3.2.3 Cross Orders

Cross Orders will only be enabled for Equity Market instruments.

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 Native order management 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.

Clients are able to flag whether a Cross Order being submitted is fixed price or adjustable price.

Cross Orders with Adjustable Price:

~~If the Cross Order is identified as a 'fixed price' Cross Order, and the submitted price does not conform to the applicable price range, the Cross Order will be rejected. However, if the Cross Order is identified as an 'adjustable price' Cross Order, the following will apply:~~
If the Cross Order is identified as an 'adjustable price' Cross Order, the following will apply:

- a) When the BBO is available, the system will validate the price against the spread within the Best Bid and Best Offer price of the instrument. If the price is out of the BBO spread, then the system will update the price of the Cross Order to be the mid-point of Best Bid and Best Offer (The system will adjust the Cross Order price to half a tick if the BBO spread is only 1 tick).
- b) If only the Best Bid (or Best Offer) is available, the Cross Order should be better than the available Best Bid (best offer). If Cross Order price is worse than best bid (or best offer), system will adjust the price of the Cross Order to be half a tick (value defined in the 'PriceImproveTks' parameter for the instrument) better than the Best Bid (or Best Offer).
- c) If both the Best Bid and Best Offer are not available, the Cross Orders will be checked against a range defined around the Dynamic Reference Price (value defined in the Ref Price Allowance (%) parameter of the instrument). The system will accept Cross Orders with a price within but not including the boundary values.

~~NOTE: Clients will be able to determine what pricing method should be applicable to the Cross Order. If traders do not want the Cross Order price to be modified they can select the 'Fixed Price' option when submitting the Cross Order. Should the submitted price not conform to the applicable price range, the Cross Order will be rejected. This will reject the Cross Order.~~

If the circuit breaker limits are breached while submitting cross orders in any of the events as above mentioned, the circuit breakers will be triggered, and the orders will be accepted or rejected based on the Circuit Breaker Cross Policy specified.

3.2.4 Pegged Orders (Hidden)

Pegged orders will only be enabled for Equity Market instruments.

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.

3.2.5 Order Capacity

The server recognizes two order capacities; agency and principal. Clients are responsible for indicating the capacity an order is submitted under and is mandatory.

3.2.6 Order Management

a) Order Ownership

The Server will associate the JSE Trader Group and JSE Trader ID combination as the owner of the order. No capacity will be provided to enter orders on behalf of another trader via Basic Native Trading Gateway.

b) Cancellation

The remainder of an Open or Parked order may be cancelled via the [Order Cancel Request](#) message of the Basic Native Trading Gateway for equity instruments.

The server will generally respond with an Execution Report or Order Cancel Reject to confirm or reject the cancellation request respectively via the Basic Native Trading Gateway. The Client Order ID specified in the message will be that of the Order Cancel Request.

The client should identify the order being cancelled by either the Original Client Order ID or Order ID. If an Order Cancel Request contains values for both Original Client OrderID and Order ID, the server will only process the Order ID.

c) Mass Cancellation

A client may mass cancel Open or Parked orders via the Order Mass Cancel Request message with an Order Sub Type of Order (0).

Clients may use the [Order Mass Cancel Request](#) message of the Basic Native Trading Gateway to mass cancel:

- a) All orders for a Firm
- b) All Firm orders for a particular instrument
- c) All Firm orders for a particular segment
- d) All orders for a given Comp ID (Interface User ID)
- e) All orders for a given Comp ID (Interface User ID) for a given instrument
- f) All orders for a given Comp ID (Interface User ID) for a given segment

The server will generally respond with an Order Mass Cancel Request to indicate, via the Status field, whether the request is successful or not. If the mass cancel is processed by multiple partitions, an Order Mass Cancel Request will be transmitted for each partition. Refer section **Error! Reference source not found.17** for any associated limitations.

The server will then transmit Execution Reports for each order that is cancelled and Order Cancel Rejects for each order that could not be cancelled. The Client Order ID of all such messages will be that of the Order Mass Cancel Request.

If the mass cancel request is rejected by a partition, the reason will be specified in the Reject Code field of the Order Mass Cancel Report.

A mass cancel request sent in by the Native Trading Gateway or the FIX Gateway, may cancel orders submitted through both gateways. In such a case, the execution reports for the order cancellation will be sent to the gateway through which, each order was submitted.

The successful Submission of an Order Mass Cancel Request to cancel all orders for a firm for a given underlying will result in two Order Mass Cancel Reports. One Rejection and One accept. The Order Mass Cancel Request will be sent out to all matching partitions in the system. Since derivatives instruments do not exist in Partition 1 this will result in an Order Mass Cancel Report with status rejected from partition 1. If request is successful for partition 2, the Comp ID that submitted the Order Mass Cancel Request will receive an Order Mass Cancel Report with status accepted for Partition 2.

d) **Amending an Order**

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

- (i) Order quantity
- (ii) Minimum Execution Size
- (iii) Limit 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 generally respond with an Execution Report or an Order Cancel Reject message to confirm or reject the cancel/replace request respectively. The Client Order ID specified in the message will be that of the Order Cancel/Replace Request.

The client should identify the order being amended by either the Original Client Order ID or Order ID. If an Order Cancel/Replace Request contains values for both Original Client OrderID and Order ID, the server will only process the Order ID.

When amending the order submitted by one comp ID of a firm, via a different comp ID of the same firm, since Client Order IDs are maintained in the system per comp ID, it is not possible to submit an amendment request by referring to the original client order ID, trader will have to refer to the order's Order ID for a successful amendment request

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

Clients may not modify orders that are fully filled.

When the quantity of a partially filled fully visible order is amended such that the remainder is zero, such requests will be rejected.

When the quantity of a partially filled hidden order is amended such that the remainder of the order is zero, the following values will be stamped in the execution reports,

- Exec Type = "Trade"
- Order Status = "Filled"

A client can amend the additional instruction submitted for a derivative instrument via the Secondary Trade Report ID field. If the amendment is accepted by the system, the order will be associated with the new Secondary Trade Report ID. This new value will be disseminated when the order is executed.

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.

3.2.8 Self-Match Prevention

Self-Match Prevention (SMP) is a mechanism that allows market participants to prevent new orders from matching with their own previously submitted orders. SMP is applicable in the order book, during continuous trading sessions (i.e., continuous trading, closing price cross sessions). A trader that requires SMP should specify a Self Trade Prevention Key in the New Order – Protocol Version 3 message with a unique identifier.

Upon a potential self-match, either the aggressing order or passive order will be expired as per the criteria that is set for the CompID. Currently the JSE setup is for the resting (passive) order to be expired before potential matching.

3.2.8.3.2.9 Trade Cancellations

The JSE may cancel trades. The server will transmit [Execution Reports](#) to the relevant clients to notify them of a trade cancellation.

If an execution received by an order is cancelled, the executed quantity will be cancelled. When the quantity is cancelled, the order will be cancelled 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 order cancellation.

3.3 Gateway Validations

This section discusses the standard gateway level validations that are present in the Basic Native Trading Gateway.

The server will generally reject a business message via an Execution Report, Cancel Reject or Mass Cancel Report message.

All client-initiated actions are subjected to two levels of gateway validations before the server receives the message.

Level one pertains to validations on the message header, data type and range defined for each field (valid values for a given field). A Reject message will be used to reject a malformed message (e.g. invalid data type, invalid value, required field missing, etc.).

If the message successfully passes the first layer of validations, the system generates an internal message to check for conditional requirements of each field and any message specific validations. This forms the second layer of the gateway validation process.

If a message fails to comply with any of the gateway level validations, a Reject message would be generated which contains a reject code, along with the reason specified. (Please refer to [Section 7.2](#) for a list of reject codes)

The Business Reject message will be used to reject a New Order message for an unknown instrument.

The Business Reject message will be used to reject messages if a partition or the entire system is suspended in the unlikely event of a process outage. The Partition ID of a Business Reject will be zero (0) if the system is suspended.

A Cancel Reject will be used to reject a Cancel Request or Order Cancel/Replace Request for an unknown instrument. The Partition ID of the Cancel Reject message will be zero (0) in these instances.

Any client-initiated message after passing gateway level validations will be subjected to internal process validations upon reaching the server. Failure to pass server level validations will be notified to clients through an Execution Report, which will indicate a reject code, to which the reason is specified in the reject code specification.

The system will validate the requests coming to the Basic Native Gateway instance of the, to ensure that they are only for the Equity instruments. If requests are for non-Equity instruments (i.e. derivatives and bonds instruments), the Basic Native Gateway will reject such requests with a 'Reject' message.

When an Order Mass Cancel Request message is sent with the Mass Cancel Request Type set to '14' or '22' to the Basic Native Gateway, such requests will be rejected by the gateway with a Reject message.

An exception to the server level rejection process is when the Instrument or Order book specified could not be found, in which case a Business Reject is generated by the system. (Please refer to [section 7.3](#) for a list of Business Reject codes)

3.4 Quotation Conventions

The limit price, stop price, bid price and offer price specified with an order or quote will be interpreted by the server in terms of the applicable quotation convention for the instrument.

In the case of equity instruments the values specified in these fields will be interpreted as the price per share.

3.5 Market Operations

3.5.1 Order Management

JSE Market Operations has the ability to submit an order, cancel request or order cancel/replace request on behalf of a client. However, only order cancellation request will be accepted as a last resort.

The client will be notified, via an Execution Report, of the order, order cancel request or order cancel/replace request submitted on its behalf if and when it is accepted. 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.

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

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 Execution ID field.

If an execution received by an order is cancelled or corrected to reduce the executed quantity, the cancelled/reduced quantity will either be cancelled or reinstated in the order book. 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 order restatement.

Market Operations also has the ability to correct the price of an execution. A trade will not be corrected to increase the executed quantity. This functionality will not be applicable to JSE processes.

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

3.6 Conditionally required fields

All fields that are conditionally required will be ignored by the server. (E.g. Stop Price field, MES field will be ignored for Limit and Market orders).

3.7 Time Stamps and Dates

Expire Time will be in YYYYMMDD-HH:MM:SS format and specified in UTC.

TransactTime is the time the Execution Report is generated by the system, this will be in Unix (Posix) time which will be the number of seconds elapsed since [midnight proleptic Coordinated Universal Time](#) (UTC) of January 1, 1970, not counting [leap seconds](#).

The first 4 bytes of the TransactTime timestamp will represent the Unix (Posix) time while the next 4 bytes will specify the micro seconds. The TransactTime will be in UTC.

NOTE: The last 4 bytes (microsecond portion) of the TransactTime will now contain a degree of accuracy up to nanoseconds. However, the last three zeroes ('000') can be ignored.

E.g. Value in Transact Time field = 2020-10-28 07:16:47.622747000 can be interpreted as

Date: 2020-10-28

Time (hours and minutes portion): 07:16

Time (seconds portion): 47

Time (microsecond portion): 622747

(Last 3 0's are ignored)

3.8 Market Operations Announcements

Clients will receive market operations announcements via the News message on the Basic 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 market operations 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, which will be disseminated to all Interface Users (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 Missed Message 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.

3.9 Mapping Native Order ID to MITCH Order ID

To convert Native Order ID to MITCH Order ID:

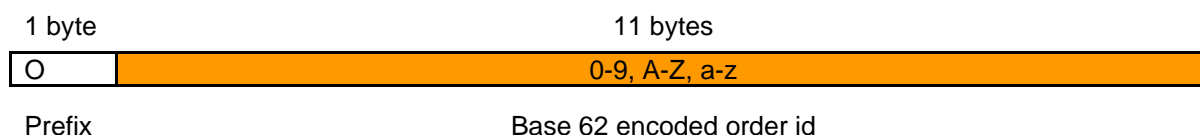
Step 1 – Remove leading O (prefix)

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

Step 3 – Convert to binary

E.g.

Order ID format (ASCII)



E.g.

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

Steps to follow:

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

- Please use 64 bit integer data types for the calculation else integers will overflow
- 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 | | |

3.10 Execution Report

The Execution Report message is used to communicate many different events to clients. The events are differentiated by the value in the Exec Type field as outlined below.

| Exec Type | Usage | Order Status | Working Indicator |
|-----------|--|--------------|---|
| 0 | <p>New Indicates that a new order side has been accepted. This message will also be sent unsolicited if an order was submitted by JSE Market Operations/Surveillance on behalf of the client.</p> <p>This message will be sent when a stop or stop limit order enters and remains unelected or a GFA/GFX/ATC/CPX order is entered and is parked.</p> <p>This message will also be sent when an unelected stop or stop limit order is elected and added to the order book without receiving an execution. This message will also be sent when an unfilled parked order with a time in force of GFX/GFA/ATC is injected and added to the order book without receiving an execution.</p> | 0 | 0 – Unset 1 - Orders elected on entry 2 – Unelected orders (Stop, Stop Limit) |
| 8 | <p>Rejected Indicates that an order has been rejected. The reason for the rejection is specified in the field Reject Code.</p> | 8 | 0 |
| F | <p>Trade Indicates that an order has been partially or fully filled. The execution details (e.g. side, price and quantity) are specified.</p> <p>This message will also be sent when an unelected order is elected and receives executions on aggression.</p> <p>This message will also be sent when a parked order with time in force of GFX/GFA/ATC/CPX is injected and receives executions on aggression.</p> | 1, 2 | 0 |
| C | <p>Expired May indicates one of the following:</p> <ul style="list-style-type: none"> a) An order has expired in terms of its time qualifier b) An order has expired due to an execution limit. c) If any remaining orders (except GTC and GTD) are expired at market close. d) When orders are expired based on the cancel on disconnect/log out feature. e) Orders expired due to triggering circuit breakers | 6 | 0 |
| 4 | <p>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/Surveillance.</p> | 4 | 0 |
| 5 | <p>Amended/Modified Indicates that an order cancel/replace request has been accepted and successfully processed.</p> | 0, 1 | 0 |
| L | <p>Triggered Indicates that a parked ATC, GFX, GFA, CPX or contingent order (i.e. stop, stop limit, trailing stop or MIT) has been activated and moved to the main container. The order is available for execution.</p> | 0, 1 | 1 |
| 9 | <p>Suspended Indicates that a GFA, GFX order that was in the main container has been parked and is no longer available for execution.</p> | 9 | 0 |

| | | | |
|---|--|------------|---|
| D | Restated (Order Cancel/Replace by Market Operations) Indicates that an order has been amended by JSE Market Operations/Surveillance or due to TIF CPX orders being re-priced during the CPX session. If an order is amended via Service Desk, it will not be assigned a new Order ID or Client Order ID. | 0, 1 | 0 |
| H | Trade Cancel Indicates that an execution has been cancelled by JSE Market Operations/Surveillance. The Execution ID of the execution being cancelled will be included. | 0, 1, 4, 6 | 0 |
| G | Trade Correct Indicates that an execution has been corrected. The Execution ID of the execution being corrected will be included along with the updated execution details (e.g. price and quantity). | 1, 2, 4, 6 | 0 |

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

3.11 Order Status

The Order Status field of 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 Order Status. The relevant order statuses are given below from the highest to lowest precedence.

| Value | Meaning |
|-------|------------------|
| 2 | Filled |
| 9 | Suspended |
| 4 | Cancelled |
| 6 | Expired |
| 1 | Partially Filled |
| 0 | New |
| 8 | Rejected |

Please refer to [Section 8](#) for process flow diagrams on the various statuses that may apply to an order.

