Trimmed Protocol for Non-NEAT Front End (NNF)

Capital Market Trading System

Version 6.4

Nov 2025

National Stock Exchange of India Ltd
Exchange Plaza, Plot No. C/1, G Block,
Bandra-Kurla Complex, Bandra (E)
Mumbai - 400 051.

Notice

Capital Market Trading System Revision History		
Version	Page No	Description
6.1		Addition of new error codes: 16730 16731 16732 16733 16738
		Corrected structure details for 6541 transcode
		Updated description for SSEC indicator
6.2		Addition of new error codes: 16735 16736
		Corrected reference structure details for 2212 transcode
		Order Flags description updated
		Order Flags table updated
6.3		Heartbeat echo back added
		Addition of new error code 17107 - Heart beat rate exceeded by the member
6.4	55,209,212,233, 236,238	Transaction codes for Immediate Ack Messages
	153-155, 157-163,248-253	Details of New Encryption Mechanism
	194-196	New Chapter 13 added for Immediate Order Acknowledgment Message

Preface

Purpose

This document describes the protocol to be used for Non-NEAT Front end (NNF) to communicate with the Capital Market Trading System and thus serves as a development guide for the NNF users.

Target Audience

This document is written for system designers and programmers of user organizations and third party software developers who are responsible for the development of software to interact with NSE's Capital Market Trading System.

Organization of This Document

This document is organized as follows:

Chapters	Description
Chapter 1	Provides a brief introduction to Non-NEAT Front end (NNF). It also details the NNF Terminal requirements.
Chapter 2	Describes the general guidelines for the designers and programmers who develop NNF. It details the data types used and also covers the Message Header that is prefaced with all the structures.
Chapter 3	Describes how a trader logs on to the trading system. It also discusses the download of the updated information on the securities, participants and the status of the markets, and describes the log on request and the system responses.
Chapter 4	Describes entering fresh orders, modifying an existing order, and canceling outstanding orders.
Chapter 5	Covers the messages that are received on the interactive connection. These messages are received by users not in response to any request.
Chapter 6	Describes the end of the trading day activities. It covers the transmission of Security Bhav Copy and Index Bhav Copy.
Chapter 7	Describes the various Broadcast messages and the Compression and Decompression algorithm of Broadcast data.
Chapter 8	Describes the Auction Inquiry and MBO Inquiry and the system responses.
Chapter 9	Encryption Decryption of Interactive Messages.

Chapters	Description
Chapter 10	Describes how member systems can directly connect to NSE for trading, while using existing formats of business messages from NNF API documents.
Chapter 11	Describes how exception at trading end should be handled.
Chapter 12	Describes the functionalities made available to CM / BM users
Chapter 13	Immediate order acknowledgement Message
Appendix	Lists the error, transaction and reason codes and also covers the various market statuses, market types and book types. Also covers security.txt, participant.txt and contract.txt structures.

Abbreviations and Acronyms Used

The abbreviations and acronyms used in this document are:

AGM	Annual General Meeting
AON	All Or None
ATO	At The Opening
AU	Auction
BCID	Broadcast Circuit ID
BM	Branch Manager
CM	Corporate Manager
DL	Dealer
DQ	Disclosed Quantity
EGM	Extraordinary General Meeting
GTC	Good Till Cancellation
GTD	Good Till Date
IOC	Immediate Or Cancel
LTP	Last Traded Price
MBO	Market By Order
MBP	Market By Price
MF	Minimum Fill
NEAT	National Exchange for Automated Trading
NNF	Non-Neat Front End
NSE	National Stock Exchange
OL	Odd Lot
RL	Regular Lot

SL	Stop Loss
ST	Special Terms
TM	Trading Member
TP	Trigger Price
TWS	Trader Workstation
VCID	Virtual Circuit ID
VV.RR.SS	Version. Release. Sub-release
WHS	Warehouse
BOVL	Branch Order Value Limit
UOVL	User Order Value Limit
PAN	Permanent Account Number
SPOS	Special Pre-Open Session

CONTENTS

Chapter 1 Introduction

The National Stock Exchange of India Ltd (NSEIL) provides a fully automated screen based trading system, enabling trading members spread across the length and breadth of India to trade directly from their offices through an extensive telecommunication network. The system is known as 'National Exchange for Automated Trading' (NEAT) system. It adopts the principles of an order driven market, based on price-time priority. The trading members can use NEAT Front end or Non-NEAT Front end (NNF) to establish a network connection with the host system of National Stock Exchange (NSE) for trading. NNF is a front end which is developed and maintained by vendors other than NSE. NSE provides the NNF users with the general guideline document of the front end whereas they are supported by their respective vendors and NSE is not responsible for the performance of the NNF.

Chapter 2 General Guidelines

Introduction

This chapter provides general guidelines for the designers and programmers who develop NNF. It also provides information on data types and their size which can help in understanding various structures.

Message Structure Details

The message structure consists of two parts namely message header and message data. The message header consists of the fields of the header which is prefaced with all the structures.

The message data consists of the actual data that is sent across to the trading system (i.e. host) or received from the trading system (i.e. host).
Transaction code, an important field of the message header, is a unique numeric identifier which is sent to or received from the trading system. This is used to identify the transaction between the TWS and the host end.

Guidelines for Designers

1. The order of the log-on messages should strictly be maintained as given in the following section (Chapter 3) of the document. Otherwise, the user cannot log on to the trading system.
2. All time fields are number of seconds from midnight January 1, 1980.
3. No host-end inquiries are permitted for NNF users.
4. All price fields must be multiplied by 100 before sending to the host end and divided by 100 while receiving from the host end as the host system processes prices in paisa.
5. All branch/user/order value limit fields must be multiplied by (100000 * 100) before sending to host end and divided by (100000 * 100) while receiving from the host end as the host system processes limits in paisa.

Guidelines for Programmers

1. If your system uses little-endian order, the data types such as UINT, SHORT, LONG and DOUBLE contained in a packet, which occupy more than one byte should be twiddled (byte reversed). Twiddling involves reversing a given number of bytes such that the byte

in 'n' position comes to the first position; the byte in (n-1) position comes to the second position and so on. For example, if the value to be sent is 1A2B (hexadecimal), reverse the bytes to 2B1A. The same applies while receiving messages. So if the value received is 02BC, the actual value is BC02. So twiddle such data types before sending and after receiving to ensure that correct data is sent and received.

Note:

Twiddling is required because of the variety in endian order—big and little. A big-endian representation has a multi-byte integer written with its most significant byte on the left. A little-endian representation, on the other hand, places the most significant byte on the right. The trading system host end uses big-endian order.

2. All alphabetical data must be converted to upper case except password before sending to the host. A combination of alphabet, numbers and special characters are allowed in the password. More details on password are explained in later chapters in this document. No NULL terminated strings should be sent to the host end. Instead, fill it with blanks before sending. The strings received from the host end are padded with blanks and are not NULL terminated.

3. All the structures should be defined in the following manner:
   - Items of type char or unsigned char, or arrays containing items of these types, are byte aligned.
   - Structures are word aligned.
   - All other types of structure members are word aligned.
   - All structures are pragma pack 2. Structures of odd size should be padded to an even number of bytes.
4. All numeric data must be set to zero (0) before sending to the host, unless a value is assigned to it.
5. All reserved fields mentioned, should be mapped to CHAR buffer and initialized to NULL.
6. Inside the broadcast packet, the first byte indicates the market type. Ignore the next 7 bytes. If the first byte is 2 it indicates Futures & Options market. The message header starts from the 9th byte. The remaining portion of the buffer has to be mapped to the broadcast structures mentioned in the document.

Note:

- The values of all the constants and transaction codes given in the document are listed in Appendix.
- The suffix IN in the transaction codes implies that the request is sent from the TWS to the host end whereas OUT implies that the message is sent from the host end to TWS

Data Types Used

Data Type	Size of Bytes	Signed / Unsigned
CHAR	1	Signed
UINT	2	Unsigned
SHORT	2	Signed
LONG	4	Signed
UNSIGNED LONG	4	Unsigned
LONG LONG	8	Signed
DOUBLE	8	Signed and Floating Point
BIT	1 bit	NA

Message Header

Each structure is prefaced with a MESSAGE_HEADER which is an interactive header. Some data in the header are fixed whereas some data are variable and set differently for each transaction code. The structure of the Message Header is as follows:

Table 1 MESSAGE HEADER

Structure Name	MESSAGE_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
LogTime	LONG	4	2
AlphaChar [2]	CHAR	2	6
TraderId	LONG	4	8
ErrorCode	SHORT	2	12
TimeStamp	LONG LONG	8	14

Structure Name	MESSAGE_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
TimeStamp1 [8]	CHAR	8	22
TimeStamp2 [8]	CHAR	8	30
MessageLength	SHORT	2	38

The fields of Message Header are described below.

Field Name	Brief Description
TransactionCode	Transaction message number. This describes the type of message received or sent.
LogTime	This field should be set to zero while sending messages.
AlphaChar [2]	This field should be set to the first two characters of Symbol if the structure contains Symbol and Series; otherwise it should be set to blank.
TraderId	This field should contain the user ID.
ErrorCode	This field should be set to zero while sending messages to the host. In the messages coming from the host, this field describes the type of error. Refer to List of Error Codes in Appendix.
TimeStamp	This field should be set to numeric zero while sending to the host. This is used in host end. For transcodes listed in appendix, time in this field will be populated in nanoseconds (from 01-Jan-1980 00:00:00). This time is stamped at the matching engine in the trading system.
TimeStamp1	This field should be set to numeric zero while sending. This is the time the message arrives at the trading system host. In TimeStamp1, time is sent in jiffies from host end. This 8 byte data needs to be typecasted as first four bytes into double variable and typecast the other four byte into another double variable. These values need to be used while requesting message area download in the same order.
TimeStamp2	This field should be set to numeric zero while sending to the host. For messages coming from the host, this field contains the machine number from which the packet is coming.

Field Name	Brief Description
	In TimeStamp2, machine number is sent from host end.
MessageLength	This field should be set to the length of the entire message, including the length of message header while sending to host.

Inner Message Header

Each structure in the Data of Update Local Database Data/Message Download Data responses is prefaced with an INNER_MESSAGE_HEADER. The structure of the Inner Message Header is as follows:

Table 2 INNER MESSAGE HEADER

Structure Name	INNER_MESSAGE_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
TraderId	LONG	4	0
LogTime	LONG	4	4
AlphaChar [2]	CHAR	2	8
TransactionCode	SHORT	2	10
ErrorCode	SHORT	2	12
TimeStamp	LONG LONG	8	14
TimeStamp1 [8]	CHAR	8	22
TimeStamp2 [8]	CHAR	8	30
MessageLength	SHORT	2	38

Note: The field descriptions are the same as MESSAGE_HEADER.

Broadcast Process Header

The broadcast messages like market open, market close, market in pre-open are prefaced with BCAST_HEADER. Some fields in the header are fixed. The remaining fields are variable and set differently for each transaction code. The structure of the BCAST_HEADER is as follows:

Table 3 BROADCAST_HEADER

Structure Name	BCAST_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	4	0
LogTime	LONG	4	4
AlphaChar	CHAR	2	8
TransCode	SHORT	2	10
ErrorCode	SHORT	2	12
BCSeqNo	LONG	4	14
Reserved	CHAR	4	18
TimeStamp2	CHAR	8	22
Filler2	CHAR	8	30
MessageLength	SHORT	2	38

Field Name	Brief Description
LogTime	This field should be set to zero while sending to host end. For messages sent from host end this field contains the time when the message was generated by the trading system host.
AlphaChar	This field is set to the first two characters of Symbol if the structure contains Symbol and Series; otherwise it is set to blank.
TransactionCode	This field contains the transaction message number. This describes the type of message received or sent.
ErrorCode	This field contains the error number which describes the type of error. Refer to List of Error Codes in Appendix.
BCSeqNo	This field contains BCAST Sequence number of the NSE host end system. The sequence number is not the unique broadcast sequence number as it has eleven set of sequence numbers for normal broadcast and six set of sequence numbers for Fast broadcast each instance of the sequence number is generated by the Individual processes in the host end. It is not an unique sequence number.
TimeStamp2	This field contains the time when message is sent from the host.
Filler2	This field contains the machine number.

Field Name	Brief Description
MessageLength	This field is set to the length of the entire message, including the length of the message header.

Note: BCAST_HEADER is prefaced with a system header which is of eight bytes

SEC_INFO

Table 4 SEC_INFO

Structure Name	SEC_INFO		
Packet Length	12 bytes		
Field Name	Data Type	Size in Byte	Offset
Symbol	CHAR	10	0
Series	CHAR	2	10

Field Name	Brief Description
Symbol	This field should contain the symbol of a security.
Series	This field should contain the series of a security.

Error Message

When the Error Code in the Message Header is having nonzero value, ERROR RESPONSE is sent. The Error Message will describe the error received. The structure is as follows:

Table 5 ERROR_RESPONSE

Structure Name	ERROR RESPONSE		
Packet Length	180 bytes		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
SEC_INFO (Refer Table 4)	STRUCT	12	40
Error Message	CHAR	128	52

Field Name	Brief Description
Symbol	This field should contain the symbol of a security.
Series	This field should contain the series of a security.

Field Name	Brief Description
ErrorMessage	Stores the error message. Refer to List of Error Codes in Appendix.

Invalid Message Length Response Transcode

If a user sends a request with improper message length, then the host will send INVALID_MSG_LENGTH_RESPONSE transcode (2322) in response. This check is not specific to the type of user and may occur for both NEAT and NNF Users.
Message length may vary from one request to the other. For example, for an Order request the Host end expects a request with the message length of 214 bytes. If the order request has any message length other than 214 bytes, it will send the above mentioned transcode with the error code – ERR_INVALID_MSG_LENGTH (defined in the error codes table previously).
Host sends the same incoming packet structure in response but with transcode populated as INVALID_MSG_LENGTH_RESPONSE (2322) and error code populated as ERR_INVALID_MSG_LENGTH.
Kindly refer to individual transocde for their corresponding message length

Communication Network Connections for NNF Users

There are two types of virtual circuit connections used to communicate with the host end. One is the Interactive Virtual Circuit ID (VCID) and the other is the Broadcast Circuit ID (BCID).
Interactive VCID follows a bidirectional path between the NNF and NEAT to host end. All the interactive / request messages and its respective response follow through this channel. Even the unsolicited message such as trade message flows from exchange (host end) to the trader terminal through this channel.
Standard implementation of TCP/IP protocol exists on the exchange's infrastructure as a result of which default features like IP fragmentation, no QoS etc. continue to be enabled and available for use by members. Default IP fragmentation a valid feature in the TCP/IP protocol works at message level and usage of same by one member connection will not block or impact the messages of other member connections.
BCID follows a unidirectional path which is from the host end to the NFF / NEAT. All the broadcast data are transmitted through this broadcast circuit from the host end for all the traders. Since this is a one way connection, the data flow is always from the exchange (host end) to the trader terminal.

Member Guide to the Gateway Router Functionality

Currently Exchange publishes a list of gateway servers (NET) in the respective segments to which members can connect. Members have the choice of connecting to any of the gateway servers.

However, the members have represented that they are required to try to login on multiple gateway server sequentially before they are able to successfully login on the Exchange for trading activity. Thus, valuable time is lost by the member for trying to access the Exchange. The same is more severe during re-login / disconnections faced by the members.

In order to address these queries the Gateway Router Functionality has been proposed to be implemented.

1. It is now proposed that members will first connect to a gateway router server in the respective segment details of which will be published by the Exchange.
2. The gateway router server will decide which gateway server is available for the member and will accordingly provide the details of the allocated gateway server to the member through the response message.
3. After getting the response message the member will need to connect to the allocated gateway server.

Thus, the process of allocating gateway servers becomes Exchange determined and highly simplified for the member.

The gateway router will decide the gateway server for the member for each trading day in the following manner:

1. The gateway router will maintain the used capacity of each gateway server. The gateway router will allocate least used gateway server (according to capacity). The capacity is based on the no. of messages allotted for each Box Id.
2. If all gateway servers have similar used capacity then a gateway server will be randomly allocated by the gateway router server.
3. Once a member has been provided session key with gateway server details by gateway router server, the member is expected to connect and login to the allocated gateway server at any time during rest of the trading day.
4. If the member gets logged off from the allocated gateway server, then the member has to request the gateway router server for getting new session key and gateway server details.

5. A member will be directed to the same gateway server by the gateway router server, once it has been allocated for the trading day.
6. Though the user will get directed to the same gateway, the user must ask the gateway router for getting the gateway details and session key as the old session key will be unique for that particular session and is cleaned up from the gateway once the user gets logged off.
7. Also, if the gateway has a failure during the day, the user will be allocated a new gateway server. This will be done transparently for the user by the gateway router server.

At the end of each trading day the gateway router server will clean up the used capacity, and will have the same capacity (full capacity) available for all gateway servers for the next day.

Chapter 3 Logon Process

Introduction

This section describes how a trader logs on to the trading system. It covers the log-on request and the system responses. This section also describes the download of the updated information on the securities, participants, and the status of the markets. It covers the structures and field descriptions of System Information Download, Local Database Download and Message Download.

The process by which a trader logs on to the trading system is called Logon Process. The trader, after issuing a sign-on request, waits for the system response. The response could be a successful logon or an error message.

Message Download Changes

- Messages will be sent through various streams (at The Exchange). The stream number will be sent in the TimeStamp2 field of the message header.

- The total number of streams from the Exchange will be specified in the first byte of alpha char field (alpha char is of 2 bytes) of the header section of SYSTEM_INFORMATION_OUT (1601) message. Streams are numbered starting from 1. E.g.: If the value in the alpha char field is 4, total number of streams from the Exchange is 4 and the stream numbers will be 1,2,3,4.

- The mechanism for message download request has changed, **Message downloads will now be served through each individual stream**. Hence, message download request needs to be sent individually for a stream by the user.

- In the message download request (Transcode 7000), first byte of alpha char field of the header section should contain the stream number for which the message download is required. If the stream no. sent in the request is invalid then exchange will drop the request. The Sequence number field must contain the sequence number value for that particular stream.
- The response of the request will be sent individually through the specified stream starting from the next sequence number specified in the request. Message download from each stream will have header, data and trailer section (same as existing format).
    - Header – This is to indicate that message download is going to commence. The first byte of alpha char field of header will contain the stream number.

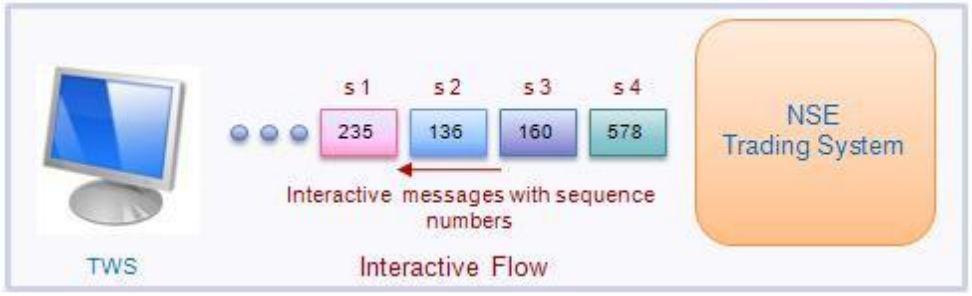
- Data – The data is wrapped in another structure. The outer header indicates that this message is a part of the Message Download Data. The inner header indicates the type of data received. The first byte of alpha char field of outer header will contain the stream number.
- Trailer – This indicates that message download is complete. The first byte of alpha char field of header will contain the stream number.

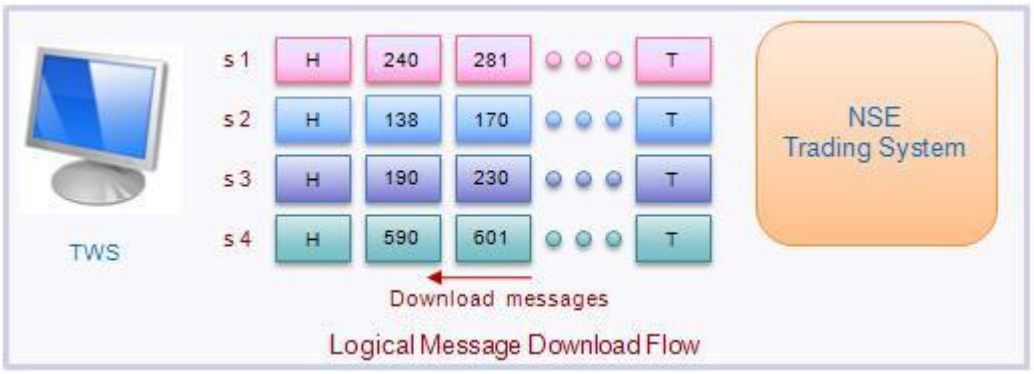
- Message download request can be made for one or more streams. It is recommended that the user requests download for all the streams.

- If the sequence number in the request is 0, then all messages for that stream will be sent. To get incremental download for any particular stream, the message download request must contain the last sequence number received from that stream.

Note:
1. Structure for message download request is not changed.
2. Structure for message download response is not changed.

Illustration: -
In the illustration given below s1, s2, s3, s4 represent separate streams





Order of Events to Be Followed During Logon and Logoff

The following sequence explains the order in which transaction codes are sent and received during log-on process.

Sequence No	Transaction Code	Sent By	Received By
1	SIGN_ON_REQUEST_IN (2300)	TWS	Host End
2	SIGN_ON_REQUEST_OUT (2301)	Host End	TWS
3	SYSTEM_INFORMATION_IN (1600)	TWS	Host End
4	SYSTEM_INFORMATION_OUT (1601)	Host End	TWS
5	UPDATE_LOCALDB_IN (7300)	TWS	Host End
6	UPDATE_LOCALDB_HEADER (7307)	Host End	TWS
7	UPDATE_LOCALDB_DATA (7304)	Host End	TWS
8	UPDATE_LOCALDB_TRAILER (7308)	Host End	TWS
9	DOWNLOAD_REQUEST (7000)	TWS	Host End
10	HEADER_RECORD (7011)	Host End	TWS
11	MESSAGE_RECORD (7021)	Host End	TWS
12	TRAILER_RECORD (7031)	Host End	TWS

The following sequence explains the order in which the transaction codes are sent and received during log-off process.

Sequence No	Transaction Code	Sent By	Received By
1	SIGN_OFF_REQUEST_IN (2320)	TWS	Host End
2	SIGN_OFF_REQUEST_OUT (2321)	Host End	TWS

Logon Request

When the user wants to establish an interactive circuit with the host, he sends this request.

Eligibility for the broker to participate in the CALL AUCTION 2 Market is being used. In SIGN_ON_REQUEST_IN, one bit from the existing reserved bit in BrokerEligibilityPerMarket structure is getting re-used for CALL AUCTION 2 market eligibility.
In the request packet sent from TWS to the Exchange, the value for these bits must be set to numerical zero, similar to other Market eligibility bits, The modified structure as per above change is given below.

Table 7 SIGNON_IN

Structure Name	SIGNON IN		
Packet Length	276 bytes		
Transaction Code	SIGN_ON_REQUEST_IN (2300)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	8	44
Password	CHAR	8	52
Reserved	CHAR	8	60
NewPassword	CHAR	8	68
TraderName	CHAR	26	76
LastPasswordChangeDateTime	LONG	4	102
BrokerId	CHAR	5	106
Reserved	CHAR	1	111
BranchId	SHORT	2	112
VersionNumber	LONG	4	114

Structure Name	SIGNON IN		
Packet Length	276 bytes		
Transaction Code	SIGN_ON_REQUEST_IN (2300)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	56	118
UserType	SHORT	2	174
SequenceNumber	DOUBLE	8	176
WorkstationNumber	CHAR	14	184
BrokerStatus	CHAR	1	198
ShowIndex	CHAR	1	199
BrokerEligibilityPerMarket (Refer Table 7.1 for Small Endian machines and Table 7.2 for Big Endian machines)	STRUCT	2	200
BrokerName	CHAR	26	202
Reserved	CHAR	16	228
Reserved	CHAR	16	244
Reserved	CHAR	16	260

For Small Endian Machines:

Table 7.1 BrokerEligibilityPerMarket

Structure Name	BrokerEligibilityPerMarket		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	2	0
Call Auction2	BIT	1	0
Call Auction1	BIT	1	0
Auction market	BIT	1	0
Spot market	BIT	1	0
Oddlot market	BIT	1	0
Normal market	BIT	1	0
Preopen	BIT	1	1
Reserved	BIT	7	1

For Big Endian Machines:

Table 7.2 BrokerEligibilityPerMarket

Structure Name	BrokerEligibilityPerMarket		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Normal market	BIT	1	0
Oddlot market	BIT	1	0
Spot market	BIT	1	0
Auction market	BIT	1	0
Call Auction1	BIT	1	0
Call Auction2	BIT	1	0
Reserved	BIT	2	0
Reserved	BIT	7	1
Preopen	BIT	1	1

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_ON_REQUEST_IN (2300).
UserId	This field should contain User ID of user/broker. This field accepts numbers only.
Password	This field should contain the password entered by the user. A combination of alphabet, numbers and special characters are allowed in the password. The user should enter the password for a successful Logon. When the user logs on for the first time the default password provided by NSE must be entered and the password should be changed by entering a new password.
NewPassword	This field should contain the new password entered by the user. This field should be entered only when the user wishes to change the password or the password has expired. Otherwise this field should be blank. The New Password should be entered along with the old password in the Password field. While logging on the system for the first time, the default password provided by NSE must be changed. the new password entered will undergo following new validations :

Field Name	Brief Description
	• The length of password should be of exact 8 characters. • The password should contain at least 1 upper case letter, 1 lower case letter, 1 numeral and 1 special characters from the list (@ # $ % & * / \). • New password must be different from previous 5 passwords. • User Id shall be locked after 3 invalid login attempts. • User shall not be allowed to set the default password as new password.
TraderName	This field when received from the host contains the user's name. This field should be sent to host as blanks.
LastPassword ChangeDateTime	This field should be set to numerical zero while log on.
BrokerId	This field should contain the trading member ID.
BranchId	This field should contain the Branch ID to which the broker belongs.
VersionNumber	This field should contain the version number of the trading system. It must be in the following format: VERSION.RELEASE.SUB_RELEASE (For example, 01.00.01) As and when these structures are changed, the version number will be changed.
UserType	This field indicates the type of user. It can take one of the following values when it is sent from the host: '0' denotes Dealer '4' denotes Corporate Manager '5' denotes Branch Manager '7' denotes Market Maker This field should be set to '0' while sending to the host.
SequenceNumber	This field should be set to numerical zero while sending the request to host.
WorkstationNumber	The network ID of the workstation should be provided. This is a seven digit number. The first five digits are fixed by the Exchange and represent the various ports / switch locations. The last two digits denote the user's PC - ID. It must be any number other than '00'.

Field Name	Brief Description
BrokerStatus	This field should be set to blank.
BrokerEligibilityPer Market	This field should be set to numerical zero.
BrokerName	This field should be set to blank

Logon Response

The response will either be **Confirmation** or **Logon Error**.

Logon Confirmation Response

A successful logon results in the Logon Confirmation Response. In SIGN_ON_REQUEST_OUT, Eligibility for the broker in CALL AUCTION 2 is being used by the existing reserved Market bit in BrokerEligibilityPerMarket structure. If the value received in these bits is 1‘, the broker is eligible to trade in respective markets. The following modified structure will be sent to the TWS from the Exchange:

Table 8 SIGNON OUT

Structure Name	SIGNON OUT		
Packet Length	276 bytes		
Transaction Code	SIGN_ON_REQUEST_OUT (2301)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	8	44
Password	CHAR	8	52
Reserved	CHAR	8	60
NewPassword	CHAR	8	68
TraderName	CHAR	26	76
LastPasswordChangeDate	LONG	4	102
BrokerId	CHAR	5	106
Reserved	CHAR	1	111
BranchId	SHORT	2	112
VersionNumber	LONG	4	114
EndTime	LONG	4	118

Structure Name	SIGNON OUT		
Packet Length	276 bytes		
Transaction Code	SIGN_ON_REQUEST_OUT (2301)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	52	122
UserType	SHORT	2	174
SequenceNumber	DOUBLE	8	176
Reserved	CHAR	14	184
BrokerStatus	CHAR	1	198
Reserved	CHAR	1	199
BrokerEligibilityPerMarket (Refer Table 8.1 for Small Endian Machines and Table 8.2 for Big Endian Machines)	STRUCT	2	200
BrokerName	CHAR	26	202
Reserved	CHAR	16	228
Reserved	CHAR	16	244
Reserved	CHAR	16	260

Table 8.1 BrokerEligibilityPerMarket (For Small Endian Machines)

Structure Name	BrokerEligibilityPerMarket		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	BIT	2	0
Call Auction2	BIT	1	0
Call Auction1	BIT	1	0
Auction market	BIT	1	0
Spot market	BIT	1	0
Oddlot market	BIT	1	0
Normal market	BIT	1	0
Preopen	BIT	1	1
Reserved	BIT	7	1

Table 8.2 BrokerEligibilityPerMarket (For Big Endian Machines)

Structure Name	BrokerEligibilityPerMarket		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Normal market	BIT	1	0
Oddlot market	BIT	1	0
Spot market	BIT	1	0
Auction market	BIT	1	0
Call Auction1	BIT	1	0
Call Auction2	BIT	1	0
Reserved	BIT	2	0
Reserved	BIT	7	1
Preopen	BIT	1	1

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_ON_REQUEST_OUT (2301).
LogTime	The current time at the trading system is sent back as number of seconds since midnight of January 1, 1980 The time at the Trader workstation must be synchronized with this.
UserId	This field contains the ID of the user.
Password	This field will be set to NULL.
NewPassword	This field will be set to NULL.
TraderName	This field contains the user name.
LastPassword ChangeDate	This filed contains the last date time when the password was changed.
BrokerId	This field contains the Trading Member ID.
BranchId	This field contains the branch ID of the particular user.
Version No	This field contains the version number of the trading system
EndTime	This field contains the time the markets last closed and is sent as the number of seconds since midnight of January 1, 1980. If this time is different from the time sent in an earlier log on, all orders, trades and messages for this trader must be deleted from the Local Database.
UserType	This field contains the type of user who is logging on: • '0' – Dealer • '4' – Corporate Manager

Field Name	Brief Description
	• '5' – Branch Manager • '7' –Market Maker
SequenceNumber	This field contains the time when the markets closed the previous trading day.
BrokerStatus	This field contains the current status of the broker: • 'S' for Suspended • 'A' for Active • 'D' for Deactivated • 'C' for Closeout or voluntary closeout
BrokerEligibility PerMarket	This structure specifies the markets that are allowed for the trading member. The trading member is eligible to enter orders in the markets that are set to 1.
BrokerName	This field contains the broker's name (trading member name).

Logon Error

In case of any error, the structure returned is:

ERROR RESPONSE (Refer to *Error Message* in Chapter 2)

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_ON_REQUEST_OUT (2301).
ErrorCode	This contains the error number. If the version number is not the same as at the host end, the version number at the host can be extracted from Error_Message in ERROR_RESPONSE (8 bytes from location 95 in the string). The format of it will be VV.RR.SS. The version number at the front end should be set to VVRRSS. Refer to List of Error Codes in Appendix.

System Information Download

The current status of the markets and the values of global variables are downloaded to the trader in response to *system information* request.

System Information Request

This request can be sent only if the user has logged on successfully. The format of the request is as follows:

Table 9 SYSTEM_INFO_REQ

Structure Name	SYSTEM_INFO_REQ		
Packet Length	40 bytes		
Transaction Code	SYSTEM_INFORMATION_IN (1600)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0

Field Name	Brief Description
TransactionCode	The transaction code is SYSTEM_INFORMATION_IN (1600).

Note: TWS User has to set time_stamp2 field present in the TWS message header to zero in SYSTEM_INFORMATION_IN message.

System Information Response

The following structure is returned as a response to the system information request:

Table 10 SYSTEM_INFORMATION_DATA

Structure Name	SYSTEM_INFORMATION_DATA		
Packet Length	94 bytes		
Transaction Code	SYSTEM_INFORMATION_OUT (1601)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
Normal	SHORT	2	40
Oddlot	SHORT	2	42
Spot	SHORT	2	44
Auction	SHORT	2	46
Call Auction1	SHORT	2	48
Call Auction2	SHORT	2	50
MarketIndex	LONG	4	52
DefaultSettlementPeriod (Normal)	SHORT	2	56
DefaultSettlementPeriod (Spot)	SHORT	2	58
DefaultSettlementPeriod (Auction)	SHORT	2	60
CompetitorPeriod	SHORT	2	62
SolicitorPeriod	SHORT	2	64
WarningPercent	SHORT	2	66
VolumeFreezePercent	SHORT	2	68

Structure Name	SYSTEM_INFORMATION_DATA		
Packet Length	94 bytes		
Transaction Code	SYSTEM_INFORMATION_OUT (1601)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	2	70
TerminalIdleTime	SHORT	2	72
BoardLotQuantity	LONG	4	74
TickSize	LONG	4	78
MaximumGtcDays	SHORT	2	82
SECURITY ELIGIBLE INDICATORS(Refer Table 10.1 for Small Endian machines and Table 10.2 for Big Endian machines)	STRUCT	2	84
DisclosedQuantityPercentAllowed	SHORT	2	86
Reserved	CHAR	6	88

Table 10.1 SECURITY ELIGIBLE INDICATORS (For Small Endian Machines)

Structure Name	SECURITY ELIGIBLE INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	5	0
Books Merged	BIT	1	0
Minimum Fill	BIT	1	0
AON	BIT	1	0
Reserved	CHAR	1	1

Table 10.2 SECURITY ELIGIBLE INDICATORS (For Big Endian Machines)

Structure Name	SECURITY ELIGIBLE INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
AON	BIT	1	0
Minimum Fill	BIT	1	0
Books Merged	BIT	1	0
Reserved	BIT	5	0

Structure Name	SECURITY ELIGIBLE INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	The transaction code is SYSTEM_INFORMATION_OUT (1601).
Alphachar	This field contains the number of streams present in the host from which Message download will be served. This field is present in the Message Header. This is totally of two bytes. Stream number will be populated in the first byte of alphachar.
MarketStatus	This field contains a value assigned for market status. Values are: '0' if it is Preopen '1' if it is Open '2' if it is Closed '3' if it is Preopen end For CALL AUCTION2 market, market status will be received as : '0' - Preopen '2' - Closed '3' - Preopen end In the pre-open state of the market, orders can only be entered but no matching takes place. The trading starts when the market is Open. No orders can be entered for a security when the market is closed.
MarketIndex	This field contains the current market index.
SettlementPeriod	This field contains the default settlement period in various markets. Default Settlement (Normal), Default Settlement (Spot) and Default Settlement (Auction).
CompetitorPeriod	This field contains the default competitor period for auction.
SolicitorPeriod	This field contains the default solicitor period for auction.
WarningPercent	This field contains the warning percentage. If a broker exceeds his turnover by this value in percent, a warning message is broadcast to all traders. Refer to Turnover Limit Exceeded Or Broker Reactivated in Chapter 7.

Field Name	Brief Description
VolumeFreezePercent	This field contains the volume freeze percentage. If a broker exceeds his turnover by this value in percent, the broker is deactivated and a message is broadcasted to all traders. Refer to Turnover Limit Exceeded Or Broker Reactivated in Chapter 7.
TerminalIdleTime	This field contains the idle time of the TWS terminal.
BoardLotQuantity	This field contains the board lot quantity. The regular lot order quantity must be a multiple of this quantity.
TickSize	This field contains the Tick size. The order price and the trigger price, if applicable, must be a multiple of this tick size.
MaximumGTCDays	This field contains the maximum GTC days, that is, the maximum number of days after which a Good Till Canceled order will be canceled.
SecurityEligibilityIndicator	If the Minimum Fill flag is set, then orders will have the Minimum Fill attribute set. If the All Or None (AON) flag is set, then orders will have the AON attribute set.
DisclosedQuantity PercentAllowed	This field contains the disclosed quantity allowed percentage. The disclosed quantity, if set, will not be lesser than this percent of the total quantity.

Update Local Database Download

The list of updated securities and participants is downloaded in response to *update local database* request. Any carried over GTC or GTD orders are also downloaded with this request. As of now GTC and GTD facilities are not allowed hence there will be no download for GTC and GTD orders.

Update Local Database Request

This message is sent to request the host end to update the local database at the front end. The structure sent is as follows:

Table 11 UPDATE_LOCALDB_IN

Structure Name	UPDATE_LOCALDB_IN		
Packet Length	62 bytes		
Transaction Code	UPDATE_LOCALDB_IN (7300)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0

Structure Name	UPDATE_LOCALDB_IN		
Packet Length	62 bytes		
Transaction Code	UPDATE_LOCALDB_IN (7300)		
Field Name	Data Type	Size in Byte	Offset
LastUpdateSecurityTime	LONG	4	40
LastUpdateParticipantTime	LONG	4	44
RequestForOpenOrders	CHAR	1	48
Reserved	CHAR	1	49
NormalMarketStatus	SHORT	2	50
OddLotMarketStatus	SHORT	2	52
SpotMarketStatus	SHORT	2	54
AuctionMarketStatus	SHORT	2	56
CallAuction1MarketStatus	SHORT	2	58
CallAuction2MarketStatus	SHORT	2	60

Field Name	Brief Description
TransactionCode	The transaction code is UPDATE_LOCALDB_IN (7300).
LastUpdateSecurityTime	This field should contain the time when the security information was last updated. This field is for each security for which information is downloaded. Further download requests can use the latest time to get updated information on the securities. Setting this time to zero results in complete download.
LastUpdateParticipantTime	This field should contain the time when the participant information was updated. This field is set for each participant for whom information is downloaded. Further download requests can use the latest time to get updated information on the participants. Setting this time to zero results in complete download.
RequestForOpenOrders	This field should be set to 'G' if GTC and GTD orders are to be downloaded. In other cases, it should be set to 'N'.
NormalMarketStatus	This field should contain the latest Normal Market status available at TWS.
OddLotMarketStatus	This field should contain the latest Odd Lot Market status available at TWS.

