Trimmed Protocol for Non-NEAT Front End (NNF)

Futures and Options Trading System

Version 9.46

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

Futures and Options Trading Systems		
Revision History		
Version	Pages Changed	Description
9.37		Introduction of new error code for user having NO trading rights.
9.38		Added new datatype unsigned long (4 bytes). Modified Total Traded volume datatype from long to unsigned long for below transaction codes. RPRT_MARKET_STATS_OUT_RPT (1833) BCAST_MBO_MBP_UPDATE (7200) BCAST_ONLY_MBP (7208) BCAST_SPD_MBP_DELTA (7211)
		Updated message length of UPDATE_LOCALDB_DATA as 548 bytes.
9.39		Added new datatype unsigned long (4 bytes). Modified Open Interest datatype from long to unsigned long for below transaction codes. RPRT_MARKET_STATS_OUT_RPT (1833) BCAST_MW_ROUND_ROBIN (7201) BCAST_TICKER_AND_MKT_INDEX (7202) MKT_MVMT_CM_OI_IN (7130)
		Field description added for User Password and New password
9.40	199	Heartbeat echo back

	235,246	Addition of new error codes - 16020 - Order price is outside the revised price range. 17107 - Heart beat rate exceeded by the member.
9.41	246	Modification of error code description: 16807 Addition of new error code: 16816
9.42	235, 236	Modified error code description: 16012 Addition of new error codes: 16730 16731 16732
9.43		Updated section –
	~~69, 83-86,93,102~~	~~Chapter 4,5,6 - Transaction codes for Immediate Ack Messages~~
	136,153-154,163-165,175-177	Chapter 9 - Introduction of new additional transcodes of Ticker and Market Index, Market Watch Update & Underlying Open Interest
	~~179-181~~	~~Chapter 10 – Details of new encryption mechanism~~
	~~183-188~~	~~Chapter 11 - Direct interface to Exchange Trading System~~
	~~240-245,265,266,268-270~~	~~Appendix for list of Transaction codes and trimmed structures~~
	~~276-277,279-281~~	~~Annexure for new Encryption/Decryption~~
	~~223-226~~	~~New Section –~~ ~~Chapter 15 Immediate order acknowledgment Message~~
		Refer Version 9.44 for immediate ack and encryption related changes

	237,238	Addition of new error code 16733, 17185, 17186
	122-126,130,240	Updated section- Chapter 8 - Introduction of new additional transcodes of Bhavcopy Broadcast
9.44	75, 91-94,101,102,111	Chapter 4,5,6 - Transaction codes for Immediate Ack Messages
	195-197	Chapter 10 – Details of new encryption mechanism
	199-204	Chapter 11 - Direct interface to Exchange Trading System
	261-263,266,267,287,289,292,293	Appendix for list of Transaction codes and trimmed structures
	299,301-305	Annexure for new Encryption/Decryption
	243-246	New Section – Chapter 15 Immediate order acknowledgment Message
9.45	31,41,56,62-63,66,69,71-72,75, ,81-82,122,124,126,158,164,173,177-180,183, 251-252,255-256,261,262,264,275,282	Changes for Pre-Open Session
9.46	105	Updated header in transcode 7309, 7341
	175,176,178	Updates for transcode 7208

Preface

Purpose

This document describes the protocol to be used for Non-NEAT Front end (NNF) to communicate with the Futures and Options Trading System (FOTS). Thus, it serves as a guide to the NNF programmers, for developing software that can communicate with FOTS.

Target Audience

This document is written for system designers and programmers of user organisations and third-party software developers, who are responsible for the development of software that interacts with NSE's FOTS.

Organisation of this Document

This document is organised as follows:

Chapters	Description
Chapter 1	Provides a brief introduction to Non-NEAT Front end (NNF). It also details the communication network connections for NNF users.
Chapter 2	Lists the 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 transactions and structures for a trader to log on to the trading system. It also provides the same information for 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 transactions and structures for entering new orders, modifying an existing order, and canceling outstanding orders.

Chapters	Description
Chapter 5	Details the order types, rules and structure of Spread order entry and the structures of the system responses. It also covers the order modification, trade modification and cancellation structures for Spread orders.
Chapter 6	Describes rules and structure of 2L and 3L order entry and the structures of the system responses. It also covers the trade modification and cancellation structures for 2L and 3L orders.
Chapter 7	Describes the messages that are received on the interactive connection. These messages are not received by users in response to any request.
Chapter 8	Discusses the end of the trading day activities. It details the messages, structures and reports related to transmission of header, data and trailer of Bhavcopy.
Chapter 9	Describes the structures of various Broadcast messages and the Compression and Decompression algorithm of Broadcast data.
Chapter 10	Encryption Decryption of Interactive Messages.
Chapter 11	Describes how member systems can directly connect to NSE for trading, while using existing formats of business messages from NNF API documents.
Chapter 12	Describes how exception at trading end should be handled.
Chapter 13	Describes the functionalities made available to CM / BM users.
Chapter 14	Describes the structures for Give up Approve and Reject Confirmation responses.
Chapter 15	Immediate order acknowledgement Message
Appendix	Lists the error, transaction and reason codes and covers the various market statuses, market types and book types.