3.12 Orders and Execution Identifiers

3.12.1 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 does not validate each Client Order ID for uniqueness. Clients must ensure unique Client Order IDs across all application messages sent under a particular User Interface ID (CompID) (e.g. New Order, Order Cancel Request etc). Given that the server supports GTD and GTC orders, clients must also ensure that Client Order IDs are unique across trading days (e.g. embed the date within the Client Order ID).

3.12.2 Execution IDs

The server will use the Execution ID field to affix a unique identifier for each Execution Report. Execution IDs will be unique across trading days in perpetuity, irrespective of the market in which the execution occurs. If an Execution Report message is used to notify a client of a trade cancellation or correction, the Execution ID will refer to the Execution ID generated for the original execution.

3.12.3 Order IDs

The server will use the Order ID field of the Execution Report to affix the order identification numbers of the trading engine. Order IDs will be unique across trading days in perpetuity, irrespective of the market in which the execution occurs.

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. As the Order ID does not change, all execution reports can be linked through the Order ID.

3.12.4 Public Order IDs

The server uses Public Order ID field of the Execution Report to affix the Public Order ID of an order which is an order identification number that will be stamped for every order that has an Order ID. Public Order ID will be the same as the Order ID for all orders that are not iceberg orders¹. For iceberg orders, the Public Order ID will renew with each replenishment. Participants should identify their orders on the market data feeds using Public Order ID as that is the identification number that will be disseminated for order book updates on market data.

3.12.5 Instrument Identification

All instruments – equity, bonds and derivatives – may be identified by the Instrument ID assigned by the Exchange to each security. The application messages transmitted by the server will always contain the Instrument ID.

3.13 Party Identification

The same party identification method is used for equity and derivative/bonds traders.

¹ [There is an exception for partially filled iceberg orders that are amended to fully visible orders. In this case, the visible order will have a public order ID different from the order ID \(Public order ID change due to earlier Pfill and replenishment while it was an iceberg order\)](#)

| ID | Description | Relevant Fields |
|------------------------------|---|-----------------|
| Trader Group + JSE Trader ID | Concatenated identifier of the Trader Group and JSE Trader ID separated by an underscore. | Trader Mnemonic |

3.14 Field Value Validations

The following field value validations will be done. If a message is rejected, it will be rejected with the *Reject* message, *Cancel Reject* message or *Execution Report*.

- The problematic field name will be specified in the *Reject Reason* field in the *Reject* message.
- The problematic message type will be specified in the Rejected Message Type field in the *Reject* message.
- The Reject code will be specified in the field Reject Code in the *Reject* message, *Cancel Reject* message or *Execution Report*.
- The Reject codes for the relevant fields will be as follows** (Please refer to section 8 for the Reject Codes).

| Message | Field | Validation | Reject Code |
|------------------------|--------------|---|--------------|
| Message Header | Length | The value has to be the actual length of the message. Otherwise reject the message. | 9901 |
| | Message Type | If the value is out of range from the defined set of values, reject the message. | 9901 |
| Logon | User Name | If a value is not specified, reject the message. | 9900 |
| | | If the value contains invalid ASCII characters (please refer section 3.10.2), reject the message. | 9901 |
| | Password | If a value is not specified, reject the message. If the value contains invalid ASCII characters, reject the message. | 9900 9901 |
| | New Password | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| Logout | Reason | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| Missed Message Request | AppID | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |

| | | | |
|-----------|------------------|---|--------------|
| | LastMsgSeqNum | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |
| New Order | Client Order ID | If a value is not specified, reject the message. If the value contains invalid ASCII characters, reject the message. | 9900 9901 |
| | Trader Mnemonic | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| | Account | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| | SecurityID | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |
| | Minimum Quantity | Minimum Execution Size that needs to be specified for a Pegged or Pegged Limit Order which must be greater than or equal to Minimum Reserve Size. This is applicable only for the Basic Native Gateway. | 134010 |
| | Order Type | If the value is out of range from the defined set of values, reject the message. | 9901 |
| | TIF | If the value is out of range from the defined set of values, reject the message. | 9901 |
| | Expire Time | The value has to be in the format of YYYYMMDD-HH:MM:SS or YYYYMMDD. Otherwise reject the message. | 9901 |
| | Side | If the value is out of range from the defined set of values, reject the message. | 9901 |
| | Order Qty | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |
| | Display Qty | The value has to be 0 or equal to the Order Quantity for Equity instruments. Otherwise reject the message via the Basic Native Gateway. | 1105 |
| | Limit Price | The value has to be greater than 0 (>0) if Order Type is Limit or Stop Limit for Equity instruments. Otherwise reject the message via the Basic Native Gateway. | 1204 |
| | Capacity | If the value is out of range from the defined set of values, reject the message. | 9901 |
| | Auto Cancel | If the value is out of range from the defined set of values, reject the message. | 9901 |

| | | | |
|-------------------------------|--------------------------|--|--------------|
| | Stop Price | The value has to be greater than 0 (>0) if Order Type is Stop or Stop Limit for Equity instruments. Otherwise reject the message via the Basic Native Gateway. | 1301 |
| | Execution Instruction | If the value is not 0, 1 or 2 on a new order message, reject the message. | 9901 |
| | Anonymity | If the value is out of range from the defined set of values, reject the message. Only applicable for the Enhanced Gateway. | 9901 |
| Order Cancel Request | Client Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| | Original Client Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| | Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| | Security ID | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |
| | Trader Mnemonic | If a value is specified and it contains invalid ASCII characters, reject the message. | 9100 |
| | Side | If the value is out of range from the defined set of values, reject the message. | 9901 |
| Order Mass Cancel Request | Client Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| | MassCancelRequestType | If the value is out of range from the defined set of values, reject the message. | 9901 |
| | Security ID | The value has to be greater than 0 (>0) if the Mass Cancel Request Type is 3 or 9 when submitted via the Basic Native Gateway. Otherwise reject the message. | 9900 |
| | Segment | If the value is not specified for Mass Cancel Request Types 4 and 15, reject the message. If the value contains invalid ASCII characters, reject the message. | 2600 9901 |
| | Order Sub Type | If the value is out of range from the defined set of values, reject the message. | 9901 |
| Order Cancel/ Replace Request | Client Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |

| | | |
|--------------------------|---|--------|
| Original Client Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| Order ID | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| Security ID | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |
| Trader Mnemonic | If a value is specified and it contains invalid ASCII characters, reject the message. | 9100 |
| Expire Time | The value has to be in the format of YYYYMMDD-HH:MM:SS or YYYYMMDD. Otherwise reject the message. | 9901 |
| Order Quantity | The value has to be greater than 0 (>0). Otherwise reject the message. | 9901 |
| Display Quantity | The value has to be 0 or equal to the Order Quantity for Equity instruments. Otherwise reject the message via the Basic Native Gateway. | 1105 |
| Minimum Quantity | Minimum Execution Size that needs to be specified for a Pegged or Pegged Limit Order which must be greater than or equal to Minimum Reserve Size for Equity instruments. Otherwise the message is rejected by the Basic Native Gateway. This is applicable only for the Basic Native Gateway. | 134010 |
| Account | If a value is specified and it contains invalid ASCII characters, reject the message. | 9901 |
| Order Type | If the value is out of range from the defined set of values, reject the message. | 9901 |
| TIF | If the value is out of range from the defined set of values, reject the message. | 9901 |
| Side | If the value is out of range from the defined set of values, reject the message. | 9901 |
| Limit Price | The value has to be greater than 0 (>0) if Order Type is Limit or Stop Limit for Equity instruments. Otherwise reject the message via the Basic Native Gateway. | 1204 |
| Stop Price | The value has to be greater than 0 (>0) if Order Type is Stop or Stop Limit for Equity instruments. Otherwise reject the message via the Basic Native Gateway. | 1301 |

3.15 Validation of ASCII characters

The values which correspond to Decimal 32 to 126 will be accepted. Any other ASCII character will be rejected.

The above valid ASCII character range is used for appropriate validations in the [section 3.14](#).

3.16 Rejection Logic

All client-initiated messages are subjected to two levels of gateway validations before the server receives the message.

Level one pertains to validations on the message header, data type and range defined for each field (valid values for a given field).

If the message successfully passes the first level of gateway validations, the system generates an internal message to check for conditional requirements of each field and any message specific validations. This forms the second level of gateway validations.

If a message fails to comply with any of gateway level validations, a Reject message will be generated which contains a reject code, along with the reason specified. The only exception to the gateway level rejection logic is when the server is unavailable in the unlikely event of an outage; a Business Reject message is generated instead of a Reject in this scenario.

Any client-initiated message after passing gateway level validations will be subjected to internal validations upon reaching the server. Failure to pass server level validations will be notified to clients via an Execution Report with a reject code and the reason which is specified in the Reject Code Specification.

An exception to the server level rejection logic is when the instrument or the order book could not be found, in which case a Business Reject is generated by the server.

3.17 Functional and Implementation Limitations

- (b) 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 AppID field in the Order Mass Cancel Report. Due to this behaviour, when a Mass Cancel Request is submitted via the Enhanced Native Trading Gateway with a request type as '14' or '22', with a underlying specified as a 'Future' instrument, the system will send a Mass Cancel Request reject message from Partition 1, as the Futures instrument is not available in that partition.
- (c) The gateway will not validate the Order ID against the type of the instrument, when an Order Cancel Request message is sent to the Basic Native Gateway without the Security ID.
- (d) When specifying the expiry time for a GTT order, a date component will also be specified along with the expiry time. The server takes the date component into consideration when validating the expiry time. I.e. if a GTT order is sent with an already elapsed expiry time but with a future date in the date component, the order will be accepted and will be expired at Market End of the current trading day. I.e. the order is treated as a DAY order.
- (e) An order mass cancel request should not be sent during Start of Trading session. If a request is sent, it will be rejected. But thereafter in a subsequent session the client will

not be able to mass cancel the same orders again. But the client can individually cancel orders.

- (f) If an order cancel/replace request is of a cancel/replace nature (a limit price change or a stop price change), the system removes the order from the relevant container (e.g. order book etc-) (Cancel) and then applies the change (replace). Hence at the time of generating the Execution Report to confirm the amendment, there is no container for the order. Due to this, the "Working Indicator" field of the Execution Report will be set to 'Unset (0)'.
- (g) If an order is successfully amended as above **Error! Reference source not found.** and an execution is resulted during the aggression, there will be no container for the order at that time. However, the system will retain the previous value that was attached to the order and stamp the same in the "Working Indicator" field of the Execution Report which is generated to communicate the execution. Once the order is added to the relevant container, the appropriate value will be tagged for the "Working Indicator" field in the Execution Reports which are generated for subsequent executions.
- (h) The server does not validate each Client Order ID for uniqueness (as mentioned in section 3.10). If a client mistakenly submitted more than one order with the same Client Order ID (within a trading day or over a couple of days if GTD/GTC is used), they will only be able to cancel/amend the most recent order (using the Client Order ID) but not the previous entries as the system maintains only one order for a Client Order ID in a map and updates/removes it once a cancel or amend is received.
- (i) It is not possible to populate the Client Order ID in the Reject message in the below scenarios:
- If the Client Order ID itself is invalid.
 - If the Client Order ID is not the first field of the message and if any field above the Client Order ID is invalid.
 - If the message header is incorrect (e.g. message type, message length).
- (j) If the original TIF was 1, 3, 4, 5, 6, 8, 9, 10 or 12 and if an Order Modification Request was sent with the TIF specified as '0' (DAY), then the amend request is accepted and not rejected; the TIF amendment will be ignored in this scenario and in the Execution Report to acknowledge the amend request, the original TIF of the order will be stamped. However if the original TIF was GTD/GTT, the Order Modification Request will be rejected unless the Order Modification Request contain the original TIF (i.e. GTD/GTT). the system cannot differentiate whether a TIF was specified in the amend request or not (as DAY is represented by 0 and when a TIF is not specified, it will also come as 0). Hence it will stamp the original TIF of the order to the amend request. Hence if a GTT order is amended to have TIF DAY, system still consider the TIF to be GTT and to have a valid expiry time; if an expiry time is not specified or an invalid expiry time is specified, the amend request will be rejected.
- (k) Cancel on disconnect is applicable only if the user is disconnected before the end of day (EOD). At EOD, Native users who are still logged in will be logged off by the system. Hence Cancel on disconnect is not applicable at EOD.. A disconnect delay parameter (set in milliseconds) is available to specify the delay applicable to a disconnect/logout before expiring the open orders/quotes. A [New Order](#) must still be submitted with the Cancel On Disconnect field set to 'Yes' for this functionality to take effect. Should the Interface User (Comp ID) be set with Mass Cancel on Logout/Disconnect and a submitted message has the Cancel On Disconnect field set to 'No', the order will not be removed from the order book on a disconnect or logout. This parameter can be set to cancel all open orders, no open orders or orders excluding ones with TIF of GTD/GTC.
- (l) It's not possible to perform negative number validations for fields with Unsigned Integer data types. For instance "Order Quantity" is of UInt64 data type. Since this field is unsigned, it is not possible to represent a negative number as "Order Quantity". This scenario could be described as below.
Field A has data type UInt8 and Field B has data type Int8

Client submits -100 on Fields A and B

Representation of -100 on Field B (Int8):

| | | | | | | | |
|----------|----|----|----|----|----|----|----|
| | | | | | | | |
| Sign Bit | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| - | 64 | 32 | 0 | 0 | 4 | 0 | 0 |

In signed integer types to represent negative values the sign bit or the leftmost bit is used. When it's set to 1 the value is negative, otherwise positive.

Representation of -100 on field A (UInt8)

In unsigned integers the sign bit is not considered as a negative value validation bit and is used as a digit to represent the number as other 7 bits do.

| | | | | | | | |
|-----|----|----|----|----|----|----|----|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 128 | 64 | 32 | 0 | 0 | 4 | 0 | 0 |

Even though we expected to represent -100, the final outcome is number 228. If we used UInt64 to represent -100, the actual value will be 9.2233720368548E+18 which most probably would fire validations such as Maximum Size.

For instance, an order submitted with -100 as the Order Quantity, would get rejected with OrderRejectCode=1003 (Invalid order size (> maximum size)) if it breaches the maximum size validation.

4 Connectivity

4.1 Interface User ID (CompIDs)

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

4.2 Passwords

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 when connecting to the Real-Time channel. 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 Reject Code of the Logon Response will indicate why the password 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 the JSE to have its password reset. The Password Expiry field of the [Login Response](#) message will indicate the number of days after which the current password will expire.

NOTE: In the event that an Interface User ID (CompID) submits a [Logon](#) message with a new password that does not comply with the password policy setup by the JSE, the system will still enable the client to connect (provided that the username and current password are submitted correctly). The [Login Response message disseminated in this instance will contain the Reject Code of '3' \(New session password does not comply with policy\)](#)

4.3 Production IP Addresses and Ports

The IP address of each client must be registered with the JSE. The IP addresses and ports of the production servers will be detailed in the Client Enablement Information Form (CEIF) after completion of an order form by the client.

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. Clients who subscribe to the colocation service will be assigned the above mentioned IP addresses and ports for the colocation gateway, whereas Clients who subscribe to the throttled Gateway will be assigned to the IP addresses and ports for the throttled gateway.

4.4 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 ports if so requested by the JSE. Please refer to the separate Client Failure and Recovery Document.