Field Name	Brief Description
SpotMarketStatus	This field should contain the latest Spot Market status available at TWS.
AuctionMarketStatus	This field should contain the latest Auction Market status available at TWS.
Call Auction1MarketStatus	This field should contain the latest CALL AUCTION1 Market status available at TWS.
Call Auction2MarketStatus	This field should contain the latest CALL AUCTION2 Market status available at TWS.

Update Local Database Response

The response will be either the database download, or a partial system information download. The latter will occur if the trader does not have the latest market status.

Partial System Information Response

This is returned if the market status sent in the UPDATE_LOCALDB_IN message is not the same at the host end or the symbols (securities) are opening. In this case the market status at the host end is sent back in the MARKET STATUS as 'wait till markets are open'. The following structure is returned:

SYSTEM INFORMATION DATA (Refer to *System Information Response* in Chapter 3)

Field Name	Brief Description
TransactionCode	The transaction code is PARTIAL_SYSTEM_INFORMATION (7321).
MarketStatus	This contains the latest market status.

Update Local Database Download

The download comprises of a header, data and the trailer. Each updated security status, participant (if selected) and GTC/GTD order will be sent as a separate message. As of now GTC and GTD facilities are not allowed hence there will be no download for GTC and GTD orders.

Update Local Database Header

This is sent only to indicate that a sign-on download is going to commence. There is no additional data sent. The header is sent in the following format:

Table 12 UPDATE_LDB_HEADER

Structure Name	UPDATE_LDB_HEADER		
Packet Length	42 bytes		
Transaction Code	UPDATE_LOCALDB_HEADER (7307)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
Reserved	CHAR	2	40

Field Name	Brief Description
TransactionCode	The transaction code is UPDATE_LOCALDB_HEADER (7307).

Update Local Database Data

The actual data is sent wrapped in another header. The outer header indicates that this message is part of the Update Local Database Data. The inner header indicates the type of data received.

The structure is as follows:

Table 13 UPDATE_LOCAL_DB_DATA

Structure Name	UPDATE_LOCAL_DB_DATA		
Packet Length	80 to 512 bytes		
Transaction Code	UPDATE_LOCALDB_DATA (7304)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
Data	CHAR	472 – (For inner Header Refer *Inner Message Header* in Chapter 2)	40

Field Name	Brief Description
TransactionCode	The transaction code is UPDATE_LOCALDB_DATA (7304).
InnerTransactionCode	The transaction codes sent are BCAST_SECURITY_MSTR_CHG. It is determined by NSE-Control whether to send this or not. (Refer to Change in Security Master in Chapter 7)

Field Name	Brief Description
	BCAST_SECURITY_STATUS_CHG. This transaction code is sent when the status of the stock is different from the expected status at the host end (Refer to Change of Security Status in Chapter 7) BCAST_PART_MSTR_CHG. If there is any change in the participant master after the time specified by the Last Update Participant Time, it is downloaded. (Refer to Change Participant Status in Chapter 7) - In all above messages, use INNER_MESSAGE_HEADER [Refer Inner Message Header in Chapter 2] instead of MESSAGE_HEADER

Update Local Database Trailer

This indicates that the download is complete. This is sent in the following format:

Table 14 UPDATE_LDB_TRAILER

Structure Name	UPDATE_LDB_**TRAILER**		
Packet Length	42 bytes		
Transaction Code	UPDATE_LOCALDB_TRAILER. (7308)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
Reserved	CHAR	2	40

Field Name	Brief Description
TransactionCode	The transaction code is UPDATE_LOCALDB_TRAILER (7308).

Message Download

This request is used to download the messages intended for the trader from the trading system. When the trader makes a request for message download, all the transactions of the trader and other important broadcasts are downloaded.

Message downloads will be served through each individual stream. Hence, message download request needs to be sent individually for a stream by the user.

Message Download Request

This message is sent for requesting message download. The structure sent to the trading system is:

Table 17 MESSAGE DOWNLOAD

Structure Name	MESSAGE DOWNLOAD		
Packet Length	48 bytes		
Transaction Code	DOWNLOAD_REQUEST (7000)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(Refer Table 1)	STRUCT	40	0
SequenceNumber	DOUBLE	8	40

Field Name	Brief Description
TransactionCode	The transaction code is DOWNLOAD_REQUEST (7000).
SequenceNumber	This contains the time last message was received by the workstation. This can be obtained from the Time Stamp1 of the MESSAGE_HEADER. To retrieve the messages from the beginning of the trading day, this field should be set to '0' or the Sequence Number received in the logon response message.
AlphaChar	This contains the stream number of the host to which it has to send the DOWNLOAD_REQUEST. The alpachar is the character array of size 2. The stream number of the host is sent in the first byte of the alphachar. The number of streams is obtained in SYSTEM_INFORMATION_OUT from host during login sequence.

Message Download Response

The download comprises of a header, data and the trailer. Each trader specific and broadcast message will be sent as a separate message.

Message Download Header

This is only to indicate that a message download is going to commence. There is no additional data sent. The header is sent in the following format:

MESSAGE HEADER (Refer to *Table 1*)

Field Name	Brief Description
TransactionCode	The transaction code is HEADER_RECORD (7011).

Message Download Data

The messages are similar to Update Local Database Data. The actual data is sent wrapped in another structure. The outer header indicates that this message is part of the Message Download Data. The inner header indicates the type of data received. The structure is shown below.

Table 18 MESSAGE HEADER

Structure Name	MESSAGE_HEADER		
Packet Length	80 to 512 bytes		
Transaction Code	MESSAGE_RECORD (7021)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
Data	CHAR	472 – (For inner Header Refer *Table 2*)	40

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter*. The transaction code is MESSAGE_RECORD (7021).
InnerData	Set of transaction codes are received. They include Trader Specific Messages • Logon / Logoff response Refer to Logon Process, Chapter 3. • Interactive message sent to the user from the NSE-Control. Refer to Unsolicited Messages, Chapter 5. • Order entry, Modification, Cancellation responses Refer to Order and Trade Management, Chapter 4 • Trade Modification, Cancellation responses Refer to Order and Trade Management, Chapter 4.

Field Name	Brief Description
	• Trade Confirmation, Stop Loss Trigger Refer to Unsolicited Messages, Chapter 5. • Broadcast Messages Market Open, Market Close, Market Pre-Open ended, Preopen Shutdown Message, Broadcast Message String, Turnover exceeded, Broker Reactivated, Broadcast message sent from NSE-Control. Refer to Broadcast, Chapter 7 • Contingency Broadcast Message Refer to Exception Handling, Chapter 11.

Message Download Trailer

This indicates that message download is completed for the particular stream. Once download is completed for one stream, DOWNLOAD_REQUEST will be sent for the next stream with its corresponding sequence number. Request will be sent until message download gets completed for all the streams. The format is as follows:

MESSAGE HEADER (Refer to *Table 1*)

Field Name	Brief Description
TransactionCode	The transaction code is TRAILER_RECORD (7031).

Logoff Request

The process by which a trader quits or signs off from the trading system is called Logoff Process. The structure sent is:

MESSAGE HEADER (Refer to *Table 1*).

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_OFF_REQUEST_IN (2320).

Logoff Confirmation Response

When the user logs on again, the user receives a packet giving the details of when he/she logged off. The structure sent is:

MESSAGE HEADER (Refer to *Table 1*)

Note: MS_SIGNOFF message is sent in the Message Header itself. The length of the packet is 40 bytes.

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_OFF_REQUEST_OUT (2321).
LogTime	This field contains the current time at the trading system is sent back as number of seconds since midnight of January 1, 1980. The time at the workstation must be synchronized with this.

Chapter 4 Order and Trade Management

Introduction

This section describes about entering new orders, modifying existing orders, and canceling outstanding orders. The trader can begin entering the orders once he has logged on to the trading system and the market is in pre-open or open state.

Please note this section is referenced in CM_DROP_COPY_PROTOCOL document. Any change here may also impact the Order Drop Copy functionality

Order Entry

Order entry allows the trader to place orders in the market. The system accepts the orders from the users and tries to match the orders with the orders in the books immediately. If the order does not match, the order is placed in the appropriate book with the price and time stamp.

NOTE:

When market status is pre-open, order entry request will be accepted only if pre-open indicator is set as '1', else orders will be rejected.

Order Types

Regular Lot

Regular Lot Orders are orders in the normal market that have none of the following terms attached: All Or None, Minimum Fill and Trigger Price.

Preopen Orders are Regular Lot orders placed when normal market is in Preopen. Pre-open orders will be identified by pre-open indicator. None of the following terms attached: DQ, All or None, Minimum Fill and Trigger Price.

Special Terms

Special Terms Orders are orders in the normal market which have special attribute attached to it. They must have Minimum Fill (MF) or All Or None (AON).

Stop Loss Orders

Stop Loss Orders are orders in normal market with Trigger Price specified. They may have the Minimum Fill or AON attribute specified.

Odd Lot Orders

Odd lot orders are orders in the Odd Lot Market with the order quantity being less than the Regular lot quantity.

Spot Orders

Spot Orders are orders in spot market where the settlement period is different from the normal market and is fixed by the exchange.

Auction Orders

Auction Orders are simple day orders and can only have the 'Day' term set to 1. ATA (at Auction) Price is not allowed for auction. A valid price has to be entered. Currently, **only those auctions that are initiated by the Exchange are allowed.** The trader has to enter the solicitor orders after the auction is initiated and before it ends (during Solicitor Period). **Auction Orders can only be cancelled. They cannot be modified.**

Call Auction

Call Auction order are orders placed in CALL AUCTION market that have none of the following terms attached: All or None, Minimum Fill and Trigger Price, Disclosed quantity.

Call Auction 1 orders are IOC orders and Call Auction 2 orders are DAY orders with limit price. Both Call Auction 1 and Call Auction 2 orders have settlement period same as Normal market.

Order Terms

Following terms and conditions can be used during order entry and order modification.

Disclosed Quantity (DQ)

This term allows the dealer to disclose only a portion of the order quantity to the market. After the initial disclosed quantity is matched, subsequent disclosed quantity is shown to the market. All the disclosures will be shown to the market with the same order number.

Trigger Price (TP)

The Stop Loss book type allows the broker to release an order into the system after the market price crosses a threshold price referred to as the trigger price. This facility is available for orders in Normal market only. For a stop loss buy order, the trigger price should not be greater than the limit price. For a stop loss sell order, the trigger price should not be less than the limit price. All the stop loss orders will be kept in a separate book till they are triggered.

Immediate or Cancel (IOC)

This term forces the order to match immediately, else be cancelled. If the order trades partially, the remaining part is cancelled.

Day

This is the default term for an order. At the end of the trading day, all outstanding Day orders are cancelled by the system.

Good till Date (GTD)

This term allows the dealer to keep an order in the system for a certain number of days. The number of days must be greater than 1 and less than or equal to the maximum number of days allowed for GTC orders. Each day is a calendar day. This facility is disabled as of now.

Good till Cancelled (GTC)

This term allows the broker to keep an order in the system until it is canceled. However, the order is canceled by the system automatically if it remains outstanding for more than the maximum number of days allowed for GTC orders. This facility is disabled as of now.

Minimum Fill (MF)

This term allows the broker to ensure that the quantity traded is at least the Minimum Fill amount specified. The minimum fill must be in multiples of the market lot and less than the order quantity. MF quantity must be less than or equal to Disclosed Quantity when the order has both MF and Disclosed Quantity attributes.

All or None (AON)

This term allows the broker to ensure that the entire order is traded and if not, nothing is traded at all. This can result in multiple trades or a single trade.

Rules of Order Entry

Order entry is not allowed in the following conditions:

- Markets are closed.
- Security is suspended.
- Security has matured.
- Security is expelled.
- Security admission date is greater than current date.
- Security is not eligible in the particular market.

- Security does not exist in the system.
- Broker is suspended.
- Broker does not exist in trading system.
- Broker is deactivated.
- User's branch order limit has exceeded.
- User is disabled.
- User is an inquiry user.
- User does not exist in trading system.
- Participant is suspended.
- Participant does not exist in trading system.
- Order price is beyond day's minimum maximum range.
- Trigger price is worse than limit price.
- Quantity is more than issued capital.
- Quantity is not equal to multiples of regular lot.
- Disclosed Quantity is less than the given percentage (determined by exchange) of order Quantity.
- Disclosed Quantity is more than order Quantity.
- Disclosed Quantity is not equal to multiples of regular lot.
- MF Quantity is more than order Quantity.
- MF Quantity is not a multiple of regular lot.
- Limit Price is not a multiple of Tick Size.
- Trigger Price is not a multiple of Tick Size.
- GTC/GTD days more than specified days.
- Spot orders with GTC/GTD.
- Auction orders with GTC/GTD/IOC.

- IOC and Disclosed Quantity combination.
- Difference between limit price and trigger price in stop loss limit orders is greater than permissible range.

Order Entry Request

The format of the order entry request is as follows:

Table 19 ORDER_ENTRY_ REQUEST

Structure Name	ORDER_ENTRY_REQUEST/RESPONSE		
Packet Length	290 bytes		
Transaction Code	BOARD_LOT_IN (2000)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(Refer Table 1)	STRUCT	40	0
ParticipantType	CHAR	1	40
Reserved	CHAR	1	41
CompetitorPeriod	SHORT	2	42
SolicitorPeriod	SHORT	2	44
ModCxlBy	CHAR	1	46
Filler9	CHAR	1	47
ReasonCode	SHORT	2	48
Reserved	CHAR	4	50
SEC_INFO (Refer Table 4)	STRUCT	12	54
AuctionNumber	SHORT	2	66
OpBrokerId	CHAR	5	68
Suspended	CHAR	1	73
OrderNumber	DOUBLE	8	74
AccountNumber	CHAR	10	82
BookType	SHORT	2	92
BuySell	SHORT	2	94
DisclosedVol	LONG	4	96
DisclosedVolRemaining	LONG	4	100
TotalVolRemaining	LONG	4	104
Volume	LONG	4	108
VolumeFilledToday	LONG	4	112

Structure Name	ORDER_ENTRY_REQUEST/RESPONSE		
Packet Length	290 bytes		
Transaction Code	BOARD_LOT_IN (2000)		
Field Name	Data Type	Size in Byte	Offset
Price	LONG	4	116
TriggerPrice	LONG	4	120
GoodTillDate	LONG	4	124
EntryDateTime	LONG	4	128
MinFillAon	LONG	4	132
LastModified	LONG	4	136
ST_ORDER_FLAGS (Refer Table 19.1 for small endian machines and Table 19.2 for big endian machines)	STRUCT	2	140
BranchId	SHORT	2	142
TraderId	LONG	4	144
BrokerId	CHAR	5	148
OERemarks	CHAR	25	153
Settlor	CHAR	12	178
ProClient	SHORT	2	190
SettlementType	SHORT	2	192
NNFField	DOUBLE	8	194
ExecTimeStamp	DOUBLE	8	202
Reserved	CHAR	4	210
PAN	CHAR	10	214
Algo ID	LONG	4	224
Reserved Filler	SHORT	2	228
LastActivityReference	LONG LONG	8	230
Reserved	CHAR	52	238

For Small Endian Machines:

Table 19.1 ST_ORDER_FLAGS

Structure Name	ST_ORDER_FLAGS
Packet Length	2 bytes

Field Name	Data Type	Size in Bit	Offset
MF	BIT	1	0
AON	BIT	1	0
IOC	BIT	1	0
GTC	BIT	1	0
Day	BIT	1	0
OnStop	BIT	1	0
Mkt	BIT	1	0
ATO	BIT	1	0
Reserved	BIT	1	1
STPC	BIT	1	1
Reserved	BIT	1	1
Preopen	BIT	1	1
Frozen	BIT	1	1
Modified	BIT	1	1
Traded	BIT	1	1
MatchedInd	BIT	1	1

For Big Endian Machines:

Table 19.2 ST_ORDER_FLAGS

Structure Name	ST_ORDER_FLAGS		
Packet Length	2 bytes		
Field Name	Data Type	Size in Bit	Offset
ATO	BIT	1	0
Mkt	BIT	1	0
OnStop	BIT	1	0
Day	BIT	1	0
GTC	BIT	1	0
IOC	BIT	1	0
AON	BIT	1	0
MF	BIT	1	0
MatchedInd	BIT	1	1
Traded	BIT	1	1
Modified	BIT	1	1

Frozen	BIT	1	1
Preopen	BIT	1	1
Reserved	BIT	1	1
STPC	BIT	1	1
Reserved	BIT	1	1

The description and values of the fields are given below.

Field Name	Brief Description
TransactionCode	The transaction code is BOARD_LOT_IN (2000).
ParticipantType	Since only exchange can initiate the auction, this field should not be set to 'I' for initiator. This should be set to 'C' for competitor order and 'S' for solicitor order.
CompetitorPeriod	This field should be set to zero.
SolicitorPeriod	This field should be set to zero.
ModCxlBy	This field denotes which person has modified or cancelled a particular order. It should contain one of the following values: • 'T' for Trader • 'B' for Branch Manager • 'M' for Corporate Manager • 'C' for Exchange
ReasonCode	This field contains the reason code for a particular order request rejection or order being frozen. This has the details regarding the error along with the error code. This field should be set to zero while sending the request to the host. Refer to Reason Codes in Appendix.
SEC_INFO	This structure should contain the Symbol and Series of the security.
AuctionNumber	Auction number is available when initiation of auction is broadcast (Auction Status Change Broadcast). For an auction order, valid auction number should be given. For other books, this field should be set to zero.
OpBrokerId	This field will always be blank.
Suspended	This field specifies whether the security is suspended or not. It should be set to blank while sending order entry request.

Field Name	Brief Description
AccountNumber	If the order is entered on behalf of a trader, the trader account number should be specified in this field. For broker's own order, this field should be set to the broker code.
BookType	This field should contain the type of order. Refer to Book Types in Appendix. MS_OE_REQUEST structure is not allowed with book type values 1 , 11 and 12 for following request transcodes 1)BOARD_LOT_IN(2000) 2)ORDER_MOD_IN(2040) 3)ORDER_CANCEL_IN(2070) Refer Trimmed Order Structure (See Appendix - Trimmed Request Structures) for placing following orders transcodes with book type 1 or 11 or 12 1)For BOARD_LOT_IN (2000), use struct MS_OE_REQUEST_TR with transcode as 20000/20400 2)For ORDER_MOD_IN (2040), use struct MS_OM_REQUEST_TR with transcode as 20040/20402 3)For ORDER_CANCEL_IN (2070), use struct MS_OM_REQUEST_TR with transcode as 20070/20404
BuySell	This field should specify whether the order is a buy or sell. It should take one of the following values. • '1' for Buy order • '2' for Sell order
DisclosedVol	This field should specify the quantity that has to be disclosed to the market. It is not applicable if the order has either the All Or None or the Immediate Or Cancel attribute set. It should not be greater than the volume of the order and not less than the Minimum Fill quantity if the Minimum Fill attribute is set. In either case, it cannot be less than the Minimum Disclosed Quantity allowed. It should be a multiple of the Regular lot.
DisclosedVolRemaining	This field contains the disclosed volume remaining from the original disclosed volume after trade(s). This should be set to zero while sending to the host.

Field Name	Brief Description
TotalVolRemaining	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry, this field should be set to Volume. Thereafter, for every response the trading system will return this value.
Volume	This field should specify the quantity of the order placed. The quantity should always be in multiples of Regular Lot except for Odd Lot orders, and be less than the issued capital. The order will go for a freeze if the quantity is greater than the freeze quantity determined by NSE-Control.
VolumeFilledToday	This field contains the total quantity traded in a day.
Price	This field should contain the price at which the order is placed. To enter a Market order, the price should be zero. The price must be a multiple of the tick size. For Stop Loss orders, the limit price must be greater than the trigger price in case of a Buy order and less if it is a Sell order. This is to be multiplied by 100 before sending to the trading system host.
TriggerPrice	This field is applicable only for a Stop Loss order and should be a multiple of the tick size. This field should contain the price at which the order is to be triggered and brought to the market. For a Stop Loss buy order, the trigger price will be less than or equal to the limit price but greater than the last traded price. For a Stop Loss sell order, the trigger price will be greater than or equal to the limit price but less than the last traded price. This is to be multiplied by 100 before sending to trading system.
GoodTillDate	This field should contain the number of days for a GTD order. This field may be set in two ways. To specify an absolute date set this field to that date in number of seconds since midnight of January 1, 1980. To specify days set this to the number of days. This can take values from 2 to the maximum days specified for GTC orders only. If this field is non-zero, the GTC flag must be off.
EntryDateTime	This field should be set to zero while sending the order entry request.
MinimumFillAon	This field should contain the minimum fill quantity when the minimum fill attribute is set for an order. It should not be greater

Field Name	Brief Description
	than either the volume of the order or the disclosed quantity and must be a multiple of the regular lot.
LastModified	If the order has been modified, this field contains the time when the order was last modified. It is the time in seconds from midnight of January 1 1980, This field should be set to zero for the order entry request (it is same as Entry Date Time.)
Order_Flags	This structure specifies the attributes of an order. The Bit fields must be set / unset by Front end. Refer section Quick Reference for Order Entry Parameters for description. In order entry request, Mkt bit should be set to 0. If the order price is 0 in the request message, then the order will be considered as market order. For market order, Mkt bit will be set as 1 in order response message.
BranchId	This field should contain the ID of the branch of the particular broker.
TraderId	This field should contain the ID of the user. This field accepts only numbers.
BrokerId	This field should contain the trading member ID.
OERemarks	This field may contain any remarks that the dealer can enter about the order in this field.
Settlor	This field contains the ID of the participants who are responsible for settling the trades through the custodians. By default, all orders are treated as broker's own orders and this field defaults to the Broker Code.
ProClient	This field should contain one of the following values based on the order entering is on behalf of the broker or a trader. '1' - represents the client's order. '2' - represents a broker's order. '4' - represents warehousing order.
SettlementType	This field contains the settlement type. It can be one of the following: '0' – T+0 settlement '1' – T+1 settlement

Field Name	Brief Description
	This field should be set to zero while sending to the host.
NNFField	This field should contain a 15 digit a unique identifier for various products deployed as per Exchange circular download ref no. 16519 dated December 14, 2010 and as updated from time to time
ExecTimeStamp	This field is used to store the time of writing to the order book. This should be set to zero while sending to the host.
PAN	This field shall contain the PAN (Permanent Account Number / PAN_EXEMPT) - This field shall be mandatory for all orders (client / participant / PRO orders).
Algo ID	For Algo order this field shall contain the Algo ID issued by the exchange. For Non-Algo order, this field shall be Zero(0)
Reserved Filler	This field is reserved for future use. This should be set to Zero (0) while sending to the exchange trading system.
LastActivityReference	This field should be set to zero while sending the order entry request.

Above changes are to be handled in Order Modification (2040) and Order Cancellation request (2070).

Order matching for the call auction2 session shall commence at the end of order collection period. Once orders are matched the outstanding orders will be carried forward to the normal market or will be cancelled by the system. The transcode ORDER_CANCEL_CONFIRMATION (2075) will be sent, in case of Order Cancelled by the System.

Order Entry Response

The response can be Order Confirmation, Order Freeze, Order Error or one of the general error responses. Order Freeze response is not expected for Auction Order Entry. Order freeze response is generated when the order placed by the trader has resulted in freeze and is waiting for the approval of the exchange. The order error response is given when the entered order is rejected by the trading system. The reason for the rejection is given by Error Code.

Order Confirmation Response

Successful order entry results in Order Confirmation Response. The confirmed order is returned to the user. When the entered order goes for a freeze and that freeze is approved, this same

transaction code is sent. This can be an unsolicited message as well. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CONFIRMATION (2073).
Suspended	This field contains 'C' if the broker is in Closeout.
OrderNumber	This field contains an Order Number assigned to the order. It is a unique identification for an order. The first two digits will contain the stream number (This will be different from the stream number for Journal Download Request-Response). The next fourteen digits will contain fourteen digit sequence number.
Price	This field contains the price of the order. If a Market order was entered when market was in Open state, the 'Market' flag in Order Terms is set and is priced at the prevailing price at the trading system. If the market order is entered when the market was in preopen, the trading system sets the 'ATO' bit in Order Terms and prices at '-1'. If it was a priced order the order gets confirmed at that price.
Order_Flags	(Refer to *Order Entry Request* in Chapter 4)
EntryDateTime	This field contains the time at which order confirmed.
LastActivityReference	This field contains a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Market Price Confirmation Response

Market Price response is generated only when the order placed by the trader is a market order and the market order entered is not fully traded at exchange. This response is not expected for the limit orders. The response packet is sent only when there is any untraded quantity left in the order.

The message sent is:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is PRICE_CONFIRMATION (2012).
Price	This field contains the price of the order. If a Market order was entered when market was in Open state, the 'Market' flag in Order Terms is set and price is set at the prevailing price at the

	trading system. If the market order is entered when the market was in preopen, this transcode is not received. For Buy order the Price will be negative but for Sell order it will be positive
Order_Flags	(Refer to *Order Entry Request* in Chapter 4)

Order Freeze Response

Order freeze response is generated when the order placed by the trader or the order after modification is awaiting approval from the exchange. This response is not expected for Auction Orders. Exchange approval of the order results in a Freeze Approval Response and rejection results in Freeze Reject Response. These responses are sent as unsolicited messages. The format sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is FREEZE_TO_CONTROL (2170).
Order_Flags	(Refer to *Order Entry Request* in Chapter 4)

Order Error Response

The order error response is sent when the entered order is rejected by the trading system. The reason for the rejection is given by the reason code and the reason string. The message sent is:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_ERROR (2231).
ErrorCode	This field contains the error number. Refer to List of Error Codes in Appendix.
Suspended	This field contains 'C' if the broker is in Closeout.

Order Modification

Order Modification enables the trader to modify unmatched orders. All order types except Auction can be modified.

Rules of Order Modification

The following modifications are not allowed:

- Buy to Sell or vice versa.
- Modifying Symbol and Series.
- Modifying Participant field.
- Modifying Pro/Cli field.
- Modifying Frozen orders.
- BM modifying CM's orders.
- DL modifying BM's orders.
- DL modifying CM's orders.
- Modifying non existing order.
- Inquiry user trying to modify.
- Modifying an order in such a way that it results in a branch order value to be exceeded.
- Modifying Auction orders.
- Modifying deactivated broker's orders.
- Changing of original data.
- Modifying AU, SP, OL book type fields.
- Difference between limit price and trigger price in stop loss limit orders is greater than permissible range.

Note: RL/ST/SL book types can be toggled between themselves only. They cannot be modified to AU or SP or OL.

Order Modification Request

The trader can modify the quantity, price and attributes of an order by specifying the order number of the order to be modified. The message sent is:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_MOD_IN (2040).
OrderNumber	This should contain order number which is the identity of the order to be modified.

Field Name	Brief Description
LastModifiedTime	This should contain time of last activity done on that order. Last activity could be order entry, order modification or last trade time of that order. It is in number of seconds from midnight of January 1, 1980,
LastActivityReference	This field should contain LastActivityReference value received in response of last activity done on that order. Last activity could be order entry confirmation, order modification confirmation or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.
Note: The other fields of order modification request are same as the fields of order entry request.	

Order Modification Confirmation Response

Successful modification of the order results in Order Modification Confirmation. When the order modification is confirmed, the order-modified time is filled and sent back. On modification, the order can result in a freeze. If the freeze is approved, order modification will be received as an "Unsolicited Message".

Unmatched ATO/ Limit Pre-open orders are carried forward to the Normal Market without any change in time priority. For unmatched ATO orders which are carried forward, derived price will be assigned, response for these orders will be sent to traders as "Unsolicited" modification response.

Unmatched Limit Pre-open orders are cancelled or carried forward to the Normal Market without any change in time priority for IPO/Relisting securities.

Unmatched limit Pre-open orders are carried forward to the next session without any change in time priority for illiquid securities

The structure sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_MOD_CONFIRMATION (2074).
LastModifiedTime	This field contains the time when the order was last modified. It is in number of seconds from midnight of January 1, 1980,
EntryDateTime	This field contains the time at which last modified by user. It is in number of seconds from midnight of January 1, 1980,

Field Name	Brief Description
ModCxlBy	This field will be set to `C` for the unmatched ATO orders, which are being carried forward to the Normal Market. This field will be set to `F` for the unmatched orders, which are being carried forward to the Normal Market from call auction 2 market for IPO/Relisting securities. Unmatched ATO orders are assigned derived price and are carried forward to the Normal Market.
Order_Flags	Preopen - This bit will be set to 1 for pre-open order modification response during pre-open market session and during Normal market session (for the carried forward orders). Preopen - This bit will be set to 1 for Call Auction 2 order modification response during Call Auction2 pre-open session and during Normal market session (for the carried forward orders) for IPO/Relisting securities. It will be set to 0 for Normal Market Open order modification response
LastActivityReference	This field contains a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Order Modification Error Response

The reason for rejection is given by the Error Code in the header. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_MOD_REJECT (2042).
Order_Flags	This bit will be set to 1 for pre-open order modification response during pre-open market session and during Normal market session (for the carried forward orders). Preopen - This bit will be set to 1 for Call Auction 2 order modification response during Call Auction2 pre-open session and during Normal market session (for the carried forward orders) for IPO/Relisting securities. It will be set to 0 for Normal Market Open order modification response

Field Name	Brief Description
Reason code	For Call Auction2, the reason code 24 will be sent. Refer to List of Reason Codes in Appendix.

Effect of Modifying the Terms of an Order on Price-Time Priority

Field Name	Can Change	Comments
Buy/Sell	No	
Order Type		Refer to Rules of Order Modification
Symbol	No	
Series	No	
Price	Yes	Changing the order price will always result in the order losing its time priority.
Quantity	Yes	The quantity of an order can be reduced any number of times without the order losing its time priority. However, increasing the quantity of an order will always result in the order losing its time priority.
PRO/CLI	No	
Account No.	No	
Day	Yes	Changing to or from a Day order retains time priority
GTC	Yes	Changing to or from a GTC order retains time priority
GTD	Yes	Changing to or from a GTD order retains time priority
Days in GTD	Yes	
DQ	Yes	Time Priority shall be lost if: - Changed DQ leads to an increase in quantity disclosed in the order book - DQ changed to non-DQ or vice versa and quantity disclosed in the order book increases
MF & AON	Yes	Changing MF to AON order or vice-versa will result in the order losing its time priority.
MF	Yes	Same as in Quantity.
SL	Yes	A SL order can be changed to a normal limit order or a Special Terms order by removing the SL attribute. The SL limit and trigger price can also be changed. In each of these cases the order loses its time priority.

Field Name	Can Change	Comments
Participant	No	
Remarks	Yes	Changing this does not change time priority.
Note: When the order quantity of an ATO or 'Market' order is modified, the order loses priority irrespective of increase or decrease in the quantity.		

Order Cancellation

The trader can cancel any unmatched/partially matched order by specifying the order number.

In after order collection period, the call auction order matching will be done. Once matching is completed the IOC orders which were not traded will get cancelled by the system, the transcode ORDER_CANCEL_CONFIRMATION (2075) will be sent.

In case of circuit hit, if Order collection phase is planned, orders related to normal market which were not traded will get cancelled by the system, the transcode ORDER_CANCEL_CONFIRMATION (2075) will be sent.

Rules of Order Cancellation

- CM can cancel BM's and DL's order, but BM and DL cannot cancel CM's order.
- BM can cancel DL's order, but DL cannot cancel BM's order.
- Deactivated broker cannot cancel his/her order.
- Auction orders cannot be cancelled after auction is finished.
- In case of CALL AUCTION 2 market, it is mandatory to mention a non-zero value in the price field.

Order Cancellation Request

The format of the message is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CANCEL_IN (2070).

Field Name	Brief Description
OrderNumber	This field should contain the order number which is the identity of the order to be cancelled.
Last ModifiedTime	This should contain time of last activity done on that order. Last activity could be order entry, order modification or last trade time of that order. It is in number of seconds from midnight of January 1, 1980,
LastActivityReference	This field should contain LastActivityReference value received in response of last activity done on that order. Last activity could be order entry confirmation, order modification confirmation or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Order Cancellation Response

The response can be Order Cancellation Confirmation, Order Cancellation Error or one of the general error responses.

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CANCEL_IN (2070).

Order Cancellation Confirmation Response

Successful cancellation of order results in Order Cancellation Confirmation Response. This will be an “Unsolicited Message” if NSE-Control cancels the order. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CANCEL_CONFIRMATION (2075).
Suspended	This field contains ‘C’ if the broker is in Closeout.
ModCxlBy	This field will be set to `C` for unmatched Pre-open orders cancelled by the Exchange. It will be blank for Pre-open orders which are cancelled by the trader in Preopen session and in Normal Market session.

Field Name	Brief Description
	This field will be set to `C` for unmatched Call Auction orders cancelled by the Exchange. It will be blank for Call Auction2 orders which are cancelled by the trader in Call Auction 2 Preopen session and in Normal Market session for IPO/Relisting securities.
Order_Flags	This bit will be set to 1 for Pre-open order cancellation response and Pre-open carried forward order cancellation response. Preopen - This bit will be set to 1 for Call Auction 2 order cancellation response during Call Auction2 pre-open session and during Normal market session (for the carried forward orders) session for IPO/Relisting securities. It will be set to 0 for Normal Market Open order cancellation response
LastActivityReference	This field contains a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Order Cancellation Error Response

The order cancellation error is sent when the cancellation request is rejected by the trading system. The reason for rejection is given by the Error Code in the header. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CANCEL_REJECT (2072).
Order_Flags	Preopen - This bit will be set to 1 for Pre-open order cancellation response and Pre-open carried forward order cancellation response. Preopen -This bit will be set to 1 for Call Auction 2 order cancellation response during Call Auction2 pre-open session and during Normal market session (for the carried forward orders) for IPO/Relisting securities.

Field Name	Brief Description
	And it will be set to 0 for Normal Market Open order cancellation response

Kill Switch

This functionality provides a facility to traders to cancel all of their orders at the same time.

Also, user can cancel all outstanding orders on particular security by specifying security information in request packet.

Kill Switch Request

The format of the message is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is KILL_SWITCH_IN (2062).
User	This field should contain the user id for which orders should be cancelled.
SEC_INFO	For cancellation of all orders, Symbol and series fields should be set as blank. For cancellation of all orders on particular security, this structure should contain the Symbol and Series of the security.

Kill Switch Response

The Quick cancel out response is sent when the kill switch is requested by the user. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is QUICK_CANCEL_OUT(2061)

Kill Switch Error Response

The kill switch error is sent when the request is rejected by the trading system. The reason for rejection will be given by the Error Code in the header. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_ERROR (2231).

Trade Modification

This functionality provides facility to traders to modify the trades. Only account number modification is allowed.

Following modifications are not allowed:

- Modifying Trade Quantity
- Modifiying Pro/Cli field
- Modifying Participant field.
- BM modifying CM's trades.
- DL modifying BM's trades.
- DL modifying CM's trades.
- Modifying non existing trade.
- Modifying Auction trades.

Trade Modification Request

The format of the message is as follows:

Table 20 TRADE_INQUIRY_DATA

Structure Name	TRADE_INQUIRY_DATA		
Packet Length	210 Bytes		
Transaction Code	TRADE_MOD_IN (5445)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(Refer Table 1)	STRUCT	40	0
SEC_INFO (Refer Table 4)	STRUCT	12	40
FillNumber	LONG	4	52
FillQty	LONG	4	56
FillPrice	LONG	4	60
MarketType	SHORT	2	64
NewVolume	LONG	4	66
Reserved	CHAR	24	70

Structure Name	TRADE_INQUIRY_DATA		
Packet Length	210 Bytes		
Transaction Code	TRADE_MOD_IN (5445)		
Field Name	Data Type	Size in Byte	Offset
BuyBrokerId	CHAR	5	94
SellBrokerId	CHAR	5	99
TraderId	LONG	4	104
RequestedBy	SHORT	2	108
BuyAccountNumber	CHAR	10	110
SellAccountNumber	CHAR	10	120
BuyPAN	CHAR	10	130
SellPAN	CHAR	10	140
Reserved	CHAR	60	150

The description and values of the fields are given below.

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_MOD_IN (5445).
SEC_INFO	This structure should contain the Symbol and Series of the security.
FillNumber	This field should contain the trade number of the trade to be modified.
FillQuantity	This field should contain the quantity that has been traded.
FillPrice	This field should contain the price at which the trade took place. This is to be multiplied by 100 before sending to the trading system host.
MarketType	This field should contain the value to denote the type of market, • '1' for Normal Market. • '2' for Odd Lot Market • '3' for Spot Market • '4' for Auction Market • '5' for CA1 • '6' for CA2
NewVolume	This field value should be same as that of FillQuantity.

Field Name	Brief Description
Buy / SellBrokerId	This field should contain the trading member ID of the broker who placed the order for the trade or the one who is responsible for the settlement.
TraderId	This field should contain the ID of the user on whose behalf request is to be made.
RequestedBy	This field indicates which side (Buy/Sell) of the trade is to be modified/cancelled. This should contain one of the following values • '1' (BUY) if the buy side is to be modified/cancelled • '2' (SELL) if the sell side is to be modified/cancelled • '3' (BUY & SELL) if both the sides are to be modified/cancelled.
BuyAccountNumber	This field should contain the Account Number of the trade on Buy side.
SellAccountNumber	This field should contain the Account Number of the trade on Sell side.
BuyPAN	This field shall contain the PAN (Permanent Account Number/PAN_EXEMPT). This field shall be mandatory for all orders (client/participant/PRO orders).
SellPAN	This field shall contain the PAN (Permanent Account Number/PAN_EXEMPT). This field shall be mandatory for all orders (client/participant/PRO orders).