Abbreviations and Acronyms

The abbreviations and acronyms used in this document are as follows:

AGM	Annual General Meeting
AON	All Or None
ATO	At The Opening
AU	Auction
BCID	Broadcast Circuit ID
BM	Branch Manager
CLI	Client
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
MWL	Market Wide Limit
NNF	Non Neat Front End
NSE	National Stock Exchange
NT	Negotiated Trade
OI	Open Interest

OL	Odd Lot
OSL	Open Strata Link
PRO	Proprietary
RL	Regular Lot
SL	Stop Loss
ST	Special Terms
TM	Trading Member
TP	Trigger Price
TWS	Trader Workstation
VCID	Virtual Circuit ID
VSAT	Very Small Aperture Terminal
VV.RR.SS	Version. Release. Sub-release
WHS	Warehouse

CONTENTS

Chapter 1 Introduction

The National Stock Exchange of India Ltd (NSEIL) provides a fully automated screen based trading system. This enables 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. For trading the trading members can use NEAT Front end or Non-NEAT Front end (NNF) to establish a network connection with the Futures and Options host system of National Stock Exchange (NSE). NNF is a front end, which is developed and maintained by vendors other than NSE. NSE provides the NNF users with the design documents of the front end. However, their respective vendors support them and NSE is not responsible for the performance of NNF.

Chapter 2 General Guidelines

This chapter provides general guidelines for the designers and programmers who develop NNF. It also provides information on data types and their size to be used in the message structure, which helps the programmer in their development activity. The sections covered in this chapter are:

- Guidelines for Designers
- Guidelines for Programmers
- Message Structure Details
- Data Types Used in Message Structure
- Message Header
- Broadcast Process Header
- Error Message
- Communication Network Connection

Guidelines for Designers

The guidelines to be followed by designers are as follows:

1. The order of the log-on messages should strictly be maintained as given in Chapter 3 Logon and Download of this document**.** Otherwise, the user will not be able to log on to the trading system.
2. All messages sent by the trading system should be time-stamped, that is the time of the message should be specified.
3. All time fields should be number of seconds from midnight January 1, 1980.
4. No host-end inquiries will be permitted for NNF users.
5. All price fields must be multiplied by 100 before sending to the host end and divided by 100 while receiving from the host end. This is because the trading system processes prices in paise (100 paise equals 1 Rupee).

Guidelines for Programmers

The guidelines to be followed by programmers are as follows:

1. If your system uses little-endian order, the data types which occupy more than one byte in a packet (such as UINT, SHORT, LONG and DOUBLE) 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. Hence, you need to twiddle such data types before sending and after receiving to ensure that correct data is sent and received.

   Note:
   Twiddling is required because endian order can be of two different types – big and little. A big-endian representation has a multibyte 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. Intel's 80x86 processors and their clones are little endian. Sun's SPARC, Motorola's 68K, and the PowerPC families are all big endian.

   All of the protocol layers in the TCP/IP suite are defined to be big endian. The trading system uses big-endian order. Hence, if your system uses little-endian order twiddle the numeric value before sending and after receiving over a TCP/IP connection.
2. All alphabetical data must be converted to upper case before sending to the host. No NULL terminated strings should be sent to the host end. Instead, terminate strings 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:

- Field of type CHAR or Unsigned CHAR, or CHAR Arrays should be byte aligned.
- Structures of odd size should be padded to an even number of bytes.
- All other types of structure members should be word aligned.

4. All numeric data must be set to zero (0) before sending to the host, unless a value is assigned to it.
5. Whenever the field name is mentioned as Reserved for example: Reserved field in Broadcast Process Header, it should be mapped to CHAR buffer and initialized to NULL.

Note:

- The values of all the constants and transaction codes given in the document are listed in the Appendix.
- The suffix IN in the transaction codes implies that the request is sent from the NNF / NEAT application to the trading system whereas the suffix OUT implies that the message is sent from the trading system to NNF / NEAT application.