4.5 Establishing a Connection

Each client will use the assigned IP address and port to establish a TCP/IP session with the server. The client will then initiate a session at the start of each trading day by sending the [Logon](#) message. If the client does not transmit a [Logon](#) message within `SESSION_HB_INTERVAL * 3<15>` seconds of establishing the TCP/IP connection, the server will break the TCP/IP connection with the client.

The client will identify itself using the Interface User ID (CompID) 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 Response](#) message with the Reject Code "0" for Successful. If the client's [Logon](#) message included a New Password, the acceptance of the login indicates that the new password has been accepted.

The client must wait for the server's [Logon Response](#) before sending additional messages. Messages received from the client before the acceptance of the login request, are rejected via the [Reject](#) message.

If a logon attempt fails because of an invalid CompID, invalid password, invalid gateway, invalid IP or CompID is not granted with required login privileges, the server will break the TCP/IP connection with the client without sending a [Logon Response](#).

If a logon attempt fails because of an expired password, a locked/suspended CompID or if logins are not currently permitted, the server will send a [Logon Response](#) message, which will include the appropriate Reject Code, and then break the TCP/IP connection with the client.

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 `MAX_CLIENT_BUFFER_LIMIT<500>`.

4.5.1 Maintaining a Connection

4.5.2 Application Sequence Numbers

The trading system consists 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 and the partition's message sequence number in the fields Partition ID and Sequence Number respectively. As the partitions operate in parallel, a sequence number is only unique per Partition ID. The sequence number of each partition is initialized to "0" at the start of each trading day.

A client will only receive a subset of the messages generated by each partition. While the sequence number of each message will be higher than that of the last message from the partition, it will not be sequential. Therefore, a client should not connect to the recovery channel and request for missed messages if the difference in Sequence No between two consecutive messages is more than one. Recovery should be requested only upon a reconnection after a session disconnection.

Sequence numbers are not maintained for client-initiated messages. Uniqueness of Client-initiated messages will be achieved through the provision of unique Client Order IDs per Interface User. It is the responsibility of the customer to ensure that a Client Order ID is unique over the life of an order.

4.5.3 Heartbeats

The client and server will use the [Heartbeat](#) message to exercise the communication line and to verify that the interfaces at each end are available. The heartbeat interval is `SESSION_HB_INTERVAL<3>` seconds.

If the server detects inactivity for a period longer than `SESSION_HB_EXPIRY_COUNT<3>` heartbeat intervals, it will break the TCP/IP connection with the client. The client is expected to employ similar logic if inactivity is detected on the part of the server.

4.6 Terminating a Connection

The client is expected to terminate each 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](#) message and will immediately break the TCP/IP connection with the user.

All open TCP/IP connections will be terminated by the server when it shuts down and a [Logout](#) will be sent. Under exceptional circumstances the server may initiate the termination of a connection during the trading day by sending the [Logout](#) message, waiting for the heartbeat interval and then breaking the TCP/IP connection with the client. In such a case, the reason for the logout will be included in the Reason field.

Either party that wishes to terminate the connection should wait for the heartbeat interval duration before breaking the TCP/IP connection, in order to ensure that the other party received 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 `MAX_CLIENT_BUFFER_LIMIT<500>`.

4.7 Message Rate Throttling

Error! Reference source not found. The JSE 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 currently defaulted to 100 messages per second per Interface User ID (CompID) and will be reviewed as and when required.

Every message that exceeds the maximum rate of an Interface User ID (CompID) will be rejected via a [Reject](#) message. The Reject Reason of such a message will be "9990".

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

4.8 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.

In addition to the above configuration, each [New Order](#) or Quote message will include an indication, via the Cancel on Disconnect field, of whether or not it will be automatically cancelled if the Interface User ID (CompID) that submitted it disconnects from the server.

If the Interface User level configuration is set to automatically cancel all orders on a disconnect, the Interface User can mark each order through its Cancel on Disconnect field; whether it should be automatically cancelled according to its preferences, should a disconnection or logout happen. For each order an Execution Report with the 'Exec Type' and 'Order Status' fields stamped with the value 'Expired' is generated, as opposed to 'Cancelled' for all 'Firm Initiated Cancellations'.

This feature does not guarantee that all outstanding marked orders will be successfully cancelled as executions that occur very near the time of disconnect may not be reported to the client.

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

4.9 Unavailability of Real-Time Channel

A [System Status](#) message with a Status of Partition Suspended (3) will be transmitted to all clients connected to the Real-Time channel in the unlikely event the real-time service for a particular partition is unavailable.

4.10 Backward Compatibility

The Millennium Exchange product supports up to 2 versions of the native protocol at a given time. This gives the flexibility for JSE to certify different customer groups to different versions of the protocol and to be able to roll-out releases to production faster. Participants can either provide the version number with the Logon message or leave the field empty. In the latter case, the system will consider the default protocol version defined in the system. If a participant submits a protocol version that is not supported by the system, the system will reject the Logon.

If a Native Gateway message, described in section 6.0 Message Formats, has 2 different versions based on the protocol, the applicable protocol version will be stated above the message (e.g. 6.6.1 Execution Report).

Currently version [42](#) is the default version with version [23](#) being the next optional version. ~~By 1 July 2022 all clients need to conform to protocol version 2 as this will then become the default protocol version number.~~

5 RECOVERY

If a client gets disconnected from the server, the recovery channel should be used to recover missed messages. The application messages (e.g. [Execution Report](#), [Cancel Reject](#), [News](#) etc.) generated during a period when a client is disconnected from the Real-Time channel will not be sent when it next reconnects.

5.1 Establishing a Connection

The client will be logged in to the Real-Time channel before it attempts to login to the Recovery channel.

Once a connection with the Real-Time channel is established, the client will use the relevant IP address and port (as outlined in Section 4.2 to establish a TCP/IP session with the Recovery channel. The client will then initiate a session with the Recovery channel by sending the [Logon](#) message. The client will identify itself using the Interface User ID (CompID) field.

The server will validate the Interface User ID (CompID), password (depending on if the password policy is enabled) and IP address (depending on if the IP address verification policy is enabled) of the client. Once the client is authenticated, the server will respond with a [Logon Response](#) message with the Reject Code Successful "0". The value, if any, in the New Password field of the [Logon](#) will be ignored.

The client must wait for the server's [Logon Response](#) before sending additional messages on the Recovery channel. Messages received from the client before the acceptance of the login request are rejected via the [Reject](#) message.

If a logon attempt fails because of an invalid CompID, the server will break the TCP/IP connection with the client without sending a [Logon Response](#).

If a logon attempt fails because of an invalid or expired password, locked CompID or if the logins are not currently permitted, the server will send a [Logon Response](#) message, which will include the appropriate Reject Code, and then break the TCP/IP connection with the client.

The Recovery channel supports a certain number of maximum concurrent logins `MAX_SESSIONS_PER_SERVICE<200>`. Once the number of logged in clients has reached this limit, the server will reject login requests from additional clients with a [Logon Response](#) and then break the TCP/IP connection. The Reject Code of such a message will be "9903".

5.2 Heartbeats

The client and server will use the [Heartbeat](#) message to exercise the communication line and to verify that the interfaces at each end are available. The heartbeat interval is `SESSION_HB_INTERVAL<5>` seconds.

If the server detects inactivity for a period longer than `SESSION_HB_EXPIRY_COUNT<5>` heartbeat intervals, it will break the TCP/IP connection with the client. The client is expected to employ similar logic if inactivity is detected on the part of the server.

5.3 Requesting Missed Messages

When a client needs to recover missed messages they must first connect to the Real Time Channel and establish a session by exchanging Logon and Logon Response messages. The client may then connect to the Recovery channel and exchange Logon and Logon Response to establish a recovery session. Any attempt to connect to the Recovery Channel without first connecting to the Real Time Channel will be rejected and the server will send a Logon Response message which will include the appropriate Reject Code. The client must ensure proper authentication (i.e. same Interface User CompID and password) when logging into both channels. Any values sent for the NewPassword field in the Logon message sent to the Recovery Channel will be ignored.

The client is expected to transmit a [Missed Message Request](#) within `SESSION_HB_INTERVAL * 3` seconds of establishing the connection.

If the client requires all messages generated by the partition for the day, it will include one (1) in the Sequence Number field. A separate [Missed Message Request](#) will be sent for each partition.

The server does not support multiple concurrent requests from the same client. [Missed Message Requests](#) submitted prior to the completion of an active request will be ignored.

When a client needs to recover missed messages he must first connect to the Real Time Channel and establish a session by exchanging [Logon](#) and [Logon Reply](#) messages. The client may then connect to the Recovery Channel and exchange [Logon](#) and [Logon Reply](#) messages to establish a recovery session. Any attempt to connect to the Recovery Channel without first connecting to the Real Time Channel shall be rejected and the server will send a [Logon Reply](#) message, which will include the appropriate Reject Code. The client must ensure proper authentication (i.e. same username and password) when logging in to both channels. Any values sent for the NewPassword field in the [Logon](#) message sent to the Recovery Channel will be ignored.

After establishing a connection with the Recovery Channel, the client may send a [Missed Message Request](#). The message should include the identifier of the partition to which the request applies along with the sequence number immediately after that of the last message received from the partition (i.e. last sequence number plus one). The user will have to send separate [Missed Message Request](#) messages to retrieve messages from each partition.

If a service interruption occurs in the Native Recovery Channel, the Native Gateway will send a System Status message to all logged in clients of that gateway's recovery channel with AppID stamped to indicate the service/partition is unavailable. When this message is received, clients can identify that the recovery service is not available for the partition indicated by AppID. They would be able to continue recovery activities on other partitions without interruptions. If the gateway was in the middle of serving a Missed Message Request, it will send a Missed Message Report message with 'Response Type' = 3 (service unavailable) to the client. If a new Missed Message Request is sent by a user, the gateway will reject the message with a 'Missed Message Request Ack' with 'ResponseType' = 3(service unavailable) to the client. Once the service is available again, the Native Gateway will send another System Status message with AppID to indicate the service availability of the partition to the clients who are still connected on to the recovery channel with 'AppStatus' = 1. When this message is received, the clients are expected to resend the request for missed messages (preferably from the point of interruption) to the gateway to resume the missed message recovery.

The message will include the identifier of the partition to which the request applies along with the sequence number immediately after that of the last message received from the partition (i.e. last sequence number plus one).

If the matching system becomes unavailable, clients will receive a Business Reject message with a value of "9998" indicating "Matching Partition Suspended." upon order entry.

5.4 Response to a Missed Message Request

The server will respond to a [Missed Message Request](#) with a [Missed Message Request Ack](#) to indicate whether the request is successful or not. A Status other than Request Accepted (0) will indicate that the request has been rejected and the reason will be specified in the field Status.

In the case of a successful recovery request, the server will retransmit the requested messages immediately after the [Missed Message Request Ack](#). Each such message will include the relevant Partition ID and Sequence Number. Once the last of these messages is sent, the server will send a [Transmission Complete](#) message. It should be noted that due to

race conditions duplicate messages may be transmitted via the recovery channel. Clients are advised to use the AppID and SeqNum to carry out duplicate discard.

A client should not send subsequent Missed Message Requests prior to receiving the Transmission Complete message, since these will be ignored by the server.

The total number of messages that will be transmitted in response to each [Missed Message Request](#) is limited to MSG_LIMIT<2000>. If the number of messages a client has missed for a particular partition exceeds this limit, the server will only send the first MSG_LIMIT<2000> messages from the AppID and Sequence No provided. The [Transmission Complete](#) message sent in such a case will include a Status of Message Limit Reached (1).

The total number of [Missed Message Requests](#) that a client may send on the Recovery channel is limited to MAX_RECOVERY_REQUEST_COUNT<1000> each day. Once this limit is reached, the server will reject any additional request via a [Missed Message Request Ack](#) with a Status of Request Limit Reached (1).

Upon receiving the [Transmission Complete message](#), the client can send a [Logout](#) message and terminate the connection or submit a new [Missed Message Request](#) for any more messages that needs to be transmitted.

5.5 Termination of the Recovery Connection

If the client does not terminate the connection within SESSION_HB_INTERVAL * 3 seconds of the transmission of the last [Transmission Complete](#) message, the server will break the TCP/IP connection with the client.

If the client has received only part of the message set that was requested, the client may send in a new Missed Message Request message for the messages that were not recovered in the first attempt. However, if such a request is not sent within SESSION_HB_INTERVAL * 3 seconds of the transmission of the last Transmission Complete message, the Server will terminate the connection. If the client is unable to send a new request within this time, the client can re-login to the Recovery Channel and send in the Missed Message Request.

If the recovery service becomes unavailable while servicing a Missed Message Request from the client, the client will be disconnected from the recovery channel. Any further Missed Message Requests sent by the client after a re-login while the recovery service is unavailable will be rejected via a Missed Message Request Ack with a Status of 3 Service Unavailable. The client can send a new Missed Message Request when the recovery service is available again to recover messages.

5.6 Unavailability of Recovery Channel

If a service interruption occurs in the Native Recovery channel (due to Order Cache outage) the Native Gateway will send a [System Status](#) message with a Status of Recovery Service Unavailable (2) to all logged in clients of that gateway's recovery channel with Partition ID stamped to indicate the service non-availability of the partition. When this message is received, the clients are expected identify that the recovery service is not available for the partition indicated by AppID. They would be able to continue recovery activities on other partitions without interruptions. If the gateway was in the middle of serving a [Missed Message Request](#), it will send a [Missed Message Report](#) message with 'ResponseType' = 3 (service unavailable) to the client.

If a new [Missed Message Request](#) is sent by a user, the gateway will reject the message with a [Missed Message Request Ack](#) with 'ResponseType' = 3 (service unavailable) to the client. Once the service is available again, Native Gateway will send another [System Status](#) message with AppID to indicate the service availability of the partition to the clients who are still connected on to the recovery channel with 'AppStatus' = 1. When this message is received, the clients are expected to resend the request for missed messages (preferably from the point of interruption) to the gateway to resume the missed message recovery.

Clients already connected to the recovery channel will get the system status message. If they connect after the message was sent and the partition is available the recovery request will be processed as normal.

NOTE: The above mentioned logic will be applicable for both Real Time and Recovery services.

6 MESSAGE FORMATS

This section provides details on the data types, header, nine administrative messages and all the application messages utilised by the server for Basic Native Trading Gateways. For each message, a description of each field is provided along with the applicable data type, offset and length (in bytes). Any message not included in this section will be rejected by the server for this service.