Trade Modification Confirmation Response

This message is sent when trade modification is confirmed by exchange trading system and corresponding new trade data is sent.

MS_TRADE_CONFIRM (Refer to *Trade Confirmation* discussed earlier in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_MODIFY_CONFIRM (2287).
LogTime (of MESSAGE_HEADER)	This will contain the activity Time i.e., the latest modified time.

Trade Modification Error

If trade modification request is rejected due to erroneous data, then the structure sent is:

MS_TRADER_INT_MSG (Refer to *Interactive/Broadcast Messages Sent from Control* discussed later in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is CTRL_MSG_TO_TRADER (5295).
ErrorCode	Refer to List of Error Codes in Appendix.

Trade Cancellation

To cancel a trade, both the parties of the trade must request for trade cancellation. As soon as the request reaches the trading system, a requested message is sent. If any error is encountered in the entered data, Trade Error message is sent. Otherwise, it goes to the NSE-Control as an alert. The counter party to the trade is notified of the trade cancellation request (Refer to *Trade Cancel Requested Notification* in Chapter 5). When both the parties of the trade ask for trade cancellation, it may be approved or rejected by the Exchange (Refer to *Trade Cancellation Confirmation* in Chapter 5).

Trade Cancellation Request

The format of the message is as follows:

TRADE_INQUIRY_DATA (*Refer to Trade Modification Request in Chapter 4*)

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CANCEL_IN (5440).
FillNumber	This field should contain the trade number of the trade to be cancelled.

Trade Cancellation Requested Response

This is an acknowledgement signifying that the request has reached the trading system.

The following structure is sent:

TRADE INQUIRY DATA (Refer to *Trade Modification Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CANCEL_OUT (5441).

Trade Cancellation Error

After the requested response, if any error is detected in the data, the following structure is sent:

TRADE INQUIRY DATA (Refer to *Trade Modification Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_ERROR (2223)
ErrorCode	Refer to List of Error Codes in Appendix.

Chapter 5 Unsolicited Messages

Introduction

This section details the unsolicited messages that are received on the interactive connection. These messages are not received by the users in response to any request.

Please note this section is referenced in CM_DROP_COPY_PROTOCOL document. Any change here may also impact the Order Drop Copy functionality.

Cancellation of Orders in Batch

GTC\GTD orders which are valid till date, if not traded, are also removed from the book. A response for the same is sent to the user. As of now GTC and GTD facilities are not allowed hence there will be GTC and GTD orders. The structure sent is:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is BATCH_ORDER_CANCEL (9002).

Stop Loss Order Triggering

When any stop loss order entered is triggered, the user who entered the order is sent the following message:

TRADE CONFIRM (Refer to *Trade Confirmation* discussed later in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is ON_STOP_NOTIFICATION (2212).

Freeze Approve Response

This message is sent when a previous order, which resulted in freeze, is approved by the Exchange. The format of the message is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction codes are: If the entered order went for a freeze, and then got freeze approval, ORDER_CONFIRMATION (2073).

	If the modified order went for a freeze, and then got freeze approval, ORDER_MOD_CONFIRMATION (2074).
LastModifiedDateTime	This field contains the time when the order was last modified.
LastActivityReference	This field contains a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Freeze Reject Response

This message is sent when a previous order, which resulted in freeze, is rejected by the Exchange. The format of the message is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction codes are: If the entered order went for a freeze, then for freeze reject ORDER_ERROR_OUT (2231). If the modified order went for a freeze, then for freeze reject ORDER_MOD_REJECT_OUT (2042).

Trade Confirmation

Trade confirmation is an unsolicited message which is generated when any order of the trader is traded. The order may trade completely or partially. In Trade confirmation message, the ST_ORDER_FLAGS structure is modified, to identify Call Auction2 session trades. In this structure Preopen indicator is defined (which will be set to 1 for trades in Call Auction2 session), this is incorporated using an existing Filler bit, in the ST_ORDER_FLAGS structure as explained below:

Table 21 MS_TRADE_CONFIRM

Structure Name	MS_TRADE_CONFIRM		
Packet Length	228 bytes		
Transaction Code	TRADE_CONFIRMATION (2222)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
ResponseOrderNumber	DOUBLE	8	40
BrokerId	CHAR	5	48
Reserved	CHAR	1	53

Structure Name	MS_TRADE_CONFIRM		
Packet Length	228 bytes		
Transaction Code	TRADE_CONFIRMATION (2222)		
Field Name	Data Type	Size in Byte	Offset
TraderNum	LONG	4	54
AccountNum	CHAR	10	58
BuySell	SHORT	2	68
OriginalVol	LONG	4	70
DisclosedVol	LONG	4	74
RemainingVol	LONG	4	78
DisclosedVolRemaining	LONG	4	82
Price	LONG	4	86
ST_ORDER_FLAGS (Refer Table 19.1 for small endian machines and Table 19.2 for big endian machines)	STRUCT	2	90
Gtd	LONG	4	92
FillNumber	LONG	4	96
FillQty	LONG	4	100
FillPrice	LONG	4	104
VolFilledToday	LONG	4	108
ActivityType	CHAR	2	112
ActivityTime	LONG	4	114
OpOrderNumber	DOUBLE	8	118
OpBrokerId	CHAR	5	126
SEC_INFO (Refer Table 4)	STRUCT	12	131
Reserved	CHAR	1	143
BookType	SHORT	2	144
NewVolume	LONG	4	146
ProClient	SHORT	2	150
PAN	CHAR	10	152
Algo ID	LONG	4	162
Reserved Filler	SHORT	2	166
LastActivityReference	LONG LONG	8	168
Reserved	CHAR	52	176

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CONFIRMATION (2222).
ResponseOrderNumber	This field contains the order number of the trader's order taking part in the trade.
BrokerId	This field contains the Trading Member ID.
TraderNum	This field contains the trader's or user ID.
AccountNum	This field contains the Account Number or Client code.
BuySell	This field contains one of the following values based on Buy/Sell. '1' for Buy '2' for Sell.
OriginalVol	This field contains the Original traded volume.
DisclosedVol	This field contains the quantity to be disclosed to the market. It is not applicable if the order has either the All Or None or the Immediate Or Cancel attribute set. It should not be greater than the volume of the order and not less than the Minimum Fill quantity if the Minimum Fill attribute is set. In either case, it cannot be less than the Minimum Disclosed Quantity allowed. It should be a multiple of the Regular lot.
RemainingVol	This field contains the volume remaining after trade(s).
DisclosedVolRemaining	This field contains the disclosed volume remaining after trade(s).
Price	This field contains the order price.
OrderFlags	(Refer to *Order Entry Request* in Chapter 4) Note : Preopen Indicator will be set as 0 for the trades happening in Normal Market session for Normal Market orders and pre-open carried forward orders Preopen indicator will be set as 1 for trades happening in the call auction 2 market.
Gtd	This field contains the number of days for a GTD Order. This field can be set in two ways as given below. To specify an absolute date, set this field to that date in number of seconds since midnight of January 1, 1980. To specify days, set this to the number of days. This can take values from 2 to the maximum days specified for GTC orders only. If this field is non-zero, the GTC flag must be off.
FillNumber	This field contains the trade number.
FillQty	This field contains the traded volume.

Field Name	Brief Description
FillPrice	This field contains the price at which order is traded.
VolFilledToday	This field contains the quantity traded today.
ActivityType	This field contains the activity type. 'B' for Buy 'S' for Sell
ActivityTime	This field contains the time when the activity took place.
OpOrderNumber	This field will always be blank.
OpBrokerId	This field will always be blank.
SEC_INFO	This field contains the Symbol and Series of the security.
BookType	This field contains the book type - RL/ ST/ SL/ OL/ SP/ AU/CA/CB.
NewVolume	This field is always set to zero for trade confirmation.
ProCli	This field is same as Pro/Client /WHS indicator having one of the following values: '1' - client's order '2' - broker's order '4' - warehousing order
PAN	This field contains the PAN
Algo ID	This field shall contain the Algo ID
Reserved Filler	This field is reserved for future use
LastActivityReference	This field contains a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Preopen

Preopen Indicator will be set as 0 for the trades happening in Normal Market session for Normal Market orders and carried forward orders.

Preopen Indicator will be set as 1 for the Preopen Trades happening in the Opening Phase.
Note: All trades for CALL AUCTION 2 market will be sent with Book type Regular Lot (1).

Trade Cancellation

Trade Cancellation Requested Notification

This message is sent when the counter party of the trade requests a trade cancellation. The structure sent is:

MS_TRADER_INT_MSG (Refer to *Interactive/Broadcast Messages Sent from Control* discussed later in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is CTRL_MSG_TO_TRADER (5295).

Trade Cancellation Confirmation Response

When NSE-Control approves the trade cancellation request the structure sent is:

TRADE CONFIRM (Refer to *Trade Confirmation* discussed earlier in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CANCEL_CONFIRM (2282).

Trade Cancellation Rejection

When NSE-Control rejects the trade cancellation alert the structure sent is:

TRADE CONFIRM (Refer to *Trade Confirmation* discussed earlier in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CANCEL_REJECT (2286).

Note: Trade cancellation will not be allowed for Preopen trades, it will be rejected from Exchange. Refer to the List of error codes:
Trade cancellation will not be allowed for Call auction 2 market trades, it will be rejected from Exchange. Refer to the List of error codes:

Interactive/Broadcast Messages Sent from Control

A message can be sent to the trader(s) from the NSE-Control Work Station. If it is sent to all the traders, it comes as a broadcast in the structure BROADCAST_MESSAGE. (Refer to *General Message Broadcast* in Chapter 7)

When the message is sent to a particular user, it comes as an interactive message in the following structure:

Table 22 MS_TRADER_INT_MSG

Structure Name	MS_TRADER_INT_MSG
Packet Length	290 bytes

Transaction Code	CTRL_MSG_TO_TRADER (5295)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(Refer Table 1)	STRUCT	40	0
TraderId	LONG	4	40
ActionCode	CHAR	3	44
Reserved	CHAR	1	47
MsgLength	SHORT	2	48
Msg	CHAR	240	50

Field Name	Brief Description
TransactionCode	The transaction codes are: CTRL_MSG_TO_TRADER (5295) for interactive messages
ActionCode	This field contains the action code to indicate the action taken. For example, 'SYS' - System 'AUI' - Auction Initiation 'AUC' - Auction Complete 'LIS' – Listing

Table 23 MS_TRADER_INT_MSG

Structure Name	MS_TRADER_INT_MSG		
Packet Length	298 bytes		
Transaction Code	BCAST_JRNL_VCT_MSG (6501)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
BranchNumber	SHORT	2	40
BrokerNumber	CHAR	5	42
ActionCode	CHAR	3	47
Reserved	CHAR	4	50
BROADCAST DESTINATION (Refer Table 23.1)	STRUCT	2	54
MsgLength	SHORT	2	56
Msg	CHAR	240	58

Table 23.1 BROADCAST DESTINATION

Structure Name	BROADCAST DESTINATION		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	BIT	7	0
TraderWs	BIT	1	0
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	BCAST_JRNL_VCT_MSG (6501) for broadcasting messages.
ActionCode	This field contains the action code to indicate the action taken. For example, 'SYS' - System 'AUI' - Auction Initiation 'AUC' - Auction Complete 'LIS' – Listing 'MAR'- Margin violation messages

Chapter 6 Bhav Copy

Introduction

This section describes the end of the trading day activities. It covers the transmission of Security Bhav Copy and Index Bhav Copy. This takes place after the markets close for the day. Broadly, the following activities are done:

- Calculation of closing price and generation of interim bhav copy (from 3.30 PM to 3. 40 PM).
- Generation of main bhav-copy will be after 4.00 PM.

Closing Batch: In closing batch, the closing price is calculated and broadcast to the traders. The interim bhav copy is also broadcast to the traders. During *closing session* traders can trade at the closing price.

Closing Session: After closing batch, the market is open for trading for 20 mins. This period is known as **Closing Session**. Traders can place orders at market price (closing price) only. Some of error codes have been introduced for closing session. Refer List of Error Codes for the same.

Security Bhav Copy

Message Stating the Transmission of Security Bhav Copy Will Start Now

This is the first message which is broadcasted saying that the bhav copy will be started now. The structure sent is:

BROADCAST MESSAGE (Refer to *General Message Broadcast* in Chapter 7)

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_JRNL_VCT_MSG (6501). Message: Security Bhav Copy is being broadcast now.

Header of Report on Market Statistics

A header precedes the actual bhav copy that is sent to the trader. The message structure sent is:

Table 24 MS_RP_HDR

Structure Name	MS_RP_HDR		
Packet Length	106 bytes		
Transaction Code	MARKET_STATS_REPORT_DATA (18201)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
MsgType	CHAR	1	40
ReportDate	LONG	4	41
UserType	SHORT	2	45
BrokerId	CHAR	5	47
BrokerName	CHAR	25	52
TraderNumber	SHORT	2	77
TraderName	CHAR	26	79

Field Name	Brief Description
TransactionCode	The transaction code is MARKET_STATS_REPORT_DATA (18201).
MsgType	This field is set to 'H' denoting Header
ReportDate	This field is set to the report date.
UserType	This field contains the type of user. This is set to '-1'.
BrokerId	This field contains Trading Member ID. This is set to blanks.
BrokerName	This field contains the name of the broker. This is set to blanks.
TraderNumber	This field contains the trader/user ID. This is set to zero.
TraderName	This field contains the name of the trader. This is set to blanks.

Report on Market Statistics

This is the actual data that is sent for the report. The structure is as follows:

Table 25 REPORT MARKET STATISTICS

Structure Name	REPORT MARKET STATISTICS		
Packet Length	478 bytes		
Transaction Code	MARKET_STATS_REPORT_DATA (18201)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER(Refer Table 1)	STRUCT	40	0
MessageType	CHAR	1	40
Reserved	CHAR	1	41

Structure Name	REPORT MARKET STATISTICS		
Packet Length	478 bytes		
Transaction Code	MARKET_STATS_REPORT_DATA (18201)		
Field Name	Data Type	Size in Byte	Offset
NumberOfRecords	SHORT	2	42
MARKET STATISTICS DATA (Refer Table 25.1)	STRUCT	434	44

Table 25.1 MARKET STATISTICS DATA

Structure Name	MARKET STATISTICS DATA		
Packet Length	62 bytes		
Field Name	Data Type	Size in Byte	Offset
SEC_INFO (Refer Table 4)	STRUCT	12	0
MarketType	SHORT	2	12
OpenPrice	LONG	4	14
HighPrice	LONG	4	18
LowPrice	LONG	4	22
ClosingPrice	LONG	4	26
TotalQuantityTraded	LONG LONG	8	30
TotalValueTraded	DOUBLE	8	38
PreviousClosePrice	LONG	4	46
FiftyTwoWeekHigh	LONG	4	50
FiftyTwoWeekLow	LONG	4	54
CorporateActionIndicator	CHAR	4	58

Field Name	Brief Description
TransactionCode	The transaction code is MARKET_STATS_REPORT_DATA (18201).
MessageType	This field is set to ‘R’ denoting Report Data.
NumberOfRecords	This field contains the number of markets for which Market Statistics is being sent. In a packet at most 7 records can be packed.
Symbol	This field contains the Symbol of the security.
Series	This field contains the series of a security.

Field Name	Brief Description
MarketType	This field contains one of the following values indicating the market type as: • '1' – Normal • '2' – Odd lot • '3' – Spot • '4' – Auction • '5' – Call Auction1 • '6' – Call Auction2 In Bhavcopy, the Market Type of Security Participating in CALL AUCTION2 will come, under Normal Market '.
OpenPrice	This field contains the open price of a security.
HighPrice	This field contains the highest trade price.
LowPrice	This field contains the lowest trade price.
ClosingPrice	This field contains the closing price of a security.
TotalQuantityTraded	This field contains the total quantity of the security that is traded today.
TotalValueTraded	This field contains the total value of the securities traded.
PreviousClosePrice	This field contains the previous day's closing price of the security.
FiftyTwoWeekHighPrice	This field contains the highest trade price in a security in the immediately previous 52 weeks.
FiftyTwoWeekLowPrice	This field contains the lowest trade price in a security in the immediately previous 52 weeks.
CorporateActionIndicator	This field contains the Corporate Action. The EGM, AGM, Interest, Bonus, Rights and Dividend flags are set depending on the corporate action.

Packet Indicating Data for Depository Securities Begins

This message indicates that hereafter the bhav copy for depository securities will be broadcast. The structure sent is:

REPORT MARKET STATISTICS (Refer to *Report on Market Statistics* discussed earlier in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is MARKET_STATS_REPORT_DATA (18201).
MessageType	This field is set to 'D' denoting Data.

Data for Depository Securities

This is same as the data packet for non-Depository securities. The structure sent is:

REPORT MARKET STATISTICS (Refer to *Report on Market Statistics* discussed earlier in this chapter)

Field Name	Brief Description
TransactionCode	The transaction code is MARKET_STATS_REPORT_DATA (18201).

Trailer Record

This indicates that the transmission of bhav copy ends here. The structure is:

Table 26 REPORT TRAILER

Structure Name	REPORT TRAILER		
Packet Length	46 bytes		
Transaction Code	MARKET_STATS_REPORT_DATA (18201)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(Refer Table 1)	STRUCT	40	0
MessageType	CHAR	1	40
NumberOfRecords	LONG	4	41
Reserved	CHAR	1	45

Field Name	Brief Description
TransactionCode	The transaction code is MARKET_STATS_REPORT_DATA (18201).
MessageType	This field is set as 'T' for trailer record.
NumberOfRecords	This field contains the number of data packets sent in the bhav copy.

Index Bhav Copy

Message Stating the Transmission of the Index Bhav Copy Will Start Now

This is the first message which is broadcast saying the bhav copy will start now. The structure sent is:

BROADCAST MESSAGE (Refer to *General Message Broadcast* in Chapter 7)

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_JRNL_VCT_MSG (6501). Message: Index Bhav Copy is being broadcast now.

Header of Report on Market Statistics

Refer to *Header of Report on Market Statistics (Security Bhav Copy)* discussed earlier in this chapter.

Field Name	Brief Description
TransactionCode	The transaction code is MKT_IDX_RPT_DATA (1836).

Report on Index

This is the actual data that is sent for index data. The structure is as follows:

Table 27 MS_RP_MARKET_INDEX

Structure Name	MS_RP_MARKET_INDEX		
Packet Length	464 bytes		
Transaction Code	MKT_IDX_RPT_DATA (1836)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(Refer Table 1)	STRUCT	40	0
MsgType	CHAR	1	40
Reserved	CHAR	1	41
NoOfIndexRecs	SHORT	2	42
MKT_INDEX [7] (Refer Table 27.1)	STRUCT	420	44

Table 27.1 MKT_INDEX

Structure Name	MKT_INDEX

Packet Length	60 bytes		
Field Name	Data Type	Size in Byte	Offset
IndName	CHAR	24	0
MktIndexPrevClose	LONG	4	24
MktIndexOpening	LONG	4	28
MktIndexHigh	LONG	4	32
MktIndexLow	LONG	4	36
MktIndexClosing	LONG	4	40
MktIndexPercent	LONG	4	44
MktIndexYrHi	LONG	4	48
MktIndexYrLo	LONG	4	52
MktIndexStart	LONG	4	56

Field Name	Brief Description
TransactionCode	The transaction code is MKT_IDX_RPT_DATA (1836).
MsgType	This field is set to 'R' denoting Report for Index Data.
NoOfIndexRecs	This field contains the number of index records in the packet.
IndName	This field contains the name of the index being broadcast. For example, CNX
MktIndexPrevClose	This field contains the previous day's closing index.
MktIndexOpening	This field contains today's opening index.
MktIndexHigh	This field contains today's high index.
MktIndexLow	This field contains today's low index.
MktIndexClosing	This field contains today's closing index.
MktIndexPercent	This field contains % change today.
MktIndexYrHi	This field contains 52-week high index.
MktIndexYrLo	This field contains 52-week low index.

Trailer of Index Data Broadcast

Refer to *Trailer Record of Security Bhav Copy* discussed earlier in this chapter.

Chapter 7 Broadcast

Introduction

This section describes the Compression and Decompression algorithm of Broadcast data and the various Broadcast messages with their structures.

Compression of the Broadcast Data

The broadcast traffic from the exchange which gives the on-line quotes to the trading terminals has been continually increasing, especially during market open and market close. To accommodate the increased broadcast traffic, the exchange has come up with a compression algorithm to compress some of the specific broadcast transaction codes, which are as follows:

Transaction Code	Represents
7201	Mkt Watch
18703	Ticker
7208	Only MBP
7214	Call Auction MBP
7215	BROADCAST CALL AUCTION MARKET WATCH
7210	Order Cancel Update

LZO compression algorithm is used to compress the above specified broadcast transaction codes. The details of the LZO compression algorithm are described below.
The LZO stands for Lempel Ziv Oberhaumer. This algorithm is freely available on the internet (URL: http://www.oberhumer.com/opensource/lzo). It is made available by free software foundation. The algorithm is tested on various operating systems like UNIX and red hat Linux.

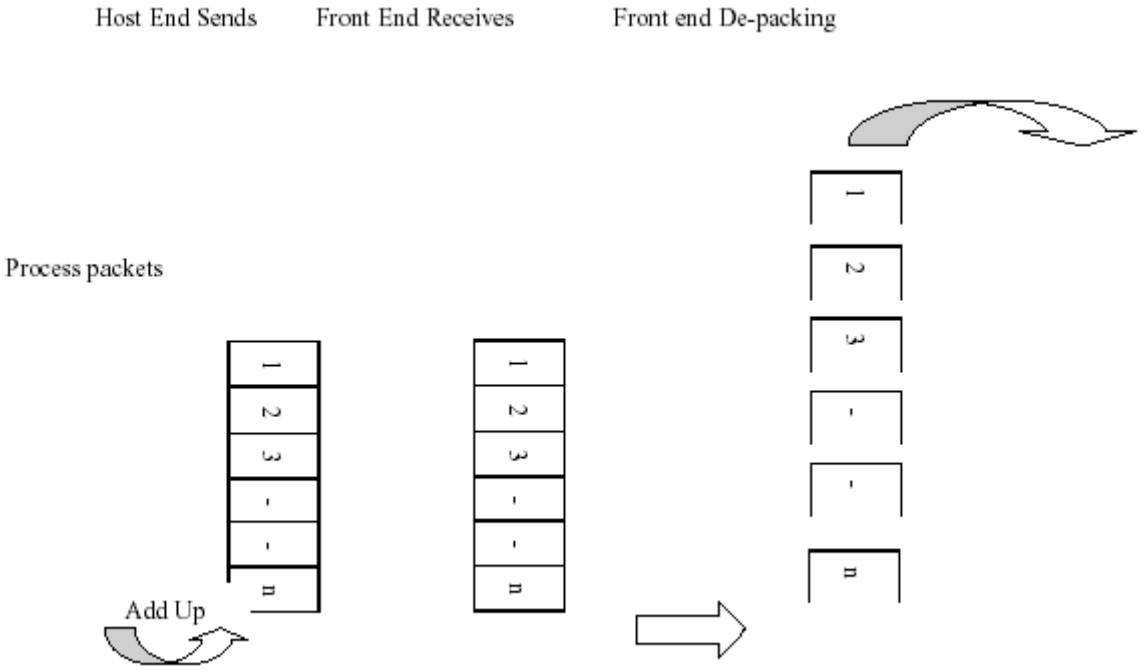
Decompression Routine

Sequential Packing

To improve the effective data transfer, the idea of sequential packing along with the lzo compression algorithm has been incorporated. At the host end, sequential packing algorithm packs the incoming data packets, which is then transmitted over the network. The data packets are packed in FIFO order.
For example,
If ‘n’ packets are packed in a buffer, they are arranged in the following order:

1st packet will be stored at the first place in the buffer, 2nd Packet will be stored at the second place, and so on.

At the front end while de packing the buffer, the packets are to be segregated in the same order, that is, isolate each packet and process each packet as per the sequence viz- first packet first and last packet at the end. The packets within a buffer may be an admixture of compressed and uncompressed data packets.



Calling Convention

The decompression routine is a C-callable routine with the following prototype:

```
Void Sigdec2 (char *ip,
              unsigned short  *ipL,
              char *op,
              unsigned short *opL,
              unsigned short *errorcode);
```

Parameters

Ip: it is the pointer to the input buffer.

IpL: It is the pointer to a short containing the length of input.

Op: it is the pointer of the output buffer.

OpL: It is the pointer to a short containing the length of output.

ErrorCode: it is the pointer to a short containing the error code.

Packet Format

Incoming packet at the front end can be interpreted by mapping onto the following structure.

```
Struct {
        CHAR   cNetId [2]
        SHORT iNoPackets
        CHAR   cPackData [512]
    }   BcastPackData
```

where,

cNetId[2]	Identifies the machine (CM broadcast or F&O Broadcast)
	Please find different values of CNetId for difference segments
	Equity: - 4
	Equity Derivative: - 2
	Currency Derivative: - 6
iNoPackets	The number of packets that are sequentially packed
cPackData	Buffer containing all the packets.

The buffer when mapped to, by the above structure, the number of packets in the buffer can be known. The next task is to segregate the packets and process the individual packets.
The packets received through the broadcast traffic have to be interpreted as follows

```
COMPRESSION_BROADCAST_DATA
{
        SHORT       CompressionLen
        CHAR        BroadcastData [ ]
}
```

Note:

- The first two bytes of the broadcast packet indicate the length of the data after compression.
- If the compression length is zero, the data received is not compressed.
- If the length is non-zero, the data following the length should be decompressed by using the decompression routine.
- Inside the broadcast data, the first 8 bytes before the message header should be ignored. The message header starts from the 9th byte.

Implementation at Front End

The lzo directory (lzo1.07) contains all the lzo source, header and library files.
These files are to be included while building an application.

lzo1z_decompress is used for decompression. This is a function of the lzo library.
An API has to be developed to encompass the above LZO decompression function.

The syntax of the call should be:
lzo_decomp (char* inp_buff, unsigned int* inp_len, char* buffer_decomp,
unsigned int *output_len, unsigned short *errorCode)
Where, lzo_decomp is a function of the API (to be developed by referring to the examples specified in the lzo 1.07 directory) that calls the lzo function for decompression "lzo1z_decompress"

Inp_buff	Specifies the input buffer (Compressed Buffer)
Inp_len	Specifies the length of input buffer (Compressed Length)
Buffer_decomp	Specifies the Buffer after decompression
output_len	Specifies the length after decompression (Output length)
errorCode	Specifies the error code

The syntax of the lzo decompress function is as follows:

lzo1z_decompress (out, decomp_inlen, in, & decomp_outlen, NULL)

Where

out	Specifies input compressed buffer
decomp_inlen	Specifies the input length of the buffer (Length of Compressed buffer)
in	Specifies the output (decompressed) buffer
decomp_outlen	Specifies the output length of the decompressed buffer

Note:
Inside the broadcast data, the first byte indicates the market type. Ignore the rest of the 7 bytes before message header. If the first byte has the value of '4', it is Capital market and if it is '2' then it is futures and options market.

The message header starts from 9th byte.

General Message Broadcast

Any general message is broadcasted in the following structure. The structure sent is:

Table 28 BROADCAST_MESSAGE

Structure Name	BROADCAST MESSAGE		
Packet Length	298 bytes		
Transaction Code	BCAST_JRNL_VCT_MSG (6501)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
BranchNumber	SHORT	2	40
BrokerNumber	CHAR	5	42
ActionCode	CHAR	3	47
Reserved	CHAR	4	50
BROADCAST DESTINATION (Refer Table No. 28.1 for small endian & Table No. 28.2 for big endian)	STRUCT	2	54
BroadcastMessageLength	SHORT	2	56
BroadcastMessage	CHAR	240	58

Note: Use any one-off following two BROADCAST DESTINATION structures:

Table 28.1 BROADCAST_DESTINATION (For Small Endian Machines)

Structure Name	BROADCAST DESTINATION		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	7	0
TraderWs	BIT	1	0
Reserved	CHAR	1	1

Table 28.2 BROADCAST_DESTINATION (For Big Endian Machines)

Structure Name	BROADCAST DESTINATION		
Packet Length	**2 bytes**		
Field Name	**Data Type**	**Size**	**Offset**
TraderWs	BIT	1	0

Structure Name	BROADCAST DESTINATION		
Packet Length	**2 bytes**		
Field Name	**Data Type**	**Size**	**Offset**
Reserved	BIT	7	0
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_JRNL_VCT_MSG (6501).
BranchNumber	This field contains the branch number of the trader or broker.
BrokerNumber	This field contains the Trading Member ID of the broker.
ActionCode	This field Indicates the action taken.
BroadcastDestination	This field contains the destination of the message, that is, Trader Workstation or Control Workstation.
BroadcastMessageLength	This field contains the length of the broadcast message.
BroadcastMessage	This field contains the broadcast message.

Change in System Status / Parameters

This message is sent when any global operating parameters are changed or status of markets is changed. The structure of the message is:

SYSTEM INFORMATION DATA (Refer to *System Information Response* in Chapter 3)

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SYSTEM_INFORMATION_OUT (7206) No of machines received in the alphachar field is 0 not the actual no of machines.

Change in Security Master

This is sent whenever the parameter of any security is changed. The structure is given below.

Table 29 SECURITY UPDATE INFORMATION

Structure Name	SECURITY UPDATE INFORMATION		
Packet Length	260 bytes		
Transaction Code	BCAST_SECURITY_MSTR_CHG (18720)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
Token	LONG	4	40
SEC_INFO (Refer Table 4)	STRUCT	12	44
InstrumentType	SHORT	2	56
PermittedToTrade	SHORT	2	58
IssuedCapital	DOUBLE	8	60
SettlementType	SHORT	2	68
FreezePercent	SHORT	2	70
CreditRating	CHAR	19	72
Reserved	CHAR	1	91
SECURITY ELIGIBILITY PER MARKET [6] (refer table 29.1 for small endian & table 29.2 for big endian)	STRUCT	24	92
SurvInd	SHORT	2	116
IssueStartDate	LONG	4	118
InterestPaymentDate	LONG	4	122
IssueMaturityDate	LONG	4	126
BoardLotQuantity	LONG	4	130
TickSize	LONG	4	134
Name	CHAR	25	138
Reserved	CHAR	1	163
ListingDate	LONG	4	164
ExpulsionDate	LONG	4	168
ReAdmissionDate	LONG	4	172
RecordDate	LONG	4	176
ExpiryDate	LONG	4	180
NoDeliveryStartDate	LONG	4	184

Structure Name	SECURITY UPDATE INFORMATION		
Packet Length	260 bytes		
Transaction Code	BCAST_SECURITY_MSTR_CHG (18720)		
Field Name	Data Type	Size in Byte	Offset
NoDeliveryEndDate	LONG	4	188
ELIGIBLITY INDICATORS (refer table 29.3 for small endian & table 29.4 for big endian)	STRUCT	2	192
BookClosureStartDate	LONG	4	194
BookClosureEndDate	LONG	4	198
PURPOSE structures (refer table 29.5 for small endian & table 29.6 for big endian)	STRUCT	2	202
LocalUpdateDateTime	LONG	4	204
DeleteFlag	CHAR	1	208
Remark	CHAR	25	209
FaceValue	LONG	4	234
ISINNumber	CHAR	12	238
MktMakerSpread	LONG	4	250
MktMakerMinQty	LONG	4	254
CallAuction1Flag	SHORT	2	258

Note: Use any one-off following two SECURITY ELIGIBILITY PER MARKET structures:

Table 29.1 SECUIRITY ELIGIBILITY PER MARKET (For Small Endian Machines)

Structure Name	SECUIRITY ELIGIBILITY PER MARKET		
Packet Length	4 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	7	0
Eligibility	BIT	1	0
Reserved	CHAR	1	1
Status	SHORT	2	2

Table 29.2 SECUIRITY ELIGIBILITY PER MARKET (For Big Endian Machines)

Structure Name	SECUIRITY ELIGIBILITY PER MARKET		
Packet Length	**4 bytes**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
Eligibility	BIT	1	0
Reserved	BIT	7	0
Reserved	CHAR	1	1
Status	SHORT	2	2

Table 29.3 ELIGIBLITY INDICATORS (For Small Endian Machines)

Structure Name	ELIGIBLITY INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	5	0
MinimumFill	BIT	1	0
AON	BIT	1	0
ParticipateInMarketIndex	BIT	1	0
Reserved	CHAR	1	1

Table 29.4 ELIGIBLITY INDICATORS (For Big Endian Machines)

Structure Name	ELIGIBLITY INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
ParticipateInMarketIndex	BIT	1	0
AON	BIT	1	0
MinimumFill	BIT	1	0
Reserved	BIT	5	0
Reserved	CHAR	1	1

Table 29.5 PURPOSE (For Small Endian Machines)

Structure Name	PURPOSE		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	BIT	2	0
EGM	BIT	1	0
AGM	BIT	1	0
Interest	BIT	1	0
Bonus	BIT	1	0
Rights	BIT	1	0
Dividend	BIT	1	0
Reserved	CHAR	1	1

Table 29.6 PURPOSE (For Big Endian Machines)

Structure Name	PURPOSE		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Dividend	BIT	1	0
Rights	BIT	1	0
Bonus	BIT	1	0
Interest	BIT	1	0
AGM	BIT	1	0
EGM	BIT	1	0
Reserved	BIT	2	0
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SECURITY_MSTR_CHG (18720).
Token	This field contains the token number of the security being updated. This is unique for a particular symbol-series combination.

Field Name	Brief Description
SecurityInformation	This field contains the Symbol and Series (EQ / IL / TT) of the security.
InstrumentType	This field contains the instrument type of the security. It can be one of the following: '0' – Equities '1' – Preference Shares '2' – Debentures '3' – Warrants '4' – Miscellaneous
PermittedToTrade	This field contains one of the following values: '0' – Listed but not permitted to trade '1' – Permitted to trade '2' - BSE listed (BSE exclusive security will be available, however trading on the same will be allowed only in case of outage at BSE)
IssuedCapital	This field contains issue size of the security.
SettlementType	This field contains the settlement type. It can be one of the following: '0' – T+0 settlement '1' – T+1 settlement
FreezePercent	This field contains the volume freeze percent w.r.t.issued capital. This field indicates the volume freeze percentage w.r.t. issued capital. This field has to be interpreted as freeze percent /10000. Eg: 41 in this field has to be interpreted as 0.0041 %
CreditRating	This field contains daily price range of the security.
Eligibility	The flag is set to '1' if the security is allowed to trade in a particular market. For Call Auction2 market (6th Market), eligibility will be set.
Status	This field contains one of the following values: '1' - Preopen (Only for Normal Market) '2' - Open '3' - Suspended

Field Name	Brief Description
	'4' - Preopen extended '6' – Price Discovery
SurvInd	Indicator for security in Surveillance Measure
IssueStartDate	This field contains the date of issue of the security.
InterestPaymentDate	This field contains the interest payment date of the issue.
IssueMaturityDate	This field contains the maturity date.
BoardLotQuantity	This field contains the Regular lot size.
TickSize	This field contains the Tick size/ Min spread size.
Name	This field contains the security name.
ListingDate	This field contains the date of listing.
ExpulsionDate	This field contains the date of expulsion.
ReAdmissionDate	This field contains the date of readmission.
RecordDate	This field contains the date of record changed.
ExpiryDate	This field contains the last date of trading before any corporate action.
NoDeliveryStartDate	This field contains the date from when physical delivery of share certificates is stopped for book closure.
NoDeliveryEndDate	This field contains the date from when physical delivery of share certificates starts after book closure.
MinimumFill	This flag is set if Minimum Fill attribute is allowed in orders of this security.
AON	This flag is set if AON attribute is allowed in orders of this security.
ParticipateInMarketIndex	This flag is set if this security participates in the market index.
BookClosureStartDate	This field contains the date when the record books in the company for shareholder names starts.
BookClosureEndDate	This field contains the date when the record books in the company for shareholder names ends.
Purpose	This field contains the EGM / AGM / Interest / Bonus / Rights / Dividend flags set depending on the corporate action.
LocalUpdateDateTime	This field contains the local database update date and time.

Field Name	Brief Description
DeleteFlag	This field contains the status of the security, that is, whether the security is deleted or not.
Remark	This field contains remarks.
FaceValue	This field contains face value of the security.
ISIN Number	This field contains ISIN number of the security.
MktMakerSpread	This field contains spread value of the security, used by Market maker user to place two-way quotes.
MktMakerMinQty	This field contains the Minimum quantity for the security, Used by Market maker user for market maker order.

Change Participant Status

This message is sent whenever there is any participant change. The structure sent is:

Table 30 Change Participant Status

Structure Name	PARTICIPANT UPDATE INFO		
Packet Length	84 bytes		
Transaction Code	BCAST_PART_MSTR_CHG (7306)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
ParticipantId	CHAR	12	40
ParticipantName	CHAR	25	52
ParticipantStatus	CHAR	1	77
ParticipantUpdateDateTime	LONG	4	78
DeleteFlag	CHAR	1	82
Reserved	CHAR	1	83

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_PART_MSTR_CHG (7306).
ParticipantId	This field contains the Participant ID.
ParticipantName	This field contains the name of the participant that is changed.
ParticipantStatus	This field contains the status of the participant which is changed:

Field Name	Brief Description
	'S' for Suspended 'A' for Active
ParticipantUpdateDateTime	This field contains the time when the participant information was changed. It is in number of seconds from January 1, 1980.