Message Structure Details

All packets (messages), flow between NNF application and trading system, consist of two parts namely, message header and message data, which are described as follows:

- The message header consists of the fields of the header which is prefaced with all the structures.

  Note: 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 NNF and the host end.
- The message data consists of the actual data that is sent across to the host or received from the host.

Data Types Used in Message Structure

The following table lists the data types to be used in message structure for NNF development.

Data Type	Size of Bytes	Signed / Unsigned
CHAR	1	Signed
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. 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	CHAR	2	6
TraderId	LONG	4	8
ErrorCode	SHORT	2	12
Timestamp	LONG LONG	8	14
TimeStamp1	CHAR	8	22
TimeStamp2	CHAR	8	30
MessageLength	SHORT	2	38

The following table provides the brief description of the various fields present in the MESSAGE_HEADER structure.

Field Name	Brief Description
TransactionCode	This field should contain the transaction message number. It describes the type of message sent or received.
LogTime	This field should be set to zero while sending messages to the host. For messages coming from the host, this contains the time when the message was generated by the trading system.

Field Name	Brief Description
AlphaChar	This field should be set to the first two characters of Symbol if the message structure contains Symbol and Series. During logon process, in the SYSTEM_INFORMATfION_OUT message response, this field should contain the number of modules. Based upon this number of modules, Frontend will populate the module_id in alpha_char field of DOWNLOAD_REQUEST packet and send to host. The module_id shall be populated in the first byte (AlphaChar[0]) and should be interpreted as an integer value and not as a character value. In other cases, it should be set to blank.
TraderId	This field should contain the user ID.
ErrorCode	This should be set to zero while sending messages to the host. For messages coming from the host, this 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 at the host end. For *transcodes listed* in appendix, time in this field will be populated in nanoseceonds (from 01-Jan-1980 00:00:00). This time is stamped at the matching engine in the trading system.
TimeStamp1	This field contains the time when the message arrives at the trading system host. This should be set to numeric zero while sending to host. Note In TimeStamp1, current time is sent in jiffies from host end. This is 8 bytes in host end. In front end, typecast the first four and the next four bytes into double and store each of these in separate variables. These values are used while requesting message area download. *Jiffy is a Unit of Time (1 second = 65536 jiffies)*
TimeStamp2	This field should be set to numeric zero while sending to the host. For messages coming from the host, this field contains the number of the machine from which the packet is coming. Note In TimeStamp2, machine number is sent from the host end. This is 8 bytes in host end and CHAR [8] in front end. In front end, if it is an interactive connection, machine number is stored in 7th position. If it is a broadcast connection, machine number is stored in 0th position.

Field Name	Brief Description
	Machine / Stream no. should be interpreted as integer value and not as character value. Values will be numeric value 1,2,3, ...,10,11 etc. and can range from 1 to 127
MessageLength	This field is set to the length of the entire message, including the length of Message Header.

Inner Message Header

Each structure in the Data of Update Local Database Data/Message Download Data responses is prefaced with a 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	CHAR	2	8
TransactionCode	SHORT	2	10
ErrorCode	SHORT	2	12
Timestamp	LONG LONG	8	14
TimeStamp1	CHAR	8	22
TimeStamp2	CHAR	8	30
MessageLength	SHORT	2	38

Broadcast Process Header

All broadcast messages like market open, market close, market in pre-open, market by price are prefaced with BCAST_HEADER. The structure of the BCAST_HEADER is as follows:

Table 3 BCAST HEADER

Structure Name	BCAST_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	2	0

Structure Name	BCAST_HEADER		
Packet Length	40 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	2	2
LogTime	LONG	4	4
AlphaChar	CHAR	2	8
TransactionCode	SHORT	2	10
ErrorCode	SHORT	2	12
BCSeqNo	LONG	4	14
Reserved	CHAR	1	18
Reserved	CHAR	3	19
TimeStamp2	CHAR	8	22
Filler2	BYTE	8	30
MessageLength	SHORT	2	38

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

Field Name	Brief Description
LogTime	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 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 for Ericcson switch. This is used for Future and Option (FO) Market By Price packet mapped to that of Capital Market structure. It is used to identify if the broadcast response is for Future & Option or for Capital market.
TimeStamp2	This field contains the time when the message is sent from the host.
Filler2	This field contains the machine number. Note The machine number is stored in 0th position. Machine / Stream no. should be interpreted as integer value.

	Values will be numeric value 1,2,3, ….,10,11 etc. and can range from 1 to 127
MessageLength	This field is set to the length of the entire message, including the length of the message header.