6.1 Data Types

The fields of the messages utilised by the server will support the data types outlined below.

| Data Type | Length | Description |
|-----------|----------|---|
| Alpha | Variable | These fields use standard ASCII character bytes. A field will be null terminated if the full fixed length is unused. The first byte will contain a null if the field is unused. |
| Byte | 1 | A single byte used to hold one ASCII character. |
| Price | 8 | Signed Little-Endian encoded eight byte signed integer field with eight implied decimal places. The maximum value supported through this would be $2^{63} / 100000000$ with eight decimal places |
| UInt8 | 1 | Little-Endian encoded 8 bit unsigned integer. |
| UInt16 | 2 | Little-Endian encoded 16 bit unsigned integer. |
| Int32 | 4 | Little-Endian encoded 32 bit signed integer. |

6.2 Message Overview

6.2.1 Administrative Messages

Administrative messages may be initiated by either the client or the server.

| Message | MsgType | Usage |
|----------------------------|---------|---|
| Logon | A | Used by the client to login to the server. |
| Logon Response | B | Used by the server to accept or reject a login request. |
| Logout | 5 | Used by the client or server to terminate a session. |
| Heartbeat | 0 | Used by the client and server to exercise the communication line during periods of inactivity and verify that the interfaces at each end are available. |
| Reject | 3 | Used by the server to reject a message that does not comply with the Native Trading Gateway specifications. |
| Missed Message Request | M | Used by the client to recover missed messages through the Recovery Channel. |
| Missed Message Request Ack | N | Used by the server to accept or reject a request for missed messages. |
| Transmission Complete | P | Used by the server to indicate that the transmission of missed messages is complete. |
| System Status | n | This message will be disseminated in the recovery channel to indicate Service Non Availability of a partition (due to order cache outage) |

6.2.2 Application Messages of Basic Gateway: Order Handling

a) Client-Initiated

| Message | MsgType | Usage |
|------------------------------|---------|--|
| New Order | 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 orders of equity instruments: (i) All Open or Parked orders (ii) All Open or Parked orders for a particular instrument. (iii) All Open or Parked orders for a particular segment. The mass cancel may apply to the orders of the logged in interface user or to all orders of the firm. |
| Order Cancel/Replace Request | G | Allows the client to modify an Open or Parked order. |
| New Order Cross | C | Allows the client to submit internal Cross Orders |

b) Server-Initiated

| Message | MsgType | Usage |
|--------------------------|---------|--|
| Execution Report | 8 | Indicates one of the following on an order submitted for an equity instrument:: (i) Order accepted. (ii) Order rejected. (iii) Order executed. (iv) Order expired. (v) Order cancelled. (vi) Order cancel/replaced. (vii) Trade cancelled. (viii) Trade corrected. (ix) Order Suspended (x) Order Restated (xi) Order Triggered |
| 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. |

6.2.3 Other Application Messages of the Basic Gateway

| Message | MsgType | Usage |
|---|---------|---|
| Business Reject | j | Indicates that an application message could not be processed and provides a description of the error. |
| Error! Reference source not found. | Z | Disseminates market announcements |

6.3 Message Header

| Field | Offset | Length | Type | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|------------------------------|--------|--------|---|-------|---------|---|-----------|---|--------|---|--------|---|------------------|---|---------------|---|-------|---|----------------|---|-----------|---|----------------------|---|------------------------------|---|------------------------|---|----------------------------|---|-----------------------|---|-----------------|---|---------------|---|---------------------------|---|--------------------------|---|------|---|-----------------|
| Start of Message | 0 | 1 | UInt8 | Indicates the start of the message. Clients will have to send the binary value of '2' at the start of each message. Server will also follow the same protocol | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Message Length | 1 | 2 | UInt16 | Length of the message from the Message Type field onwards. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Message Type | 3 | 1 | Byte | <table border="1"> <thead> <tr> <th>ASCII</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Heartbeat</td> </tr> <tr> <td>3</td> <td>Reject</td> </tr> <tr> <td>5</td> <td>Logout</td> </tr> <tr> <td>8</td> <td>Execution Report</td> </tr> <tr> <td>9</td> <td>Cancel Reject</td> </tr> <tr> <td>A</td> <td>Logon</td> </tr> <tr> <td>B</td> <td>Logon Response</td> </tr> <tr> <td>D</td> <td>New Order</td> </tr> <tr> <td>F</td> <td>Order Cancel Request</td> </tr> <tr> <td>G</td> <td>Order Cancel/Replace Request</td> </tr> <tr> <td>M</td> <td>Missed Message Request</td> </tr> <tr> <td>N</td> <td>Missed Message Request Ack</td> </tr> <tr> <td>P</td> <td>Transmission Complete</td> </tr> <tr> <td>j</td> <td>Business Reject</td> </tr> <tr> <td>n</td> <td>System Status</td> </tr> <tr> <td>q</td> <td>Order Mass Cancel Request</td> </tr> <tr> <td>r</td> <td>Order Mass Cancel Report</td> </tr> <tr> <td>Z</td> <td>News</td> </tr> <tr> <td>C</td> <td>New Order Cross</td> </tr> </tbody> </table> | ASCII | Meaning | 0 | Heartbeat | 3 | Reject | 5 | Logout | 8 | Execution Report | 9 | Cancel Reject | A | Logon | B | Logon Response | D | New Order | F | Order Cancel Request | G | Order Cancel/Replace Request | M | Missed Message Request | N | Missed Message Request Ack | P | Transmission Complete | j | Business Reject | n | System Status | q | Order Mass Cancel Request | r | Order Mass Cancel Report | Z | News | C | New Order Cross |
| ASCII | Meaning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Heartbeat | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Reject | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Logout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | Execution Report | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | Cancel Reject | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | Logon | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B | Logon Response | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D | New Order | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F | Order Cancel Request | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| G | Order Cancel/Replace Request | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M | Missed Message Request | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| N | Missed Message Request Ack | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| P | Transmission Complete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| j | Business Reject | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| n | System Status | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| q | Order Mass Cancel Request | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| r | Order Mass Cancel Report | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Z | News | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | New Order Cross | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

6.4 Administrative Messages

6.4.1 Logon

| Field | Offset | Length | Type | Description |
|------------------|--------|--------|-------|---|
| Header | | | | |
| CompID | 4 | 6 | Alpha | Interface User ID (CompID) assigned to the client. |
| Password | 10 | 25 | Alpha | Password assigned to the Interface User ID (CompID). |
| New Password | 35 | 25 | Alpha | New password for Interface User ID (CompID). |
| Protocol Version | 60 | 4 | Int32 | Specifies the version of protocol. The system will only accept values from 1-9. Current applicable values: 4 ; 2 ; 3 If left empty, the system will consider the default value of 4 . |

6.4.2 Logon Response

| Field | Offset | Length | Type | Description |
|-----------------|--------|--------|-------|---|
| Header | | | | |
| Reject Code | 4 | 4 | Int32 | Will be "0" or '3' if login is accepted/successful. The field will contain a code specifying the reason if login is rejected. Please refer to Section 7.1 for a list of reject codes. |
| Password Expiry | 8 | 4 | Int32 | Number of days for password expiry. This field will be ignored if it contains a negative value. |

6.4.3 Logout

| Field | Offset | Length | Type | Description |
|---------------|--------|--------|-------|--|
| Header | | | | |
| Reason | 4 | 20 | Alpha | Reason for the logout. (For a normal client logout the reason will be "User logout received"). |

6.4.4 Heartbeat

| Field | Offset | Length | Type | Description |
|---------------|--------|--------|------|-------------|
| Header | | | | |

6.4.5 Reject

| Field | Offset | Length | Type | Description |
|-----------------|--------|--------|-------|---|
| Header | | | | |
| Reject Code | 4 | 4 | Int32 | Code specifying the reason for the reject. Please refer to Section 7.2 for a list of reject codes as well as the full list of reject codes in JSE Reject Codes Specification. |
| Reject Reason | 8 | 30 | Alpha | Reason for the reject. If the rejection is due to an issue with a particular field its name will be specified. |
| Message Type | 38 | 1 | Byte | Type of message rejected. |
| Client Order ID | 39 | 20 | Alpha | Client specified identifier of the rejected message if it is available. |

6.4.6 Missed Message Request

| Field | Offset | Length | Type | Description |
|-----------------|--------|--------|-------|---|
| Header | | | | |
| Partition ID | 4 | 1 | UInt8 | Identity of the matching partition the request relates to. |
| Sequence Number | 5 | 4 | Int32 | Sequence number immediately after that of the last message received from the partition. |

6.4.7 Missed Message Request Ack

| Field | Offset | Length | Type | Description | |
|---------------|--------|--------|-------|-----------------------------|-----------------------------|
| Header | | | | | |
| Status | 4 | 1 | UInt8 | Value Meaning | |
| | | | | 0 | Request Accepted/Successful |
| | | | | 1 | Request Limit Reached |
| | | | | 2 | Invalid Partition ID |
| | | | | 3 | Service Unavailable |

6.4.8 Transmission Complete

| Field | Offset | Length | Type | Description | |
|---------------|--------|--------|-------|-----------------------------|--------------------------|
| Header | | | | | |
| Status | 4 | 1 | UInt8 | Value Meaning | |
| | | | | 0 | All Messages Transmitted |
| | | | | 1 | Message Limit Reached |
| | | | | 3 | Service Unavailable |

6.4.9 System Status

| Field | Offset | Length | Type | Description | |
|---------------|--------|--------|-------|--|------------------------------|
| Header | | | | | |
| Partition ID | 4 | 1 | UInt8 | Identity of the matching partition the message relates to. | |
| Status | 5 | 1 | UInt8 | Value Meaning | |
| | | | | 1 | Recovery Service Resumed |
| | | | | 2 | Recovery Service Unavailable |
| | | | | 3 | Partition Suspended |

6.5 Application Messages of the Basic Gateway: Client-Initiated

6.5.1 New Order – Protocol Version 2

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> | | | | | | | | | | | | | | |
|------------------------|---------------------------|---------------|--------------|---|--------------|----------------|----------|---------------------|----------|--------------------|----------|-------------------|----------|-------------------------|-----------|---------------|-----------|---------------------------|
| Header | | | | | | | | | | | | | | | | | | |
| <u>Client Order ID</u> | <u>4</u> | <u>20</u> | <u>Alpha</u> | <u>Client specified identifier of the order.</u> | | | | | | | | | | | | | | |
| <u>Security ID</u> | <u>24</u> | <u>4</u> | <u>Int32</u> | <u>Numeric Identifier of the instrument for which the order is submitted.(Instrument ID)</u> | | | | | | | | | | | | | | |
| <u>Trader Mnemonic</u> | <u>28</u> | <u>17</u> | <u>Alpha</u> | <u>This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID.</u> | | | | | | | | | | | | | | |
| <u>Account</u> | <u>45</u> | <u>10</u> | <u>Alpha</u> | <u>Client Account information. This is the Client Account of the firm who is sending the order. Only Numeric values will be allowed.</u> | | | | | | | | | | | | | | |
| <u>Order Type</u> | <u>55</u> | <u>1</u> | <u>UInt8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>1</u></td> <td><u>Market Order</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Limit Order</u></td> </tr> <tr> <td><u>3</u></td> <td><u>Stop Order</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Stop Limit Order</u></td> </tr> <tr> <td><u>50</u></td> <td><u>Pegged</u></td> </tr> <tr> <td><u>51</u></td> <td><u>Pegged Limit Order</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>1</u> | <u>Market Order</u> | <u>2</u> | <u>Limit Order</u> | <u>3</u> | <u>Stop Order</u> | <u>4</u> | <u>Stop Limit Order</u> | <u>50</u> | <u>Pegged</u> | <u>51</u> | <u>Pegged Limit Order</u> |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Market Order</u> | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Limit Order</u> | | | | | | | | | | | | | | | | | |
| <u>3</u> | <u>Stop Order</u> | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Stop Limit Order</u> | | | | | | | | | | | | | | | | | |
| <u>50</u> | <u>Pegged</u> | | | | | | | | | | | | | | | | | |
| <u>51</u> | <u>Pegged Limit Order</u> | | | | | | | | | | | | | | | | | |

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------|--|---------------|--------------|--|--------------|----------------|----------|------------|----------|-------------------------------|----------|----------------------------------|----------|---------------------------|----------|--------------------------|----------|-----------------------------|----------|-----------------------------|----------|-------------------------------|-----------|--|-----------|--|-----------|---------------------------|-----------|----------------------------------|
| <u>Time In Force</u> | <u>56</u> | <u>1</u> | <u>Uint8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>0</u></td> <td><u>Day</u></td> </tr> <tr> <td><u>1</u></td> <td><u>Good Till Cancel (GTC)</u></td> </tr> <tr> <td><u>3</u></td> <td><u>Immediate or Cancel (IOC)</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Fill or Kill (FOK)</u></td> </tr> <tr> <td><u>5</u></td> <td><u>At the Open (OPG)</u></td> </tr> <tr> <td><u>6</u></td> <td><u>Good Till Date (GTD)</u></td> </tr> <tr> <td><u>8</u></td> <td><u>Good Till Time (GTT)</u></td> </tr> <tr> <td><u>9</u></td> <td><u>Good for Auction (GFA)</u></td> </tr> <tr> <td><u>50</u></td> <td><u>Good for EOD Volume Auction Uncross (GDX)</u></td> </tr> <tr> <td><u>51</u></td> <td><u>Good for Intraday Auction (GFX)</u></td> </tr> <tr> <td><u>10</u></td> <td><u>At the Close (ATC)</u></td> </tr> <tr> <td><u>12</u></td> <td><u>Closing Price Cross (CPX)</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>0</u> | <u>Day</u> | <u>1</u> | <u>Good Till Cancel (GTC)</u> | <u>3</u> | <u>Immediate or Cancel (IOC)</u> | <u>4</u> | <u>Fill or Kill (FOK)</u> | <u>5</u> | <u>At the Open (OPG)</u> | <u>6</u> | <u>Good Till Date (GTD)</u> | <u>8</u> | <u>Good Till Time (GTT)</u> | <u>9</u> | <u>Good for Auction (GFA)</u> | <u>50</u> | <u>Good for EOD Volume Auction Uncross (GDX)</u> | <u>51</u> | <u>Good for Intraday Auction (GFX)</u> | <u>10</u> | <u>At the Close (ATC)</u> | <u>12</u> | <u>Closing Price Cross (CPX)</u> |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>0</u> | <u>Day</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Good Till Cancel (GTC)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>3</u> | <u>Immediate or Cancel (IOC)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Fill or Kill (FOK)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>5</u> | <u>At the Open (OPG)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>6</u> | <u>Good Till Date (GTD)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>8</u> | <u>Good Till Time (GTT)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>9</u> | <u>Good for Auction (GFA)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>50</u> | <u>Good for EOD Volume Auction Uncross (GDX)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>51</u> | <u>Good for Intraday Auction (GFX)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>10</u> | <u>At the Close (ATC)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>12</u> | <u>Closing Price Cross (CPX)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Expire Time</u> | <u>57</u> | <u>17</u> | <u>Alpha</u> | <p><u>Expire time will be in the YYYYMMDD-HH:MM:SS format and specified in UTC. Only the YYYYMMDD is allowed if the Time in Force Type is GTD (6).</u></p> <p><u>This field will be ignored if the Time in Force Type is not GTD (6), or GTT (8).</u></p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Side</u> | <u>74</u> | <u>1</u> | <u>Uint8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>1</u></td> <td><u>Buy</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Sell</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>1</u> | <u>Buy</u> | <u>2</u> | <u>Sell</u> | | | | | | | | | | | | | | | | | | | | |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Buy</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Sell</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Order Quantity</u> | <u>75</u> | <u>4</u> | <u>Int32</u> | <u>Total order quantity.</u> | | | | | | | | | | | | | | | | | | | | | | | | | | |