Change of Security Status

This message is sent whenever the status of any security changes. The structure sent is:

Table 31 Change of Security Status

Structure Name	SECURITY STATUS UPDATE INFORMATION		
Packet Length	442 bytes		
Transaction Code	BCAST_SECURITY_STATUS_CHG (18130) OR BCAST_SECURITY_STATUS_CHG_PREOPEN (18707)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
TOKEN AND ELIGIBILITY [25] (Refer table 31.1)	STRUCT	400	42

Table 31.1 TOKEN AND ELIGIBILITY

Structure Name	TOKEN AND ELIGIBILITY		
Packet Length	16 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
SECURITY STATUS PER MARKET[6] (Refer table 31.2)	STRUCT	12	4

Table 31.2 SECURITY STATUS PER MARKET

Structure Name	SECURITY STATUS PER MARKET		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Status	Short	2	0

Field Name	Brief Description
TransactionCode	The transaction code is: When the status of the security changes BCAST_SECURITY_STATUS_CHG (18130). BCAST_SECURITY_STATUS_CHG_PREOPEN (18707).
NumberOfRecords	This field contains the number of records of the structure TOKEN AND ELIGIBILITY.
Token	This field contains the token number of the security which has been changed.
Status	This field contains the new status of the security. This can take one of the following values: '1' - Preopen '2' - Open '3' - Suspended '4' - Preopen extended '6' – Price Discovery This will include Call Auction2 Market data at the 6th position.

Turnover Limit Exceeded or Broker Reactivated

When a broker's turnover limit exceeds, the broker is deactivated and a message is broadcast to all workstations. The same structure is also sent when any broker is reactivated. The structure is:

Table 32 Turnover Limit Exceeded or Broker Reactivated

Structure Name	BROADCAST LIMIT EXCEEDED		
Packet Length	77 bytes		
Transaction Code	BCAST_TURNOVER_EXCEEDED (9010) OR BROADCAST_BROKER_REACTIVATED (9011)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
BrokerCode	CHAR	5	40

Structure Name	BROADCAST LIMIT EXCEEDED		
Packet Length	77 bytes		
Transaction Code	BCAST_TURNOVER_EXCEEDED (9010) OR BROADCAST_BROKER_REACTIVATED (9011)		
Field Name	Data Type	Size in Byte	Offset
CounterBroker Code	CHAR	5	45
WarningType	SHORT	2	50
SEC_INFO (Refer Table 4)	STRUCT	12	52
TradeNumber	LONG	4	64
TradePrice	LONG	4	68
TradeVolume	LONG	4	72
Final	CHAR	1	76

Field Name	Brief Description
TransactionCode	The transaction code is: BCAST_TURNOVER_EXCEEDED (9010), if the broker turnover is about to exceed or has already exceeded. BROADCAST_BROKER_REACTIVATED (9011), if the broker is reactivated after being deactivated.
BrokerCode	This field contains the Broker code who is about to exceed or has already exceeded his turnover limit. If the transaction code is BROADCAST_BROKER_REACTIVATED, then this broker is reactivated.
CounterBrokerCode	This field is not in use.
WarningType	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. The value is '1' if the turnover limit is about to exceed, '2' if turnover limit is exceeded. In the latter case the broker is deactivated.
Symbol	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This contains the symbol of the security in which the broker has last traded.

Field Name	Brief Description
Series	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This contains the series of the security.
TradeNumber	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This is the trade number in which the broker has last traded
TradePrice	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This contains the price of the trade.
TradeVolume	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This contains the trade quantity of the trade.
Final	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This indicates whether it is the final auction trade.

Auction Activity Message

This structure is sent whenever there is any auction related activity. This includes any change in Auction MBO. The structure is:

Table 33 Auction Activity Message

Structure Name	MS_AUCTION_INQ_DATA		
Packet Length	76 bytes		
Transaction Code	BCAST_AUCTION_INQUIRY_OUT (18700).		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
ST_AUCTION_INQ_INFO (Refer Table 33.1)	STRUCT	36	40

Table 33.1 Auction Activity Message

Structure Name	ST_AUCTION_INQ_INFO		
Packet Length	36 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
AuctionNumber	SHORT	2	4
AuctionStatus	SHORT	2	6
InitiatorType	SHORT	2	8
TotalBuyQty	LONG	4	10
BestBuyPrice	LONG	4	14
TotalSellQty	LONG	4	18
BestSellPrice	LONG	4	22
AuctionPrice	LONG	4	26
AuctionQty	LONG	4	30
SettlementPeriod	SHORT	2	34

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_AUCTION_INQUIRY_OUT (18700).
Token	This field contains the token number of the security in which the auction is started.
AuctionNumber	This field contains the number of the auction.
AuctionStatus	Refer to Auction Status in Appendix.
InitiatorType	This field specifies whether auction is initiated by trader or control. This field is set to control since only Exchange initiated auctions are permitted now.
TotalBuyQty	This field contains the total Buy Quantity for the auction.
BestBuyPrice	This field contains the best Buy price. This is the highest price for a Buy auction.
TotalSellQty	This field contains the total Sell quantity for the auction.
BestSellPrice	This field contains the best Sell price. This is the lowest price for a Sell auction.
AuctionPrice	This field contains the price at which auction trade has taken place.
AuctionQty	This field contains the quantity of securities that have been auctioned.

Field Name	Brief Description
SettlementPeriod	This field contains the period by which settlement between the parties should take place. This value is defaulted by the Exchange and cannot be modified by the user.

Change of Auction Status

When the status of an auction changes (from pending to active or, competitor period or solicitor period is ended or started) a message is broadcast to all workstations with the following structure and transaction codes:

Table 34 Change of Auction Status

Structure Name	AUCTION STATUS CHANGE		
Packet Length	302 bytes		
Transaction Code	BC_AUCTION_STATUS_CHANGE (6581)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
SEC_INFO	STRUCT	12	40
AuctionNumber	SHORT	2	52
AuctionStatus	CHAR	1	54
ActionCode	CHAR	3	55
BROADCAST_DESTINATION (Refer Table 34.1 for small endian & Table 34.2 for big endian)	STRUCT	2	58
BroadcastMessageLength	SHORT	2	60
BroadcastMessage	CHAR	240	62

Table 34.1 BROADCAST_DESTINATION (For Small Endian Machines)

Structure Name	BROADCAST DESTINATION		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	7	0
TraderWs	BIT	1	0
Reserved	CHAR	1	1

Table 34.2 BROADCAST_DESTINATION (For Big Endian Machines)

Structure Name	BROADCAST DESTINATION		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
TraderWs	BIT	1	0
Reserved	BIT	7	0
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	The transaction code is BC_AUCTION_STATUS_CHANGE (6581).
Symbol	This field contains the symbol of the security.
Series	This field contains the series of the security.
AuctionNumber	This field contains the auction number.
AuctionStatus	This field contains the status of the auction. Refer to Auction Status in Appendix.
ActionCode	This field contains the action code to indicate the action taken.
BroadcastDestination	This field contains the destination of the message.
BroadcastMessageLength	This field contains the length of the broadcast message.
BroadcastMessage	This field contains the contents of the broadcast message.

Change of Market Status

Whenever the status of the market changes, the following structure is sent:

Table 35 Change of Market Status

Structure Name	BCAST_VCT_MESSAGES		
Packet Length	298 bytes		
Transaction Code	BC_OPEN_MESSAGE (6511) OR BC_CLOSE_MESSAGE (6521) OR BC_PREOPEN_SHUTDOWN_MSG (6531) OR BC_NORMAL_MKT_PREOPEN_ENDED (6571) OR BC_CLOSING_START(6583) OR BC_CLOSING_END(6584)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
SEC_INFO(Refer Table 4)	STRUCT	12	40
MarketType	SHORT	2	52
BROADCAST_DESTINATION (Refer Table 34.1 for small endian & Table 34.2 for big endian)	STRUCT	2	54
BroadcastMessageLength	SHORT	2	56
BroadcastMessage	CHAR	240	58

Field Name	Brief Description
TransactionCode	The transaction codes are as follows: BC_OPEN_MESSAGE (6511). This is sent when the market is opened. BC_CLOSE_MESSAGE (6521). This is sent when the market is closed. BC_PREOPEN_SHUTDOWN_MSG (6531). This is sent when the market is preopened. BC_NORMAL_MKT_PREOPEN_ENDED (6571). This is sent when the preopen period ends.

Field Name	Brief Description
	BC_CLOSING_START (6583). This is sent when closing session is opened. BC_CLOSING_END (6584). This is sent when closing session is closed.
SecurityInformation	This field contains the symbol and series of a security.
MarketType	This field indicates the type of market. It contains one of the following values: '1' - Normal '2' - Odd Lot '3' - Spot '4' – Auction '5' – Call auction1 '6' – Call auction2
BroadcastDestination	This field is set to '1' if it signifies that the message is for the Trader Workstation.
BroadcastMessageLength	This field contains the length of the broadcast message.
BroadcastMessage	This field contains the contents of the broadcast message.

In case of security level trading/Market status change following separate broadcast messages will be sent to trader.
BCAST_JRNL_VCT_MSG (6501) refer Table 22.
BC_SYMBOL_STATUS_CHANGE_ACTION (7764).

Security Level Trading/Market Status Change Message

Security level trading/market status change messages are sent separately in following structure and transcode.

SECURITY LEVEL TRADING STATUS CHANGE

Structure Name	BCAST_SYMBOL_STATUS_CHANGE _ACTION		
Packet Length	58 bytes		
Transaction Code	BC_SYMBOL_STATUS_CHANGE_ACTION (7764)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 2)	STRUCT	40	0
SEC_INFO(Refer Table 3)	STRUCT	12	40
MarketType	SHORT	2	52
Reserved	SHORT	2	54
ActionCode	SHORT	2	56

Field Name	Brief Description
TransactionCode	The transaction code is BC_SYMBOL_STATUS_CHANGE_ACTION (7764)
SecurityInformation	This field contains the symbol and series of a security.
MarketType	This field indicates the type of market. It contains one of the following values: '1' - Normal '2' - Odd Lot '3' - Spot '4' – Auction '5' – Call auction1 '6' – Call auction2.
ActionCode	It contains of the following values: 6531(BC_PREOPEN_SHUTDOWN_MSG) - This action code is set when the security is preopened. 6571(BC_NORMAL_MKT_PREOPEN_ENDED) - This action code is set when the security's preopen period ends. 6511(BC_OPEN_MESSAGE) - This action code is set when the security is opened. 6521(BC_CLOSE_MESSAGE) - This action code is set when the security is closed. 6583(BC_CLOSING_START) - This action code is set when the security's closing session is opened. 6584(BC_CLOSING_END) - This action code is set when the security's closing session is closed

Ticker and Market Index

Ticker and market index information is sent in the following structure:

Table 36 Ticker and Market Index

Structure Name	TICKER TRADE DATA		
Packet Length	546 bytes		
Transaction Code	BCAST_TICKER_AND_MKT_INDEX (18703)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
TICKER INDEX INFORMATION [28] (Refer to TABLE 36.1)	STRUCT	504	42

Table 36.1 TICKER INDEX INFORMATION

Structure Name	TICKER INDEX INFORMATION		
Packet Length	18 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
MarketType	SHORT	2	4
FillPrice	LONG	4	6
FillVolume	LONG	4	10
MarketIndexValue	LONG	4	14

Field Name	Brief Description
TransactionCode	The transaction code sent is BCAST_TICKER_AND_MKT_INDEX (18703).
NumberOfRecords	This field indicates the number of times (Maximum 28) the structure TICKER INDEX INFORMATION is repeated.
Token	This field contains the token number—a unique number given to a particular symbol-series combination.
MarketType	This field contains the market type.
FillPrice	This field contains the price at which the order has been traded.

FillVolume	This field contains the quantity of security traded.
MarketIndexValue	This field contains the value of the market index.

Market by Order / Market by Price Update

The information regarding the best buy orders and the best sell orders is given in the following format:

Table 37 Market by Order / Market by Price Update

Structure Name	BROADCAST MBO MBP		
Packet Length	482 bytes		
Transaction Code	BCAST_MBO_MBP_UPDATE (7200)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
INTERACTIVE MBO DATA (Refer Table 37.1)	STRUCT	240	40
MBPBuffer [size of (MBP INFORMATION) * 10] (Refer MBP_INFORMATION in Table 37.7))	CHAR	160	280
BbTotalBuyFlag	SHORT	2	440
BbTotalSellFlag	SHORT	2	442
TotalBuyQuantity	LONG LONG	8	444
TotalSellQuantity	LONG LONG	8	452
MBO MBP INDICATOR (Refer Table 37.2 for Small Endian & Table 37.3 for Big Endian)	STRUCT	2	460
ClosingPrice	LONG	4	462
OpenPrice	LONG	4	466
HighPrice	LONG	4	470
LowPrice	LONG	4	474
Reserved	CHAR	4	478

Table 37.1 INTERACTIVE MBO DATA

Structure Name	INTERACTIVE MBO DATA		
Packet Length	240 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
BookType	SHORT	2	4
TradingStatus	SHORT	2	6
VolumeTradedToday	LONG LONG	8	8
LastTradedPrice	LONG	4	16
NetChangeIndicator	CHAR	1	20
Filler	CHAR	1	21
NetPriceChangeFromClosingPrice	LONG	4	22
LastTradeQuantity	LONG	4	26
LastTradeTime	LONG	4	30
AverageTradePrice	LONG	4	34
AuctionNumber	SHORT	2	38
AuctionStatus	SHORT	2	40
InitiatorType	SHORT	2	42
InitiatorPrice	LONG	4	44
InitiatorQuantity	LONG	4	48
AuctionPrice	LONG	4	52
AuctionQuantity	LONG	4	56
MBOBuffer [size of (MBO INFORMATION) * 10] (Refer MBO_INFORMATION in Table 37.4)	STRUCT	180	60

Table 37.2 MBO MBP INDICATOR (For Small Endian Machines)

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	4	0
Sell	BIT	1	0

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Buy	BIT	1	0
LastTradeLess	BIT	1	0
LastTradeMore	BIT	1	0
Reserved	CHAR	1	1

Table 37.3 MBO MBP INDICATOR (For Big Endian Machines)

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
LastTradeMore	BIT	1	0
LastTradeLess	BIT	1	0
Buy	BIT	1	0
Sell	BIT	1	0
Reserved	BIT	4	0
Reserved	CHAR	1	1

Table 37.4 MBO INFORMATION

Structure Name	MBO INFORMATION		
Packet Length	18 bytes		
Field Name	Data Type	Size in Byte	Offset
TraderId	LONG	4	0
Qty	LONG	4	4
Price	LONG	4	8
ST MBO MBP TERMS (Refer Table 37.5 for small endian & Table 37.6 for big endian)	STRUCT	2	12
MinFillQty	LONG	4	14

Table 37.5 ST MBO MBP TERMS (For Small Endian Machines)

Structure Name	ST MBO MBP TERMS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved1	BIT	6	0
Aon	BIT	1	0
Mf	BIT	1	0
Reserved2	BIT	8	1

Table 37.6 ST MBO MBP TERMS (For Big Endian Machines)

Structure Name	ST MBO MBP TERMS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Mf	BIT	1	0
Aon	BIT	1	0
Reserved1	BIT	6	0
Reserved2	BIT	8	1

Table 37.7 MBP INFORMATION

Structure Name	MBP INFORMATION		
Packet Length	16 bytes		
Field Name	Data Type	Size in Byte	Offset
Quantity	LONG LONG	8	0
Price	LONG	4	8
NumberOfOrders	SHORT	2	12
BbBuySellFlag	SHORT	2	14

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_MBO_MBP_UPDATE (7200).
Token	This field contains the token number—a unique number given to a particular symbol-series combination.

Field Name	Brief Description
BookType	This field contains the book type—RL / ST / SL / OL / SP / AU
TradingStatus	This field contains the trading status of the security: '1' - Preopen '2' - Open '3' - Suspended '4' - Preopen Extended '6' – Price Discovery
VolumeTradedToday	This field contains the total quantity of a security traded on the current day.
LastTradedPrice	This field contains the price at which the latest trade in a security has taken place.
NetChangeIndicator	This field is a flag which indicates any change of the order price from the LTP. '+' for increase '-' for decrease
NetPriceChange	This field contains the net change between the order price and the LTP.
LastTradeQuantity	This field contains the quantity at which the last trade took place in a security.
LastTradeTime	This field contains the time when the last trade took place in a security.
AverageTradePrice	This field contains the average price of all the trades in a security.
AuctionNumber	This field contains the auction number. The maximum value this can take is 9999. In other cases, it is set to zero.
AuctionStatus	Refer to Auction Status in Appendix.
InitiatorType	This field contains the initiator type—control or trader. Presently initiator type is control, since only the Exchange can initiate an auction. Otherwise it is blank.
InitiatorPrice	This field contains the price of the security of the initiator's auction order. Otherwise it is set to zero.

Field Name	Brief Description
InitiatorQuantity	This field contains the quantity of the security of the initiator's auction order. Otherwise it is set to zero.
AuctionPrice	This field contains the price at which auction in a security takes place. Otherwise it is set to zero.
AuctionQuantity	This field contains the quantity at which auction in a security takes place. Otherwise it is set to zero.
RecordBuffer (MBO INFORMATION)	This field contains five best Buy orders and five best Sell orders from the order book. First five contains Buy orders and next five contains Sell orders.
RecordBuffer (MBP INFORMATION)	This field contains five best Buy prices and five best Sell prices from the order book .First five are for Buy and next five for Sell.
BbTotalBuyFlag	This field contains value '1' if there is a buyback order in the buy side else its value is zero. This is useful if the buyback order is not amongst the top five.
BbTotalSellFlag	Currently, its value is set to zero.
TotalBuyQuantity	This field contains the total quantity of buy orders in a security.
TotalSellQuantity	This field contains the total quantity of sell orders in a security.
Indicator	This structure contains flags which can be set to indicate Buy, Sell and latest trade less than or greater than the immediately previous LTP.
ClosingPrice	This field contains the closing price of a security.
OpenPrice	This field contains the open price of a security.
HighPrice	This field contains the highest trade price.
LowPrice	This field contains the lowest trade price.
MBOInformation	This field contains the quantity and price for a maximum of five best prices.
MBPInformation	This field contains the quantity, price and number of orders for a maximum of five best prices.

Only Market by Price Update

The information regarding the best buy orders and the best sell orders is given in the following format:

Table 38 BROADCAST ONLY MBP

Structure Name	BROADCAST ONLY MBP		
Packet Length	566 bytes		
Transaction Code	BCAST_ONLY_MBP (7208)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NoOfRecords	SHORT	2	40
INTERACTIVE ONLY MBP DATA [2] (Refer Table 38.1)	STRUCT	524	42

Table 38.1 INTERACTIVE ONLY MBP DATA

Structure Name	INTERACTIVE ONLY MBP DATA		
Packet Length	262 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
BookType	SHORT	2	4
TradingStatus	SHORT	2	6
VolumeTradedToday	LONG LONG	8	8
LastTradedPrice	LONG	4	16
NetChangeIndicator	CHAR	1	20
Filler	CHAR	1	21
NetPriceChangeFromClosingPrice	LONG	4	22
LastTradeQuantity	LONG	4	26
LastTradeTime	LONG	4	30
AverageTradePrice	LONG	4	34
AuctionNumber	SHORT	2	38
AuctionStatus	SHORT	2	40
InitiatorType	SHORT	2	42
InitiatorPrice	LONG	4	44

Structure Name	INTERACTIVE ONLY MBP DATA		
Packet Length	262 bytes		
Field Name	Data Type	Size in Byte	Offset
InitiatorQuantity	LONG	4	48
AuctionPrice	LONG	4	52
AuctionQuantity	LONG	4	56
RecordBuffer [size of (MBP INFORMATION) * 10] (Refer Table 38.4)	CHAR	160	60
BbTotalBuyFlag	SHORT	2	220
BbTotalSellFlag	SHORT	2	222
TotalBuyQuantity	LONG LONG	8	224
TotalSellQuantity	LONG LONG	8	232
MBP INDICATOR (Refer Table 38.2 for Small Endian & Refer Table 38.3 Big Endian)	STRUCT	2	240
ClosingPrice	LONG	4	242
OpenPrice	LONG	4	246
HighPrice	LONG	4	250
LowPrice	LONG	4	254
IndicativeClosePrice	LONG	4	258

Table 38.2 MBP INDICATOR (For Small Endian Machines)

Structure Name	MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved [4]	BIT	4	0
Sell	BIT	1	0
Buy	BIT	1	0
LastTradeLess	BIT	1	0
LastTradeMore	BIT	1	0

Structure Name	MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	CHAR	1	1

Table 38.3 MBP INDICATOR (For Big Endian Machines)

Structure Name	MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
LastTradeMore	BIT	1	0
LastTradeLess	BIT	1	0
Buy	BIT	1	0
Sell	BIT	1	0
Reserved	BIT	4	0
Reserved	CHAR	1	1

Table 38.4 MBP INFORMATION

Structure Name	MBP INFORMATION		
Packet Length	16 bytes		
Field Name	Data Type	Size in Byte	Offset
Quantity	LONG LONG	8	0
Price	LONG	4	8
NumberOfOrders	SHORT	2	12
BbBuySellFlag	SHORT	2	14

Field Name	Brief Description
TransactionCode	The transaction code set for the purpose is BCAST_ONLY_MBP (7208).
NoOfRecords	This field contains the number of securities sent.
Token	This field contains the token number—a unique number given to a particular symbol-series combination.
BookType	This field contains the book type—RL / ST / SL / SP / AU

Field Name	Brief Description
TradingStatus	This field specifies trading status of the security. It contains one of the following values. '1' - Preopen '2' - Open '3' - Suspended '4' - Preopen Extended '6' – Price Discovery Trading Status for a Security will be '6' during pre-open session. It will be '2' when Normal Market opens.
VolumeTradedToday	This field contains the total quantity of a security traded on the current day. During Preopen this field will contain Indicative Equilibrium Quantity. Once matching starts it contains total quantity traded for that security. If field value exceeds unsigned long max value (i.e. 4294967295), the value of the field will be wrapped up, i.e. start from 0. If field is read as LONG (signed LONG) and if field value exceeds signed long max value (i.e. 214748364), then the value will be negative.
LastTradedPrice	This field contains the price at which the latest trade in a security has taken place. During 1st preopen, LTP field will display Previous day's value in MBP screen. For next preopen sessions it will show the last traded price of security that was last updated during the market status open or Pre-Open. Once matching starts it contains the LTP of the security.
NetChangeIndicator	This field is a flag which indicates any change of the order price from the LTP. '+' for increase '-' for decrease.

Field Name	Brief Description
	During Preopen it will indicate any change in Indicative Open Price from previous day's close price. Once matching starts it will indicate the change in trade price from previous day's close price.
NetPriceChange	This field contains the net change between the order price and the LTP. During Preopen it will contain net % change between previous day's close price and the indicative open price. Once matching starts it will contain net % change between previous day's close price and trade price.
LastTradeQuantity	This field contains the quantity at which the last trade took place in a security. During preopen, for securities which are in Price Discovery, LTQ field will display as previous day's value. Once matching starts this field contains the quantity at which the last trade took place in a security
LastTradeTime	This field contains the time when the last trade took place in a security. During preopen, for securities which are in Price Discovery, LTT field will display as previous day's value. Once matching starts it contains the Last Trade Time.
AverageTradePrice	This field contains the average price of all the trades in a security. During 1st Preopen session it will always be zero. For next preopen sessions, it will have the average traded price that was last updated during the market status open or Pre-Open. Once matching starts it will contain the Average Trade Price.
AuctionNumber	This field contains the auction number. The maximum value this can take is 9999. Otherwise it is set to zero. During Preopen it will always be zero.
AuctionStatus	Refer to Auction Status in Appendix.

Field Name	Brief Description
	During Preopen it will always be zero.
InitiatorType	This field contains the initiator type—control or trader. Presently initiator type is control, since only the Exchange can initiate an auction. Otherwise it is set to blank. During Preopen it will always be blank.
InitiatorPrice	This field contains the price of the security of the initiator's auction order. Otherwise it is set to zero. During Preopen it will always be zero.
InitiatorQuantity	This field contains the quantity of the security of the initiator's auction order. Otherwise it is set to zero. During Preopen it will always be zero.
AuctionPrice	This field contains the price at which auction in a security takes place. Otherwise it is set to zero. During Preopen it will always be zero.
AuctionQuantity	This field contains the quantity at which auction in a security takes place. Otherwise it is zero. During Preopen it will always be zero.
Record Buffer (MBP INFORMATION)	This field contains five best Buy prices and five best Sell prices from the order book. First five are for buy and next five for sell. During Preopen order collection period (till pre-open end), in this structure the first four rows for Buy and Sell contains the four Limit orders and the last row of both sides is reserved for ATO orders. During Preopen order collection period (till pre-open end), if ATO order exists then in Price field -1 will be sent in the last row of both sides.
BbTotalbuyFlag	The field contains the values to represent buy back orders, market maker order or both.The values will be as below. "0" Non Market Maker and Non Buy back orders "1" Buy back orders "2" Market Maker Orders "3" Market Maker and Buy Back Order This is useful if the buyback order is not amongst the top five.

Field Name	Brief Description
	The values in this field will be according to the flag value table given below.
BbTotalsellFlag	The field contains the values to represent buy back orders; market maker order or both.The values will be as below. "0" Non Market Maker and Non Buy back orders "1" Buy back orders "2" Market Maker Orders "3" Market Maker and Buy Back Order This is useful if the buyback order is not amongst the top five. The values in this field will be according to the flag value table given below.
TotalBuyQuantity	This field contains the total quantity of buy orders in a security.
TotalSellQuantity	This field contains the total quantity of sell orders in a security.
Indicator	This field contains flags which can be set to indicate Buy, Sell and Latest trade less than or greater than the immediately previous LTP. LastTradeMore During Preopen session: Indicate change from the Last received Indicative Open Price. If received open price is more than the last received open price, then it will be set to 1, else it will be 0. During Matching: Indicate change from the Last received Trade Price. If received open price is more than the last received trade price, then it will be set to 1, else it will be 0. Vice versa for LastTradeLess Buy / SELL: This BIT will be set to 0
ClosingPrice	This field contains the closing price of a security.
OpenPrice	This field contains the open price of a security. This field contains the Indicative opening price of a security for that Preopen session and Final Open Price of a security for Matching Phase.

Field Name	Brief Description
	When normal market opens, Final open price will be available in this field.
HighPrice	This field contains the highest trade price. During 1st Preopen session it will always be zero. For next preopen sessions, it will have the high price that was last updated during the market status open or Pre-Open. Once matching starts it will be updated.
LowPrice	This field contains the lowest trade price.
MBPInformation	This structure contains the quantity, price and number of orders for a maximum of five best prices. This field contains the quantity, price and number of orders for max of 5 orders out of which first four orders are best limit and the last ATO order. If there are less than 4 limit orders, ATO order will still be at the 5th place During Preopen order collection period (till pre-open end),if ATO order exists then in Price field -1 will be sent in the last row of both sides.
Quantity	This field contains the quantity at the price point. If field value exceeds unsigned long max value (i.e. 4294967295), the value of the field will be wrapped up, i.e. start from 0. If field is read as LONG (signed LONG) and if field value exceeds signed long max value (i.e. 214748364), then the value will be negative.
Price	The price point in the MBP array.
NumberOfOrders	The number of orders at the price point.
BbBuySellFlag	This field contains the values to indicate whether there is a buyback order or market maker order in the buy or sell side at the price point. The values in this field will be according to the flag value table.

When the Normal Market opens, the final open price will be available in the Normal Market broadcast transcode BCAST_ONLY_MBP (7208) in OpenPrice field of the structure BROADCAST ONLY MBP.

Market Watch Update

The market watch information gives the best buy order and its quantity, best sell order and its quantity and the last trade price. The structure sent for the purpose is:

Table 39 BROADCAST INQUIRY RESPONSE

Structure Name	BROADCAST INQUIRY RESPONSE		
Packet Length	466 bytes		
Transaction Code	BCAST_MW_ROUND_ROBIN (7201)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
MARKETWATCHBROADCAST [4] (Refer table 39.1)	STRUCT	424	42

Table 39.1 MARKETWATCHBROADCAST

Structure Name	MARKETWATCHBROADCAST		
Packet Length	106 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
MARKET WISE INFORMATION [3] (Refer Table 39.2)	STRUCT	102	4

Table 39.2 MARKET WISE INFORMATION

Structure Name	MARKET WISE INFORMATION		
Packet Length	34 bytes		
Field Name	Data Type	Size in Byte	Offset
MBO MBP INDICATOR (Refer table 39.3 for small endian & table 39.4 for big endian)	STRUCT	2	0
BuyVolume	LONG LONG	8	2

Structure Name	MARKET WISE INFORMATION		
Packet Length	34 bytes		
Field Name	Data Type	Size in Byte	Offset
BuyPrice	LONG	4	10
SellVolume	LONG LONG	8	14
SellPrice	LONG	4	22
LastTradePrice	LONG	4	26
LastTradeTime	LONG	4	30

Table 39.3 MBO MBP INDICATOR (For Small Endian Machines)

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	4	0
Sell	BIT	1	0
Buy	BIT	1	0
LastTradeLess	BIT	1	0
LastTradeMore	BIT	1	0
Reserved	CHAR	1	1

Table 39.4 MBO MBP INDICATOR (For Big Endian Machines)

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
LastTradeMore	BIT	1	0
LastTradeLess	BIT	1	0
Buy	BIT	1	0
Sell	BIT	1	0
Reserved	BIT	4	0
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	The transaction code set for the purpose is BCAST_ONLY_MBP (7201).
NumberOfRecords	This field contains the number of times the structure MARKET WATCH BROADCAST is repeated.
Token	This field contains the token number—a unique number given to a particular symbol-series combination.
Indicator	This structure contains the flags which can be set to indicate Buy, Sell and Last trade less than or greater than previous LTP.
BuyVolume	This field contains the quantity of the best Buy order.
BuyPrice	This field contains the price of the best Buy order.
SellVolume	This field contains the quantity of the best Sell order.
SellPrice	This field contains the price of the best Sell order.
LastTradePrice	This field contains the latest trade price of a security. During preopen it contains the indicative open price of that security.
LastTradeTime	This field contains the latest trade time of a security.

CALL AUCTION MBP Broadcast

During Call Auction2 pre-open session, market data will b BROADCAST CALL AUCTION MBP e sent based on the order activity during the order collection period. Indicative opening price will be computed based on the order activity. When Call Auction2 pre-open session ends, order activity will be stopped and the final open price will be computed for all Call-Auction2 securities. Final open price will be available in the market data.

After computation of final open price, orders will be matched based on the final open price. Trades related data will be available in market data once the matching is started.

Once the FOP is calculated and matching is over for a token, the MBP data for that token will be received in the existing MBP broadcast packet (7208).

The transaction code to disseminate the Call Auction2 market data during Preopen session is BCAST_CALL AUCTION_MBP (7214).

The structure on the transcode is as show below:

Table 40 BROADCAST CALL AUCTION MBP

Structure Name	BROADCAST CALL AUCTION MBP		
Transaction Code	BCAST_CALL AUCTION_MBP (7214)		
Packet Length	538 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NoOfRecords	SHORT	2	40
INTERACTIVE CALL AUCTION MBP DATA [2] (Refer Table 40.1)	STRUCT	496	42

Table 40.1 INTERACTIVE CALL AUCTION MBP DATA

Structure Name	INTERACTIVE CALL AUCTION MBP DATA		
Packet Length	248 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
BookType	SHORT	2	4
TradingStatus	SHORT	2	6
VolumeTradedToday	LONG LONG	8	8
IndicativeTradedQty	LONG LONG	8	16
LastTradedPrice	LONG	4	24
NetChangeIndicator	CHAR	1	28
Filler	CHAR	1	29
NetPriceChangeFromClosingPrice	LONG	4	30
LastTradeQuantity	LONG	4	34
LastTradeTime	LONG	4	38
AverageTradePrice	LONG	4	42
FirstOpenPrice	LONG	4	46
RecordBuffer [size of (MBP INFORMATION) * 10] (Refer Table 40.4)	CHAR	160	50
BbTotalBuyFlag	SHORT	2	210
BbTotalSellFlag	SHORT	2	212
TotalBuyQuantity	LONG LONG	8	214

Structure Name	INTERACTIVE CALL AUCTION MBP DATA		
Packet Length	248 bytes		
Field Name	Data Type	Size in Byte	Offset
TotalSellQuantity	LONG LONG	8	222
MBP INDICATOR (Refer Table 40.2 for small endian & Table 40.3 for Big endian)	STRUCT	2	230
ClosingPrice	LONG	4	232
OpenPrice	LONG	4	236
HighPrice	LONG	4	240
LowPrice	LONG	4	244

For Small Endian Machines:

Table 40.2 MBP INDICATOR

Structure Name	MBP INDICATOR		
Packet Length	2 Bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	4	0
Sell	BIT	1	0
Buy	BIT	1	0
LastTradeLess	BIT	1	0
LastTradeMore	BIT	1	0
Reserved	CHAR	1	1

For Big Endian Machines:

Table 40.3 MBP INDICATOR

Structure Name	MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
LastTradeMore	BIT	1	0
LastTradeLess	BIT	1	0
Buy	BIT	1	0
Sell	BIT	1	0
Reserved	BIT	4	0
Reserved	CHAR	1	1

Table 40.4 MBP INFORMATION

Structure Name	MBP INFORMATION		
Packet Length	16 bytes		
Field Name	Data Type	Size in Byte	Offset
Quantity	LONG LONG	8	0
Price	LONG	4	8
NumberOfOrders	SHORT	2	12
BbBuySellFlag	SHORT	2	14

Field Name	Brief Description
TransactionCode	The transaction code set for the purpose is BCAST_CALL AUCTION_MBP (7214).
NoOfRecords	This field contains the number of securities sent.
Token	This field contains the token number—a unique number given to a particular symbol-series combination.
BookType	This field contains the book type—RL / ST / SL / SP / AU / CA/ CB For CALL AUCTION1 session book type will be CA(11) For CALL AUCTION2 session book type will be CB(12)
TradingStatus	This field specifies trading status of the security. It contains one of the following values. '1' – Preopen '2' – Open '3' – Suspended '4' – Preopen Extended '6' – Price Discovery Trading Status for a Security will be '6' during pre-open session and opening session
VolumeTradedToday	This field contains the total quantity of a security traded on the current day. During Preopen this field will contain Indicative Equilibrium Quantity. Once matching starts it contains total quantity traded for that security.
LastTradedPrice	This field contains the price at which the latest trade in a security has taken place.

Field Name	Brief Description
	During Preopen as well as During matching, it contains LTP of the security.
NetChangeIndicator	This field is a flag which indicates any change of the IOP or LTP from previous day's close price. '+' for increase '-' for decrease. During Preopen it will indicate any change in Indicative Open Price from previous day's close price. Once matching starts it will indicate the change in trade price from previous day's close price.
NetPriceChange	This field contains the net change between the IOP or LTP from previous day's close price. During Preopen it will contain net % change between previous day's close price and the indicative open price. Once matching starts it will contain net % change between previous day's close price and trade price.
LastTradeQuantity	This field contains the quantity at which the last trade took place in a security. During Preopen as well as During matching, it contains the quantity at which the last trade took place in a security.
LastTradeTime	This field contains the time when the last trade took place in a security. During Preopen as well as During matching, it contains the Last Trade Time.
AverageTradePrice	This field contains the average price of all the trades in a security. During 1st Preopen session it will always be zero. For next preopen sessions, it will have the average traded price that was last updated during the market status opening. Once matching starts it will contain the Average Trade Price.
FirstOpenPrice	This field contains the First trade open price for call auction security. During first call auction- order collection period, this field will be zero. Once matching starts it will contain the First Trade Price. Once updated, for all subsequent call auctions, it will not change.

Field Name	Brief Description
	This field may remain zero till the first trade happens.
Record Buffer (MBP INFORMATION)	This field contains five best Buy prices and five best Sell prices from the order book. First five are for buy and next five for sell. During Preopen order collection period (till pre-open end), in this structure the first five rows for Buy and Sell contains the five Limit orders.
BbTotalbuyFlag	This field contains the values to indicate whether there is a buyback order or market maker order in the buy side .This is useful if the buyback order or market maker order is not amongst the top five. During Preopen and matching, value will always be zero.
BbTotalsellFlag	This field contains the values to indicate whether there is a buyback order or market maker order in the sell side .This is useful if the buyback order or market maker order is not amongst the top five. During Preopen and matching, value will always be zero.
TotalBuyQuantity	This field contains the total quantity of buy orders in a security.
TotalSellQuantity	This field contains the total quantity of sell orders in a security.
Indicator	This field contains flags which can be set to indicate Buy, Sell and Latest trade less than or greater than the immediately previous LTP. LastTradeMore During Preopen session: Indicate change from the Last received Indicative Open Price. If received open price is more than the last received open price, then it will be set to 1, else it will be 0. During Matching: Indicate change from the Last received Trade Price. If received open price is more than the last received trade price, then it will be set to 1, else it will be 0. Vice versa for LastTradeLess Buy / SELL This BIT will be set to 0
ClosingPrice	This field contains the closing price of a security.
OpenPrice	This field contains the open price of a security. This field contains the Indicative opening price of a security for that Preopen session and Final Open Price of a security for Matching Phase.

Field Name	Brief Description
	When normal market opens, Final open price will be available in this field.
ClosingPrice	This field contains the closing price of a security.
OpenPrice	This field contains the Indicative opening price of a security for that Preopen session and Final Open Price of a security for Matching Phase. When normal market opens, Final open price will be available in this field.
HighPrice	This field contains the highest trade price. During 1st Preopen session it will always be zero. For next preopen sessions, it will have the high price that was last updated during the market status opening. Once matching starts it will be updated.
LowPrice	This field contains the lowest trade price. During 1st Preopen session it will always be zero. For next preopen sessions, it will have the low price that was last updated during the market status opening. Once matching starts it will be updated.
MBPInformation	This field contains the quantity, price and number of orders for a maximum of five best prices. For CALL AUCTION1 This field contains the quantity, price and number of orders for max of 5 orders out of which first four orders are best limit and the last ATO order. If there are less than 4 limit orders, ATO order will still be at the 5th place During Preopen order collection period (till pre-open end), if ATO order exists then in Price field -1 will be sent in the last row of both sides. For CALL AUCTION2 This field contains the quantity, price and number of orders for max of 5 best Limit orders.
Quantity	This field contains the quantity at the price point.
Price	The price point in the MBP array.
NumberOfOrders	The number of orders at the price point.