Error Message

When the Error Code in the MESSAGE_HEADER is not zero, the structure sent is ERROR RESPONSE. The Error Message will describe the error received. The structure is as follows:

Table 4 MS ERROR RESPONSE

Structure Name	MS_ERROR_RESPONSE		
Packet Length	182 bytes		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE HEADER in Chapter 2)*	STRUCT	40	0
Key	CHAR	14	40
ErrorMessage	CHAR	128	54

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

Field Name	Brief Description
Key	This field contains the token number of the Contract.
ErrorMessage	This field contains 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 NNF / 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.

Message rate control

Currently exchange trading system controls the Order message rate per second for each connected BOX id. Member systems must count the number of messages sent in a second and not exceed the message rate allocated by the member. (Member systems must maintain the message rate per second which the member has subscribed for). If there is breach, members shall experience disconnection from Exchange Trading System as the Exchange shall logoff the Box id. Effectively all user ids connected to the Box id shall also be disconnected.

The message flow control mechanism as applicable irrespective of “NEAT Adaptor” or “Direct Connection” mode of connecting to the Exchange is as follows:

For Members connecting on LAN i.e. through Colocation IP’s:

The message threshold is currently set at “configured message rate per second +10%”. Accordingly, the messages received beyond subscribed rate up and to 110% of subscribed rate are rejected by the Exchange. Further the Box id is disconnected if the messages exceed 110% of the subscribed rate for the respective segments.

For Members connecting on WAN i.e. through Non-Colocation IP’s:

The message threshold is currently set at “configured message rate per second * X factor”. Accordingly, the Box id is disconnected if the messages exceed the threshold calculated as mentioned above of the subscribed rate for the respective segments.

The X factor is internally decided by the Exchange in order to account for the network propagation delay faced by the members for connection via non-colocation facility.

For e.g., if the order rate decided is 100 msgs/second, and the order rate is being counted separately by the trading system and the exchange host, it may happen sometimes that the order rate as seen by the exchange host exceeds 100 msgs/second, due to different time window calculations at both ends. Clock time is not to be considered for message-rate. Member systems are expected to control their message rate per second (the ‘second’ is not of the clock, but of running time, i.e. sliding window). I.e., when an order is being sent, check how many messages are already sent in t-minus- 1000 milliseconds or t-minus-1000000 microseconds.

Currently this validation is done every second.

Additional message rate control at Milisecond level

As per proposed change

An additional check will be done for the message rate at milli second granularity.
The number of messages should not exceed an absolute limit of 120 for every 100 milli seconds ie. 120 messages in 100 milli seconds will be allowed.

The messages which are exceeding the milli second threshold will be rejected.
The counting of messages is based on sliding window protocol which is already followed for box rate validation every second.
The additional check is applicable for only connections which are originating from servers in LAN i.e. Co location IPs.
The additional threshold limit is same for all Co-location connections irrespective of the box rate limit.

Chapter 3 Logon and Download

This chapter describes the message structures and its fields for a trader to log on to the trading system. It discusses the logon request and the system responses. This chapter 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 sections covered in this chapter are:

- Order of Transaction Code Exchanges during Logon and Logoff
- Logon Request
- Logon Response
- System Information Download
- Update Local Database Download
- Message Download
- Logoff Request
- Logoff Confirmation Response

Order of Transaction Code Exchanges during Logon and Logoff

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

Sequence No	Transaction Code	Sent By	Received By
1	SIGN_ON_REQUEST_IN	NNF	Host End
2	SIGN_ON_REQUEST_OUT	Host End	NNF
3	SYSTEM_INFORMATION_IN	NNF	Host End
4	SYSTEM_INFORMATION_OUT	Host End	NNF
5	UPDATE_LOCALDB_IN	NNF	Host End
6	UPDATE_LOCALDB_HEADER	Host End	NNF
7	UPDATE_LOCALDB_DATA	Host End	NNF
8	UPDATE_LOCALDB_TRAILER	Host End	NNF

Sequence No	Transaction Code	Sent By	Received By
9	DOWNLOAD_REQUEST (Module 1)	NNF	Host End
10	HEADER_RECORD	Host End	NNF
11	MESSAGE_RECORD	Host End	NNF
12	TRAILER_RECORD	Host End	NNF
13	DOWNLOAD_REQUEST (Module 2)	NNF	Host End
14	HEADER_RECORD	Host End	NNF
15	MESSAGE_RECORD	Host End	NNF
16	TRAILER_RECORD	Host End	NNF
17	DOWNLOAD_REQUEST (Module 3)	NNF	Host End
18	HEADER_RECORD	Host End	NNF
19	MESSAGE_RECORD	Host End	NNF
20	TRAILER_RECORD	Host End	NNF
21	DOWNLOAD_REQUEST (Module 4)	NNF	Host End
22	HEADER_RECORD	Host End	NNF
23	MESSAGE_RECORD	Host End	NNF
24	TRAILER_RECORD	Host End	NNF