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> |
|------------------------------|---------------|---------------|--------------|--|
| <u>Display Quantity</u> | <u>79</u> | <u>4</u> | <u>Int32</u> | <u>Maximum quantity that may be displayed.</u> <u>The intended display quantity has to be inserted as this is a mandatory field. This will be zero for a Pegged, Pegged Limit or GDX Order.</u> |
| <u>Minimum Quantity</u> | <u>83</u> | <u>4</u> | <u>Int32</u> | <u>Minimum Execution Size that needs to be specified for a Pegged or Pegged Limit Order which must be greater than or equal to Minimum Reserve Size.</u> |
| <u>Limit Price</u> | <u>87</u> | <u>8</u> | <u>Price</u> | <u>Limit price. This field will be ignored if the Order Type is not Limit (2) or Stop Limit (4).</u> |
| <u>Stop Price</u> | <u>95</u> | <u>8</u> | <u>Price</u> | <u>Stop price or Hard Limit. This field will be ignored if the Order Type is not Stop (3), Stop Limit (4) or Pegged Limit Order (51).</u> <u>Hard Limit Price of a Pegged Limit Order will be specified here.</u> |
| <u>Capacity</u> | <u>103</u> | <u>1</u> | <u>Uint8</u> | <u>Value Meaning</u> <hr/> <u>2</u> <u>Principal</u> <hr/> <u>3</u> <u>Agency</u> |
| <u>Cancel on Disconnect</u> | <u>104</u> | <u>1</u> | <u>Uint8</u> | <u>Value Meaning</u> <hr/> <u>0</u> <u>Do Not Cancel</u> <hr/> <u>1</u> <u>Cancel</u> |
| <u>Order Book</u> | <u>105</u> | <u>1</u> | <u>Uint8</u> | <u>Value Meaning</u> <hr/> <u>1</u> <u>Regular</u> |
| <u>Execution Instruction</u> | <u>106</u> | <u>1</u> | <u>Int8</u> | <u>Value Meaning</u> <hr/> <u>0</u> <u>None</u> <hr/> <u>2</u> <u>Include in EOD Volume Auction Uncross</u> |

| <u>Order Sub Type</u> | <u>107</u> | <u>1</u> | <u>UInt8</u> | <p><u>Whether the order is a pegged order.</u> <u>If the client submits a Pegged Order with a limit price (a so called "Hard Limit"), the Order Type has to be Pegged Limit (51) along with Order Sub Type with a value 50, 51 or 52. If the client submits a Pegged without a limit price, the Order Type has to be Pegged (50) along with Order Sub Type with a value 50, 51 or 52."</u></p> <p><u>If Side = 1 (Buy) the valid Order Sub Types will be 50 and 51.</u> <u>If Side = 2 (Sell) the valid Order Sub Types will be 50 and 52.</u></p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>50</u></td> <td><u>Pegged to Mid</u></td> </tr> <tr> <td><u>51</u></td> <td><u>Pegged to Bid</u></td> </tr> <tr> <td><u>52</u></td> <td><u>Pegged to Offer</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>50</u> | <u>Pegged to Mid</u> | <u>51</u> | <u>Pegged to Bid</u> | <u>52</u> | <u>Pegged to Offer</u> |
|-----------------------|------------------------|----------|--------------|---|--------------|----------------|-----------|----------------------|-----------|----------------------|-----------|------------------------|
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | |
| <u>50</u> | <u>Pegged to Mid</u> | | | | | | | | | | | |
| <u>51</u> | <u>Pegged to Bid</u> | | | | | | | | | | | |
| <u>52</u> | <u>Pegged to Offer</u> | | | | | | | | | | | |

6.5.2 New Order – Protocol Version 3

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> | | | | | | | | | | | | | | |
|------------------------|---------------------------|---------------|--------------|---|--------------|----------------|----------|---------------------|----------|--------------------|----------|-------------------|----------|-------------------------|-----------|---------------|-----------|---------------------------|
| <u>Header</u> | | | | | | | | | | | | | | | | | | |
| <u>Client Order ID</u> | <u>4</u> | <u>20</u> | <u>Alpha</u> | <u>Client specified identifier of the order.</u> | | | | | | | | | | | | | | |
| <u>Security ID</u> | <u>24</u> | <u>4</u> | <u>Int32</u> | <u>Numeric Identifier of the instrument for which the order is submitted.(Instrument ID)</u> | | | | | | | | | | | | | | |
| <u>Trader Mnemonic</u> | <u>28</u> | <u>17</u> | <u>Alpha</u> | <u>This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID.</u> | | | | | | | | | | | | | | |
| <u>Account</u> | <u>45</u> | <u>10</u> | <u>Alpha</u> | <u>Client Account information. This is the Client Account of the firm who is sending the order. Only Numeric values will be allowed.</u> | | | | | | | | | | | | | | |
| <u>Order Type</u> | <u>55</u> | <u>1</u> | <u>UInt8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>1</u></td> <td><u>Market Order</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Limit Order</u></td> </tr> <tr> <td><u>3</u></td> <td><u>Stop Order</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Stop Limit Order</u></td> </tr> <tr> <td><u>50</u></td> <td><u>Pegged</u></td> </tr> <tr> <td><u>51</u></td> <td><u>Pegged Limit Order</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>1</u> | <u>Market Order</u> | <u>2</u> | <u>Limit Order</u> | <u>3</u> | <u>Stop Order</u> | <u>4</u> | <u>Stop Limit Order</u> | <u>50</u> | <u>Pegged</u> | <u>51</u> | <u>Pegged Limit Order</u> |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Market Order</u> | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Limit Order</u> | | | | | | | | | | | | | | | | | |
| <u>3</u> | <u>Stop Order</u> | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Stop Limit Order</u> | | | | | | | | | | | | | | | | | |
| <u>50</u> | <u>Pegged</u> | | | | | | | | | | | | | | | | | |
| <u>51</u> | <u>Pegged Limit Order</u> | | | | | | | | | | | | | | | | | |

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------|--|---------------|--------------|--|--------------|----------------|----------|------------|----------|-------------------------------|----------|----------------------------------|----------|---------------------------|----------|--------------------------|----------|-----------------------------|----------|-----------------------------|----------|-------------------------------|-----------|--|-----------|--|-----------|---------------------------|-----------|----------------------------------|
| <u>Time In Force</u> | <u>56</u> | <u>1</u> | <u>UInt8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>0</u></td> <td><u>Day</u></td> </tr> <tr> <td><u>1</u></td> <td><u>Good Till Cancel (GTC)</u></td> </tr> <tr> <td><u>3</u></td> <td><u>Immediate or Cancel (IOC)</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Fill or Kill (FOK)</u></td> </tr> <tr> <td><u>5</u></td> <td><u>At the Open (OPG)</u></td> </tr> <tr> <td><u>6</u></td> <td><u>Good Till Date (GTD)</u></td> </tr> <tr> <td><u>8</u></td> <td><u>Good Till Time (GTT)</u></td> </tr> <tr> <td><u>9</u></td> <td><u>Good for Auction (GFA)</u></td> </tr> <tr> <td><u>50</u></td> <td><u>Good for EOD Volume Auction Uncross (GDX)</u></td> </tr> <tr> <td><u>51</u></td> <td><u>Good for Intraday Auction (GFX)</u></td> </tr> <tr> <td><u>10</u></td> <td><u>At the Close (ATC)</u></td> </tr> <tr> <td><u>12</u></td> <td><u>Closing Price Cross (CPX)</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>0</u> | <u>Day</u> | <u>1</u> | <u>Good Till Cancel (GTC)</u> | <u>3</u> | <u>Immediate or Cancel (IOC)</u> | <u>4</u> | <u>Fill or Kill (FOK)</u> | <u>5</u> | <u>At the Open (OPG)</u> | <u>6</u> | <u>Good Till Date (GTD)</u> | <u>8</u> | <u>Good Till Time (GTT)</u> | <u>9</u> | <u>Good for Auction (GFA)</u> | <u>50</u> | <u>Good for EOD Volume Auction Uncross (GDX)</u> | <u>51</u> | <u>Good for Intraday Auction (GFX)</u> | <u>10</u> | <u>At the Close (ATC)</u> | <u>12</u> | <u>Closing Price Cross (CPX)</u> |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>0</u> | <u>Day</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Good Till Cancel (GTC)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>3</u> | <u>Immediate or Cancel (IOC)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Fill or Kill (FOK)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>5</u> | <u>At the Open (OPG)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>6</u> | <u>Good Till Date (GTD)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>8</u> | <u>Good Till Time (GTT)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>9</u> | <u>Good for Auction (GFA)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>50</u> | <u>Good for EOD Volume Auction Uncross (GDX)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>51</u> | <u>Good for Intraday Auction (GFX)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>10</u> | <u>At the Close (ATC)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>12</u> | <u>Closing Price Cross (CPX)</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Expire Time</u> | <u>57</u> | <u>17</u> | <u>Alpha</u> | <p><u>Expire time will be in the YYYYMMDD-HH:MM:SS format and specified in UTC. Only the YYYYMMDD is allowed if the Time in Force Type is GTD (6).</u></p> <p><u>This field will be ignored if the Time in Force Type is not GTD (6), or GTT (8).</u></p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Side</u> | <u>74</u> | <u>1</u> | <u>UInt8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>1</u></td> <td><u>Buy</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Sell</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>1</u> | <u>Buy</u> | <u>2</u> | <u>Sell</u> | | | | | | | | | | | | | | | | | | | | |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Buy</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Sell</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Order Quantity</u> | <u>75</u> | <u>4</u> | <u>Int32</u> | <u>Total order quantity.</u> | | | | | | | | | | | | | | | | | | | | | | | | | | |

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> |
|------------------------------|---------------|---------------|--------------|--|
| <u>Display Quantity</u> | <u>79</u> | <u>4</u> | <u>Int32</u> | <u>Maximum quantity that may be displayed.</u> <u>The intended display quantity has to be inserted as this is a mandatory field. This will be zero for a Pegged, Pegged Limit or GDX Order.</u> |
| <u>Minimum Quantity</u> | <u>83</u> | <u>4</u> | <u>Int32</u> | <u>Minimum Execution Size that needs to be specified for a Pegged or Pegged Limit Order which must be greater than or equal to Minimum Reserve Size.</u> |
| <u>Limit Price</u> | <u>87</u> | <u>8</u> | <u>Price</u> | <u>Limit price. This field will be ignored if the Order Type is not Limit (2) or Stop Limit (4).</u> |
| <u>Stop Price</u> | <u>95</u> | <u>8</u> | <u>Price</u> | <u>Stop price or Hard Limit. This field will be ignored if the Order Type is not Stop (3), Stop Limit (4) or Pegged Limit Order (51).</u> <u>Hard Limit Price of a Pegged Limit Order will be specified here.</u> |
| <u>Capacity</u> | <u>103</u> | <u>1</u> | <u>UInt8</u> | <u>Value Meaning</u> <hr/> <u>2 Principal</u> <hr/> <u>3 Agency</u> |
| <u>Cancel on Disconnect</u> | <u>104</u> | <u>1</u> | <u>UInt8</u> | <u>Value Meaning</u> <hr/> <u>0 Do Not Cancel</u> <hr/> <u>1 Cancel</u> |
| <u>Order Book</u> | <u>105</u> | <u>1</u> | <u>UInt8</u> | <u>Value Meaning</u> <hr/> <u>1 Regular</u> |
| <u>Execution Instruction</u> | <u>106</u> | <u>1</u> | <u>Int8</u> | <u>Value Meaning</u> <hr/> <u>0 None</u> <hr/> <u>2 Include in EOD Volume Auction Uncross</u> |

| <u>Order Sub Type</u> | <u>107</u> | <u>1</u> | <u>UInt8</u> | <p><u>Whether the order is a pegged order.</u> <u>If the client submits a Pegged Order with a limit price (a so called "Hard Limit"), the Order Type has to be Pegged Limit (51) along with Order Sub Type with a value 50, 51 or 52. If the client submits a Pegged without a limit price, the Order Type has to be Pegged (50) along with Order Sub Type with a value 50, 51 or 52."</u></p> <p><u>If Side = 1 (Buy) the valid Order Sub Types will be 50 and 51.</u> <u>If Side = 2 (Sell) the valid Order Sub Types will be 50 and 52.</u></p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>50</u></td> <td><u>Pegged to Mid</u></td> </tr> <tr> <td><u>51</u></td> <td><u>Pegged to Bid</u></td> </tr> <tr> <td><u>52</u></td> <td><u>Pegged to Offer</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>50</u> | <u>Pegged to Mid</u> | <u>51</u> | <u>Pegged to Bid</u> | <u>52</u> | <u>Pegged to Offer</u> |
|--|------------------------|-----------|--------------|---|--------------|----------------|-----------|----------------------|-----------|----------------------|-----------|------------------------|
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | |
| <u>50</u> | <u>Pegged to Mid</u> | | | | | | | | | | | |
| <u>51</u> | <u>Pegged to Bid</u> | | | | | | | | | | | |
| <u>52</u> | <u>Pegged to Offer</u> | | | | | | | | | | | |
| <u>Following field(s) have been introduced by Protocol v3</u> | | | | | | | | | | | | |
| <u>Self Trade Prevention Key</u> | <u>108</u> | <u>12</u> | <u>Alpha</u> | <u>Client specified identifier of order relevant to self-match.</u> | | | | | | | | |

6.4.106.5.3 Order Cancel Request

| Field | Offset | Length | Type | Description | | | | | | |
|----------------------|---------|--------|-------|--|-------|---------|---|---------|---|------|
| Header | | | | | | | | | | |
| Client Order ID | 4 | 20 | Alpha | Client specified identifier of the request. | | | | | | |
| Orig Client Order ID | 24 | 20 | Alpha | Client specified identifier of the order being cancelled. | | | | | | |
| Order ID | 44 | 12 | Alpha | Unique identifier of the order assigned by the matching system. | | | | | | |
| Security ID | 56 | 4 | Int32 | Unique numeric identifier of the instrument (Instrument ID) being cancelled. | | | | | | |
| Trader Mnemonic | 60 | 17 | Alpha | This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID. | | | | | | |
| Side | 77 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Buy</td> </tr> <tr> <td>2</td> <td>Sell</td> </tr> </tbody> </table> | Value | Meaning | 1 | Buy | 2 | Sell |
| Value | Meaning | | | | | | | | | |
| 1 | Buy | | | | | | | | | |
| 2 | Sell | | | | | | | | | |
| Order Book | 78 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular | | |
| Value | Meaning | | | | | | | | | |
| 1 | Regular | | | | | | | | | |