Field Name	Brief Description
BbBuySellFlag	This field contains the values to indicate whether there is a buyback order or market maker order in the buy or sell side at the price point. During Preopen and matching, value will always be zero.

This transcode will be sent only for the securities which are eligible to take part in CALL AUCTION 2 sessions.

Note: The sent Packet will be LZO compressed packet.

Flag Value Table

The values of buyback flags in MBP array and total order buyback values in both buy and sell sides will be according to the following table:

Buy_back order	Market maker order	bb_buy_flag/ bb_sell_flag/ bb_total_buy_flag/ bb_total_sell_flag
NO	NO	0
YES	NO	1
NO	YES	2
YES	YES	3

Market Watch Update

The market watch information gives the best buy order and its quantity, best sell order and its quantity and the last trade price. The market watch data for Call Auction market is sent through new transcode (7215). The structure sent for the purpose is:

Table 41 BROADCAST CALL AUCTION MARKET WATCH

Structure Name	BROADCAST CALL AUCTION MARKET WATCH		
Transaction Code	BCAST_CA_MW (7215)		
Packet Length	482 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NoOfRecords	SHORT	2	40
MARKETWATCHBROADCAST[11] (Refer Table 41.1)	STRUCT	440	42

Table 41.1 MARKETWATCHBROADCAST

Structure Name	MARKETWATCHBROADCAST		
Packet Length	40 Bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
Mkt Type	SHORT	2	4
MBO MBP INDICATOR (Refer Table 37.2 for small endian and Table 37.3 for big endian)	STRUCT	2	6
BuyVolume	LONG LONG	8	8
BuyPrice	LONG	4	16
SellVolume	LONG LONG	8	20
SellPrice	LONG	4	28
LastTradePrice	LONG	4	32
LastTradeTime	LONG	4	36

For Small Endian Machines:

Table 41.2 MARKETWATCH_BROADCAST

Structure Name	MBO MBP INDICATOR		
Packet Length	2 Bytes		
Field Name	Data Type	Size	Offset
Reserved	BIT	4	0
Sell	BIT	1	0
Buy	BIT	1	0
LastTradeLess	BIT	1	0
LastTradeMore	BIT	1	0
Reserved	CHAR	1	1

For Big Endian Machines:

Table 41.3 MARKETWATCHBROADCAST

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
LastTradeMore	BIT	1	0

Structure Name	MBO MBP INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
LastTradeLess	BIT	1	0
Buy	BIT	1	0
Sell	BIT	1	0
Reserved	BIT	4	0
Reserved	CHAR	1	1

Field Name	Brief Description
TransactionCode	The transaction code sent is BCAST_CA_MW (7215).
NumberOfRecords	This field contains the number of times the structure MARKET WATCH BROADCAST is repeated.
Token	This field contains the token number—a unique number given to a particular symbol-series combination.
Mkt Type	This field contains the market type For CALL AUCTION1, market type 5 will be received For CALL AUCTION2, market type 6 will be received
Indicator	This structure contains the flags which can be set to indicate Buy, Sell and Last trade less than or greater than previous LTP.
BuyVolume	This field contains the quantity of the best Buy order.
BuyPrice	This field contains the price of the best Buy order.
SellVolume	This field contains the quantity of the best Sell order.
SellPrice	This field contains the price of the best Sell order.
LastTradePrice	This field contains the latest trade price of a security.
LastTradeTime	This field contains the latest trade time of a security.

Security Open Message

Note: The Following transcode SECURITY_OPEN_PRICE 6013) will not be sent by exchange.

When the market opens the open price of the security is sent in the following structure:

Table 42 MS_SEC_OPEN_MSGS

Structure Name	MS_SEC_OPEN_MSGS		
Transaction Code	SECURITY_OPEN_PRICE (6013)		
Packet Length	58 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
SEC_INFO (Refer Table 4)	STRUCT	12	40
Token	SHORT	2	52
OpeningPrice	LONG	4	54

Field Name	Brief Description
TransactionCode	The transaction code sent is SECURITY_OPEN_PRICE (6013).
SEC_INFO	This structure contains the symbol and series for a particular security.
Token	This field contains a unique number that is given to a particular symbol-series combination.
OpeningPrice	This field contains open price of the security.

Broadcast Circuit Check

If there has been no data on the broadcast circuit for a stipulated time period, then a pulse is sent. This time is nine seconds now but it can be changed by NSE–Control. This is only to intimate that the circuit is still there but there is no data to send. The structure sent is:

BCAST_HEADER (*Refer to Broadcast Header in chapter 2*)

Field Name	Brief Description
TransactionCode	The transaction code sent is BC_CIRCUIT_CHECK (6541).

Multiple Index Broadcast

The multiple index broadcast structure is as follows:

Table 43 BROADCAST INDICES

Structure Name	BROADCAST INDICES		
Transaction Code	BCAST_INDICES (7207)		
Packet Length	474 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
Indices[6] (Refer Table 43.1)	STRUCT	426	42

Table 43.1 Indices

Structure Name	INDICES		
Packet Length	71 Bytes		
Field Name	Data Type	Size in Byte	Offset
IndexName	CHAR	21	0
IndexValue	LONG	4	21
HighIndexValue	LONG	4	25
LowIndexValue	LONG	4	29
OpeningIndex	LONG	4	33
ClosingIndex	LONG	4	37
PercentChange	LONG	4	41
YearlyHigh	LONG	4	45
YearlyLow	LONG	4	49
NoOfUpmoves	LONG	4	53
NoOfDownmoves	LONG	4	57
MarketCapitalisation	DOUBLE	8	61
NetChangeIndicator	CHAR	1	69
FILLER	CHAR	1	70

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_INDICES (7207)
NoOfRecords	This field contains the number of indices currently supported by the system. Depending upon this number, there will be records filled up in subsequent Indices structure.
Indices	This field is an array of structure. The attributes of this structure are given below in this table itself.
IndexName	This field contains Name of the index. For example, Nifty

Field Name	Brief Description
IndexValue	This field contains the online market index value at that instance of broadcast.
HighIndexValue	This field contains the day's highest index value at the time of broadcast.
LowIndexValue	This field contains day's lowest index value at the time of broadcast.
OpeningIndex	This field contains the opening index value at the time of market open. In Preopen, Indicative Index value will be computed on indicative opening price. Once the final open price is computed, the final index value will be sent.
ClosingIndex	If market is open, this field it is set to previous day's closing index. After completion of day's batch processing, this field value shows today's close.
PercentChange	This field contains the percent change in current index with respect to yesterday's closing index.
YearlyHigh	This field contains the highest index in the year.
YearlyLow	This field contains the lowest index in the year.
NoOfupmoves	This field contains the number of time index has moved up with respect to previous index.
NoOfdownmoves	This field contains the number of time index has moved down with respect to previous index.
MarketCapitalization	This field contains the Market Capitalization of securities participating in the index.
NetChange Indicator	This field contains one of the following values. • '+' - if the current index is greater than previous index. • '-' - if the current index is less than previous index. • ' ' - if the current index is equal to previous index.

Multiple Indicative Index Broadcast

The Indicative Index Broadcast messages will start arriving half an hour before the market close. The multiple indicative index broadcast structure is as follows:

BROADCAST INDICATIVE INDICES

Structure Name	BROADCAST INDICATIVE INDICES		
Transaction Code	BCAST_INDICATIVE_INDICES (8207)		
Packet Length	474 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
IndicativeIndices[6] (Refer Indicative Indices Table)	STRUCT	426	42

Indicative Indices

Structure Name	INDICATIVE INDICES		
Packet Length	71 Bytes		
Field Name	Data Type	Size in Byte	Offset
IndexName	CHAR	21	0
IndicativeCloseValue	LONG	4	21
Reserved	LONG	4	25
Reserved	LONG	4	29
Reserved	LONG	4	33
ClosingIndex	LONG	4	37
PercentChange	LONG	4	41
Reserved	LONG	4	45
Reserved	LONG	4	49
Change	LONG	4	53
Reserved	LONG	4	57
MarketCapitalization	DOUBLE	8	61
NetChange Indicator	CHAR	1	69
FILLER	CHAR	1	70

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_INDICATIVE_INDICES (8207)
NoOfRecords	This field contains the number of indicative indices currently supported by the system. Depending upon this number, there will be records filled up in subsequent Indicative Indices structure.
IndicativeIndices	This field is an array of structure. The attributes of this structure are given below in this table itself.

Field Name	Brief Description
IndexName	This field contains Name of the indicative index. For example, Nifty
IndicativeCloseValue	This field contains the indicative index close value.
ClosingIndex	If market is open, this field it is set to zero. After completion of day's batch processing, this field value shows closing value of the index.
PercentChange	This field contains the difference between the Indicative closing value and previous day's closing value of the index in percentage format.
Change	This field contains the absolute difference between the Indicative closing value and previous day's closing value of the index.
MarketCapitalization	This field contains the Market Capitalization of securities participating during the indicative close session.
NetChange Indicator	This field contains one of the following values. • '+' - if the current index is greater than previous indicative close index. • '-' - if the current index is less than previous indicative close index. • ' ' - if the current index is equal to previous indicative close index.

Multiple Index Broadcast for INDIA VIX

The multiple index broadcast structure for INDIA VIX is as follows:

Table 44 BROADCAST INDICES VIX

Structure Name	BROADCAST INDICES VIX		
Transaction Code	BCAST_INDICES_VIX(7216)		
Packet Length	474 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
Indices[6] (Refer Table 44.1)	STRUCT	426	42

Table 44.1 INDICES

Structure Name	INDICES		
Packet Length	71 Bytes		
Field Name	Data Type	Size in Byte	Offset
IndexName	CHAR	21	0
IndexValue	LONG	4	21
HighIndexValue	LONG	4	25
LowIndexValue	LONG	4	29
OpeningIndex	LONG	4	33
ClosingIndex	LONG	4	37
PercentChange	LONG	4	41
YearlyHigh	LONG	4	45
YearlyLow	LONG	4	49
NoOfUpmoves	LONG	4	53
NoOfDownmoves	LONG	4	57
MarketCapitalisation	DOUBLE	8	61
NetChangeIndicator	CHAR	1	69
FILLER	CHAR	1	70

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_INDICES (7216)
NoOfRecords	This field contains the number of indices currently supported by the system. Depending upon this number, there will be records filled up in subsequent Indices structure.
Indices	This field is an array of structure. The attributes of this structure are given below in this table itself.
IndexName	This field contains Name of the index. It will be India VIX
IndexValue	This field contains the online market index value at that instance of broadcast.
HighIndexValue	This field contains the day's highest index value at the time of broadcast.
LowIndexValue	This field contains day's lowest index value at the time of broadcast.
OpeningIndex	This field contains the opening index value at the time of market open.
ClosingIndex	If market is open, this field it is set to previous day's closing index. After completion of day's batch processing, this field value shows today's close.

Field Name	Brief Description
PercentChange	This field contains the percent change in current index with respect to yesterday's closing index.
YearlyHigh	This field contains the highest index in the year.
YearlyLow	This field contains the lowest index in the year.
NoOfupmoves	This field contains the number of time index has moved up with respect to previous index.
NoOfdownmoves	This field contains the number of time index has moved down with respect to previous index.
MarketCapitalization	This field contains the Market Capitalization of securities participating in the index.
NetChange Indicator	This field contains one of the following values. '+' - if the current index is greater than previous index. '-' - if the current index is less than previous index. ' ' - if the current index is equal to previous index.

NOTE: Fields marked as * requires to be divided by 10000 for correct interpretation.

Broadcast industry index

This Packet contains the index values of 17 Indices with name. The structure is as follows:

Table 45 BROADCAST INDUSTRY INDICES

Structure Name	BROADCAST INDUSTRY INDICES		
Transaction Code	BCAST_IND_INDICES (7203)		
Packet Length	484 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
Indices[17] (Refer Table 45.1)	STRUCT	425	42

Table 45.1 INDICES

Structure Name	INDICES		
Packet Length	25 Bytes		
Field Name	Data Type	Size in Byte	Offset
Industry Name[21]	CHAR	21	0

Structure Name	INDICES		
Packet Length	25 Bytes		
Field Name	Data Type	Size in Byte	Offset
IndexValue	LONG	4	21

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_IND_INDICES (7203).
NoOfRecords	This field contains the number of indices currently supported by the system. Depending upon this number, there will be records filled up in subsequent Indices structure.
Indices	This field is an array of structure. The attributes of this structure are given below in this table itself
IndexName	This field contains Name of the index. For example, Defty, CNX IT
IndexValue	This field contains the online market index value at that instance of broadcast.

Broadcast buy back Information

This packet will contain the buyback Information which are running on that day. This will be broadcasted for every one hour from Market open till market closes on that day. The structure is as follows:

Table 46 BROADCAST BUY_BACK

Structure Name	BROADCAST BUY_BACK		
Transaction Code	BCAST_BUY_BACK (18708)		
Packet Length	426 Bytes		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
BuyBackData [6] (Refer Table 46.1)	STRUCT	384	42

Table 46.1 BUYBACKDATA

Structure Name	BUYBACKDATA		
Packet Length	64 Bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
Symbol	CHAR	10	4
Series	CHAR	2	14
PdayCumVol	DOUBLE	8	16
PdayHighPrice	LONG	4	24
PdayLowPrice	LONG	4	28
PdayWtAvg	LONG	4	32
CdayCumVol	DOUBLE	8	36
CdayHighPrice	LONG	4	44
CdayLowPrice	LONG	4	48
CdayWtAvg	LONG	4	52
StartDate	LONG	4	56
EndDate	LONG	4	60

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_BUY_BACK (18708)
NoOfRecords	This field contains the number of times the structure BuyBackData is repeated.
BuyBackData	This field is an array of structure. The attributes of this structure are given below in this table itself.
Token	This field contains a unique number that is given to a particular symbol-series combination.
Symbol	This field contains the symbol of the security.
Series	This field contains the series of the security.
PDayCumVolume	This field contains previous day cumulative Volume
PDayHighPrice	This field contains Previous day's High Price
PDayLowPrice	This field contains Previous day's Low Price
PDayWeightAvg	This field contains Previous day's Weighted Average Price
CDayCummulativeVolume	This field contains current day's cumulative Volume
CDayHighPrice	This field contains current day's High Price

Field Name	Brief Description
CDayLowPrice	This field contains current day's Low Price
CDayWeightAvg	This field contains current day's Weighted Average Price
StartDate	This field contains Start Date of Buy back period
EndDate	This field contains End Date of Buy back period

CALL AUCTION Order Cancel Update

In case of Special Preopen Session (SPOS) for IPO/Relist, order cancellation statistics will be sent to users during order collection period.

Order cancel statistics will be sent only for securities which are eligible to take part in Special Preopen Session.

The cancellation statistics will solely reflect order cancellation initiated by market participant.

Order cancelled by system/exchange will be excluded from cancellation statistics.

The transaction code to disseminate the order cancel statistics data during call auction session is BCAST_CALL AUCTION_ORD_CXL_UPDATE (7210).

The structure on the transcode is as show below:

BROADCAST CALL AUCTION ORD CXL UPDATE

Structure Name	BROADCAST CALL AUCTION ORD CXL UPDATE		
Packet Length	490 bytes		
Transaction Code	BCAST_CALL AUCTION_ORD_CXL_UPDATE (7210)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer Table 3)	STRUCT	40	0
NoOfRecords	SHORT	2	40
INTERACTIVE ORD CXL DETAILS [8] (Refer Table 69.1)	STRUCT	448	42

INTERACTIVE ORD CXL DETAILS

Structure Name	INTERACTIVE ORD CXL DETAILS		
Packet Length	56 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
Filler	CHAR	4	4
BuyOrdCxlCount	LONG LONG	8	8
BuyOrdCxlVol	LONG LONG	8	16
SellOrdCxlCount	LONG LONG	8	24
SellOrdCxlVol	LONG LONG	8	32
Reserved	CHAR	16	40

Field Name	Brief Description
TransactionCode	The transaction code set for the purpose is BCAST_CALL AUCTION_ORD_CXL_UPDATE (7210).
NoOfRecords	This field contains the number of securities sent.
Token	This field contains the token number—a unique number given to a particular symbol-series combination.
BuyOrdCxlCount	This field contains the total count of buy orders cancelled for the security during SPOS session.
BuyOrdCxlVol	This field contains the total quantity of buy orders cancelled for the security during SOPS session.
SellOrdCxlCount	This field contains the total count of sell orders cancelled for the security during SPOS session.
SellOrdCxlVol	This field contains the total quantity of sell orders cancelled for the security during SPOS session.

Chapter 8 Inquiry

Introduction

This section describes the Auction Inquiry and the system responses for the same.

Auction Inquiry Request

The format of the message sent in a structure is as follows:

Table 47 MS_AUCTION_INQ_REQ

Structure Name	MS_AUCTION_INQ_REQ		
Transaction Code	AUCTION_INQUIRY_IN (18016)		
Packet Length	55 Bytes		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 1)	STRUCT	40	0
SEC_INFO (Refer Table 4)	STRUCT	12	40
AuctionNo	SHORT	2	52
PageIndicator	CHAR	1	54

Field Name	Brief Description
TransactionCode	The transaction code is AUCTION_INQUIRY_IN (18016)
SEC_INFO	This structure should contain the symbol and series for a particular security
AuctionNo	This field should contain the auction number. It is optional to specify symbol and series.
PageIndicator	This field is to help the user browse through various pages of information. It contains the values of 'U', 'D', 'H', 'E', 'F' for Up, Down, Home, End, and First respectively

Auction Inquiry Response

As soon as the auction inquiry request reaches the system, it sends back the structure of response in the MESSAGE HEADER (Refer to *Message Header* in Chapter 2). The response can be either an error code or the requested response.

Field Name	Brief Description
TransactionCode	The transaction code is AUCTION_INQUIRY_OUT (18017).
ErrorCode	This field contains the error code. If this error code is not '0' then error has occurred, if this is zero, then auction inquiry is successful. In case of error, symbol, series or auction number may be wrong or the auction inquiry as a whole may be wrong. In this case, the same structure is sent back in which the message header is present.
NumberOf Records	This field contains the number of records that are sent in the Inquiry Data structure which follows this field.
InquiryData	This is an array of structure. It contains the inquiry data. Refer to Auction Activity Message in Chapter 7 for details of fields in the Inquiry Data structure

Note: If the auction inquiry request is correct, the following structure is sent:

Table 48 AUCTION INQUIRY RESPONSE

Structure Name	AUCTION INQUIRY RESPONSE		
Packet Length	222 bytes		
Transaction Code	AUCTION_INQUIRY_OUT (18017)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
InquiryData[5] (Refer Table 48.1)	STRUCT	180	42

Table 48.1 INQUIRYDATA

Structure Name	INQUIRYDATA		
Packet Length	36 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
AuctionNumber	SHORT	2	4
AuctionStatus	SHORT	2	6
InitiatorType	SHORT	2	8
TotalBuy	LONG	4	10
BestBuyPrice	LONG	4	14
TotalSell	LONG	4	18
BestSellPrice	LONG	4	22

Structure Name	INQUIRYDATA		
Packet Length	36 bytes		
Field Name	Data Type	Size in Byte	Offset
AuctioinPrice	LONG	4	26
AuctionQuantity	LONG	4	30
SettlementPeriod	SHORT	2	34

Chapter 9 Encryption Decryption of Interactive Messages

Background

NSE provides a pan-India trading platform to its trading members. Members connect to this platform using client-server architecture. Connections are made using TCP/IP protocol and messages are exchanged using NSE's own messaging format (also known as NNF format). Messages exchanged are binary in nature. Currently these messages are not encrypted, exchange now proposes to encrypt them. Exchange now proposes to encrypt the messages using AES 256 bits GCM encryption with authentication. This section of document provides an overview of the implementation approach that exchange has finalized, for doing the same.

Overview

Interactive messages which are exchanged between member applications and the exchange today use the NNF protocol published by exchange. As for every trading platform, similarly in this case as well availability, reliability and speed are the key considerations in the protocol. In order to enhance the security posture, it is now proposed to encrypt these messages on an end-to-end basis. While encryption of messages within member environment towards their clients will need to be done by respective members. For the communication that happens between member applications and exchange, a few changes into NNF protocol are being proposed. Changes have been envisaged considering the following attributes.

(i) Secure communication

(ii) Availability

(iii) Reliability

(iv) Speed

Minimal changes in member applications

Proposed Methodology

Exchange proposes a combination of TLS 1.3 security protocol and AES-256 bits-based symmetric encryption approach. Following is an overview.

1st Step: Member applications will connect initially to Exchange Gateway Router server using TCP with TLS 1.3 security protocol and will receive unique session key from the Exchange through the secured connection.

2nd Step: Member applications will then connect to allocated Exchange Gateway server through TCP, and each and every message will be encrypted/decrypted using the same session key (symmetric cryptography AES 256 bits GCM mode) at both member end and Exchange end. For the new encryption method, AES 256 bits GCM encryption with authentication would be used.

Below are the details of the methodology

(i) Exchange will generate self-signed CA certificates on periodic basis. CA certificate will remain common for all members and shall be distributed as and when generated via extranet.

(ii) On a daily basis when member applications need to connect to trading platform they will need to do the following

a. Member applications will connect to Exchange Gateway Router server on TCP using TLS 1.3 security protocol. As part of TLS 1.3 security protocol, it is recommended that member applications verify Gateway Router server authenticity using the CA certificate provided by the Exchange.

b. GR request and GR response messages will be sent and received by member applications using TLS 1.3 security protocol.

c. A unique 32-byte session key will be provided to member applications as part of GR response message.

(iii) Post successful communication with Gateway router server, member applications will establish a new TCP connection with the allocated gateway server of Exchange. The first message after connecting through TCP will be a non-encrypted special

registration message (SECURE_BOX_REGISTRATION_REQUEST) to indicate that member application is using encryption. All the messages, after the first message, that are exchanged on this connection from both sides (member applications and Exchange) will be encrypted and decrypted using the 32-byte session key that was provided from Exchange at the time of Gateway Router handshake. GCM mode of symmetric cryptography AES 256 bits will be used by member applications and Exchange.

For new encryption methodology, unique 32-byte session key, 16-byte IV (Initialization Vector) (8-byte Static and 8-byte Dynamic) and a 12-bytes additional key value will be provided as part of GR response message.

In the member application, encryption and decryption operations are performed using a combination of **static** and **dynamic Initialization Vectors (IVs)**. The static and dynamic IV is taken from **GR response message** received from exchange. Static IV remains unchanged, however the dynamic IV is modified for each message. The member must **maintain two separate copies of the dynamic IV**: ensuring that for every message the dynamic part of the IV is **incremented by 1** before encryption and **decremented by 1** before decryption. In the event of a **box disconnection**, the IVs are reset at exchange end, and a **new static and dynamic IV** is provided in GR response message to a fresh GR query.

(iv) In case of new login or disconnection and then re login, the above-mentioned steps will be repeated

The new encryption data flow and implementation will co-exist with the current live implementation; however, the member applications which are migrating to the new encryption need to connect to a new port on the Gateway Router server located at the Exchange End. The first message will remain unencrypted in either encryption methodology used, consistent with the current encryption framework. No changes are required for members continuing the existing encryption mechanism.

We envisage minimal changes in member applications. Sample function calls which could be considered for encryption-decryption for the above proposed approaches are provided in annexure for Encryption/Decryption.

Chapter 10 Direct Interface to Exchange Trading System

This chapter describes how member systems can directly connect to NSE for trading, while using existing formats of business messages from NNF API documents.

To directly connect to NSE for trading, member systems will have carry out the changes specified herein.

Message Formats

Change to packet format

Length (2 bytes)	Sequence number (4 bytes)	Checksum(MD5/Authentication Tag) for Message data (16 bytes)	Message Data (Variable length)

- Max length will be the predefined value of 1024 bytes.

  Length = size of length field (2 bytes) +
  size of sequence number field (4 bytes) +
  size of the checksum field (16 bytes) +
  size of Message data (variable number of bytes as per the transcode)
- For members connecting on encrypted mode, the sequence number received in the request message for Order related interactive messages will be echoed back in the sequence number field of corresponding response messages. It is recommended to send an incremental sequence number.
- For members connecting on non-encrypted mode, there is no change in sequence number. Sequence number will be sent as 0 in all the packets.
- Message data will be of variable length
- For existing encryption methodology, the checksum algorithm used will be MD5. Checksum is applied only on the Message data field and not on the entire packet.
- For the new encryption with authentication, the MD5 checksum will be used only for the initial message, "Secure Box Registration Request". For subsequent communications, this field will contain the authentication tag.

- The authentication tag received as part of the message header will be verified against the tag obtained after decrypting the Message Data using the new encryption method.
- If the checksum (MD5 / authentication tag) does not match, a box sign-off message with error code (19031) will be sent to the member before disconnection.

  For more details on MD5 refer: RFC 1321 (rfc1321) - The MD5 Message-Digest Algorithm ()
- In case checksum is not matched, packet will be dropped at Exchange end

Change to structure for 'MESSAGE_HEADER'

MESSAGE_HEADER

Structure Name	MESSAGE_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
Transaction Code	SHORT	2	0
LogTime	LONG	4	2
AlphaChar	CHAR	2	6
User Id	LONG	4	8
ErrorCode	SHORT	2	12
Timestamp	LONG LONG	8	14
TimeStamp1	CHAR	8	22
TimeStamp2	CHAR	8	30
MessageLength	SHORT	2	38

Note: Member systems must populate relevant User ID field in the header.

Connecting to NSE for Trading

Sequence to be followed by the member for login

1. Member to connect (TCP/IP, SSL connection) to the IP and port provided by the exchange and send the GR_REQUEST using OpenSSL (Version 1.1.1 for existing encryption and 3.4.0 for new encryption) library calls with TLS versions 1.3 (TLS1_3_VERSION). Refer annexure for Encryption/Decryption.

2. Exchange will send the GR_RESPONSE to the member containing the IP address, Port and the Session key and cryptographic key & cryptographic IV (Initialization Vector) on SSL connection. For new encryption with authentication, GR_RESPONSE will contain IP address, Port, Session key, cryptographic key, cryptographic IV (Initialization Vector) and cryptographic additional key. If there is any error, then ErrorCode field in MESSAGE_HEADER will be populated with relevant error code in the GR_RESPONSE.
3. Member applications will then make a new TCP connection with the allocated Gateway server (IP and port provided in the GR_RESPONSE) and send SECURE_BOX_REGISTRATION_REQUEST. BoxID (received in GR_RESPONSE) is to be populated in SECURE_BOX_REGISTRATION_REQUEST
4. Exchange will send the SECURE_BOX_REGISTRATION_RESPONSE. If there is any error, then ErrorCode field in MESSAGE_HEADER will be populated with relevant error code in the SECURE_BOX_REGISTRATION_RESPONSE and the Box connection will be terminated.
5. If there is no error in SECURE_BOX_REGISTRATION_RESPONSE, member should do encryption and decryption initialization to create encryption and decryption contexts Please refer (annexure for Encryption/Decryption) for existing and new encryption mechanism respectively. This initialization should be done only once. Once initialized, all further messages between member application and allocated Gateway server will be encrypted and decrypted using same encryption and decryption contexts respectively. Further member should send the BOX_SIGN_ON_REQUEST_IN.

   For existing encryption methodology - BoxID, BrokerID and Session key (received in GR_RESPONSE) is to be populated in BOX_SIGN_ON_REQUEST_IN. MD5 Algorithm to be performed on plain messages. That means, while sending the messages to Trading system, MD5 is to be performed first and then encryption. Encrypted message length + 22 (sizeof(Header)) will have to be written in first 2 bytes of header, Sequence Number in next 4 bytes and MD5 value (of plain message) will be written in last 16 bytes of Header and the header will have to be prepended to the encrypted message. This message will

be sent out to Trading System. While receiving the messages from Trading System, decryption should be done first and then MD5 is to be applied on decrypted buffer. Decryption should be done on message excluding first 22 bytes of header.

For new encryption methodology - BoxID, BrokerID and Session key (received in GR_RESPONSE) is to be populated in BOX_SIGN_ON_REQUEST_IN. MD5 to be applied only on the first message - SECURE_BOX_REGISTRATION_REQUEST and above flow to be followed. For the remaining messages, Encrypted message length + 22 (sizeof(Header)) will have to be written in first 2 bytes of header, Sequence Number in next 4 bytes and authentication tag will be written in last 16 bytes of Header and the header will have to be prepended to the encrypted message. This message will be sent out to Trading System. While receiving the messages from Trading System, decryption should be done on message excluding the first 22 bytes of header and then authentication tag received as part for message header should be verified against authentication tag obtained after decryption of the message. For the first message where MD5 was used, decryption should be done first and then MD5 is to be applied on decrypted buffer. Decryption should be done on message excluding first 22 bytes of header.

6. Exchange will send the BOX_SIGN_ON_REQUEST_OUT. If there is any error, then ErrorCode field in MESSAGE_HEADER will be populated with relevant error code in the BOX_SIGN_ON_REQUEST_OUT and the Box connection will be terminated.
   Note: Multiple BOX_SIGN_ON_REQUEST_IN requests on a successfully established box connection will lead to the existing box connection termination.
7. Once a connection for a particular Box ID is established, all users linked with this Box ID can login using the SIGNON_IN structure. Refer Chapter 3 for login request and response using SIGNON_IN structure.
8. For further flow refer to existing protocol defined in Chapter 3 of Protocol Document

Gateway Router Request

MS_GR_REQUEST

Structure Name	MS_GR_REQUEST		
Packet Length	48 bytes		
Transaction Code	GR_REQUEST (2400)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0
Box ID	SHORT	2	40
BrokerID	CHAR	5	42
Filler	CHAR	1	47

Field Name	Brief Description
Transaction Code	This field is the part of Message Header. The transaction code is 2400.
Box ID	Exchange provided Box ID to be used for this connection
BrokerID	This field should contain the trading member ID

Gateway Router Response

MS_GR_RESPONSE (Existing encryption)

Structure Name	MS_GR_RESPONSE		
Packet Length	124 bytes		
Transaction Code	GR_RESPONSE(2401)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0
Box ID	SHORT	2	40
BrokerID	CHAR	5	42
Filler	CHAR	1	47
IP Address	CHAR	16	48
Port	LONG	4	64
Session Key	CHAR	8	68
Cryptographic Key	CHAR	32	76
Cryptographic IV (Initialization Vector)	CHAR	16	108

Field Name	Brief Description
Transaction Code	This field is the part of Message Header. The transaction code is 2401
Error Code	This field is the part of Message Header. Error Code will be set if the query is unsuccessful. Refer to List of Error Codes in Appendix.
Box ID	Exchange provided Box ID used for this connection
BrokerID	This field should contain the trading member ID
IP Address	IP address assigned by exchange
Port	Port Number given by exchange
Session Key	Session key to be used for authentication
Cryptographic Key	Cryptographic key for both the encryption and decryption of all messages between member application and allocated Gateway Server.
Cryptographic IV (Initialization Vector)	Cryptographic IV (Initialization Vector) for both the encryption and decryption of all messages between member application and allocated Gateway Server.

MS_GR_RESPONSE (New encryption)

Structure Name	MS_GR_RESPONSE		
Packet Length	136 bytes		
Transaction Code	GR_RESPONSE (2401)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER	STRUCT	40	0
Box ID	SHORT	2	40
BrokerID	CHAR	5	42
Filler	CHAR	1	47
IP Address	CHAR	16	48
Port	LONG	4	64
Session Key	CHAR	8	68
Cryptographic Key	CHAR	32	76
Static Cryptographic IV	CHAR	8	108
Dynamic Cryptographic IV	LONG LONG	8	116
Cryptographic Additional Key	CHAR	12	124

Field Name	Brief Description
Transaction Code	This field is part of Message Header. The transaction code is 2401

Field Name	Brief Description
Error Code	This field is the part of Message Header. Error Code will be set if the query is unsuccessful. Refer to List of Error Codes in Appendix.
Box ID	Exchange provided Box ID used for this connection
BrokerID	This field should contain the trading member ID
IP Address	IP address assigned by exchange
Port	Port Number given by exchange
Session Key	Session key to be used for authentication
Cryptographic Key	Cryptographic key for both the encryption and decryption of all messages between member application and allocated Gateway Server.
Static Cryptographic IV	Static Cryptographic IV for both the encryption and decryption of all messages between member application and allocated Gateway Server.
Dynamic Cryptographic IV	Dynamic Cryptographic IV for both encryption and decryption of all messages between member application and allocated Gateway Server.
Cryptographic Additional Key	Cryptographic Additional Key for both the encryption and decryption of all messages between member application and allocated Gateway Server.

Secure Box Registration Request

SECURE_BOX_REGISTRATION_REQUEST

Structure Name	MS_ SECURE_BOX_REGISTRATION_REQUEST_IN		
Packet Length	42 bytes		
Transaction Code	SECURE_BOX_REGISTRATION_REQUEST_IN (23008)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0
BoxId	SHORT	2	40

Field Name	Brief Description
Transcode	This field is the part of Message Header. The transaction code is 23008
BoxId	Exchange provided Box ID to be used for this connection

Secure Box Registration Response

SECURE_BOX_REGISTRATION_RESPONSE

Structure Name	MS_ SECURE_BOX_REGISTRATION_RESPONSE_OUT		
Packet Length	40 bytes		
Transaction Code	SECURE_BOX_REGISTRATION_REQUEST_OUT (23009)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0

Field Name	Brief Description
Transcode	This field is the part of Message Header. The transaction code is 23009
ErrorCode	This field is the part of Message Header. Error Code will be set if the query is unsuccessful. *Refer to List of Error Codes in Appendix*

Box Sign on Request

MS_BOX_SIGN_ON_REQUEST_IN

Structure Name	MS_BOX_SIGN_ON_REQUEST_IN		
Packet Length	60 bytes		
Transaction Code	BOX_SIGN_ON_REQUEST_IN(23000)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0
BoxId	SHORT	2	40
BrokerID	CHAR	5	42
Reserved	CHAR	5	47
SessionKey	CHAR	8	52

Field Name	Brief Description
Transcode	This field is the part of Message Header.The transaction code is 23000
BoxId	Exchange provided Box ID to be used for this connection

BrokerID	This field should contain the trading member ID
SessionKey	Session key received in GR_RESPONSE(2401)

Box Sign on Response

MS_BOX_SIGN_ON_REQUEST_OUT

Structure Name	MS_BOX_SIGN_ON_REQUEST_OUT		
Packet Length	52 bytes		
Transaction Code	BOX_SIGN_ON_REQUEST_OUT(23001)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0
BoxId	SHORT	2	40
Reserved	CHAR	10	42

Field Name	Brief Description
Transaction Code	This field is the part of Message Header. The transaction code is 23001
Error Code	This field is the part of Message Header. Error Code will be set if the query is unsuccessful. Refer to List of Error Codes in Appendix.
BoxId	Exchange provided Box ID used for this connection

SignOn In

Members systems must send other messages immediately using existing protocol defined in Chapter 3 of Protocol Document. A few fields in the Logon message have to be populated differently for direct connection:

Field Name	Brief Description
TransactionCode	The transaction code is MS_SIGNON (2300).
ShowIndex	'T' = to use Trimmed-NNF protocol with Total Traded Quantity and Value Data Type Change Note: Only Trimmed-NNF protocol is supported by Direct Interface

Note: Rest of the fields of SIGNON IN to be populated as prescribed in Chapter 3 of protocol document.

If authentication information is correct, member systems will receive a successful SIGNON_OUT response.

How to Logoff?

To logoff from the exchange trading system, there is no change and use the existing protocol defined in Chapter 3 of protocol document.

Heartbeat exchange

Member systems must send heartbeat message to the exchange trading system during periods of inactivity and the same will be echoed back by the exchange. Trading Host will consider the member system as inactive after missing two heartbeats in succession and disconnect the socket connection. If a member sends more than one heartbeat message within the same interval, the exchange will disregard the extra messages and increase the drop counter by 1 for every additional heartbeat message received. The drop counter tracks the number of ignored heartbeat messages from a member connection. If the drop counter reaches the threshold value set by the exchange, the member connection will be disconnected from the exchange trading system. The exchange will also logoff the box id of the member, which means that all the users linked to that box id will be disconnected. The drop counter will be reset to zero for the respective box id after every disconnection.

Heartbeats will carry following data in MessageData segment of the message. Heartbeat is to be sent only if there is inactivity for 30 seconds. The format is MESSAGE_HEADER with following detail.

HEARTBEAT

Structure Name	HEARTBEAT
Packet Length	40 bytes
Transaction Code	23506

Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0

Field Name	Description
TransactionCode	The transaction code is (23506).

Recovering from disconnections

If member system detects a loss of TCP connection with the exchange trading system, please perform the same operations for starting a fresh login given above.

Performing Trading activities

Once authenticated connection is successfully established, member systems can send any business message to exchange as described in NNF protocol documents. Care should be taken to use MESSAGE_HEADER described in this chapter wherever applicable in front of business messages.

Connection Termination

When connection is terminated by exchange, BOX_SIGN_OFF (20322) message with appropriate error code will be sent.

Box Sign Off

MS_BOX_SIGN_OFF

Structure Name	MS_BOX_SIGN_OFF		
Packet Length	42 bytes		
Transction code	BOX_SIGN_OFF (20322)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Message Header structure)	STRUCT	40	0
BoxId	SHORT	2	40

Field Name	Brief Description
TransactionCode	This field is the part of Message Header.The transaction code is 20322.

Error Code	This field is the part of Message Header.Error Code will be set if the query is unsuccessful. Refer to List of Error Codes in Appendix.
BoxId	Exchange provided Box ID used for this connection

Chapter 11 Exception Handling

Introduction

NSE's trading system constitutes of multiple matching engines (streams). Each stream hosts a range of contracts on which trading is allowed. In case of an exception single/multiple streams will get impacted. It is necessary that relevant information is disseminated in such events so that necessary action can be taken at member's end to bring their systems into a consistent state.