6.4.116.5.4 Order Mass Cancel Request

| Field | Offset | Length | Type | Description | | | | | | | | | | | | | | |
|--------------------------|--|--------|-------|---|-------|---------|---|--------------------------------|---|-----------------------------|---|---|---|---------------------|---|--|----|---|
| Header | | | | | | | | | | | | | | | | | | |
| Client Order ID | 4 | 20 | Alpha | Client specified identifier of the cancel request. | | | | | | | | | | | | | | |
| Mass Cancel Request Type | 24 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>All Firm orders for Instrument</td> </tr> <tr> <td>4</td> <td>All Firm orders for Segment</td> </tr> <tr> <td>7</td> <td>All orders for Client (Interface User ID)</td> </tr> <tr> <td>8</td> <td>All orders for Firm</td> </tr> <tr> <td>9</td> <td>Client (Interface User ID) orders for Instrument</td> </tr> <tr> <td>15</td> <td>Client (Interface User ID) orders for Segment</td> </tr> </tbody> </table> | Value | Meaning | 3 | All Firm orders for Instrument | 4 | All Firm orders for Segment | 7 | All orders for Client (Interface User ID) | 8 | All orders for Firm | 9 | Client (Interface User ID) orders for Instrument | 15 | Client (Interface User ID) orders for Segment |
| Value | Meaning | | | | | | | | | | | | | | | | | |
| 3 | All Firm orders for Instrument | | | | | | | | | | | | | | | | | |
| 4 | All Firm orders for Segment | | | | | | | | | | | | | | | | | |
| 7 | All orders for Client (Interface User ID) | | | | | | | | | | | | | | | | | |
| 8 | All orders for Firm | | | | | | | | | | | | | | | | | |
| 9 | Client (Interface User ID) orders for Instrument | | | | | | | | | | | | | | | | | |
| 15 | Client (Interface User ID) orders for Segment | | | | | | | | | | | | | | | | | |
| Security ID | 25 | 4 | Int32 | Numeric identifier of instrument (Instrument ID) of orders being cancelled. Required if Mass Cancel Request Type = 3 or 9. Else this field will be ignored and can be null. | | | | | | | | | | | | | | |
| Segment | 29 | 6 | Alpha | Identifier of the segment for which orders will be cancelled. Please refer to 09 - JSE Reference Data Management for the valid segments. Required if MassCancelRequestType = 4 or 15. Else this field will be ignored and can be null. | | | | | | | | | | | | | | |
| Order Sub Type | 35 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Order</td> </tr> </tbody> </table> | Value | Meaning | 0 | Order | | | | | | | | | | |
| Value | Meaning | | | | | | | | | | | | | | | | | |
| 0 | Order | | | | | | | | | | | | | | | | | |
| Order Book | 36 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular | | | | | | | | | | |
| Value | Meaning | | | | | | | | | | | | | | | | | |
| 1 | Regular | | | | | | | | | | | | | | | | | |

6.4.126.5.5 Order Cancel/ Replace Request

| Field | Offset | Length | Type | Description | | | | | | | | | | | | | | |
|--------------------------|--------------------|--------|--------|---|-------|---------|---|--------------|---|-------------|---|------------|---|------------------|----|--------|----|--------------------|
| Header | | | | | | | | | | | | | | | | | | |
| Client Order ID | 4 | 20 | Alpha | Client specified identifier of the request. | | | | | | | | | | | | | | |
| Original Client Order ID | 24 | 20 | Alpha | Client specified identifier of the order being amended. | | | | | | | | | | | | | | |
| Order ID | 44 | 12 | String | Unique identifier of the order assigned by the matching system | | | | | | | | | | | | | | |
| Security ID | 56 | 4 | Int32 | Identifier of the Instrument (Instrument ID) of the order being amended. | | | | | | | | | | | | | | |
| TraderMnemonic | 60 | 17 | Alpha | This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID. | | | | | | | | | | | | | | |
| Account | 77 | 10 | Alpha | Client Account information. This is the Client Account of the firm who is sending the order. | | | | | | | | | | | | | | |
| Order Type | 87 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Market Order</td> </tr> <tr> <td>2</td> <td>Limit Order</td> </tr> <tr> <td>3</td> <td>Stop Order</td> </tr> <tr> <td>4</td> <td>Stop Limit Order</td> </tr> <tr> <td>50</td> <td>Pegged</td> </tr> <tr> <td>51</td> <td>Pegged Limit Order</td> </tr> </tbody> </table> | Value | Meaning | 1 | Market Order | 2 | Limit Order | 3 | Stop Order | 4 | Stop Limit Order | 50 | Pegged | 51 | Pegged Limit Order |
| Value | Meaning | | | | | | | | | | | | | | | | | |
| 1 | Market Order | | | | | | | | | | | | | | | | | |
| 2 | Limit Order | | | | | | | | | | | | | | | | | |
| 3 | Stop Order | | | | | | | | | | | | | | | | | |
| 4 | Stop Limit Order | | | | | | | | | | | | | | | | | |
| 50 | Pegged | | | | | | | | | | | | | | | | | |
| 51 | Pegged Limit Order | | | | | | | | | | | | | | | | | |

| Time In Force | 88 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Day</td> </tr> <tr> <td>1</td> <td>Good Till Cancel (GTC)</td> </tr> <tr> <td>3</td> <td>Immediate or Cancel (IOC)</td> </tr> <tr> <td>4</td> <td>Fill or Kill (FOK)</td> </tr> <tr> <td>5</td> <td>At the Opening</td> </tr> <tr> <td>6</td> <td>Good Till Date (GTD)</td> </tr> <tr> <td>8</td> <td>Good Till Time (GTT)</td> </tr> <tr> <td>9</td> <td>Good for Auction (GFA)</td> </tr> <tr> <td>50</td> <td>Good for EOD Volume Auction Uncross (GDX)</td> </tr> <tr> <td>51</td> <td>Good for Intraday Auction (GFX)</td> </tr> <tr> <td>10</td> <td>At the Close (ATC)</td> </tr> <tr> <td>12</td> <td>Closing Price Cross (CPX)</td> </tr> </tbody> </table> | Value | Meaning | 0 | Day | 1 | Good Till Cancel (GTC) | 3 | Immediate or Cancel (IOC) | 4 | Fill or Kill (FOK) | 5 | At the Opening | 6 | Good Till Date (GTD) | 8 | Good Till Time (GTT) | 9 | Good for Auction (GFA) | 50 | Good for EOD Volume Auction Uncross (GDX) | 51 | Good for Intraday Auction (GFX) | 10 | At the Close (ATC) | 12 | Closing Price Cross (CPX) |
|------------------|---|----|-------|---|-------|---------|---|-----|---|------------------------|---|---------------------------|---|--------------------|---|----------------|---|----------------------|---|----------------------|---|------------------------|----|---|----|---------------------------------|----|--------------------|----|---------------------------|
| Value | Meaning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Day | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Good Till Cancel (GTC) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Immediate or Cancel (IOC) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Fill or Kill (FOK) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | At the Opening | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Good Till Date (GTD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | Good Till Time (GTT) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | Good for Auction (GFA) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 50 | Good for EOD Volume Auction Uncross (GDX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 51 | Good for Intraday Auction (GFX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | At the Close (ATC) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | Closing Price Cross (CPX) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Expire Time | 89 | 17 | Alpha | <p>This field will indicate the date/time the order expires on.</p> <p>Expire time will be in the YYYYMMDD-HH:MM:SS date and time format and specified in UTC. Only the YYYYMMDD will be considered if the Time in Force Type is Good Till Date (6). Field will be ignored if the Time in Force Type is not Good Till Date (6) or Good Till Time (8).</p> <p>It is mandatory to specify a valid value in this field for GTD/GTT orders. If 0 is specified for GTD/GTT orders, the request will be rejected. For non GTD/GTT orders, the value in this field will be ignored.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Side | 106 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Buy</td> </tr> <tr> <td>2</td> <td>Sell</td> </tr> </tbody> </table> | Value | Meaning | 1 | Buy | 2 | Sell | | | | | | | | | | | | | | | | | | | | |
| Value | Meaning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Buy | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Sell | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Order Quantity | 107 | 4 | Int32 | Total order quantity. | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Display Quantity | 111 | 4 | Int32 | <p>Maximum quantity to be displayed.</p> <p>The value should be 0 for a Pegged ,Pegged Limit, GDX order.</p> <p>The value should equal the order quantity for a visible order.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Minimum Quantity | 115 | 4 | Int32 | Minimum execution size that needs to be specified for a Pegged or Pegged Limit Order. | | | | |
|------------------|---------|---|-------|---|-------|---------|---|---------|
| Limit Price | 119 | 8 | Price | Limit price. Field will be ignored if Order Type is not Limit (2) or Stop Limit (4). | | | | |
| Stop Price | 127 | 8 | Price | Stop price. Field will be ignored if Order Type is not Stop (3) ,Stop Limit (4) or Pegged Limit Order (51). Hard Limit Price of a Pegged Limit Order will be specified here. The same stop price should be specified if this field is not being amended | | | | |
| Order Book | 135 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular |
| Value | Meaning | | | | | | | |
| 1 | Regular | | | | | | | |

6.4.136.5.6 New Order Cross

| Field | Offset | Length | Type | Description |
|---------------------------|--------|--------|-------|---|
| Header | | | | |
| Cross ID | 4 | 20 | Alpha | An identifier of the Cross Order. This will be unique across the trading day. Required for Cross Orders. Only Alpha numeric values will be allowed in this field. No special characters will be allowed. |
| Cross Type | 24 | 1 | UInt8 | The type of the Cross Order: Value Meaning 5 Internal Cross 50 Internal Cross (Price Adjustable) Any other value will be rejected via a Reject message |
| Buy Side Client Order ID | 25 | 20 | Alpha | Client specified identifier of the buy side. |
| Buy Side Capacity | 45 | 1 | UInt8 | Capacity of the buy side. Value Meaning 2 Principal 3 Agency Any other value will be rejected via a Reject message. |
| Buy Side Trader Mnemonic | 46 | 17 | Alpha | This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID. |
| Buy Side Account | 63 | 10 | Alpha | Client Account information of the buy side This is the Client Account of the firm who is sending the buy side of the Cross Order. Only numeric values will be allowed. |
| Sell Side Client Order ID | 73 | 20 | Alpha | Client specified identifier of the sell side. |
| Sell Side Capacity | 93 | 1 | UInt8 | Capacity of the sell side. Value Meaning 2 Principal 3 Agency Any other value will be rejected via a Reject message. |
| Sell Side Trader Mnemonic | 94 | 17 | Alpha | This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID. |
| Sell Side Account | 111 | 10 | Alpha | Client Account information of the sell side. This is the Client Account of the firm who is sending the sell side of the Cross Order. Only numeric values will be allowed. |
| Security ID | 121 | 4 | Int32 | Numeric Identifier of the instrument for which the Cross Order is submitted.(Instrument ID) |
| Order Type | 125 | 1 | UInt8 | Type of the order. Value Meaning 2 Limit Any other value will be rejected via a Reject message. |

| | | | | |
|----------------|-----|---|-------|---|
| Time In Force | 126 | 1 | UInt8 | Time qualifier of the order. Only DAY TIF is allowed for Cross Orders. Value Meaning 0 Day |
| Limit Price | 127 | 8 | Price | Limit price. This field will be ignored if the Order Type is not Limit (2) or Stop Limit (4). |
| Order Quantity | 135 | 4 | Int32 | Total order quantity of the Cross Order |

6.6 Application Messages of the Basic Gateway: Server-Initiated

6.6.1 Execution Report - Protocol Version: 2

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|-------------------------|---------------|--------------|--|--------------|----------------|----------|------------|----------|------------------|----------|-------------------------|----------|-----------------|----------|------------------|----------|----------------|----------|--------------|----------|----------------------|----------|---------------------|----------|-----------------|----------|------------------|
| Header | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Partition ID</u> | <u>4</u> | <u>1</u> | <u>UInt8</u> | <u>Identity of the matching partition.</u> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Sequence Number</u> | <u>5</u> | <u>4</u> | <u>Int32</u> | <u>Message sequence number of the matching partition.</u> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Execution ID</u> | <u>9</u> | <u>21</u> | <u>Alpha</u> | <p><u>Unique Identifier of the Execution Report. Unique across all partitions and all trading days.</u></p> <p><u>On Order confirmation, amendment or cancellation a unique Execution ID will be generated.</u></p> <p><u>Identifier of the Exec ID of the execution report for the trade being cancelled or corrected if Execution Type is Trade Cancel (H) or Trade Correct (G).</u></p> <p><u>This will also match to the Side ExecID field in a Trade Capture Report generated for a trade.</u></p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Client Order ID</u> | <u>30</u> | <u>20</u> | <u>Alpha</u> | <p><u>Client specified identifier of the order, order cancel request or order cancel/replace request.</u></p> <p><u>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.</u></p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Order ID</u> | <u>50</u> | <u>12</u> | <u>Alpha</u> | <u>Server specified identifier of the order.</u> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Execution Type</u> | <u>62</u> | <u>1</u> | <u>Alpha</u> | <p><u>Execution Type of the order.</u></p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>0</u></td> <td><u>New</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Cancelled</u></td> </tr> <tr> <td><u>5</u></td> <td><u>Amended/Modified</u></td> </tr> <tr> <td><u>8</u></td> <td><u>Rejected</u></td> </tr> <tr> <td><u>9</u></td> <td><u>Suspended</u></td> </tr> <tr> <td><u>C</u></td> <td><u>Expired</u></td> </tr> <tr> <td><u>F</u></td> <td><u>Trade</u></td> </tr> <tr> <td><u>G</u></td> <td><u>Trade Correct</u></td> </tr> <tr> <td><u>H</u></td> <td><u>Trade Cancel</u></td> </tr> <tr> <td><u>D</u></td> <td><u>Restated</u></td> </tr> <tr> <td><u>L</u></td> <td><u>Triggered</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>0</u> | <u>New</u> | <u>4</u> | <u>Cancelled</u> | <u>5</u> | <u>Amended/Modified</u> | <u>8</u> | <u>Rejected</u> | <u>9</u> | <u>Suspended</u> | <u>C</u> | <u>Expired</u> | <u>F</u> | <u>Trade</u> | <u>G</u> | <u>Trade Correct</u> | <u>H</u> | <u>Trade Cancel</u> | <u>D</u> | <u>Restated</u> | <u>L</u> | <u>Triggered</u> |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>0</u> | <u>New</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Cancelled</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>5</u> | <u>Amended/Modified</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>8</u> | <u>Rejected</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>9</u> | <u>Suspended</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>C</u> | <u>Expired</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>F</u> | <u>Trade</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>G</u> | <u>Trade Correct</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>H</u> | <u>Trade Cancel</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>D</u> | <u>Restated</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>L</u> | <u>Triggered</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | |
|--------------------------|------------|-----------|--------------|--|
| <u>Order Status</u> | <u>63</u> | <u>1</u> | <u>UInt8</u> | <p><u>Value</u> <u>Meaning</u></p> <hr/> <p><u>0</u> <u>New</u></p> <hr/> <p><u>1</u> <u>Partially Filled</u></p> <hr/> <p><u>2</u> <u>Filled</u></p> <hr/> <p><u>4</u> <u>Cancelled</u></p> <hr/> <p><u>6</u> <u>Expired</u></p> <hr/> <p><u>8</u> <u>Rejected</u></p> <hr/> <p><u>9</u> <u>Suspended</u></p> |
| <u>Reject Code</u> | <u>64</u> | <u>4</u> | <u>Int32</u> | Code specifying the reason for the reject. Please refer Section 8.2 for a list of reject codes as well as the full list of reject Codes Specification. Field will be ignored if Execution Type is not Rejected (8). |
| <u>Executed Price</u> | <u>68</u> | <u>8</u> | <u>Price</u> | Executed price of the trade in ZAC. This field will be ignored if Execution Type is not Trade (F) or Trade Correct (G). |
| <u>Executed Quantity</u> | <u>76</u> | <u>4</u> | <u>Int32</u> | Executed quantity. This field will be ignored if Execution Type is not Trade (F) or Trade Correct (G). This will always be 0 on a Same Day On Book trade cancellation. |
| <u>Leaves Quantity</u> | <u>80</u> | <u>4</u> | <u>Int32</u> | Quantity available for further execution. It is the remaining quantity of the order. |
| <u>Working Indicator</u> | <u>84</u> | <u>1</u> | <u>UInt8</u> | Whether the order is currently being worked on. Will only be applicable when Order Status is New (0). <u>Value</u> <u>Meaning</u> <hr/> <u>0</u> <u>Unset</u> <hr/> <u>1</u> <u>Order is being worked</u> <hr/> <u>2</u> <u>Order is not currently in a working state</u> |
| <u>Security ID</u> | <u>85</u> | <u>4</u> | <u>Int32</u> | Identifier of the instrument the Execution Report is sent for. |
| <u>Side</u> | <u>89</u> | <u>1</u> | <u>UInt8</u> | <u>Value</u> <u>Meaning</u> <hr/> <u>1</u> <u>Buy</u> <hr/> <u>2</u> <u>Sell</u> |
| <u>TraderMnemonic</u> | <u>90</u> | <u>17</u> | <u>Alpha</u> | This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID. |
| <u>Account</u> | <u>107</u> | <u>10</u> | <u>Alpha</u> | Client Account information. This is the Client Account of the firm who submitted the order. |