Exception handling:

1. At the start of the outage message will be sent on broadcast channel with StreamNumber and status as 1 (start of outage) and members may get disconnected from the exchange (Member can also receive this message through journal download).
2. On receiving message in step 1, members should clear outstanding orders at their end for the respective streams. Exchange would also cancel all the outstanding orders and no cancellation messages will be sent for these orders.
3. Once exchange has restored the stream, message will be sent on broadcast channel with StreamNumber and status as 0 (end of outage) (Member can also receive this message through journal download).
4. On receiving the message in step 3, Members can reconnect to the exchange in case they have got disconnected in step 1.

Message structure

Message structure is as follows:

Structure Name	MS_BCAST_CONT_MESSAGE		
Packet Length	244 bytes		
Transaction Code	BCAST_CONT_MSG (5294)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
StreamNumber	SHORT	2	40
Status	SHORT	2	42
Reserved	CHAR	200	44

The following table provides details of the various fields present in above Message structure.

Field Name	Brief Description
StreamNumber	0 – All streams are impacted or impacted stream number (eg 1, 2, 3, 4...)
Status	1 – Start of outage 0 – End of outage
Reserved	Reserved for future use

DR 45 Initiative

NSE trading system provides high availability of its services by having identical setup available at NSE DR Site.

Please find below list of point to be considered in case of switchover to DR site

1. Members will have to reconnect to trading system, as they will be disconnected once the primary site is unavailable
2. Member should continue to use existing connectivity parameter for connecting to NSE trading system at DR site
3. Member on reconnecting at DR site will receive start of outage message as a part of journal download.

   The message sent in the following format

   (MS_BCAST_CONT_MESSAGE) (refer to Exception handling)
4. Exchange shall not carry forward outstanding orders from primary site to DR site and no cancellation messages will be sent for these orders. Accordingly members are advised to clear outstanding orders at their end.
5. Exchange shall publish streamwise trade number of the last trade (Exchange trade number) available at DR site. Member may note that streamwise trades upto the last trade number shall only be considered.
6. Exchange shall broadcast streamwise last trade number.

   The message sent in the following format

(MS_TRADER_INT_MSG) (refer to Interactive/broadcast messages sent from control)

7. Member shall be able to perform trade modification or trade cancellation on trades which are available at DR site.
8. In case member is connected after switchover, they will receive end of outage message. The message sent in the following format
   (MS_BCAST_CONT_MESSAGE) (refer to Exception handling)
   In case member is not connected, they will receive this message as a part of journal download post reconnecting to NSE trading system at DR site.
   The message sent in the following format
   (MS_BCAST_CONT_MESSAGE) (refer to Exception handling)
9. Journal download information before switchover shall not be available ,
10. Used limit value in User Order Value Limit (UOVL) and Branch Order Value Limit (BOVL) will be reset to zero after switchover to DR site.

Chapter 12 CM-BM Functionalities

Introduction

This section describes about functionalities available to corporate manager and branch manager users for risk management and admin related activities.

Branch Order Limit

Corporate manager can set limits on total value of buy/sell orders entered by specific branch within trading member's firm.
Branch order value limit will be applicable to users available in the branch.

Branch Order Value Limit Update Request

The format of the message is as follows:

Structure Name	BRANCH_ORDER_VAL_LIMIT_UPDATE		
Packet Length	104 bytes		
Transaction Code	BRANCH_ORDER_VAL_UPDATE_IN (5716)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
BrokerId	CHAR	5	40
Reserved	CHAR	25	45
Branch	SHORT	2	70
BRANCH_LIMIT	STRUCT	32	72

Structure Name	BRANCH_LIMIT		
Packet Length	32 bytes		
Field Name	Data Type	Size in Byte	Offset
BranchBuyValueLimit	DOUBLE	8	0
Reserved	CHAR	8	8
BranchSellValueLimit	DOUBLE	8	16
Reserved	CHAR	8	24

Field Name	Brief Description
TransactionCode	The transaction code is BRANCH_ORDER_VAL_LIMIT_UPDATE_IN (5716)
BrokerId	This field should contain the Trading Member ID
Branch	This field should contain the branch number for which limit to be set
BranchBuyValueLimit	This field should contain branch buy limit to be set (in lakhs) Valid values: 0 to 9999999.99 This is to be multiplied by (100000*100) before sending to the trading system host
BranchSellValueLimit	This field should contain branch sell limit to be set (in lakhs) Valid values: 0 to 9999999.99 This is to be multiplied by (100000*100) before sending to the trading system host

Branch Order Value Limit Update Response

On successful branch limit updates, exchange will send Branch Order Limit Update Response to

- Corporate manager
- Branch manager(of branch id mentioned in request)

The structure is sent as follows:

BRANCH_ORDER_VAL_LIMIT_UPDATE (Refer to Branch Order Value Limit Request structure)

Field Name	Brief Description
TransactionCode	The transaction code is BRANCH_ORDER_LIMIT_UPDATE_OUT (5717)
ErrorCode	This field contains error code. If error code field value is zero (0) then user order value limit update is done successfully.

If branch order value limit update request is rejected by trading system, then ERROR RESPONSE (Refer Table 5) packet will be sent to user who has sent limit update request. Reason for rejection will be given by ErrorCode in the header.

User Order Limit

Corporate manager can set limit on total value of buy/sell orders entered by specific user within trading member's firm. Similarly, Branch manager can set limit on total value of buy/sell orders entered by specific user within the branch.

User Order Value Limit Update Request

The format of the message is as follows:

Structure Name	USER_ORDER_VAL_LIMIT_UPDATE		
Packet Length	142 bytes		
Transaction Code	USER_ORDER_VAL_UPDATE_IN (5719)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
BrokerId	CHAR	5	40
Reserved	CHAR	1	45
Branch	SHORT	2	46
Reserved	CHAR	26	48
UserId	LONG	4	74
USER_LIMITS	STRUCT	64	78

Structure Name	USER_LIMITS		
Packet Length	64 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	16	0
UserOrderBuyValueLimit	DOUBLE	8	16
Reserved	CHAR	24	24
UserOrderSellValueLimit	DOUBLE	8	48
Reserved	CHAR	8	56

Field Name	Brief Description
TransactionCode	The transaction code is USER_ORDER_LIMIT_UPDATE_IN (5719)
BrokerId	This field should contain the Trading Member ID

Field Name	Brief Description
Branch	This field should contain the branch number of user for which limit to be set
UserId	This field should contain the user ID of the user for which limit to be set
UserOrderBuyValueLimit	This field should contain user buy limit to be set (in lakhs) Valid values: 0 to 9999999.99 This is to be multiplied by (100000*100) before sending to the exchange trading system
UserOrderSellValueLimit	This field should contain user sell limit to be set (in lakhs) Valid values: 0 to 9999999.99 This is to be multiplied by (100000*100) before sending to the exchange trading system

User Order Value Limit Update Response

On successful user limit updates, exchange will send User Order Limit Update Response to

- user who has sent limit update request
- user for which limit has been set
- Corporate manager (if branch manager tries to update limit for user within branch).

The structure is sent as follows:

USER_ORDER_VAL_LIMIT_UPDATE (Refer to User Order Value Limit Request structure)

Field Name	Brief Description
TransactionCode	The transaction code is USER_ORDER_LIMIT_UPDATE_OUT (5720)
ErrorCode	This field contains error code. If error code field value is zero (0) then user order value limit update is done successfully.

If user order value limit update request is rejected by trading system, then ERROR RESPONSE (Refer Table 5) packet will be sent to user who has sent limit update request. Reason for rejection will be given by ErrorCode in the header.

Order Limit

This functionality provides facility to specify maximum quantity per order and maximum value per order that user can enter in order entry/order modification request.

Corporate manager can set limit on order quantity and order value of an order, entered by user within trading member's firm. Similarly Branch manager can set limit on order quantity and order value of an order entered by user within the branch.

Order Limit Update Request

The format of the message is as follows:

Structure Name	ORDER_LIMIT_UPDATE		
Packet Length	68 bytes		
Transaction Code	DEALER_LIMIT_UPDATE_IN (5721)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
BrokerId	CHAR	5	40
Reserved	CHAR	1	45
UserId	LONG	4	46
OrderQtyLimit	DOUBLE	8	50
OrderValLimit	DOUBLE	8	58
Reserved	CHAR	2	66

Field Name	Brief Description
TransactionCode	The transaction code is DEALER_LIMIT_UPDATE_IN (5721)
BrokerId	This field should contain the Trading Member ID
UserId	This field should contain the User ID for which limit to be set
QuantityValLimit	This field should contain Order Quantity limit to be Set Valid values : 1 to 999999999
OrderValLimit	This field should contain Order Limit to be Set in lakhs Valid values: 0 to 9999999.99 This is to be multiplied by (100000*100) before sending to the trading system host

Order Limit Update Response

On successful order limit updates, exchange will send Order Limit Update Response to

- user who has sent limit update request
- user for which limit has been set
- Corporate manager (if branch manager tries to update limit for user within branch).

The structure is sent as follows:

ORDER_LIMIT_UPDATE (Refer to Order Limit Update_Request structure)

Field Name	Brief Description
TransactionCode	The transaction code is DEALER_LIMIT_UPDATE_IN (5722)
ErrorCode	This field contains error code. If error code field value is zero (0) then order limit update is done successfully.

If order limit update request is rejected by trading system, then ERROR RESPONSE (Refer Table 5) packet will be sent to user who has sent limit update request. Reason for rejection will be given by ErrorCode in the header.

Reset UserId

This functionality enables the Corporate Manager to terminate the active session for users within trading member's firm. Similarly, Branch Manager can terminate the active session for users within the branch.

User Reset Request

The format of the message is as follows:

SIGNON IN (Refer to Logon Structure in Chapter 3)

Field Name	Brief Description
TransactionCode	The transaction code is RESET_USERID_REQ (5723).
UserId	This field should contain User ID of user to be reset. This field accepts numbers only.

User Reset Response

In below mentioned scenarios, exchange trading system will send User Reset Response to user who has sent user reset request,

- On Successful user session reset

The structure is sent as follows:

SIGNON IN (Refer to Logon Structure in Chapter 3)

Field Name	Brief Description
TransactionCode	The transaction code is RESET_USERID_RESP (5724).
ErrorCode	This field contains error code. If error code field value is zero (0) then reset user is done successfully.

If User Reset request is rejected by exchange trading system, then ERROR RESPONSE (Refer Table 5) packet will be sent to user who has sent user reset request. Reason for rejection will be given by ErrorCode in the header.

Reset Password

Corporate manager can reset password of users within trading member's firm.

- The user's password will reset to "Neat@CM1" i.e. default password.
- User whose password is to be reset should be 'Disabled' or 'Inactive'
- On resetting the password of disabled user, status of the user will be changed to inactive.
- The Corporate Manager will not be allowed to reset his own password.

User Password Reset Request

The format of the message is as follows:

Structure Name	RESET_PASSWORD		
Packet Length	58 bytes		
Transaction Code	RESET_PASSWORD_IN (5738)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	14	44

Field Name	Brief Description
TransactionCode	The transaction code is RESET_PASSWORD_IN (5738)
UserId	This field should contain user id for which password to be reset

User Password Reset Response

In below mentioned scenarios, exchange trading system will send User password reset response to user who has sent user password reset request

- On Successful user password reset
- If user password reset request is rejected by exchange trading system (Reason for rejection will be given by ErrorCode in the header.)

The structure is sent as follows:
RESET_PASSWORD (Refer to User Password Reset Request structure)

Field Name	Brief Description
TransactionCode	The transaction code is RESET_PASSWORD_OUT (5739)
ErrorCode	This field contains error code. If error code field value is zero (0) then reset password for user is done successfully. If error code field value is non-zero, then reset password request for user is rejected. Refer to List of Error Codes in Appendix.

Cancel On Logout (COL) Status

This functionality if enabled provides facility to traders to cancel all their outstanding orders when user logs off from exchange trading system.

Corporate manager can enable/disable COL status for the users within trading member's firm.

User COL Status Update Request

The format of the message is as follows:

Structure Name	COL_ USER_STATUS_CHANGE_REQ		
Packet Length	52 bytes		
Transaction Code	COL_USER_STATUS_CHANGE _IN (5790)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
ColoUserBit	CHAR	1	44
Reserved	CHAR	7	45

Field Name	Brief Description
TransactionCode	The transaction code is COL_USER_STATUS_CHANGE_IN (5790)
UserId	This field should contain user id for which COL status to be set
ColoUserBit	This field should contain user's COL status to be set. It should contain one of the following values. • '0' for Disable COL status • '1' for Enable COL status

User COL Status Update Response

In below mentioned scenarios, exchange trading system will send User COL Status Update response to user who has sent status update request

- On Successful COL status updates
- If User COL status update request is rejected by exchange trading system (Reason for rejection will be given by ErrorCode in the header.)

The structure is sent as follows:

Structure Name	COL_USER_STATUS_CHANGE_RESP		
Packet Length	46 bytes		
Transaction Code	COL_USER_STATUS_CHANGE _OUT (5791)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
ColoUserBit	CHAR	1	44
Reserved	CHAR	1	45

Field Name	Brief Description
TransactionCode	The transaction code is COL_USER_STATUS_CHANGE_OUT (5791)
ErrorCode	This field contains error code. If error code field value is zero (0) then user's COL status update is done successfully. If error code field value is non-zero, then request for user's COL status update is rejected. Refer to List of Error Codes in Appendix.
UserId	This field will contain user id for which COL status is set
ColoUserBit	This field will contain user's COL status is set. It will contain one of the following values. • '0' for Disable COL status • '1' for Enable COL status

Also, in case of successful COL status update, trading system will send interactive message to

- user who has sent status update request
- user for which status has been updated
- Branch manager (if the status update is done for the dealer under his branch).
- Other Branch managers of same branch if status update is done for Branch manager.

The message sent will be of the following format:
MS_TRADER_INT_MSG *(Refer to Interactive/Broadcast Messages Sent from Control)*

The following table provides the details of the various fields present in the MS_TRADER_INT_MSG Structure.

Field Name	Brief Description
TransactionCode	The transaction code is CTRL_MSG_TO_TRADER (5295).
BroadCastMessage Length	This field contains Message Length
BroadCastMessage	This field contains actual Message

Trade Cancellation Status

Corporate manager can enable/disable Trade Cancellation Status for the users within trading member's firm.

If Trade Cancellation status for user is enabled, then user will be allowed to send Trade cancellation request to exchange trading system.

User TRD-CXL Status Update Request

The format of the message is as follows:

Structure Name	USER_ TRD_MOD_CXL_CHANGE_REQ		
Packet Length	52 bytes		
Transaction Code	USER_ TRD_MOD_CXL_CHANGE _IN (5792)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
TrdModCxlBit	CHAR	1	44
Reserved	CHAR	7	45

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD_CXL_CHANGE_IN (5792)
AlphaChar	To identify status change for Trade Cancellation, AlphaChar values to be set as below • AlphaChar[0] = 'T'

Field Name	Brief Description
	• AlphaChar[1] = 'X'
UserId	This field should contain user id for which trade cancel status to be set.
TrdModCxlBit	This field should contain user's Trade Cancellation Status to be set. It should contain one of following values, • 'Y' for Enable Trade Cancellation Status • 'N' for Disable Trade Cancellation Status

User TRD-CXL Status Update Response

On successful Trade CXL status updates, exchange trading system will send User TRD-CXL Status Update Response to the user who has sent status update request as well as to the user for which TRD-CXL status has been set.

If User TRD-CXL status update request is rejected by exchange trading system, then status update response packet will be sent to user who has sent status update request.
Reason for rejection will be given by ErrorCode in the header.

The structure is sent as follows:

Structure Name	USER_TRD_MOD_CXL_CHANGE_RESP		
Packet Length	46 bytes		
Transaction Code	USER_TRD_MOD_CXL_CHANGE _OUT (5793)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
TrdModCxlBit	CHAR	1	44
Reserved	CHAR	1	45

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD_CXL_CHANGE_OUT (5793)
ErrorCode	This field contains error code.

Field Name	Brief Description
	If error code field value is zero (0) then user's Trade Cxl status update is done successfully. If error code field value is non-zero, then request for user's Trade Cxl status update is rejected. Refer to List of Error Codes in Appendix.
UserId	This field will contain user id for which trade cancel status is set.
TrdModCxlBit	This field will contain user's Trade Cancellation Status is set. It will contain one of following values, • 'Y' for Enable Trade Cancellation Status • 'N' for Disable Trade Cancellation Status

Also, in case of successful TRD-CXL status update, trading system will send interactive message to

- user who has sent status update request
- user for which status has been updated
- Branch manager (if the status update is done for the dealer under his branch).
- Other Branch managers of same branch if status update is done for Branch manager

The message sent will be of the following format:
MS_TRADER_INT_MSG *(Refer to Interactive/Broadcast Messages Sent from Control)*

The following table provides the details of the various fields present in the MS_TRADER_INT_MSG Structure.

Field Name	Brief Description
TransactionCode	The transaction code is CTRL_MSG_TO_TRADER (5295).
BroadCastMessage Length	This field contains Message Length
BroadCastMessage	This field contains actual Message

Trade Modification Status

Corporate manager can enable/disable Trade Modification Status for the users within trading member's firm.

If Trade Modification status for user is enabled, then user will be allowed to send Trade modification request to exchange trading system.

User TRD-MOD Status Update Request

The message sent will be of the following format:

USER_ TRD_MOD_CXL_CHANGE_REQ (*Refer to User TRD-CXL Status Update Request structure*)

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD_CXL_CHANGE_IN (5792)
UserId	This field should contain user id for which trade modification status to be set.
TrdCxlBit	This field should contain user's Trade Modification Status to be set. It should contain one of following values, • 'Y' for Enable Trade Modification Status • 'N' for Disable Trade Modification Status

User TRD-MOD Status Update Response

On successful Trade MOD status updates, exchange trading system will send User TRD-MOD Status Update Response to the user who has sent status update request as well as to the user for which TRD-MOD status has been set.

If User TRD-MOD status update request is rejected by exchange trading system, then status update response packet will be sent to user who has sent status update request.
Reason for rejection will be given by ErrorCode in the header.

The message sent will be of the following format:

USER_ TRD_MOD_CXL_CHANGE_RESP (*Refer to User TRD-CXL Status Update Response structure*)

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD_CXL_CHANGE_OUT (5793)
ErrorCode	This field contains error code. If error code field value is zero (0) then user's Trade Mod status update is done successfully. If error code field value is non-zero, then request for user's Trade Mod status update is rejected. Refer to List of Error Codes in Appendix.
UserId	This field will contain user id for which trade modification status is set.
TrdModCxlBit	This field will contain user's Trade Modification Status is set. It will contain one of following values, • 'Y' for Enable Trade Modification Status • 'N' for Disable Trade Modification Status

Also, in case of successful TRD-MOD status update, trading system will send interactive message to

- user who has sent status update request
- user for which status has been updated
- Branch manager (if the status update is done for the dealer under his branch).
- Other Branch managers of same branch if status update is done for Branch manager

The message sent will be of the following format:
MS_TRADER_INT_MSG *(Refer to Interactive/Broadcast Messages Sent from Control)*

The following table provides the details of the various fields present in the MS_TRADER_INT_MSG Structure.

Field Name	Brief Description
TransactionCode	The transaction code is CTRL_MSG_TO_TRADER (5295).
BroadCastMessage Length	This field contains Message Length
BroadCastMessage	This field contains actual Message

Unlock User

Corporate manager can send unlock request for the users within trading member's firm.
As soon as User Unlock request reaches trading system, User Unlock Requested Response message is sent to user who has sent Unlock User Request. This in turn generates alert to NSE-Control user. This alert may be approved or rejected by exchange.

User Unlock Request

The format of the message is as follows:

Structure Name	USER_ADDR_UNLOCK_REQ		
Packet Length	68 bytes		
Transaction Code	USER_ADDR_UNLOCK_IN (5424)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	24	44

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_IN (5424)
UserId	This field should contain user id for which unlock request to be made

User Unlock Requested Response

This is an acknowledgement signifying that the User Unlock Request has reached the trading system. If any error is encountered in the User Unlock Request data, then appropriate error code will be set.

The structure is sent as follows:

Structure Name	USER_ADDR_UNLOCK_RESP		
Packet Length	44 bytes		
Transaction Code	USER_ADDR_UNLOCK_OUT (5425)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_OUT (5425)
ErrorCode	This field contains error code. If error code field value is zero (0) then user unlock request for user is made to exchange successfully. If error code field value is non-zero, then user unlock request for user is rejected. Refer to List of Error Codes in Appendix.

User Unlock Approval/Rejection Response

On approval of user unlock request by exchange trading system, exchange trading system will send user unlock response to user who has sent user unlock request.

The structure is sent as follows:

Structure Name	USER_ADDR_UNLOCK_APP_REJ_RESP		
Packet Length	44 bytes		
Transaction Code	USER_ADDR_UNLOCK_APPROVE_OUT (5575)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE HEADER (Refer Table 1)	STRUCT	40	0
UserId	LONG	4	40

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_APPROVE_OUT (5575)
ErrorCode	This field contains error code. If error code field value is non-zero, then user unlock request for user is rejected. Refer to List of Error Codes in Appendix.

On rejection of user unlock request by exchange trading system, exchange trading system will send user unlock response to user who has sent user unlock request,

The structure is sent as follows:

USER_ADDR_UNLOCK_REJECT RESP (Refer to User Unlock Approval/Rejection Response structure)

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_REJECT_OUT (5579)
ErrorCode	This field contains error code. If error code field value is non-zero, then user unlock request for user is rejected. Refer to List of Error Codes in Appendix.

Trading Member Level Kill Switch

This functionality provides a facility to Corporate Manager, to cancel all pending orders of all the users under trading member's firm at the same time.

Also, user can cancel all outstanding orders on particular security by specifying security information in request packet.

Member Level Kill Switch Request

The format of the message is as follows:

ORDER_ENTRY_REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is KILL_SWITCH_IN (2062).
User	This field should contain 0 for Trading Member level kill switch request.
SEC_INFO	For cancellation of all orders, Symbol and series fields should be set as blank. For cancellation of all orders on particular security, this structure should contain the Symbol and Series of the security.

Member Level Kill Switch Response

The Quick cancel out response is sent when the member level kill switch is requested by the corporate manager.

The message sent is as follows:

ORDER_ENTRY_REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description

TransactionCode	The transaction code is QUICK_CANCEL_OUT(2061)

Member Level Kill Switch Error Response

The kill switch error is sent when the request is rejected by the trading system. The reason for rejection will be given by the Error Code in the header.

The message sent is as follows:

ORDER_ENTRY_REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_ERROR (2231).
ErrorCode	This field contains the error number. Refer to List of Error Codes in Appendix.

User Level Kill Switch

This functionality provides a facility to Corporate Manager and Branch Manager to cancel all of their orders at the same time.

Also, they can cancel all of their outstanding orders on particular security by specifying security information in request packet.

User Level Kill Switch Request

The format of the message is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is KILL_SWITCH_IN (2062).
User	This field should contain the user id for which orders should be cancelled.
SEC_INFO	For cancellation of all orders, Symbol and series fields should be set as blank. For cancellation of all orders on particular security, this structure should contain the Symbol and Series of the security.

User Level Kill Switch Response

The Quick cancel out response is sent when the kill switch is requested by the user. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is QUICK_CANCEL_OUT(2061)

User Level Kill Switch Error Response

The kill switch error is sent when the request is rejected by the trading system. The reason for rejection will be given by the Error Code in the header. The message sent is as follows:

ORDER ENTRY REQUEST (Refer to *Order Entry Request* in Chapter 4)

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_ERROR (2231).

Order and Trade

Order Entry

This functionality enables the Corporate Manager and Branch Manager to place orders in the market.

Please refer Trimmed Order Entry Request Structure from Trimmed Structures section for further details.

Order Modification

This functionality enables the Corporate Manager and Branch Manager to modify their unmatched orders by specifying the order number of the order to be modified. All order types except Auction can be modified. Corporate Manager can modify his own order and also for his Branch Manager and Dealers/Traders. Branch Manager can modify his own order and also for his Dealers/Traders.

Please refer Trimmed Order Mod/Cxl Request Structure from Trimmed Structures section for further details.

Order Cancellation

The functionality enables the Corporate Manager and Branch Manager to cancel their any unmatched/partially matched order by specifying the order number. Corporate Manager can

cancel his own order and also for this Branch Managers and Dealers/Traders. Branch Manager can cancel his own order and also for his Dealers/Traders.

Please refer Trimmed Order Mod/Cxl Request Structure from Trimmed Structures section for further details.

Trade Modification

This functionality enables the Corporate Manager and Branch Manager to modify their trades. Only account number modification is allowed. Corporate Manager can modify his own trade and also for his Branch Managers and Dealers/Traders. Branch Manager can modify his own trade and also for his Dealers/Traders.

Please refer *Trade Modification* section (in Chapter 4) for further details.

Trade Cancellation

This functionality enables the Corporate Manager and Branch Manager to cancel their trades. But to cancel a trade, both the parties of the trade must request for trade cancellation. Corporate Manager can cancel his own trade and also for his Branch Managers and Dealers/Traders. Branch Manager can cancel his own trade and also for his Dealers/Traders.

Please refer *Trade Cancellation* section (in Chapter 4) for further details.

Close Out Order Entry

This facility is provided to trading members in closeout mode to place an opposite order with intent to reduce the open positions. Close out orders entered shall be Regular Lot (RL) and Immediate or Cancel (IOC) orders.

Clearing members can place order entry on behalf of the linked trading members. A close out order entry can be placed by Corporate Manager of member type PCM (Professional clearing member) or PCM+TM (Professional clearing member which is also a Trading member).

Order Confirmation/Cancellation messages shall be sent to Corporate Manager of clearing member and Corporate Manager of trading member, on whose behalf the order was placed.

If the order is rejected by the close out system, the rejection message shall be sent only to the clearing member. If the order is matched, the trade confirmation shall be sent to the clearing member and the trading member on whose behalf order was placed.

The format for closeout order entry please refer Trimmed Order Entry Request Structure from Trimmed Structures section for further details.

The UserId and BrokerId field has to be the one given below in case of close out order entry.

Field Name	Brief Description
UserId	This field should be zero.
BrokerId	This field should contain the trading member ID on whose behalf the order is being placed

Chapter 13 Immediate Order Acknowledgement Message

Background

NSE provides confirmation or rejection for all order-related packets received from its trading members. The Exchange now proposes to introduce an additional, immediate acknowledgment for order-related messages. This section outlines the implementation details of this new feature.

Overview

The trading system accepts incoming orders from users and attempts to match them against existing orders in the order book maintained at the host end. Post which the host end generates and sends a confirmation or rejection message to the respective trading members. As an enhancement, the Exchange proposes the introduction of an additional acknowledgment message that will be sent immediately upon receipt of the order at the Exchange. This immediate acknowledgment serves as an indication that the order has been successfully received by the Exchange. The existing process of sending a final confirmation or rejection message will continue unchanged.

Implementation Approach

Member applications that wish to receive immediate acknowledgments must send the designated immediate acknowledgment request transcodes for all applicable order-related transactions, as described below. Upon receiving such a request, the Exchange will respond with an immediate acknowledgment, indicating that the order has been successfully received at the Exchange. Subsequently, the standard order confirmation or rejection message will be sent once the order is validated and processed by the trading system.

This new request must be transmitted to the Exchange via a separate communication channel (New port on Gateway router server) and must utilize the new GCM-encrypted channel with new additional authentication tag as mentioned in chapter 10.

Immediate Ack Request

Transaction Code	Code	Structure
TRIMMED_BOARD_LOT_ACK_IN	20400	*Refer to ORDER_ENTRY_REQUEST_TR*
TRIMMED_ORDER_MOD_ACK_IN TRIMMED_ORDER_CANCEL_ACK_IN	20402 20404	*Refer to ORDER_OM_REQUEST_TR*

Immediate Ack Response

Table 61 MS_ACK_RESPONSE

Structure Name	MS_ACK_RESPONSE		
Packet Length	22 bytes		
Transaction Code	QUICK_ACK_OE_RESP (20401)/ QUICK_ACK_OM_RESP (20403)/ QUICK_ACK_OC_RESP (20405)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
TraderId	LONG	4	2
TimeStamp	LONG LONG	8	6
Reference	LONG	4	14
ErrorCode	SHORT	2	18
MessageLength	SHORT	2	20

Field Name	Brief Description
TransactionCode	The transaction code is QUICK_ACK_OE_RESP (20401), QUICK_ACK_OM_RESP (20403), QUICK_ACK_OC_RESP (20405)
TraderId	This field should contain the ID of the user. This field accepts only numbers.
TimeStamp	This field will contain a unique value for current activity. Currently the same shall be in nanoseconds and stamped at the Gateway.
Reference	This field value will be echoed back as the value received in respective order packets in filler/reference field.
ErrorCode	This contains the error number. *Refer to List of Error Codes in Appendix.*
MessageLength	This field is set to the length of the entire message

Co-Existence Approach

Immediate ack request transcodes will co-exist with the existing order-related request transcodes. Immediate ack request transcodes described above will be accepted exclusively via separate communication channel (New port on Gateway router server) that supports enhanced GCM-based encryption with authentication.

Members who wish to continue to use the existing order message formats will also have the option to migrate to the enhanced GCM-based authentication encryption. However, this migration must be implemented over a separate connection or port, distinct from the one used for the existing setup. Details of all port information will be communicated via separate circular.

To ensure co-existence for all members, we will provide three different ports for the following three scenarios.

Scenarios	Channel	Transcodes
Existing encryption	Existing Port of Gateway Router	Existing order messages
New encryption with authentication	New Port1 of Gateway Router	Existing order messages
Immediate Acknowledgement with new encryption methodology	New Port2 of Gateway Router	New Immediate Ack Request messages

Appendix

Please note the details in appendix are also directly or indirectly referenced in CM_DROP_COPY_PROTOCOL document. Any change here may also impact the Order Drop Copy functionality.

List of Error Codes

Error Code ID	Error Code Value	Description of Error Code
ERR_MARKET_NOT_OPEN	16000	The trading system is not available for trading.
ERR_INVALID_USER_TYPE	16001	Invalid User Type OR Reset User Password not requested by Corporate manager
ERR_BAD_TRANSACTION_CODE	16003	Erroneous transaction code received.
ERR_USER_ALREADY_SIGNED_ON	16004	User already signed on.
ERR_INVALID_SIGNON	16006	Invalid Box/User sign-on, please try again.
ERR_SIGNON_NOT_POSSIBLE	16007	Signing on to the trading system is restricted. Please try later on.
ERR_INVALID_SYMBOL	16012	Invalid symbol/series.
ERR_INVALID_ORDER_NUMBER	16013	Invalid order number
NOT_YOUR_FILL	16015	Invalid trade cancel request.
ERR_SECURITY_NOT_AVAILABLE	16035	Security is unavailable for trading at this time. Please try later.
ERR_INVALID_BROKER_OR_BRANCH	16041	Trading Member does not exist in the system.
ERR_USER_NOT_FOUND	16042	Dealer does not exist in the system.
ERR_TRD_MOD_REJ_END_OF_DAY_PROCESSING_STARTED	16050	Trade modification request rejected as end of the day processing started.
FUNCTION_NOT_AVAILABLE	16052	When Preopen trade cancel request is rejected

Error Code ID	Error Code Value	Description of Error Code
		OR BOVL/UOVL Limits not allowed to be set as unlimited OR BOVL update not requested by Corporate Manager OR Inconsistent data for BOVL update OR Branch Manager not allowed UOVL update for self/CM/other BM/users of other branch. OR Branch Manager not allowed Dealer Limit update for self. OR User Unlock Request not requested by Corporate Manager OR User Unlock Request not allowed for Corporate Manager OR User level COL disabled
ERR_PASSWORD_HAS_EXPIRED	16053	Your password has expired, must be changed.
ERR_INVALID_BRANCH	16054	Branch does not exist in the system. OR Inconsistent data for UOVL update
ERR_PROGRAM_ERROR	16056	Program error.
ORDER_NOT_FOUND	16060	Modified/Cancelled order not found
ERROR_INVALID_STATUS_CHANGE	16063	Requested status change not allowed
ERROR_NOTHING_CHANGED	16070	Data has not changed
ERR_INVALID_BUYER_USER_ID	16098	Invalid trader ID for buyer.

Error Code ID	Error Code Value	Description of Error Code
ERR_INVALID_SELLER_USER_ID	16099	Invalid trader ID for seller.
ERR_INVALID_SYSTEM_VERSION	16100	Your system version has not been updated.
ERR_SYSTEM_ERROR	16104	System could not complete your transaction - ADMIN notified.
ERR_MOD_CAN_REJECT	16115	Order Modification/ Cancellation rejected by the system.
ERR_CANT_COMPLETE_YOUR_REQUEST	16123	System not able to complete your request. Please try again.
ERR_USER_IS_DISABLED	16134	This Dealer is disabled. Please call the Exchange.
OE_INVALID_STOCK_STATUS	16145	Security is not eligible to trade in Preopen.
ERR_INVALID_USER_ID	16148	Invalid Dealer ID entered.
ERR_INVALID_TRADER_ID	16154	Invalid Trader ID entered.
ERR_ATO_IN_OPEN	16169	Order priced ATO cannot be entered when a security is open.
ORD_NOT_ALLOWED_IN_PREOPEN	16197	Order Entry or Modification not allowed in preopen.
ERROR_PRO_PARTICIPANT_INVALID	16233	Proprietary requests cannot be made for participant.
INVALID_PRICE	16247	Invalid price in the price field.
ERR_TRADE_MOD_DIFF_VOL	16251	Trade modification with different quantities is received.
CXLD_TRADE_MOD_REQUEST	16252	Cancelled the trade modify request.
OE_DELETED_BUT_EXISTS	16260	Record is there in master file but delete flag is set.
ERROR_ALREADY_DELETED	16264	The member has already been deleted.
ERR_NOT_FOUND	16273	Record does not exist.
ERR_MARKETS_CLOSED	16278	The markets have not been opened for trading.

Error Code ID	Error Code Value	Description of Error Code
ERR_SECURITY_NOT_ADMITTED	16279	The security has not yet been admitted for trading.
ERR_SECURITY_MATURED	16280	The security has matured.
ERR_SECURITY_EXPELLED	16281	The security has been expelled.
ERR_QUANTITY_EXCEEDS_ISSUED_CAPITAL	16282	The order quantity is greater than the issued capital.
ERR_PRICE_NOT_MULT_TICK_SIZE	16283	The order price is not multiple of the tick size.
ERR_PRICE_EXCEEDS_DAY_MIN_MAX	16284	The order price is out of the day's price range.
ERR_BROKER_NOT_ACTIVE	16285	The broker is not active.
ERROR_INVALID_SYSTEM_STATUS	16300	System is in a wrong state to make the requested change.
OE_AUCTION_PENDING	16303	Request denied. Pending auctions.
ERR_ QUANTITY_FREEZE_CANCELLED	16307	The order is canceled due to quantity freeze.
ERR_PRICE_FREEZE_CANCELLED	16308	The order is canceled due to price freeze.
AON_VOLUME_NOT_ENOUGH	16310	AON volume not enough
ERR_SOLICITOR_PERIOD_OVER	16311	The Solicitor period for the Auction is over.
ERR_COMPETITIOR_PERIOD_OVER	16312	The Competitor period for the Auction is over.
OE_AUC_PERIOD_GREATER	16313	The Auction period will cross Market Close time.
OE_AUC_NOT_CAN	16314	The Auction cannot be cancelled.
ERR_LIMIT_WORSE_TRIGGER	16315	The limit price is worse than the trigger price.
ERR_TRG_PRICE_NOT_MULT_TICK_SIZE	16316	The trigger price is not a multiple of tick size.
ERR_NO_AON_IN_LIMITS	16317	AON attribute not allowed.
ERR_NO_MF_IN_LIMITS	16318	MF attribute not allowed.

Error Code ID	Error Code Value	Description of Error Code
ERR_NO_AON_IN_SECURITY	16319	AON attribute not allowed at security level.
ERR_NO_MF_IN_SECURITY	16320	MF attribute not allowed at security level.
ERR_MF_EXCEEDS_DQ	16321	MF quantity is greater than Disclosed quantity.
ERR_MF_NOT_MULT_BOARD_LOT	16322	MF quantity is not a multiple of regular lot.
ERR_MF_EXCEEDS_ORIGINAL_ QUANTITY	16323	MF quantity is greater than Original quantity.
ERR_DQ_EXCEEDS_ORIGINAL_ QUANTITY	16324	Disclosed quantity is greater than Original quantity.
ERR_DQ_NOT_MULT_BOARD_LOT	16325	Disclosed quantity is not a multiple of regular lot.
ERR_GTD_EXCEEDS_LIMIT	16326	GTD is greater than that specified at System.
OE_QUANTITY_GERATER_RL	16327	Quantity is greater than Regular lot size.
ERR_QUANTITY_NOT_MULT_BOARD_L OT	16328	Quantity is not a multiple of regular lot.
ERR_BROKER_NOT_PERMITTED_IN_M KT	16329	Trading Member not permitted in the market.
ERR_SECURITY_IS_SUSPENDED	16330	Security is suspended.
CXL_REMAIN_ACTIVE_ORDER	16332	Remaining passive order has to be cancelled.
ERR_BRANCH_LIMIT_EXCEEDED	16333	Branch Order Value Limit is exceeded.
OE_ORD_CAN_CHANGED	16343	The order to be cancelled has changed.
OE_ORD_CANNOT_CANCEL	16344	The order cannot be cancelled.
OE_INIT_ORD_CANCEL	16345	Initiator order cannot be cancelled.
OE_ORD_CANNOT_MODIFY	16346	Order cannot be modified.