| IsMarketOpsRequest | 117 | 1 | UInt8 | <p>This field indicates whether a New Order, Cancel Request or Order Cancel or Replace Request was submitted by Market Operations.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No</td> </tr> <tr> <td>1</td> <td>Yes</td> </tr> </tbody> </table> | Value | Meaning | 0 | No | 1 | Yes |
|---------------------------------------|---|--------------------|------------------------|---|-----------------------|-------------------------|-------------------|--------------------------------|--------------------|---|
| Value | Meaning | | | | | | | | | |
| 0 | No | | | | | | | | | |
| 1 | Yes | | | | | | | | | |
| Transact Time | 118 | 8 | UInt64 | Time the Execution Report was generated. | | | | | | |
| Order Book | 126 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular | | |
| Value | Meaning | | | | | | | | | |
| 1 | Regular | | | | | | | | | |
| Execution Instruction | 127 | 1 | Int8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>None</td> </tr> <tr> <td>2</td> <td>Include in EOD Volume Auction Uncross</td> </tr> </tbody> </table> <p>This will have a default value of 0 if Order Type is not Limit or 2 if the Time in Force type is not GTT (8).</p> | Value | Meaning | 0 | None | 2 | Include in EOD Volume Auction Uncross |
| Value | Meaning | | | | | | | | | |
| 0 | None | | | | | | | | | |
| 2 | Include in EOD Volume Auction Uncross | | | | | | | | | |
| Cross ID | 128 | 20 | Alpha | <p>The unique ID of the Cross Order. Only populated for execution report messages generated for Internal Cross Orders.</p> <p>The value submitted with the New Order Cross Message will be populated.</p> | | | | | | |
| Cross Type | 148 | 1 | UInt8 | <p>The type of the Cross Order. Only populated for execution report messages generated Internal Cross Orders.</p> <p>The value submitted with the New Order Cross Message will be populated.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>Internal Cross</td> </tr> <tr> <td>50</td> <td>Internal Cross (Price Adjustable)</td> </tr> </tbody> </table> | Value | Meaning | 5 | Internal Cross | 50 | Internal Cross (Price Adjustable) |
| Value | Meaning | | | | | | | | | |
| 5 | Internal Cross | | | | | | | | | |
| 50 | Internal Cross (Price Adjustable) | | | | | | | | | |
| Display Quantity | 149 | 4 | Int32 | Current visible quantity. | | | | | | |
| Public Order ID | 153 | 12 | Alpha | Server specified public order identifier of the order. | | | | | | |

| Indicator Flags | 165 | 1 | Bit field | <p>Denotes whether a given side of the trade is the aggressor side or the passive side. Applicable only when the Exec Type is Trade or Trade Correct. Should be ignored for other Exec Types. During Auctions, both side ERs of a trade will have '0' stamped.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Name</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0th</td> <td>Aggressor Indicator</td> <td>0 : Order initiator is passive</td> </tr> <tr> <td></td> <td></td> <td>1 : Order initiator is aggressor</td> </tr> </tbody> </table> | Bit | Name | Meaning | 0th | Aggressor Indicator | 0 : Order initiator is passive | | | 1 : Order initiator is aggressor | | | | | | | | | |
|-------------------------------------|--|--|---------------------------|---|-----------------------|-------------------------|-------------------------|--------------------------------|-------------------------------------|--|-------------------|-----------------------------------|--|---|--|---|-------------------|---|--|-------------------------------|-------------------|---|
| Bit | Name | Meaning | | | | | | | | | | | | | | | | | | | | |
| 0th | Aggressor Indicator | 0 : Order initiator is passive | | | | | | | | | | | | | | | | | | | | |
| | | 1 : Order initiator is aggressor | | | | | | | | | | | | | | | | | | | | |
| Liquidity Indicator | 166 | 1 | UInt8 | <p>Indicator to identify whether this fill was a result of a liquidity provider providing or liquidity taker taking the liquidity. Required if Order Status is Partially Filled (1) or Filled (2) and Execution Type is Trade (F).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Unset</td> </tr> <tr> <td>1</td> <td>Added Liquidity</td> </tr> <tr> <td>2</td> <td>Removed Liquidity</td> </tr> <tr> <td>4</td> <td>Auction</td> </tr> </tbody> </table> | Value | Meaning | 0 | Unset | 1 | Added Liquidity | 2 | Removed Liquidity | 4 | Auction | | | | | | | | |
| Value | Meaning | | | | | | | | | | | | | | | | | | | | | |
| 0 | Unset | | | | | | | | | | | | | | | | | | | | | |
| 1 | Added Liquidity | | | | | | | | | | | | | | | | | | | | | |
| 2 | Removed Liquidity | | | | | | | | | | | | | | | | | | | | | |
| 4 | Auction | | | | | | | | | | | | | | | | | | | | | |
| TypeofTrade | 167 | 1 | UInt8 | <p>Indicates whether the executed portion is visible or hidden. Valid only if ExecType (150) = F. Ignore value in all other cases.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td></td> <td>Visible</td> </tr> <tr> <td>0</td> <td>The executed portion was passive and visible when the trade occurred</td> </tr> <tr> <td></td> <td>Hidden</td> </tr> <tr> <td>1</td> <td>The executed portion was passive and hidden when the trade occurred</td> </tr> <tr> <td></td> <td>Not specified (i.e. Ignore this field)</td> </tr> <tr> <td>2</td> <td>The executed portion was aggressive when the trade occurred</td> </tr> <tr> <td></td> <td>Pegged</td> </tr> <tr> <td>3</td> <td>The executed portion was passive and pegged when the trade occurred</td> </tr> </tbody> </table> | Value | Meaning | | Visible | 0 | The executed portion was passive and visible when the trade occurred | | Hidden | 1 | The executed portion was passive and hidden when the trade occurred | | Not specified (i.e. Ignore this field) | 2 | The executed portion was aggressive when the trade occurred | | Pegged | 3 | The executed portion was passive and pegged when the trade occurred |
| Value | Meaning | | | | | | | | | | | | | | | | | | | | | |
| | Visible | | | | | | | | | | | | | | | | | | | | | |
| 0 | The executed portion was passive and visible when the trade occurred | | | | | | | | | | | | | | | | | | | | | |
| | Hidden | | | | | | | | | | | | | | | | | | | | | |
| 1 | The executed portion was passive and hidden when the trade occurred | | | | | | | | | | | | | | | | | | | | | |
| | Not specified (i.e. Ignore this field) | | | | | | | | | | | | | | | | | | | | | |
| 2 | The executed portion was aggressive when the trade occurred | | | | | | | | | | | | | | | | | | | | | |
| | Pegged | | | | | | | | | | | | | | | | | | | | | |
| 3 | The executed portion was passive and pegged when the trade occurred | | | | | | | | | | | | | | | | | | | | | |

6.6.2 Execution Report - Protocol Version: 3

| <u>Field</u> | <u>Offset</u> | <u>Length</u> | <u>Type</u> | <u>Description</u> | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|-------------------------|---------------|--------------|--|--------------|----------------|----------|------------|----------|------------------|----------|-------------------------|----------|-----------------|----------|------------------|----------|----------------|----------|--------------|----------|----------------------|----------|---------------------|----------|-----------------|----------|------------------|
| Header | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Partition ID</u> | <u>4</u> | <u>1</u> | <u>UInt8</u> | <u>Identity of the matching partition.</u> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Sequence Number</u> | <u>5</u> | <u>4</u> | <u>Int32</u> | <u>Message sequence number of the matching partition.</u> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Execution ID</u> | <u>9</u> | <u>21</u> | <u>Alpha</u> | <p><u>Unique Identifier of the Execution Report. Unique across all partitions and all trading days.</u></p> <p><u>On Order confirmation, amendment or cancellation a unique Execution ID will be generated.</u></p> <p><u>Identifier of the Exec ID of the execution report for the trade being cancelled or corrected if Execution Type is Trade Cancel (H) or Trade Correct (G).</u></p> <p><u>This will also match to the Side ExecID field in a Trade Capture Report generated for a trade.</u></p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Client Order ID</u> | <u>30</u> | <u>20</u> | <u>Alpha</u> | <p><u>Client specified identifier of the order, order cancel request or order cancel/replace request.</u></p> <p><u>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.</u></p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Order ID</u> | <u>50</u> | <u>12</u> | <u>Alpha</u> | <u>Server specified identifier of the order.</u> | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>Execution Type</u> | <u>62</u> | <u>1</u> | <u>Alpha</u> | <p><u>Execution Type of the order.</u></p> <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>0</u></td> <td><u>New</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Cancelled</u></td> </tr> <tr> <td><u>5</u></td> <td><u>Amended/Modified</u></td> </tr> <tr> <td><u>8</u></td> <td><u>Rejected</u></td> </tr> <tr> <td><u>9</u></td> <td><u>Suspended</u></td> </tr> <tr> <td><u>C</u></td> <td><u>Expired</u></td> </tr> <tr> <td><u>F</u></td> <td><u>Trade</u></td> </tr> <tr> <td><u>G</u></td> <td><u>Trade Correct</u></td> </tr> <tr> <td><u>H</u></td> <td><u>Trade Cancel</u></td> </tr> <tr> <td><u>D</u></td> <td><u>Restated</u></td> </tr> <tr> <td><u>L</u></td> <td><u>Triggered</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>0</u> | <u>New</u> | <u>4</u> | <u>Cancelled</u> | <u>5</u> | <u>Amended/Modified</u> | <u>8</u> | <u>Rejected</u> | <u>9</u> | <u>Suspended</u> | <u>C</u> | <u>Expired</u> | <u>F</u> | <u>Trade</u> | <u>G</u> | <u>Trade Correct</u> | <u>H</u> | <u>Trade Cancel</u> | <u>D</u> | <u>Restated</u> | <u>L</u> | <u>Triggered</u> |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>0</u> | <u>New</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Cancelled</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>5</u> | <u>Amended/Modified</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>8</u> | <u>Rejected</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>9</u> | <u>Suspended</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>C</u> | <u>Expired</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>F</u> | <u>Trade</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>G</u> | <u>Trade Correct</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>H</u> | <u>Trade Cancel</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>D</u> | <u>Restated</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <u>L</u> | <u>Triggered</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| <u>Order Status</u> | <u>63</u> | <u>1</u> | <u>UInt8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>0</u></td> <td><u>New</u></td> </tr> <tr> <td><u>1</u></td> <td><u>Partially Filled</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Filled</u></td> </tr> <tr> <td><u>4</u></td> <td><u>Cancelled</u></td> </tr> <tr> <td><u>6</u></td> <td><u>Expired</u></td> </tr> <tr> <td><u>8</u></td> <td><u>Rejected</u></td> </tr> <tr> <td><u>9</u></td> <td><u>Suspended</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>0</u> | <u>New</u> | <u>1</u> | <u>Partially Filled</u> | <u>2</u> | <u>Filled</u> | <u>4</u> | <u>Cancelled</u> | <u>6</u> | <u>Expired</u> | <u>8</u> | <u>Rejected</u> | <u>9</u> | <u>Suspended</u> |
|--------------------------|--|-----------|--------------|--|--------------|----------------|----------|--------------|----------|------------------------------|----------|--|----------|------------------|----------|----------------|----------|-----------------|----------|------------------|
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | |
| <u>0</u> | <u>New</u> | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Partially Filled</u> | | | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Filled</u> | | | | | | | | | | | | | | | | | | | |
| <u>4</u> | <u>Cancelled</u> | | | | | | | | | | | | | | | | | | | |
| <u>6</u> | <u>Expired</u> | | | | | | | | | | | | | | | | | | | |
| <u>8</u> | <u>Rejected</u> | | | | | | | | | | | | | | | | | | | |
| <u>9</u> | <u>Suspended</u> | | | | | | | | | | | | | | | | | | | |
| <u>Reject Code</u> | <u>64</u> | <u>4</u> | <u>Int32</u> | Code specifying the reason for the reject. Please refer Section 8.2 for a list of reject codes as well as the full list of reject Codes Specification. Field will be ignored if Execution Type is not Rejected (8). | | | | | | | | | | | | | | | | |
| <u>Executed Price</u> | <u>68</u> | <u>8</u> | <u>Price</u> | Executed price of the trade in ZAC. This field will be ignored if Execution Type is not Trade (F) or Trade Correct (G). | | | | | | | | | | | | | | | | |
| <u>Executed Quantity</u> | <u>76</u> | <u>4</u> | <u>Int32</u> | Executed quantity. This field will be ignored if Execution Type is not Trade (F) or Trade Correct (G). This will always be 0 on a Same Day On Book trade cancellation. | | | | | | | | | | | | | | | | |
| <u>Leaves Quantity</u> | <u>80</u> | <u>4</u> | <u>Int32</u> | Quantity available for further execution. It is the remaining quantity of the order. | | | | | | | | | | | | | | | | |
| <u>Working Indicator</u> | <u>84</u> | <u>1</u> | <u>UInt8</u> | Whether the order is currently being worked on. Will only be applicable when Order Status is New (0). <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>0</u></td> <td><u>Unset</u></td> </tr> <tr> <td><u>1</u></td> <td><u>Order is being worked</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Order is not currently in a working state</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>0</u> | <u>Unset</u> | <u>1</u> | <u>Order is being worked</u> | <u>2</u> | <u>Order is not currently in a working state</u> | | | | | | | | |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | |
| <u>0</u> | <u>Unset</u> | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Order is being worked</u> | | | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Order is not currently in a working state</u> | | | | | | | | | | | | | | | | | | | |
| <u>Security ID</u> | <u>85</u> | <u>4</u> | <u>Int32</u> | Identifier of the instrument the Execution Report is sent for. | | | | | | | | | | | | | | | | |
| <u>Side</u> | <u>89</u> | <u>1</u> | <u>UInt8</u> | <table border="1"> <thead> <tr> <th><u>Value</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td><u>1</u></td> <td><u>Buy</u></td> </tr> <tr> <td><u>2</u></td> <td><u>Sell</u></td> </tr> </tbody> </table> | <u>Value</u> | <u>Meaning</u> | <u>1</u> | <u>Buy</u> | <u>2</u> | <u>Sell</u> | | | | | | | | | | |
| <u>Value</u> | <u>Meaning</u> | | | | | | | | | | | | | | | | | | | |
| <u>1</u> | <u>Buy</u> | | | | | | | | | | | | | | | | | | | |
| <u>2</u> | <u>Sell</u> | | | | | | | | | | | | | | | | | | | |
| <u>TraderMnemonic</u> | <u>90</u> | <u>17</u> | <u>Alpha</u> | This will be the concatenated identifier of the Trader Group and the JSE Trader ID. (Mandatory). The concatenation will be done by using an underscore between the JSE Trader Group and Trader ID. | | | | | | | | | | | | | | | | |
| <u>Account</u> | <u>107</u> | <u>10</u> | <u>Alpha</u> | Client Account information. This is the Client Account of the firm who submitted the order. | | | | | | | | | | | | | | | | |

| IsMarketOpsRequest | 117 | 1 | UInt8 | <p>This field indicates whether a New Order, Cancel Request or Order Cancel or Replace Request was submitted by Market Operations.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No</td> </tr> <tr> <td>1</td> <td>Yes</td> </tr> </tbody> </table> | Value | Meaning | 0 | No | 1 | Yes |
|---------------------------------------|---|--------------------|------------------------|---|-----------------------|-------------------------|-------------------|--------------------------------|--------------------|---|
| Value | Meaning | | | | | | | | | |
| 0 | No | | | | | | | | | |
| 1 | Yes | | | | | | | | | |
| Transact Time | 118 | 8 | UInt64 | Time the Execution Report was generated. | | | | | | |
| Order Book | 126 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular | | |
| Value | Meaning | | | | | | | | | |
| 1 | Regular | | | | | | | | | |
| Execution Instruction | 127 | 1 | Int8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>None</td> </tr> <tr> <td>2</td> <td>Include in EOD Volume Auction Uncross</td> </tr> </tbody> </table> <p>This will have a default value of 0 if Order Type is not Limit or 2 if the Time in Force type is not GTT (8).</p> | Value | Meaning | 0 | None | 2 | Include in EOD Volume Auction Uncross |
| Value | Meaning | | | | | | | | | |
| 0 | None | | | | | | | | | |
| 2 | Include in EOD Volume Auction Uncross | | | | | | | | | |
| Cross ID | 128 | 20 | Alpha | <p>The unique ID of the Cross Order. Only populated for execution report messages generated for Internal Cross Orders.</p> <p>The value submitted with the New Order Cross Message will be populated.</p> | | | | | | |
| Cross Type | 148 | 1 | UInt8 | <p>The type of the Cross Order. Only populated for execution report messages generated Internal Cross Orders.</p> <p>The value submitted with the New Order Cross Message will be populated.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>Internal Cross</td> </tr> <tr> <td>50</td> <td>Internal Cross (Price Adjustable)</td> </tr> </tbody> </table> | Value | Meaning | 5 | Internal Cross | 50 | Internal Cross (Price Adjustable) |
| Value | Meaning | | | | | | | | | |
| 5 | Internal Cross | | | | | | | | | |
| 50 | Internal Cross (Price Adjustable) | | | | | | | | | |
| Display Quantity | 149 | 4 | Int32 | Current visible quantity. | | | | | | |
| Public Order ID | 153 | 12 | Alpha | Server specified public order identifier of the order. | | | | | | |

| Indicator Flags | 165 | 1 | Bit field | <p>Denotes whether a given side of the trade is the aggressor side or the passive side. Applicable only when the Exec Type is Trade or Trade Correct. Should be ignored for other Exec Types. During Auctions, both side ERs of a trade will have '0' stamped.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Name</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0th</td> <td>Aggressor Indicator</td> <td>0 : Order initiator is passive</td> </tr> <tr> <td></td> <td></td> <td>1 : Order initiator is aggressor</td> </tr> </tbody> </table> | Bit | Name | Meaning | 0th | Aggressor Indicator | 0 : Order initiator is passive | | | 1 : Order initiator is aggressor | |
|---|---|--|---------------------------|---|-----------------------|-------------------------|-------------------------|---|-------------------------------------|---|-------------------|---|--|---|
| Bit | Name | Meaning | | | | | | | | | | | | |
| 0th | Aggressor Indicator | 0 : Order initiator is passive | | | | | | | | | | | | |
| | | 1 : Order initiator is aggressor | | | | | | | | | | | | |
| Liquidity Indicator | 166 | 1 | UInt8 | <p>Indicator to identify whether this fill was a result of a liquidity provider providing or liquidity taker taking the liquidity. Required if Order Status is Partially Filled (1) or Filled (2) and Execution Type is Trade (F).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Unset</td> </tr> <tr> <td>1</td> <td>Added Liquidity</td> </tr> <tr> <td>2</td> <td>Removed Liquidity</td> </tr> <tr> <td>4</td> <td>Auction</td> </tr> </tbody> </table> | Value | Meaning | 0 | Unset | 1 | Added Liquidity | 2 | Removed Liquidity | 4 | Auction |
| Value | Meaning | | | | | | | | | | | | | |
| 0 | Unset | | | | | | | | | | | | | |
| 1 | Added Liquidity | | | | | | | | | | | | | |
| 2 | Removed Liquidity | | | | | | | | | | | | | |
| 4 | Auction | | | | | | | | | | | | | |
| TypeofTrade | 167 | 1 | UInt8 | <p>Indicates whether the executed portion is visible or hidden. Valid only if ExecType (150) = F. Ignore value in all other cases.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Visible The executed portion was passive and visible when the trade occurred</td> </tr> <tr> <td>1</td> <td>Hidden The executed portion was passive and hidden when the trade occurred</td> </tr> <tr> <td>2</td> <td>Not specified (i.e. Ignore this field) The executed portion was aggressive when the trade occurred</td> </tr> <tr> <td>3</td> <td>Pegged The executed portion was passive and pegged when the trade occurred</td> </tr> </tbody> </table> | Value | Meaning | 0 | Visible The executed portion was passive and visible when the trade occurred | 1 | Hidden The executed portion was passive and hidden when the trade occurred | 2 | Not specified (i.e. Ignore this field) The executed portion was aggressive when the trade occurred | 3 | Pegged The executed portion was passive and pegged when the trade occurred |
| Value | Meaning | | | | | | | | | | | | | |
| 0 | Visible The executed portion was passive and visible when the trade occurred | | | | | | | | | | | | | |
| 1 | Hidden The executed portion was passive and hidden when the trade occurred | | | | | | | | | | | | | |
| 2 | Not specified (i.e. Ignore this field) The executed portion was aggressive when the trade occurred | | | | | | | | | | | | | |
| 3 | Pegged The executed portion was passive and pegged when the trade occurred | | | | | | | | | | | | | |
| Following field(s) have been introduced by Protocol v3 | | | | | | | | | | | | | | |
| Self Trade Prevention Key | 168 | 12 | Alpha | Client specified identifier of order relevant to self-match. | | | | | | | | | | |

6.4.146.6.3 Order Cancel Reject

| Field | Offset | Length | Type | Description | | | | |
|-----------------|---------|--------|--------|---|-------|---------|---|---------|
| Header | | | | | | | | |
| Partition ID | 4 | 1 | UInt8 | Identity of the matching partition. | | | | |
| Sequence Number | 5 | 4 | Int32 | Message sequence number of the matching partition. | | | | |
| Client Order ID | 9 | 20 | Alpha | Client specified identifier of the rejected cancel or cancel/replace request. | | | | |
| Order ID | 29 | 12 | Alpha | Server specified identifier of the order for which the cancel or cancel/replace was submitted. | | | | |
| Transact Time | 41 | 8 | UInt64 | Time the Order Cancel Reject occurred. | | | | |
| Reject Code | 49 | 4 | Int32 | Code specifying the reason for the reject. Please refer to Section 8.2 for a list of reject codes as well as the full list of reject codes in Volume 10 – JSE Reject Codes Specification. | | | | |
| Order Book | 53 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular |
| Value | Meaning | | | | | | | |
| 1 | Regular | | | | | | | |

6.4.156.6.4 Order Mass Cancel Report

| Field | Offset | Length | Type | Description | | | | | | |
|-----------------|----------|--------|--------|---|-------|---------|---|----------|---|----------|
| Header | | | | | | | | | | |
| Partition ID | 4 | 1 | UInt8 | Identity of the matching partition. | | | | | | |
| Sequence Number | 5 | 4 | Int32 | Message sequence number of the matching partition. | | | | | | |
| Client Order ID | 9 | 20 | Alpha | Client specified identifier of the mass cancel request. | | | | | | |
| Status | 29 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Rejected</td> </tr> <tr> <td>7</td> <td>Accepted</td> </tr> </tbody> </table> | Value | Meaning | 0 | Rejected | 7 | Accepted |
| Value | Meaning | | | | | | | | | |
| 0 | Rejected | | | | | | | | | |
| 7 | Accepted | | | | | | | | | |
| Reject Code | 30 | 4 | Int32 | Code specifying the reason for the reject. Please refer to the Section 8.2 for a list of reject codes as well as the full list of reject codes in Volume 10 – Reject Codes and Reasons Specification. This field will be ignored if Status is not Rejected (0). | | | | | | |
| Transact Time | 34 | 8 | UInt64 | Time the order mass cancel report was generated. | | | | | | |
| Order Book | 42 | 1 | UInt8 | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Regular</td> </tr> </tbody> </table> | Value | Meaning | 1 | Regular | | |
| Value | Meaning | | | | | | | | | |
| 1 | Regular | | | | | | | | | |

6.56.7 Other Application Messages of the Basic Gateway: Server-Initiated

6.5.16.7.1 News

| Field | Offset | Length | Type | Description | | | | | | | | |
|------------------------|---------------|--------|-------|--|-------|---------|---|---------|---|---------------|---|--------------|
| Header | | | | | | | | | | | | |
| Partition ID | 4 | 1 | UInt8 | Identity of the partition. | | | | | | | | |
| Sequence Number | 5 | 4 | Int32 | Message sequence number of the partition. | | | | | | | | |
| OrigTime | 9 | 24 | Alpha | Time the announcement was published which will be specified in UTC and in the YYYYMMDD-HH:MM:SS.uuuuuu format. | | | | | | | | |
| Urgency | 33 | 1 | Byte | <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Regular</td> </tr> <tr> <td>1</td> <td>High Priority</td> </tr> <tr> <td>2</td> <td>Low Priority</td> </tr> </tbody> </table> | Value | Meaning | 0 | Regular | 1 | High Priority | 2 | Low Priority |
| Value | Meaning | | | | | | | | | | | |
| 0 | Regular | | | | | | | | | | | |
| 1 | High Priority | | | | | | | | | | | |
| 2 | Low Priority | | | | | | | | | | | |
| Headline | 34 | 100 | Alpha | Headline or subject of market operations announcement. | | | | | | | | |
| Text | 134 | 750 | Alpha | Text of the market operations announcement. | | | | | | | | |
| Instruments | 884 | 100 | Alpha | Pipe separated list of symbols of the instruments market operations announcements relate to. | | | | | | | | |
| Underlying Instruments | 984 | 100 | Alpha | Pipe separated list of symbols of the underlying instruments. | | | | | | | | |
| Firm List | 1084 | 54 | Alpha | Pipe separated list of firms that the announcement should be sent to. | | | | | | | | |
| User List | 1138 | 54 | Alpha | Pipe separated list of users that the announcement should be sent to. | | | | | | | | |

6.5.26.7.2 Business Reject

| Field | Offset | Length | Type | Description |
|-----------------|--------|--------|--------|--|
| Header | | | | |
| Partition ID | 4 | 1 | UInt8 | Identity of the matching partition. |
| Sequence Number | 5 | 4 | Int32 | Message sequence number of the matching partition. |
| Reject Code | 9 | 4 | Int32 | Code specifying the reason for the reject. Please refer to Section 7.3 for a list of reject codes. |
| Client Order ID | 13 | 20 | Alpha | Client specified identifier of the rejected message if it is available. |
| OrderID | 33 | 12 | Alpha | Server specified identifier of the order for which the cancel or cancel/replace was submitted. |
| Transact Time | 45 | 8 | UInt64 | Time the order mass cancel report was generated. |

7 REJECT CODES

Some of the key reject codes for the Login Response, Reject and Business Reject messages are provided in this section.

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

7.1 Login Response

| Reject Code | Description |
|-------------|--|
| 0 | Login successful |
| 1 | Invalid CompID or password |
| 3 | New session password does not comply with policy |
| 100 | Not logged into real-time channel |
| 9903 | Concurrent login limit reached |

7.2 Reject

| Reject Code | Description |
|-------------|-------------------------------|
| 105 | Login request being processed |
| 107 | Not logged in |
| 9900 | Required field missing |
| 9901 | Invalid value in field |
| 9990 | Maximum message rate exceeded |

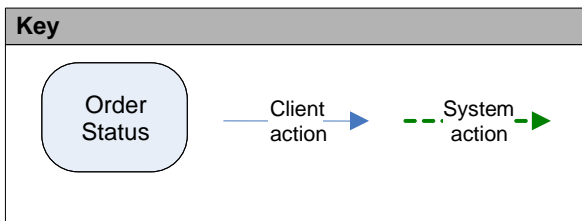
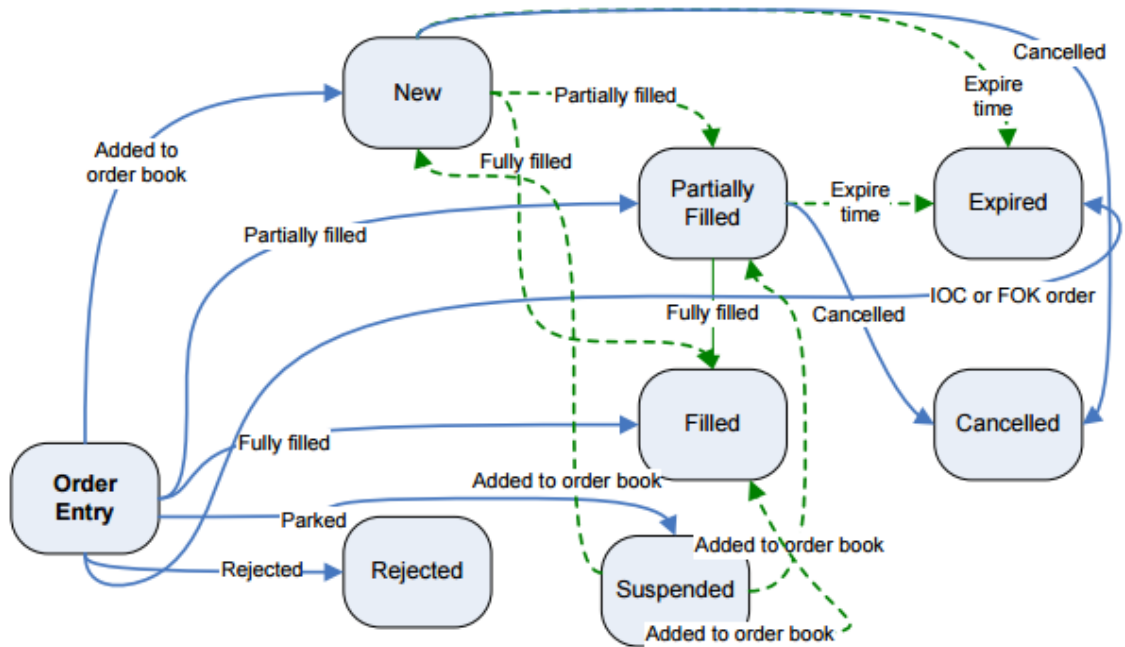
7.3 Business Reject

| Reject Code | Description |
|-------------|------------------------------|
| 9000 | Unknown instrument |
| 9998 | Matching partition suspended |
| 9999 | System suspended |

8 PROCESS FLOWS

8.1 Order Status Changes

8.1.1 General



8.1.2 Market Operations Actions