Error Code ID	Error Code Value	Description of Error Code
ERR_TRADING_NOT_ALLOWED	16348	Trading is not allowed in this market.
CHG_ST_EXISTS	16363	New status requested should not be same as existing one.
OE_SECURITY_IN_PREOPEN	16369	The security status is preopen.
ERR_USER_TYPE_INQUIRY	16372	Order entry not allowed for user as it is of inquiry type.
ERR_SOLICITION_NOT_ALLOWED	16379	The broker is not allowed to enter soliciting orders.
ERR_AUCTION_FINISHED	16383	Trading in this auction is finished.
ERR_NO_TRADING_IN_SECURITY	16387	Security is not allowed to trade in this market.
ERR_FOK_ORDER_CANCELLED	16388	When Preopen unmatched orders are cancelled by the system after preopen session ends. When normal market unmatched orders are cancelled by the system if order collection phase is planned after circuit hit. When IOC unmatched orders are cancelled by the system.
ERR_TURNOVER_LIMIT_NOT_SET	16392	Turnover limit not provided. Please contact Exchange.
ERR_DQ_EXCEEDS_LIMIT	16400	DQ has exceeded limit set in control.
ERR_WRONG_LOGIN_ADDRESS	16403	You are trying to sign on from a different location. Sign on is not allowed.
ERR_ADMIN_SUSP_CANCELLED	16404	Order is cancelled due to freeze admin suspension.
ERR_INVALID_PRO_CLIENT	16411	Pro-client can be either Pro or Client only.

Error Code ID	Error Code Value	Description of Error Code
ERR_INVALID_NEW_VOLUME	16412	New volume should be less than the traded volume.
ERR_INVALID_BUY_SELL	16413	Requested by can be BUY or SELL or BOTH.
ERR_INVALID_INST	16414	Invalid combination of book type and instructions (order_type).
ERR_INVALID_ORDER_PARAM	16415	Invalid combination of MF / AON / Disclosed Volume.
ERR_INVALID_CP_ID	16416	Invalid counter broker Id.
ERR_NNF_REQ_EXCEEDED	16417	Number of NNF requests exceeded.
ERR_INVALID_ORDER	16418	Order entered has invalid data.
ERR _CXLED_TRADE_CXL_REQ	16419	Cancelled trade cancel request.
ERR_INVALID_ALPHA_CHAR	16420	Alpha char must be the same as first two chars of symbol.
ERR_TRADER_CANT_INIT_AUCTION	16421	Only control can initiate auctions, not trader.
ERR_INVALID_BOOK_TYPE	16422	Book type should be between 1(RL) and 7(AU).
ERR_INVALID_TRIGGER_PRICE	16423	Invalid trigger price entered.
ERR_INVALID_MSG_LENGTH	16424	Message length is invalid.
ERR_INVALID_PARTICIPANT	16425	Participant does not exist.
ERR_PARTICIPANT_AND_VOLUME_ CHANGED	16426	Participant and volume cannot be changed simultaneously.
ERR_BROKER_SUSP_TRD_MOD_REJ	16427	Trade modification rejected due to broker suspension
INVALID_AUCTION_INQUIRY	16430	Invalid auction inquiry request.
INVALID_ACCOUNT	16431	Invalid Account in the Account field
ORDER_VALUE_LIMIT_EXCEEDED	16436	The order value limit has exceeded
DQ_NOT_ALLOWED_IN_PREOPEN	16439	DQ Orders are not allowed in preopen.
SERIES_NOT_ALLOWED_IN_PREOPEN	16440	Order Entry is not allowed in preopen for the series.

Error Code ID	Error Code Value	Description of Error Code
ST_NOT_ALLOWED_IN_PREOPEN	16441	ST Orders are not allowed in preopen.
ORDER_VALUE_EXCEEDS_ORDER_VALUE_LIMIT	16442	The current placed order's value is more than users order value limit
ERROR_SL_LMT_RSNBLTY_CHECK	16448	Difference between limit price and trigger price is beyond permissible range
ACCOUNT_MANDATORY	16450	Account number is mandatory in Account field
OE_BL_MKT_ORDERS_IN_CLOSING	16473	Only board lot market orders are allowed in Closing Session.
ORDER_CANCELED_DUE_TO_SECURITY_ SUSPENSION	16482	The order has been cancelled as security has been suspended
ORDER_CANCELED_DUE_TO_PARTICIPANT_ SUSPENSION	16483	The order has been cancelled as participant has been suspended
ERR_FUNCTION_NOT_FOR_INQ_USER	16493	Functionality not available for Inquiry user
ERR_PRICE_OUTSIDE_REVISED_PRICE_RANGE	16521	Order price is outside the revised price range.
BUY_ORDER_VALUE_LIMIT_EXCEEDED	16530	Users buy order value limit has exceeded.
SELL_ORDER_VALUE_LIMIT_ EXCEEDED	16531	The order value limit for the sell quantity has exceeded its limit
ERR_BR_BUY_ORD_VAL_LIMIT_EXCEEDED	16532	Branch buy order limit has been exceeded
ERR_BR_SELL_ORD_VAL_LIMIT_EXCEEDED	16533	Branch sell order limit has been exceeded
NO_BUY_BACK_RUNNING	16534	No buyback running for that security.

Error Code ID	Error Code Value	Description of Error Code
PARTIAL_ORDER_REJECTED	16535	Order partially rejected. Remaining order quantity specified rejected due to system error.
QUICK_CXL_REJECTED	16536	Quick Cancel request rejected due to system error. Retry Quick Cancel Request
ERR_CANNOT_LOGOFF_SELF	16560	Not allowed to reset user's own login session
ERR_USER_ALREADY_SIGNED_OFF	16562	Requested user is already signed off
ERR_NO_PRIVILEGE_FOR_USER	16563	No privilege to execute functionality
ERR_FRZ_REJECT_FOR_CLOSEOUT	16567	This error code will be returned when a Close out order goes into freeze.
ERR_CLOSEOUT_NOT_ALLOWED	16568	This error code is returned when a Close out order entry is not allowed.
ERR_CLOSEOUT_ORDER_REJECT	16569	This error code is returned when a Close out order is rejected by the system.
ERR_CLOSEOUT_TRDMOD_REJECT	16571	This error code will be returned when a user under a broker in 'Close out' state tries to modify Trade.
INVALID_MSG_LENGTH	16573	Message length is invalid.
ERR_MAX_UOVL_VALUE_EXCEEDED	16576	Maximum UOVL exceeded
ERR_MAX_BOVL_VALUE_EXCEEDED	16577	Maximum BOVL exceeded
ERR_USER_IP_REC_NOT_FOUND	16588	User does not exist
ERR_SYS_REJECT	16592	Order Entry is not allowed
rms_order_reject	16597	Order entry / Modification rejected by the Exchange
ERR_SEC_REJECT	16598	Order Entry is not allowed
ERR_ORD_VAL_EXCEEDED	16600	The order value has exceeded maximum permissible limit
ERR_PREOPEN_ORDER_REJECT	16601	Request Rejected by the exchange

Error Code ID	Error Code Value	Description of Error Code
MARKET_ORDER_NOT_ALLOWED_IN_BT_SESSION	16603	Market order not allowed in Block Trade session
DQ_ORDER_NOT_ALLOWED_IN_CLOSING	16604	Disclosed Quantity (DQ) order not allowed in closing session
ERR_INVALID_CLIENT	16606	Client order not allowed for market maker user
ERR_INST_PARTICIPANT_ORD_NOT_ALLOWED	16700	INST Participant orders not allowed for T+0 settlement
ERROR_ALGOID_NNFID_MISMATCH_1	16730	NNF id & Algo id mismatch - Algo ID entered is 0 in order request.
ERROR_ALGOID_NNFID_MISMATCH_2	16731	NNF id & Algo id mismatch - For Non-Algo orders Algo id should be 0 (zero) in order request.
ERROR_ALGO_MKT_NOT_ALLOWED	16732	Market order not allowed for Algo order.
ERROR_INVALID_NNF_ID	16733	13th digit of NNF id is invalid.
ERROR_ORDREJ_AS_ALGOID_DISABLED	16735	Order rejected as Algo ID is disabled by the Exchange
ERROR_ORDCXL_AS_ALGOID_DISABLED	16736	Order cancelled as Algo ID is disabled by the Exchange
ERROR_BL_ORD_TIMED_OUT	16738	Block Deal order timed out.
ERR_ORD_LIM_EXCEEDS_SET_ORD_VAL_LIM	16750	Order Limit exceeds the set User Order Value Limit
ERROR_USER_ALREADY_UNLOCKED	16752	User already unlocked
ERROR_DUPLICATE_UNLOCK_ALERT	16753	Duplicate user unlock request
ERR_ACCNT_DISABLE_TRADING	16761	The account is disabled from trading as per directions of SEBI/Statutory Authority.
ERR_NEW_PWD_INVALID	16778	Password set is not in lines of the password policy
ERR_ACCNT_DISABLE_TRADING_FOR_PIT	16910	Account is disabled for trading in the scrip during the Trading Window closure period (SEBI PIT Reg).

Error Code ID	Error Code Value	Description of Error Code
		Please contact the company for more details.
ERR_STATUS_CHANGE_NOT_ALLOWED	17015	Status change not allowed. User should be Dealer/Branch Manager/Inquiry
ERROR_INVALID_PACKET	17101	The packet has invalid transaction code OR Packet has invalid data
ERR_HEARTBEAT_NOT_RECEIVED	17102	Heart Beat not received
ERR_INVALID_BOX_ID	17104	Invalid Box Id
ERR_SEQ_NUM_MISMATCH	17105	Sequence number mismatch found
ERROR_BOX_RATE_EXCEEDED	17106	Box Rate has been exceeded by the Member
ERROR_HB_RATE_EXCEEDED	17107	Heart beat rate exceeded by the member
ERR_VOLUNTARY_CLOSEOUT_ORDR_REJECT	17017	Order Cancelled due to Voluntary Closeout.
ERR_ACTV_NUM_OF_USRS_IN_BRNCH_EXCEEDED	17022	Number for active users in branch exceeded
ERR_ORD_COULD_RESULT_IN_SELF_TRADE	17080	The order could have resulted in self-trade.
ERR_MAX_USR_LOGIN_EXCEEDED	17142	Maximum user login allowed per box has been exceeded
ERR_INVALID_PAN_ID	17177	Invalid PAN Id
ERR_INVALID_ALGO_ID	17179	Invalid Algo Id
ERR_INVALID_RESERVED_FILLER	17180	Invalid value in the Reserved Filler field
ERR_MKT_ORD_NOT_ALLOWED	17182	Security not traded. Market order not allowed.

Error Code ID	Error Code Value	Description of Error Code
ERR_TRADE_BEYOND_MARKUP_PRICE	17183	Order could have resulted in trade beyond mark-up price.
ERR_USER_HAVING_NULL_RIGHTS	17184	Order rejected as user has NO trading rights
ERR_CHECKSUM_FAILED_GR	19028	Checksum verification failed at Gateway Router.
ERR_MULTIPLE_GR_QUERY_RCV	19029	Multiple GR_QUERY request received.
ERR_CANNOT_MOD_AUC_ORDER	16397	Modifying Auction Order not allowed
ERR_ENCRYPTION_FLAG_MISMATCH	19030	Encryption Flag Mismatch
ERR_MD5_CHECKSUM_FAILURE	19031	MD5 Checksum Failed

Reason Codes

The reason codes and the corresponding values are given below.

Reason Code	Value
Security	5
Broker	6
Branch	7
User	8
Participant	9
Counter Party	10
Order Number	11
Auction Number	15
Order Type	16
Price Freeze	17
Quantity Freeze	18
Call Auction 1	23
Call Auction 2	24

List of Transaction Codes

Transaction Code	Code	Structure	Size	I/B*
SYSTEM_INFORMATION_IN	1600	MESSAGE_HEADER	40	I
SYSTEM_INFORMATION_OUT	1601	SYSTEM_INFORMATION_DATA	90	I
BOARD_LOT_IN	2000	ORDER_ENTRY_REQUEST	290	I
BOARD_LOT_OUT	2001	ORDER_ENTRY_REQUEST	214	I
PRICE_CONFIRMATION	2012	ORDER_ENTRY_REQUEST	290	I
ORDER_MOD_IN	2040	ORDER_ENTRY_REQUEST	290	I
ORDER_MOD_REJECT	2042	ORDER_ENTRY_REQUEST	290	I
QUICK_CANCEL_OUT	2061	ORDER_ENTRY_REQUEST	290	I
KILL_SWITCH_IN	2062	ORDER_ENTRY_REQUEST	290	I
ORDER_CANCEL_IN	2070	ORDER_ENTRY_REQUEST	290	I
ORDER_CANCEL_OUT	2071	ORDER_ENTRY_REQUEST	214	I
ORDER_CANCEL_REJECT	2072	ORDER_ENTRY_REQUEST	290	I
ORDER_CONFIRMATION	2073	ORDER_ENTRY_REQUEST	290	I
ORDER_MOD_CONFIRMATION	2074	ORDER_ENTRY_REQUEST	290	I
ORDER_CANCEL_CONFIRMATION	2075	ORDER_ENTRY_REQUEST	290	I
FREEZE_TO_CONTROL	2170	ORDER_ENTRY_REQUEST	290	I
ON_STOP_NOTIFICATION	2212	TRADE_CONFIRM	228	I
TRADE_CONFIRMATION	2222	TRADE_CONFIRM	228	I
TRADE_ERROR	2223	TRADE_INQUIRY_DATA	210	I
ORDER_ERROR	2231	ORDER_ENTRY_REQUEST	290	I
TRADE_CANCEL_CONFIRM	2282	TRADE_CONFIRM	228	I
TRADE_CANCEL_REJECT	2286	TRADE_CONFIRM	228	I
TRADE_MODIFY_CONFIRM	2287	TRADE_CONFIRM	228	I
SIGN_ON_REQUEST_IN	2300	SIGNON_IN	276	I
SIGN_ON_REQUEST_OUT	2301	SIGNON_OUT	276	I
ERROR_RESPONSE_OUT	2302	ERROR_RESPONSE	180	I
SIGN_OFF_REQUEST_OUT	2321	MESSAGE HEADER	40	I
GR_REQUEST	2400	MS_GR_REQUEST	48	I
GR_RESPONSE	2401	MS_GR_RESPONSE	124 136	I
BCAST_CONT_MSG	5294	MS_BCAST_CONT_MESSAGE	244	B
CTRL_MSG_TO_TRADER	5295	MS_TRADER_INT_MSG	290	I
USER_ADDR_UNLOCK_IN	5424	USER_ADDR_UNLOCK_REQ	68	I
USER_ADDR_UNLOCK_OUT	5425	USER_ADDR_UNLOCK_RESP	44	I
TRADE_CANCEL_IN	5440	TRADE_INQUIRY_DATA	210	I

Transaction Code	Code	Structure	Size	I/B*
TRADE_CANCEL_OUT	5441	TRADE_INQUIRY_DATA	210	I
TRADE_MOD_IN	5445	TRADE_INQUIRY_DATA	210	I
USER_ADDR_UNLOCK_APPROVE_OUT	5575	USER_ADDR_UNLOCK_APP_REJ_RESP	44	I
USER_ADDR_UNLOCK_REJECT_OUT	5579	USER_ADDR_UNLOCK_APP_REJ_RESP	44	I
BRANCH_ORDER_LIMIT_UPDATE_IN	5716	BRANCH_ORDER_VAL_LIMIT_UPDATE	104	I
BRANCH_ORDER_LIMIT_UPDATE_OUT	5717	BRANCH_ORDER_VAL_LIMIT_UPDATE	104	I
USER_ORDER_LIMIT_UPDATE_IN	5719	USER_ORDER_VAL_LIMIT_UPDATE	142	I
USER_ORDER_LIMIT_UPDATE_OUT	5720	USER_ORDER_VAL_LIMIT_UPDATE	142	I
DEALER_LIMIT_UPDATE_IN	5721	ORDER_LIMIT_UPDATE	68	I
DEALER_LIMIT_UPDATE_OUT	5722	ORDER_LIMIT_UPDATE	68	I
SIGN_OFF_TRADER_IN	5723	SIGNON IN	276	I
SIGN_OFF_TRADER_OUT	5724	SIGNON IN	276	I
RESET_PASSWORD_IN	5738	RESET_PASSWORD	58	I
RESET_PASSWORD_OUT	5739	RESET_PASSWORD	58	I
COL_USER_STATUS_CHANGE _IN	5790	COL_ USER_STATUS_CHANGE_REQ	52	I
COL_USER_STATUS_CHANGE _OUT	5791	COL_USER_STATUS_CHANGE_RESP	46	I
TRD_MOD_CXL_STATUS_CHANGE _IN	5792	USER_ TRD_MOD_CXL_CHANGE_REQ	52	I
TRD_MOD_CXL_STATUS_CHANGE _OUT	5793	USER_TRD_MOD_CXL_CHANGE_RESP	46	I
BCAST_JRNL_VCT_MSG	6501	BCAST_VCT_MESSAGES	298	B
BC_OPEN_MESSAGE	6511	BCAST_VCT_MESSAGES	298	B
BC_CLOSE_MESSAGE	6521	BCAST_VCT_MESSAGES	298	B
BC_PREOPEN_SHUTDOWN_MSG	6531	BCAST_VCT_MESSAGES	298	B
BC_CIRCUIT_CHECK	6541	BCAST_HEADER	40	B
BC_NORMAL_MKT_PREOPEN_ENDED	6571	BCAST_VCT_MESSAGES	298	B
BC_AUCTION_STATUS_CHANGE	6581	AUCTION_STATUS_CHANGE	302	B
DOWNLOAD_REQUEST	7000	MESSAGE_DOWNLOAD	48	I
HEADER_RECORD	7011	MESSAGE HEADER	40	I
MESSAGE_RECORD	7021	MESSAGE HEADER	40	I
TRAILER_RECORD	7031	MESSAGE HEADER	40	I
BROADCAST_MBO_MBP	7200	BROADCAST MBO MBP	482	B
BCAST_MW_ROUND_ROBIN	7201	BROADCAST INQUIRY RESPONSE	466	B
BCAST_SYSTEM_INFORMATION_OUT	7206	SYSTEM_INFORMATION_DATA	90	B
BCAST_ONLY_MBP	7208	BROADCAST ONLY MBP	566	B
BCAST_CALL AUCTION_ORD_CXL_UPDATE	7210	BROADCAST CALL AUCTION ORD CXL UPDATE	490	B
BCAST_CALL AUCTION_MBP	7214	BROADCAST CALL AUCTION MBP	538	B

Transaction Code	Code	Structure	Size	I/B*
BCAST_CA_MW	7215	BROADCAST CALL AUCTION MARKET WATCH	482	B
BCAST_INDICES	7207	BROADCAST INDICES	474	B
BCAST_INDICES_VIX	7216	BROADCAST INDICES VIX	474	B
UPDATE_LOCALDB_IN	7300	UPDATE_LOCAL_DATABASE	58	I
UPDATE_LOCALDB_DATA	7304	Packet of size >80 and <=512	512	I
BCAST_PART_MSTR_CHG	7306	PARTICIPANT_UPDATE_INFO	84	B
UPDATE_LOCALDB_HEADER	7307	UPDATE_LDB_HEADER	42	I
UPDATE_LOCALDB_TRAILER	7308	UPDATE_LDB_TRAILER	42	I
PARTIAL_SYSTEM_INFORMATION	7321	SYSTEM_INFORMATION_DATA	90	I
BC_SYMBOL_STATUS_CHANGE _ACTION	7764	BCAST_SYMBOL_STATUS_CHANGE _ACTION	58	B
BCAST_INDICATIVE_INDICES	8207	BROADCAST INDICATIVE INDICES	474	B
BATCH_ORDER_CANCEL	9002	ORDER_ENTRY_REQUEST	290	I
BCAST_TURNOVER_EXCEEDED	9010	BROADCAST_LIMIT_EXCEEDED	77	B
BROADCAST_BROKER_REACTIVATED	9011	BROADCAST_LIMIT_EXCEEDED	77	B
AUCTION_INQUIRY_IN	18016	MS_AUCTION_INQ_REQ	55	I
AUCTION_INQUIRY_OUT	18017	AUCTION INQUIRY RESPONSE	222	I
MARKET_STATS_REPORT_DATA	18201	MS_RP_HDR REPORT MARKET STATISTICS REPORT TRAILER	106 478 46	B
BCAST_AUCTION_INQUIRY_OUT	18700	MS_AUCTION_INQ_DATA	76	B
BCAST_TICKER_AND_MKT_INDEX	18703	TICKER TRADE DATA	546	B
BCAST_SECURITY_STATUS_CHG_PREOPEN	18707	SECURITY STATUS UPDATE INFORMATION	442	I/B
BCAST_BUY_BACK	18708	BROADCAST BUY_BACK	426	B
BCAST_SECURITY_MSTR_CHG	18720	SECURITY UPDATE INFORMATION	260	I/B
BCAST_SECURITY_STATUS_CHG	18130	SECURITY STATUS UPDATE INFORMATION	442	I/B
BOARD_LOT_IN_TR	20000	ORDER_ENTRY_ REQUEST _TR	136	I
BOARD_LOT_OUT_TR	20001	MS_OM_REQUEST_TR	132	I
ORDER_MOD_IN_TR	20040	ORDER_OM_ REQUEST _TR	180	I
ORDER_MOD_OUT_TR	20041	MS_OM_REQUEST_TR	132	I
ORDER_MOD_REJECT_TR	20042	ORDER_OM_ RESPONSE_TR	216	I
ORDER_CANCEL_IN_TR	20070	ORDER_OM_ REQUEST _TR	180	I
ORDER_CANCEL_REJECT_TR	20072	ORDER_OM_ RESPONSE_TR	216	I
ORDER_CONFIRMATION_TR	20073	ORDER_OM_ RESPONSE_TR	216	I

Transaction Code	Code	Structure	Size	I/B*
ORDER_MOD_CONFIRMATION_TR	20074	ORDER_OM_ RESPONSE_TR	216	I
ORDER_CXL_CONFIRMATION_TR	20075	ORDER_OM_ RESPONSE_TR	216	I
ORDER_ERROR_TR	20231	ORDER_OM_ RESPONSE_TR	216	I
PRICE_CONFIRMATION_TR	20012	ORDER_OM_ RESPONSE_TR	216	I
TRADE_CONFIRMATION_TR	20222	MS_TRADE_CONFIRM_TR	192	I
TRIMMED_BOARD_LOT_ACK_IN	20400	ORDER_ENTRY_ REQUEST _TR	136	I
TXN_EXT_QUICK_ACK_OE_RESP	20401	MS_ACK_RESPONSE	22	I
TRIMMED_ORDER_MOD_ACK_IN	20402	ORDER_OM_ REQUEST _TR	180	I
TXN_EXT_QUICK_ACK_OM_RESP	20403	MS_ACK_RESPONSE	22	I
TRIMMED_ORDER_CANCEL_ACK_IN	20404	ORDER_OM_ REQUEST _TR	180	I
TXN_EXT_QUICK_ACK_OC_RESP	20405	MS_ACK_RESPONSE	22	I
BOX_SIGN_ON_REQUEST_IN	23000	MS_BOX_SIGN_ON_REQUEST_IN	60	I
BOX_SIGN_ON_REQUEST_OUT	23001	MS_BOX_SIGN_ON_REQUEST_OUT	52	I
SECURE_BOX_REGISTRATION_REQUEST_IN	23008	MS_SECURE_BOX_REGISTRATION_REQUEST_IN	42	I
SECURE_BOX_REGISTRATION_RESPONSE_OUT	23009	MS_SECURE_BOX_REGISTRATION_RESPONSE_OUT	40	I
BOX_SIGN_OFF	20322	MS_BOX_SIGN_OFF	42	I

* Interactive/Broadcast

List of Transaction Codes Containing Timestamp in Nanoseconds

The transaction codes that will contain timestamp in nanoseconds from 01-Jan-1980 00:00:00 are listed in following table:

Transaction Code	Code
PRICE_CONFIRMATION	2012
ORDER_MOD_REJECT	2042
ORDER_CANCEL_REJECT	2072
ORDER_CONFIRMATION	2073
ORDER_MOD_CONFIRMATION	2074
ORDER_CANCEL_CONFIRMATION	2075
FREEZE_TO_CONTROL	2170
ON_STOP_NOTIFICATION	2212
TRADE_CONFIRMATION	2222

Transaction Code	Code
ORDER_ERROR	2231
BATCH_ORDER_CANCEL	9002
PRICE_CONFIRMATION_TR	20012
ORDER_MOD_REJECT_TR	20042
ORDER_CANCEL_REJECT_TR	20072
ORDER_CONFIRMATION_TR	20073
ORDER_MOD_CONFIRMATION_TR	20074
ORDER_CXL_CONFIRMATION_TR	20075
TRADE_CONFIRMATION_TR	20222
ORDER_ERROR_TR	20231

Quick Reference for Order Entry Parameters

The order flags are given below.

Order Terms:

Order Flags	Input/Output	Attributes Represent
MF	Input, to be set when the min fill quantity is given	If set to 1, represents Minimum Fill attribute
AON	Input	If set to 1, represents All or None attribute
IOC	Input	If set to 1, represents Immediate or Cancel attribute
GTC	Input	If set to 1, represents Good Till Cancel attribute
Day	Input	If set to 1, represents Day attribute. This is default attribute
SL	Input	If set to 1, represents Stop Loss attribute
Market	Output	If set to 1, represents Market order
ATO	Output	If set to 1, represents Market order in PREOPEN or CALL AUCTION1 or CALL AUCTION 2 market.
STPC	Input	• If set to 0, represents order resulting in self-trade to be cancelled as per default action by the exchange.

Order Flags	Input/Output	Attributes Represent
		• If set to 1, represents active order resulting in self-trade to be cancelled. • Order modification will be rejected if this bit is modified. In case of triggered stop loss order, bit selected during order entry will be considered
Preopen	Input	• If set to 1, represents the order is a Preopen session order and • If set to 0, represents Normal Market Open order.
Frozen	Output	If set to 1, represents the order has gone for a freeze
Modified	Output	If set to 1, represents the order is modified
Traded	Output	If set to 1, represents the order is traded partially or fully.
MatchedInd	Output	Reserved

Status	Market	Book Type	Order Terms and Other Characteristic Fields
Preopen	Normal Market	RL**	Non-zero value of Good Till Date/DAY/GTC mandatory, mutually exclusive, input ATO output, set if Market order, value of order price returned is '-1'.
Open	Normal Market	RL**	Non-zero value of Good Till Date/DAY/ GTC/ IOC mandatory, mutually exclusive, input MKT output, set if it is Market order.
Open	Normal Market	SL**	SL mandatory, input Non-zero value of Good Till Date/DAY/ GTC/ IOC mandatory, mutually exclusive, input MF/ AON mutually exclusive, input MKT output, set if Market order Trigger Price is mandatory.

Status	Market	Book Type	Order Terms and Other Characteristic Fields
Open	Normal Market	ST**	Non-zero value of Good Till Date /DAY/ GTC/ IOC mandatory, mutually exclusive, input MF/ AON mandatory, mutually exclusive, input MKT output, set if it is Market order.
Open	Odd Lot Market	OL**	Non-zero value of Good Till Date/DAY/ GTC/ IOC mandatory, mutually exclusive, input. Volume is less than Board Lot quantity.
Open	Auction Market	AU**	DAY mandatory, input. Auction Number and Participant Type are mandatory.
Preopen	Call Auciton 1 Market	CA	Non-zero value of IOC /DAY mandatory, mutually exclusive, input. ATO output, set if Market order, value of order price returned is '-1'.
Preopen	Call Auciton 2 Market	CB	Value of IOC set as 0 mandatory, mutually exclusive, input. ATO output set as 0, as Market Order Not allowed. Value of DAY set as 1 mandatory, mutually exclusive, input.
Close			Order entry is not allowed.

** Other input flags in the order terms are not allowed, hence should not be set.

Market Type

The market types are:

Status	Market Status ID
Normal Market	1
Odd Lot Market	2
Spot Market	3
Auction Market	4
Call auction1 Market	5
Call auction2 Market	6

Market Status

The market can be in one of the following statuses:

Status	Market Status ID
PreOpen (only for Normal Market)	0
Open	1
Closed	2
Preopen ended	3

Book Types

There are seven books. These books fall in four markets.

Book Type	Book ID	Market Type
Regular Lot Order	1	Normal Market
Special Terms Order	2	Normal Market
Stop Loss Order	3	Normal Market
Odd Lot Order	5	Odd Lot Market
Spot Order	6	Spot Market
Auction Order	7	Auction Market
Call Auction1	11	Call auction1 market
Call Auction2	12	Call auction2 market

Auction Status

Status	Value Sent in Packet	ID	Description
AUCTION_PENDING_APPROVAL		1	If the auction is initiated by the trader an alert is generated at the CWS. The auction status is in pending for approval.

Status	Value Sent in Packet	ID	Description
AUCTION_PENDING	'P'	2	If any auction in the particular security is already going on, the status of the auction entered next is pending.
OPEN_COMPETITIOR_PERIOD	'C'	3	When the auction gets initiated, this is the status.
OPEN_SOLICITOR_PERIOD	'S'	4	Auction enters solicitor period.
AUCTION_MATCHING	'M'	5	After solicitor period ends, the auction enters matching state. The matching of auction orders takes place.
AUCTION_FINISHED	'F'	6	Status after matching of orders is done and auction trades are generated.
AUCTION_CXLED	'X'	7	Auction is cancelled by NSE-Control.

Security Status

Status	Status ID
Preopen	1
Open	2
Suspended	3
Preopen Extended	4
Price Discovery	6

Activity Types

The activity types that are sent in reports are given below.

Activity Type	Description	Code
ORIGINAL_ORDER	This is the order that was entered. GTC/GTD orders still in the book also come with this activity type.	1
ACTIVITY_TRADE	The trade was done.	2
ACTIVITY_ORDER_CANCEL	The order was cancelled.	3

Activity Type	Description	Code
ACTIVITY_ORDER_MODIFY	The order was modified.	4
ACTIVITY_TRADE_MOD	The trade was modified.	5
ACTIVITY_TRADE_CXL_1	Trade cancellation was requested.	6
ACTIVITY_TRADE_CXL_2	Action has been taken on this request.	7
ACTIVITY_BATCH_ORDER_CANCEL	At the end of the day, all untraded Day orders are cancelled. GTC/GTD orders due for cancellation are also cancelled.	8

Pipe Delimited File Structures

The upload files have a header record at the beginning of the file followed by the detail records. All the fields in both the header and detail records are separated by pipe ('|').The fields are not of fixed width. Any two fields are separated by a '|' symbol.

Security File Structure

Header

Table 51 SECURITY_FILE_HEADER

Structure Name	SECURITY_FILE_HEADER		
Packet Length	19 bytes		
Field Name	Data Type	Size in Byte	Offset
NEATCM	CHAR	6	0
Reserved	CHAR	1	6
VersionNumber	CHAR	7	7
Reserved	CHAR	1	14
DATE	LONG	4	15

Stock Structure

Table 52 STOCK_STRUCTURE

Structure Name	STOCK_STRUCTURE		
Packet Length	270 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0

Structure Name	STOCK_STRUCTURE		
Packet Length	270 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	4
Symbol	CHAR	10	5
Reserved	CHAR	1	15
Series	CHAR	2	16
Reserved	CHAR	1	18
InstrumentType	SHORT	2	19
Reserved	CHAR	1	21
IssuedCapital	DOUBLE	8	22
Reserved	CHAR	1	30
PermittedToTrade	SHORT	2	31
Reserved	CHAR	1	33
CreditRating	CHAR	19	34
Reserved	CHAR	1	53
ST_SEC_ELIGIBILITY_ PER_ MARKET [6] (Refer Table 52.1)	STRUCT	30	54
BoardLotQuantity	LONG	4	84
Reserved	CHAR	1	88
TickSize	LONG	4	89
Reserved	CHAR	1	93
Name	CHAR	25	94
Reserved	CHAR	1	119
SurvInd	SHORT	2	120
Reserved	CHAR	1	122
IssueStartDate	LONG	4	123
Reserved	CHAR	1	127
IssueIPDate	LONG	4	128
Reserved	CHAR	1	132
MaturityDate	LONG	4	133
Reserved	CHAR	1	137
FreezePercent	SHORT	2	138
Reserved	CHAR	1	140
ListingDate	LONG	4	141

Structure Name	STOCK_STRUCTURE		
Packet Length	270 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	145
ExpulsionDate	LONG	4	146
Reserved	CHAR	1	150
ReAdmissionDate	LONG	4	151
Reserved	CHAR	1	155
ExDate	LONG	4	156
Reserved	CHAR	1	160
RecordDate	LONG	4	161
Reserved	CHAR	1	165
NoDeliveryDateStart	LONG	4	166
Reserved	CHAR	1	170
NoDeliveryDateEnd	LONG	4	171
Reserved	CHAR	1	175
ParticipantInMktIndex	CHAR	1	176
Reserved	CHAR	1	177
AON	CHAR	1	178
Reserved	CHAR	1	179
MF	CHAR	1	180
Reserved	CHAR	1	181
SettlementType	SHORT	2	182
Reserved	CHAR	1	184
BookClosureStartDate	LONG	4	185
Reserved	CHAR	1	189
BookClosureEndDate	LONG	4	190
Reserved	CHAR	1	194
Dividend	CHAR	1	195
Reserved	CHAR	1	196
Rights	CHAR	1	197
Reserved	CHAR	1	198
Bonus	CHAR	1	199
Reserved	CHAR	1	200
Interest	CHAR	1	201

Structure Name	STOCK_STRUCTURE		
Packet Length	270 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	202
AGM	CHAR	1	203
Reserved	CHAR	1	204
EGM	CHAR	1	205
Reserved	CHAR	1	206
MMSpread	LONG	4	207
Reserved	CHAR	1	211
MMMinQty	LONG	4	212
Reserved	CHAR	1	216
SSEC	SHORT	2	217
Reserved	CHAR	1	219
Remarks	CHAR	25	220
Reserved	CHAR	1	245
LocalDBUpdateDateTime	LONG	4	246
Reserved	CHAR	1	250
DeleteFlag	CHAR	1	251
Reserved	CHAR	1	252
FaceValue	LONG	4	253
Reserved	CHAR	1	257
ISIN Number	CHAR	12	258

Table 52.1 ST_SEC_ELIGIBILITY_PER_MARKET

Structure Name	ST_SEC_ELIGIBILITY_PER_MARKET		
Packet Length	6 bytes		
Field Name	Data Type	Size in Byte	Offset
Security Status	SHORT	2	0
Reserved	CHAR	1	2
Eligibility	CHAR	1	3
Reserved	CHAR	2	4

Field Name	Brief Description
Token	Token number of the security being updated. This is unique for a particular symbol-series combination.
Symbol	This field should contain the symbol of a security.
Series	This field should contain the series of a security
InstrumentType	This field contains the instrument type of the security. It can be one of the following: ▪ '0' – Equities ▪ '1' – Preference Shares ▪ '2' – Debentures ▪ '3' – Warrants ▪ '4' – Miscellaneous
IssuedCapital	Issue size of the security.
PermittedToTrade	• '0' - Listed but not permitted to trade • '1' - Permitted to trade • '2' – BSE listed (BSE exclusive security will be available, however trading on the same will be allowed only in case of outage at BSE)
CreditRating	This field contains daily price range of the security.
SecurityStatus	• '1' - Preopen (Only for Normal market) • '2' - Open • '3' - Suspended • '4' - Preopen extended • '5' - Stock Open With Market • '6' – Price Discovery This will contain the Call Auction2 Market security status at 6th position The values can be : 1' : Preopen 3' : Suspended 6': Price Discovery.
Eligibility	• 0' – for Stocks not eligible in current market • '1' – for stocks eligible in current Market 6th Position represents eligibility for Call Auction 2 Market.

Field Name	Brief Description
BoardLotQuantity	Regular lot size.
TickSize	Tick size/ Min spread size.
Name	Security name.
SurvInd	Indicator for security in Surveillance Measure
IssueStartDate	Date of issue of the security.
IssueIPDate	Interest Payment Date
IssueMaturityDate	Maturity Date.
FreezePercent	Freeze percent. This field indicates the volume freeze percentage w.r.t. issued capital. This field has to be interpreted as freeze percent /10000. E.g.: 41 in this field has to be interpreted as 0.0041 %
ListingDate	Date of listing.
ExpulsionDate	Date of expulsion.
ReAdmissionDate	Date of readmission.
ExDate	Last date of trading before any corporate action.
RecordDate	Date of record changed.
NoDeliveryStartDate	Date from when physical delivery of share certificates is stopped for book closure.
NoDeliveryEndDate	No delivery end date.
ParticipateInMktIndex	'1' – Security is present in NIFTY Index. '0' – Security is not present in NIFTY Index.
AON	'1'- AON is allowed. '0'- AON is not allowed
MF	'1'- MF is allowed. '0'- MF is not allowed
SettlementType	This field contains the settlement type. It can be one of the following: '0' – T+0 settlement '1' – T+1 settlement
BookClosureStartDate	Date at which the record books in the company for shareholder names starts.
BookClosureEndDate	Date at which the record books in the company for shareholder names ends.

Field Name	Brief Description
Dividend	'1' – Dividend '0' – No Dividend
Rights	'1' – Rights '0' - No Rights
Bonus	'1' – Rights '0' - No Rights
Interest	'1' – Interest '0' - No Interest
AGM	'1' – AGM '0' - No AGM
EGM	'1' – EGM '0' – No EGM
MMSpread	This is the spread value per security.
MMMinQty	This field contains the Minimum quantity for the security, Used by Market maker user for market maker order.
SSEC	'1' – Securities (except SME) eligible in Normal market and Odd Lot markets. '2' – IPO Session is held security (including SME) '3' – Re-list Session is held security (including SME). '4' – Illiquid security eligible for Call Auction session (CA2) (including SME). '5' – SME securities eligible in normal market. This is used as an identifier for different market securities.
Remark	Remarks
LocalLDBUpdateDateTime	This is the local database update date-time.
DeleteFlag	This indicates the status of the security, whether the security is deleted or not. • 'N' : Active • 'Y' : Deleted
FaceValue	This field contains face value of the security
ISIN Number	This field contains the ISIN Number of the security.

Contract File Structure

Header

Table 53 CONTRACT_FILE_HEADER

Structure Name	CONTRACT_FILE_HEADER		
Packet Length	13 bytes		
Field Name	Data Type	Size in Byte	Offset
NEATFO	CHAR	6	0
Reserved	CHAR	1	6
VersionNumber	CHAR	5	7
Reserved	CHAR	1	12

Stock Structure

Table 54 STOCK_STRUCTURE

Structure Name	STOCK_STRUCTURE		
Packet Length	322 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
Reserved	CHAR	1	4
AssetToken	LONG	4	5
Reserved	CHAR	1	9
InstrumentName	CHAR	6	10
Reserved	CHAR	1	16
Symbol	CHAR	10	17
Reserved	CHAR	1	27
Series	CHAR	2	28
Reserved	CHAR	2	30
ExpiryDate (in seconds from January 1,1980)	LONG	4	32
Reserved	CHAR	1	36
StrikePrice	LONG	4	37
Reserved	CHAR	1	41
OptionType	CHAR	2	42

Structure Name	STOCK_STRUCTURE		
Packet Length	322 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	44
Category	CHAR	1	45
Reserved	CHAR	1	46
CALevel	SHORT	2	47
Reserved	CHAR	2	49
PermittedToTrade	SHORT	2	51
Reserved	CHAR	1	53
IssueRate	SHORT	2	54
Reserved	CHAR	1	56
ST_SEC_ELIGIBILITY_ PER_ MARKET [4] (Refer Table 54.1)	STRUCT	24	57
IssueStartDate	LONG	4	81
Reserved	CHAR	1	85
InterestPaymentDate	LONG	4	86
Reserved	CHAR	1	90
Issue Maturity Date	LONG	4	91
Reserved	CHAR	1	95
MarginPercentage	LONG	4	96
Reserved	CHAR	1	100
MinimumLotQuantity	LONG	4	101
Reserved	CHAR	1	105
BoardLotQuantity	LONG	4	106
Reserved	CHAR	1	110
TickSize	LONG	4	111
Reserved	CHAR	1	115
IssuedCapital	DOUBLE	8	116
Reserved	CHAR	1	124
FreezeQuantity	LONG	4	125
Reserved	CHAR	1	129
WarningQuantity	LONG	4	130
Reserved	CHAR	1	134
ListingDate	LONG	4	135

Structure Name	STOCK_STRUCTURE		
Packet Length	322 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	139
ExpulsionDate	LONG	4	140
Reserved	CHAR	1	144
ReadmissionDate	LONG	4	145
Reserved	CHAR	1	149
RecordDate	LONG	4	150
Reserved	CHAR	1	154
NoDeliveryStartDate	LONG	4	155
Reserved	CHAR	1	159
NoDeliveryEndDate	LONG	4	160
Reserved	CHAR	1	164
LowPriceRange	LONG	4	165
Reserved	CHAR	1	169
HighPriceRange	LONG	4	170
Reserved	CHAR	1	174
ExDate	LONG	4	175
Reserved	CHAR	1	179
BookClosureStartDate	LONG	4	180
Reserved	CHAR	1	184
BookClosureEndDate	LONG	4	185
Reserved	CHAR	1	189
LocalLDBUpdateDateTime	LONG	4	190
Reserved	CHAR	1	194
ExerciseStartDate	LONG	4	195
Reserved	CHAR	1	199
ExerciseEndDate	LONG	4	200
Reserved	CHAR	1	204
TickerSelection	SHORT	2	205
Reserved	CHAR	1	207
OldTokenNumber	LONG	4	208
Reserved	CHAR	1	212
CreditRating	CHAR	12	213

Structure Name	STOCK_STRUCTURE		
Packet Length	322 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	225
Name	CHAR	25	226
Reserved	CHAR	1	251
EGMAGM	CHAR	1	252
Reserved	CHAR	1	253
InterestDivident	CHAR	1	254
Reserved	CHAR	1	255
RightsBonus	CHAR	1	256
Reserved	CHAR	1	257
MFAON	CHAR	1	258
Reserved	CHAR	1	259
Remarks	CHAR	24	260
Reserved	CHAR	1	284
ExStyle	CHAR	1	285
Reserved	CHAR	1	286
ExAllowed	CHAR	1	287
Reserved	CHAR	1	288
ExRejectionAllowed	CHAR	1	289
Reserved	CHAR	1	290
PlAllowed	CHAR	1	291
Reserved	CHAR	1	292
CheckSum	CHAR	1	293
Reserved	CHAR	1	294
IsCorporateAdjusted	CHAR	1	295
Reserved	CHAR	1	296
SymbolForAsset	CHAR	10	297
Reserved	CHAR	1	307
InstrumentOfAsset	CHAR	6	308
Reserved	CHAR	1	314
BasePrice	LONG	4	315
Reserved	CHAR	1	319
DeleteFlag	CHAR	1	320

Table 54.1 ST_SEC_ELIGIBILITY_PER_MARKET

Structure Name	ST_SEC_ELIGIBILITY_PER_MAKRET		
Packet Length	**6 bytes**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
Security Status	SHORT	2	0
Reserved	CHAR	1	2
Eligibility	CHAR	1	3
Reserved	CHAR	2	4

Field Name	Brief Description
Token	Token number of the security being updated. This is unique for a particular symbol-series combination.
AssetToken	Token number of the asset.
SecurityInformation	This contains the Instrument Name, Symbol & Series (EQ / IL / TT), Expiry date, Strike Price, Option Type, Corporate Action level of the security
PermittedToTrade	This field can have any one of the following value: • '0' - Listed but not permitted to trade • '1' - Permitted to trade
Reserved Identifier	This field can have any one of the following value: • '0' – Unreserved Contract • '1' – Reserved Contract
IssueRate	Price of the issue.
Eligibility	The flag is set to 1 if the security is allowed to trade in a particular market.
SecurityStatus	This field can have any one of the following value: • '1' - Preopen (Only for Normal market) • '2' - Open • '3' - Suspended • '4' - Preopen extended

Field Name	Brief Description
	• '5' - Stock Open with Market
IssueStartDate	Date of issue of the security.
InterestPaymentDate	Interest payment date
IssueMaturityDate	Maturity date.
MarginPercent	It is an initial margin percent to be collected on a contract.
MinimumLotQuantity	It is minimum lot of the order which can be placed.
BoardLotQuantity	Regular lot size.
TickSize	Tick size/ Min spread size.
IssuedCapital	Issue size of the security.
FreezeQuantity	Freeze quantity.
WarningQuantity	Warning quantity.
ListingDate	Date of listing.
ExpulsionDate	Date of expulsion.
ReAdmissionDate	Date of readmission.
RecordDate	Date of record changed.
NoDeliveryStartDate	Date from when physical delivery of share certificates is stopped for book closure.
NoDeliveryEndDate	No delivery end date.
LowPriceRange	Minimum price at which order can be placed without causing a price freeze.
HighPriceRange	Maximum price at which order can be placed without causing a price freeze.
ExDate	Last date of trading before any corporate action.
BookClosureStartDate	Date at which the record books in the company for shareholder names starts.
BookClosureEndDate	Date at which the record books in the company for shareholder names ends.
LocalLDBUpdateDateTime	This is the local database update date-time.
ExerciseStartDate	This is the starting date for exercise.
ExerciseEndDate	This is the last date for exercise.
OldTokenNumber	Not used.

Field Name	Brief Description
CreditRating	This field contains daily price range of the security.
Name	Security name.
EGM/AGM	This field can have any one of the following value: • '0' - No EGM/AGM • '1' - EGM • '2' - AGM • '3' - Both EGM and AGM
InterestDividend	This field can have any one of the following value: • '0' - No Interest/ Dividend • '1' - Interest • '2' - Dividend
RightsBonus	This field can have any one of the following value: • '0' - No Rights/Bonus • '1' - Rights • '2' - Bonus • '3' - Both Rights and Bonus
MFAON	This field can have any one of the following value: • '0' - MF/AON not allowed • '1' - MF allowed • '2' - AON allowed • '3' - MF and AON allowed
Remark	Remarks
ExStyle	This field can have any one of the following value: • 'A' - American style Exercise allowed • 'E' - European style Exercise allowed
ExAllowed	Exercise is allowed on this contract if this flag is set to true.
ExRejectionAllowed	Exercise rejection is allowed on this contract if this bit is set to true.
PlAllowed	Position liquidation is allowed on this contract if this flag is set to true.
CheckSum	Not used.

Field Name	Brief Description
IsCorporateAdusted	This field shows whether this contract is corporate adjusted.
AssetName	Name of the underlying asset. **Note:** For example, NIFTY.
InstrumentIDOfAsset	ID of the instrument for the underlying asset of this contract.
AssetInstrument	Underlying asset type. **Note:** For example, INDEX.
BasePrice	Base price of the security.
DeleteFlag	This flag indicates the status of the security, whether the security is deleted or not. This field can have any one of the following value: • 'N' : Active • 'Y' : Deleted

Participant Structure

Header

Table 55 PARTICIPANT_FILE_HEADER

Structure Name	PARTICIPANT_FILE_HEADER		
Packet Length	20 bytes		
Field Name	Data Type	Size in Byte	Offset
NEATCM	CHAR	6	0
Reserved	CHAR	1	6
VersionNumber	CHAR	7	7
Reserved	CHAR	1	14
DATE	LONG	4	15
Reserved	CHAR	1	19

Structure

Table 56 PARTICIPANT_STRUCTURE

Structure Name	PARTICIPANT_STRUCTURE		
Packet Length	47 bytes		
Field Name	Data Type	Size in Byte	Offset
ParticipantId	CHAR	12	0
Reserved	CHAR	1	12
ParticipantName	CHAR	25	13
Reserved	CHAR	1	38
ParticipantStatus	CHAR	1	39
Reserved	CHAR	1	40
DeleteFlag	CHAR	1	41
Reserved	CHAR	1	42
LastUpdateTime	LONG	4	43

Field Name	Brief Description
ParticipantId	ID of the participant.
ParticipantName	Name of the participant.
ParticipantStatus	If this field is set to ‘S’ then the participant is suspended. If this is field is set to ‘A’ then the participant is active.
DeleteFlag	If this field is set to ‘Y’ then the participant is deleted from the system, else he/she is present in the system.
LastUpdateTime	The last time this record was modified.

Trimmed Structures

Trimmed Order Entry Request structure

Table 57 ORDER_ENTRY_REQUEST

Structure Name	ORDER_ENTRY_ REQUEST _TR		
Transaction Code	BOARD_LOT_IN_TR (20000) TRIMMED_BOARD_LOT_ACK_IN (20400)		
Packet Length	136 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
Transcode	SHORT	2	0
TraderId	LONG	4	2
SEC_INFO (Refer Table 4)	STRUCT	12	6
AccountNumber [10]	CHAR	10	18
BookType	SHORT	2	28
BuySell	SHORT	2	30
DisclosedVol	LONG	4	32
Volume	LONG	4	36
Price	LONG	4	40
GoodTillDate	LONG	4	44
ST_ORDER_FLAGS (Refer Table 57.1 for small endian machines and Table 57.2 for big endian machines)	STRUCT	2	48
BranchId	SHORT	2	50
UserId	LONG	4	52
BrokerId [5]	CHAR	5	56
Suspended	CHAR	1	61
Settlor [12]	CHAR	12	62
ProClient	SHORT	2	74
NNFField	DOUBLE	8	76
TransactionId	LONG	4	84
PAN	CHAR	10	88
Algo ID	LONG	4	98
Reserved Filler	SHORT	2	102
Reserved	CHAR	32	104

For Small Endian Machines:

Table 57.1 ST_ORDER_FLAGS

Structure Name	ST_ORDER_FLAGS		
Packet Length	2 bytes		
Field Name	Data Type	Size in Bit	Offset
MF	BIT	1	0
AON	BIT	1	0
IOC	BIT	1	0
GTC	BIT	1	0
Day	BIT	1	0
OnStop	BIT	1	0
Mkt	BIT	1	0
ATO	BIT	1	0
Reserved	BIT	1	1
STPC	BIT	1	1
Reserved	BIT	1	1
Preopen	BIT	1	1
Frozen	BIT	1	1
Modified	BIT	1	1
Traded	BIT	1	1
MatchedInd	BIT	1	1

For Big Endian Machines:

Table 57.2 ST_ORDER_FLAGS

Structure Name	ST_ORDER_FLAGS		
Packet Length	2 bytes		
Field Name	Data Type	Size in Bit	Offset
ATO	BIT	1	0
Mkt	BIT	1	0
OnStop	BIT	1	0
Day	BIT	1	0
GTC	BIT	1	0
IOC	BIT	1	0
AON	BIT	1	0

Structure Name	ST_ORDER_FLAGS		
Packet Length	2 bytes		
Field Name	Data Type	Size in Bit	Offset
MF	BIT	1	0
MatchedInd	BIT	1	1
Traded	BIT	1	1
Modified	BIT	1	1
Frozen	BIT	1	1
Preopen	BIT	1	1
Reserved	BIT	1	1
STPC	BIT	1	1
Reserved	BIT	1	1

Field Name	Brief Description
TransactionCode	The transaction code is BOARD_LOT_IN_TR (20000)/ TRIMMED_BOARD_LOT_ACK_IN (20400).
TraderId	This field should contain the user ID of the user.
SEC_INFO	This structure should contain the Symbol and Series of the security.
AccountNumber	If the order is entered on behalf of a trader, the trader account number should be specified in this field. For broker's own order, this field should be set to the broker code.
BookType	This field should contain the type of order. BOARD_LOT_IN_TR (20000)/TRIMMED_BOARD_LOT_ACK_IN (20400). must have BookType 1 or 11 or 12.
BuySell	This field should specify whether the order is a buy or sell. It should take one of the following values. • '1' for Buy order • '2' for Sell order
DisclosedVol	This field should specify the quantity that has to be disclosed to the market. It is not applicable if the order has either the All Or None or the Immediate Or Cancel attribute set. It should not be greater than the volume of the order and not less than the Minimum Fill quantity if the Minimum Fill attribute is set. In either case, it cannot be less than the Minimum Disclosed Quantity allowed. It should be a multiple of the Regular lot.

Field Name	Brief Description
Volume	This field should specify the quantity of the order placed. The quantity should always be in multiples of Regular Lot except for Odd Lot orders and be less than the issued capital. The order will go for a freeze if the quantity is greater than the freeze quantity determined by NSE-Control.
Price	This field should contain the price at which the order is placed. To enter a Market order, the price should be zero. The price must be a multiple of the tick size. For Stop Loss orders, the limit price must be greater than the trigger price in case of a Buy order and less if it is a Sell order. This is to be multiplied by 100 before sending to the trading system host.
GoodTillDate	This field should contain the number of days for a GTD order. This field may be set in two ways. To specify an absolute date set this field to that date in number of seconds since midnight of January 1, 1980. To specify days set this to the number of days. This can take values from 2 to the maximum days specified for GTC orders only. If this field is non-zero, the GTC flag must be off.
Order_Flags	This structure specifies the attributes of an order. The Bit fields must be set / unset by Front end. Refer Quick Reference for Order Entry Parameters section for description. In order entry request, Mkt bit should be set to 0. If the order price is 0 in the request message, then the order will be considered as market order. For market order, Mkt bit will be set as 1 in order response message.
BranchId	This field should contain the ID of the branch of the particular broker.
UserId	This field should contain the ID of the user. This field accepts only numbers.
BrokerId	This field should contain the trading member ID.
Suspended	This field specifies whether the security is suspended or not. It should be set to blank while sending order entry request.
Settlor	This field contains the ID of the participants who are responsible for settling the trades through the custodians. By default, all

Field Name	Brief Description
	orders are treated as broker's own orders and this field defaults to the Broker Code.
ProClient	This field should contain one of the following values based on the order entering is on behalf of the broker or a trader. '1' - represents the client's order. '2' - represents a broker's order. '4' - represents warehousing order.
NNFField	This field should contain a 15 digit a unique identifier for various products deployed as per Exchange circular download ref no. 16519 dated December 14, 2010, and as updated from time to time
PAN	This field shall contain the PAN (Permanent Account Number / PAN_EXEMPT) - This field shall be mandatory for all orders (client / participant / PRO orders).
Algo ID	For Algo order this field shall contain the Algo ID issued by the exchange. For Non-Algo order, this field shall be Zero(0)
Reserved Filler	This field is reserved for future use. This should be set to Zero (0) while sending to the exchange trading system.

Trimmed Order Mod/Cancel Request Structure

Table 58 ORDER_OM_REQUEST

Structure Name	ORDER_OM_ REQUEST _TR		
Transaction Code	ORDER_MOD_IN_TR (20040)/ TRIMMED_ORDER_MOD_ACK_IN (20402) ORDER_CANCEL_IN_TR (20070)/ TRIMMED_ORDER_CANCEL_ACK_IN (20404)		
Packet Length	180 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
LogTime	LONG	4	2
UserId	LONG	4	6

Structure Name	ORDER_OM_ REQUEST _TR		
Transaction Code	ORDER_MOD_IN_TR (20040)/ TRIMMED_ORDER_MOD_ACK_IN (20402) ORDER_CANCEL_IN_TR (20070)/ TRIMMED_ORDER_CANCEL_ACK_IN (20404)		
Packet Length	180 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
ErrorCode	SHORT	2	10
TimeStamp1	LONG LONG	8	12
TimeStamp2	CHAR	1	20
Modified / Cancelled By	CHAR	1	21
ReasonCode	SHORT	2	22
SEC_INFO (Refer Table 4)	STRUCT	12	24
OrderNumber	DOUBLE	8	36
AccountNumber [10]	CHAR	10	44
BookType	SHORT	2	54
BuySell	SHORT	2	56
DisclosedVol	LONG	4	58
DisclosedVolRemaining	LONG	4	62
TotalVolRemaining	LONG	4	66
Volume	LONG	4	70
VolumeFilledToday	LONG	4	74
Price	LONG	4	78
EntryDateTime	LONG	4	82
LastModified	LONG	4	86
ST_ORDER_FLAGS (Refer Table 19.1 for small endian machines and Table 19.2 for big endian machines)	STRUCT	2	90
BranchId	SHORT	2	92
UserId	LONG	4	94
BrokerId [5]	CHAR	5	98
Suspended	CHAR	1	103
Settlor [12]	CHAR	12	104

Structure Name	ORDER_OM_ REQUEST _TR		
Transaction Code	ORDER_MOD_IN_TR (20040)/ TRIMMED_ORDER_MOD_ACK_IN (20402) ORDER_CANCEL_IN_TR (20070)/ TRIMMED_ORDER_CANCEL_ACK_IN (20404)		
Packet Length	180 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
ProClient	SHORT	2	116
SettlementType	SHORT	2	118
NNFField	DOUBLE	8	120
TransactionId	LONG	4	128
PAN	CHAR	10	132
Algo ID	LONG	4	142
Reserved Filler	SHORT	2	146
LastActivityReference	LONG LONG	8	148
Reserved	CHAR	24	156

Trimmed Order Mod/Cancel Response Structure

Table 59 ORDER_OM_RESPONSE

Structure Name	ORDER_OM_ RESPONSE_TR		
Transaction Code	ORDER_MOD_REJECT_TR (20042) ORDER_CANCEL_REJECT_TR (20072) ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075) ORDER_ERROR_TR (20231) PRICE_CONFIRMATION_TR (20012)		
Packet Length	216 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
LogTime	LONG	4	2
UserId	LONG	4	6

Structure Name	ORDER_OM_ RESPONSE_TR		
Transaction Code	ORDER_MOD_REJECT_TR (20042) ORDER_CANCEL_REJECT_TR (20072) ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075) ORDER_ERROR_TR (20231) PRICE_CONFIRMATION_TR (20012)		
Packet Length	216 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
ErrorCode	SHORT	2	10
TimeStamp1	LONG LONG	8	12
TimeStamp2	CHAR	1	20
Modified / Cancelled By	CHAR	1	21
ReasonCode	SHORT	2	22
SEC_INFO (Refer Table 4)	STRUCT	12	24
OrderNumber	DOUBLE	8	36
AccountNumber [10]	CHAR	10	44
BookType	SHORT	2	54
BuySell	SHORT	2	56
DisclosedVol	LONG	4	58
DisclosedVolRemaining	LONG	4	62
TotalVolRemaining	LONG	4	66
Volume	LONG	4	70
VolumeFilledToday	LONG	4	74
Price	LONG	4	78
EntryDateTime	LONG	4	82
LastModified	LONG	4	86
ST_ORDER_FLAGS (Refer Table 19.1 for small endian machines and Table 19.2 for big endian machines)	STRUCT	2	90
BranchId	SHORT	2	92
UserId	LONG	4	94

Structure Name	ORDER_OM_ RESPONSE_TR		
Transaction Code	ORDER_MOD_REJECT_TR (20042) ORDER_CANCEL_REJECT_TR (20072) ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075) ORDER_ERROR_TR (20231) PRICE_CONFIRMATION_TR (20012)		
Packet Length	216 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
BrokerId [5]	CHAR	5	98
Suspended	CHAR	1	103
Settlor [12]	CHAR	12	104
ProClient	SHORT	2	116
SettlementType	SHORT	2	118
NNFField	DOUBLE	8	120
TransactionId	LONG	4	128
Timestamp	LONG LONG	8	132
PAN	CHAR	10	140
Algo ID	LONG	4	150
Reserved Filler	SHORT	2	154
LastActivityReference	LONG LONG	8	156
Reserved	CHAR	52	164

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_MOD_REJECT_TR (20042) ORDER_CANCEL_REJECT_TR (20072) ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075) ORDER_ERROR_TR (20231) PRICE_CONFIRMATION_TR (20012)

Field Name	Brief Description
TraderId	This field should contain the user ID of the user.
TimeStamp2	This field contains the number of the machine from which the packet is coming.
ModCxlBy	This field denotes which person has modified or cancelled a particular order. It should contain one of the following values: • 'T' for Trader • 'B' for Branch Manager • 'M' for Corporate Manager • 'C' for Exchange
ReasonCode	This field contains the reason code for a particular order request rejection or order being frozen. This has the details regarding the error along with the error code. This field should be set to zero while sending the request to the host. Refer to Reason Codes in Appendix.
SEC_INFO	This structure should contain the Symbol and Series of the security.
AccountNumber	If the order is entered on behalf of a trader, the trader account number should be specified in this field. For broker's own order, this field should be set to the broker code.
BookType	This field should contain the type of order. Refer to Book Types in Appendix. The Request messages in transaction codes mentioned above must have BookType 1 or 11 or 12.
BuySell	This field should specify whether the order is a buy or sell. It should take one of the following values. • '1' for Buy order • '2' for Sell order
DisclosedVol	This field should specify the quantity that has to be disclosed to the market. It is not applicable if the order has either the All Or None or the Immediate Or Cancel attribute set. It should not be greater than the volume of the order and not less than the Minimum Fill quantity if the Minimum Fill attribute is set. In either case, it cannot be less than the Minimum Disclosed Quantity allowed. It should be a multiple of the Regular lot.

Field Name	Brief Description
DisclosedVolRemaining	This field contains the disclosed volume remaining from the original disclosed volume after trade(s). This should be set to zero while sending to the host.
TotalVolRemaining	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry, this field should be set to Volume. Thereafter, for every response the trading system will return this value.
Volume	This field should specify the quantity of the order placed. The quantity should always be in multiples of Regular Lot except for Odd Lot orders and be less than the issued capital. The order will go for a freeze if the quantity is greater than the freeze quantity determined by NSE-Control.
VolumeFilledToday	This field contains the total quantity traded in a day.
Price	This field should contain the price at which the order is placed. To enter a Market order, the price should be zero. The price must be a multiple of the tick size. For Stop Loss orders, the limit price must be greater than the trigger price in case of a Buy order and less if it is a Sell order. This is to be multiplied by 100 before sending to the trading system host.
EntryDateTime	This field should be set to zero while sending the order entry request.
LastModified	If the order has been modified, this field contains the time when the order was last modified. It is the time in seconds from midnight of January 1 1980, This field should be set to zero for the order entry request (it is same as Entry Date Time.)
Order_Flags	This structure specifies the attributes of an order. The Bit fields must be set / unset by Front end. Refer section Quick Reference for Order Entry Parameters for description. In order entry request, Mkt bit should be set to 0. If the order price is 0 in the request message, then the order will be considered as market order. For market order, Mkt bit will be set as 1 in order response message.

Field Name	Brief Description
BranchId	This field should contain the ID of the branch of the particular broker.
TraderId	In Request packet, this field should contain the ID of the user on whose behalf order is to be modified/cancelled. This field accepts only numbers.
BrokerId	This field should contain the trading member ID.
Suspended	This field specifies whether the security is suspended or not. It should be set to blank while sending order entry request.
ProClient	This field should contain one of the following values based on the order entering is on behalf of the broker or a trader. '1' - represents the client's order. '2' - represents a broker's order. '4' - represents warehousing order.
SettlementType	This field contains the settlement type. It can be one of the following: '0' – T+0 settlement '1' – T+1 settlement This field should be set to zero while sending to the host.
NNFField	This field should contain a 15 digit a unique identifier for various products deployed as per Exchange circular download ref no. 16519 dated December 14, 2010, and as updated from time to time
Timestamp	Time in this field will be populated in nanoseconds (from 01-Jan-1980 00:00:00). This time is stamped at the matching engine in the trading system.
PAN	This field shall contain the PAN (Permanent Account Number / PAN_EXEMPT) - This field shall be mandatory for all orders (client / participant / PRO orders).
Algo ID	For Algo order this field shall contain the Algo ID issued by the exchange. For Non-Algo order, this field shall be Zero(0)
Reserved Filler	This field is reserved for future use. This should be set to Zero (0) while sending to the exchange trading system.
LastActivityReference	For Order Modification/Cancellation request, this field should contains LastActivityReference value received in response of last

Field Name	Brief Description
	activity done on that order. Last activity could be order entry confirmation, order modification confirmation or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Trimmed Trade Confirmation Structure

Table 60 MS_TRADE_CONFIRM

Structure Name	MS_TRADE_CONFIRM_TR		
Transaction Code	TRADE_CONFIRMATION_TR (20222)		
Packet Length	192 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
LogTime	LONG	4	2
UserId	LONG	4	6
TimeStamp	LONG LONG	8	10
TimeStamp1	CHAR	8	18
ResponseOrderNumber	DOUBLE	8	26
TimeStamp2	CHAR	1	34
BrokerId [5]	CHAR	5	35
TraderNum	LONG	4	40
BuySell	SHORT	2	44
AccountNum [10]	CHAR	10	46
OriginalVol	LONG	4	56
DisclosedVol	LONG	4	60
RemainingVol	LONG	4	64
DisclosedVolRemaining	LONG	4	68
Price	LONG	4	72
ST_ORDER_FLAGS (Refer Table 19.1 for small endian machines and Table 19.2 for big endian machines)	STRUCT	2	76

Structure Name	MS_TRADE_CONFIRM_TR		
Transaction Code	TRADE_CONFIRMATION_TR (20222)		
Packet Length	192 bytes		
Usage	PRAGMA Pack(2)		
Field Name	Data Type	Size in Byte	Offset
FillNumber	LONG	4	78
FillQty	LONG	4	82
FillPrice	LONG	4	86
VolFilledToday	LONG	4	90
ActivityType [2]	CHAR	2	94
ActivityTime	LONG	4	96
SEC_INFO (Refer Table 4)	STRUCT	12	110
BookType	SHORT	2	112
ProClient	SHORT	2	114
PAN	CHAR	10	116
Algo ID	LONG	4	126
Reserved Filler	SHORT	2	130
LastActivityReference	LONG LONG	8	132
Reserved	CHAR	52	140

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CONFIRMATION_TR (20222).
Timestamp	Time in this field will be populated in nanoseconds (from 01-Jan-1980 00:00:00). This time is stamped at the matching engine in the trading system.
PAN	This field shall contain the PAN
Algo ID	This field shall contain the Algo ID
Reserved Filler	This field is reserved for future use
LastActivityReference	This field shall contain a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Note: The other field descriptions are the same as MS_TRADE_CONFIRM.

Annexure for Encryption/Decryption

Sr. No.	The following are sample function calls of OpenSSL library in Linux (for reference)
1	***Note –*** • Openssl Library version used is OpenSSL 1.1.1 for existing encryption and 3.4.0 for new encryption with authentication. • TLS protocol version has been set to 1.3 (TLS1_3_VERSION). Following are the system library calls for TLS1.3- **SSL/TLS library initialization →** 1. **SSL_library_init**() - Initialize SSL library by registering algorithms. 2. **OpenSSL_add_all_algorithms**() - Adds all algorithms to the table (digests and ciphers) 3. **SSL_load_error_strings**() - Registers the error strings for all libcrypto and libssl error strings. 4. **SSL_CTX_new**(**TLS_client_method**()) - Create a new SSL_CTX object as framework for TLS/SSL enabled functions. 5. **SSL_CTX_set_min_proto_version**(SSL_CTX *ctx, int version) - Set the minimum protocol versions to TLS1_3_VERSION. 6. **SSL_CTX_set_max_proto_version**(SSL_CTX *ctx, int version) - Set the maximum protocol versions to TLS1_3_VERSION. **Establishing the SSL/TLS connection→** 1. **socket**(PF_INET, SOCK_STREAM, 0) - Create TCP socket. 2. **connect**(int sockfd, const struct sockaddr *addr, socklen_t addrlen) - Initiate the TCP/IP connection with server. 3. **SSL_new**(SSL_CTX *ctx) - Create new SSL connection state. 4. **SSL_set_fd**(SSL *ssl, int fd) - Attach the socket descriptor. 5. **SSL_connect**(SSL *ssl) - Perform the SSL connection. **Validating the Gateway Router server certificate →** 1. **SSL_get_peer_certificate**(const SSL *ssl) - Get the server's certificate. 2. **X509_STORE_new**() - This function returns a new X509_STORE.

	3. **X509_STORE_CTX_new**() - This function returns a newly initialised X509_STORE_CTX. 4. **X509_STORE_load_locations**(X509_STORE *ctx, const char *file, const char *dir) - Configure files and directories used by a certificate store. The path of CA certificate (gr_ca_cert1.pem) will be used in this function. The CA certificate (gr_ca_cert1.pem) will be provided by the Exchange for validation of Gateway Router certificate. 5. **X509_STORE_CTX_init**(X509_STORE_CTX *ctx, X509_STORE *trust_store, X509 *target, STACK_OF(X509) *untrusted) - This function returns a newly initialised X509_STORE_CTX structure. 6. **X509_verify_cert**(X509_STORE_CTX *ctx) - This function builds and verify X509 certificate chain. **Send and Receive messages on SSL/TLS connection →** 1. **SSL_write**(SSL *ssl, const void *buf, int num) - Send message on SSL. 2. **SSL_read**(SSL *ssl, void *buf, int num) - Receive message from SSL.

2	**For symmetric encryption/decryption methodology –**
	
	Existing encryption mechanism -
	
	Initialization→
	void encrypt_EVP_aes_256_cbc_init(EVP_CIPHER_CTX **ctx, unsigned char *key, unsigned char *iv)
	{
	if(!(*ctx = **EVP_CIPHER_CTX_new()**))
	handleErrors();
	
	if(1 != **EVP_EncryptInit_ex**(*ctx**, EVP_aes_256_gcm()**, NULL, key, iv))
	handleErrors();
	}
	
	Encryption→
	void encrypt(EVP_CIPHER_CTX *ctx, unsigned char *plaintext, int plaintext_len, unsigned char *ciphertext, int *ciphertext_len)
	{
	int len;

```
    if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
      handleErrors();
    *ciphertext_len = len;
  }
```

Decryption:

Initialization→

```
  void decrypt_EVP_aes_256_cbc_init(EVP_CIPHER_CTX **ctx, unsigned char *key,
  unsigned char *iv)
  {
    if(!(*ctx = EVP_CIPHER_CTX_new()))
      handleErrors();

    if(1 != EVP_DecryptInit_ex(*ctx, EVP_aes_256_gcm(), NULL, key, iv))
      handleErrors();
  }
```

Decryption→

```
  int decrypt(EVP_CIPHER_CTX *ctx, unsigned char *ciphertext, int ciphertext_len,
  unsigned char *plaintext, int *plaintext_len)
  {
    int len;

    if(1 != EVP_DecryptUpdate(ctx, plaintext, &len, ciphertext, ciphertext_len))
      handleErrors();
    *plaintext_len = len;
  }
```

New encryption with authentication mechanism-

GCM_IV_LEN 16
aad_len 12
GCM_TAG_LEN 16

Encryption Block:

Initialization→

```
void encrypt_init(EVP_CIPHER_CTX **enc_ctx,
                  const unsigned char *key,
                  const unsigned char *iv)
{
   int retv = 0;
   if (!*enc_ctx)
      *enc_ctx = EVP_CIPHER_CTX_new();
    retv = EVP_EncryptInit (*enc_ctx, EVP_aes_256_gcm(), NULL, NULL);
    retv = EVP_CIPHER_CTX_ctrl (*enc_ctx, EVP_CTRL_GCM_SET_IVLEN, GCM_IV_LEN,
NULL);
    retv = EVP_EncryptInit(*enc_ctx, NULL, key, NULL);
}
```

Encryption→

```
void encrypt_data(EVP_CIPHER_CTX *enc_ctx,
                  const unsigned char *iv,
                  const unsigned char *plaintext,
                  int plaintext_len,
                  const unsigned char *aad,
                  int aad_len,
                  unsigned char *ciphertext,
                  unsigned char *tag)
{
   int len = 0;
   int len2=0;
   int retv = 0;
   retv = EVP_EncryptInit(enc_ctx, NULL, NULL, iv);
   retv = EVP_EncryptUpdate(enc_ctx, NULL, &len, aad, aad_len);
   retv = EVP_EncryptUpdate(enc_ctx, ciphertext, &len, plaintext, plaintext_len);
   OSSL_PARAM params[2] = {
      OSSL_PARAM_END, OSSL_PARAM_END
   };
   retv = EVP_EncryptFinal_ex(enc_ctx, ciphertext, &len2);
   params[0] = OSSL_PARAM_construct_octet_string(OSSL_CIPHER_PARAM_AEAD_TAG,
tag, GCM_TAG_LEN);
   retv = EVP_CIPHER_CTX_get_params(enc_ctx, params);
}
```

Decryption Block:

Initialization→

```
void decrypt_init(EVP_CIPHER_CTX **dec_ctx,
                  const unsigned char *key,
                  const unsigned char *iv)
{   int retv = 0;
    if (!*dec_ctx)
        *dec_ctx = EVP_CIPHER_CTX_new();
     retv = EVP_DecryptInit(*dec_ctx, EVP_aes_256_gcm(), NULL, NULL);
     retv = EVP_CIPHER_CTX_ctrl(*dec_ctx, EVP_CTRL_GCM_SET_IVLEN, GCM_IV_LEN,
NULL);
     retv = EVP_DecryptInit(*dec_ctx, NULL,key, NULL);
}
```

Decryption→

```
void decrypt_data(EVP_CIPHER_CTX *dec_ctx,
                  const unsigned char *iv,
                  const unsigned char *ciphertext,
                  int ciphertext_len,
                  const unsigned char *aad,
                  int aad_len,
                  const unsigned char *tag,
                  unsigned char *plaintext)
{
    int len = 0;
    int len2 = 0;
    int retv;

    retv = EVP_DecryptInit(dec_ctx, NULL, NULL, iv);
    retv = EVP_DecryptUpdate(dec_ctx, NULL, &len, aad, aad_len);
    retv = EVP_DecryptUpdate(dec_ctx, plaintext, &len, ciphertext, ciphertext_len);
    OSSL_PARAM params[2] = {
        OSSL_PARAM_END, OSSL_PARAM_END
    };
    params[0] = OSSL_PARAM_construct_octet_string(OSSL_CIPHER_PARAM_AEAD_TAG,
tag, GCM_TAG_LEN);
    retv = EVP_CIPHER_CTX_set_params(dec_ctx, params);
    retv = EVP_DecryptFinal_ex(dec_ctx,plaintext, &len2);
    if (retv <= 0)
        printf("!!!Decryption Failed!!!\n");;
```

```
    else
        printf("!!!Decryption Successful!!!\n");
}
```

Note –

- The ones highlighted in bold are OpenSSL library functions.
- plaintext is the actual message buffer.
- ciphertext is the encrypted message buffer.

```
//=====Pseudocode Dynamic IV changes=====
// Define the IV structure
typedef struct {
  char       caStaticIv[8];    // Static IV (8 bytes)
  long long  lDynamicIv;       // Dynamic IV (64-bit integer)
} CRYPTOGRAPHIC_IV_KEY;

// Original IV received from GR response
CRYPTOGRAPHIC_IV_KEY sIv;

// Separate copies for encryption and decryption
CRYPTOGRAPHIC_IV_KEY sEncCryptoGraphicIv;
CRYPTOGRAPHIC_IV_KEY sDecCryptoGraphicIv;

// Step 1: Initialize from GR response
sIv = get_iv_from_gr_response();  // sIv is populated with the static and dynamic IV values

// Step 2: Create two copies - One for encryption and One for decryption.
sEncCryptoGraphicIv = sIv;
sDecCryptoGraphicIv = sIv;

// Step 3: Before Encryption - The dynamic IV is incremented by 1.
sEncCryptoGraphicIv.lDynamicIv += 1;
encrypted_data = encrypt(data, &sEncCryptoGraphicIv);

// Step 4: Before Decryption - The dynamic IV is decremented by 1.
sDecCryptoGraphicIv.lDynamicIv -= 1;
decrypted_data = decrypt(encrypted_data, &sDecCryptoGraphicIv);
```