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

Sequence No	Transaction Code	Sent By	Received By
1	SIGN_OFF_REQUEST_IN	NNF	Host End
2	SIGN_OFF_REQUEST_OUT	Host End	NNF

The structure given below is part of Logon request and response structure. It specifies the markets that are allowed for the trading member to place an order. Note to NNF Developer: Use any one of following two structures:

Table 5 ST_BROKER_ELIGIBILITY_PER_MKT

<table>
<tr><td>Structure Name</td><td colspan="3">ST_BROKER_ELIGIBILITY_PER_MKT</td></tr>
<tr><td>Packet Length</td><td colspan="3">2 bytes</td></tr>
<tr><td>Field Name</td><td>Data Type</td><td>Size</td><td>Offset</td></tr>
<tr><td colspan="4">For Small Endian Machines</td></tr>
<tr><td>Reserved</td><td>BIT</td><td>4</td><td>0</td></tr>
</table>

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			
Normal Market	BIT	1	0
Oddlot Market	BIT	1	0
Spot Market	BIT	1	0
Auction Market	BIT	1	0
Reserved	BIT	4	0
Reserved	BIT	7	1
PreOpen	BIT	1	1

Logon Request

The process by which a trader logs on to the trading system is called Logon Process. User needs to send the request as per the structure given below for establishing an interactive circuit with the host end:

Table 6 MS_SIGNON

Structure Name	MS_SIGNON		
Packet Length	278 bytes		
Transaction Code	SIGN_ON_REQUEST_IN (2300)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to Message Header in Chapter 2)*	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

Structure Name	MS_SIGNON		
Packet Length	278 bytes		
Transaction Code	SIGN_ON_REQUEST_IN (2300)		
Field Name	Data Type	Size in Byte	Offset
BranchID	SHORT	2	112
VersionNumber	LONG	4	114
Batch2StartTime	LONG	4	118
HostSwitchContext	CHAR	1	122
Colour	CHAR	50	123
Reserved	CHAR	1	173
UserType	SHORT	2	174
SequenceNumber	DOUBLE	8	176
WsClassName	CHAR	14	184
BrokerStatus	CHAR	1	198
ShowIndex	CHAR	1	199
ST_BROKER_ELIGIBILITY_PER_MKT	STRUCT	2	200
MemberType	SHORT	2	202
ClearingStatus	CHAR	1	204
BrokerName	CHAR	25	205
Reserved	CHAR	16	230
Reserved	CHAR	16	246
Reserved	CHAR	16	262

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

Field Name	Brief Description
TransactionCode	This field is part of MESSAGE_HEADER Structure *(Refer to MESSAGE_HEADER in Chapter 2).* The value should be SIGN_ON_REQUEST_IN (2300).
UserId	This field should contain the user ID of the member/broker. It accepts numbers only.
Password	The password should be of exact eight characters in length.The password should be alphanumeric i.e password should contain 1 upper case letter,1 lower case letter,1 numeral and 1 special character from the list @#$%&*/\. The trader should enter the password for a successful Logon. When the trader logs on for the first time the default password

Field Name	Brief Description
	provided by NSE i.e Neat@FO1must be entered and the password should be changed by entering a new password.
NewPassword	This field should be entered only when the trader wishes to change the password or the password has expired. The new password should be of eight characters. The new password should be alphanumeric i.e password should contain 1 upper case letter,1 lower case letter,1 numeral and 1 special character from the list @#$%&*/\.The new password entered should not be from the last 5 passwords.Otherwise this field should be blank. The New Password should be entered along with the old password in the Password field. While logging on to the system for the first time, the default password provided by NSE i.e Neat@FO1 must be changed.
TraderName	This field should be set to blank while sending to the host. In the response from host, it will contain the user's name.
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. Note Branch ID can be of 3 digits.
VersionNumber	This field should contain the version number of the trading system. The format is VERSION.RELEASE.SUB_RELEASE. (For example, 7.02.00)
Batch2StartTime	This field should be set to numerical zero.
HostSwitchContext	This field should be set to blank.
Colour	This field should be set to blank.
UserType	This field can take one of the following values. • '0' denotes Dealer • '4' denotes Corporate Manager • '5' denotes Branch Manager This field should be set to zero while sending to the host.
SequenceNumber	This field should be set to numerical zero while sending the request to host.
WorkstationNumber	This field should contain the network ID of the workstation. This is a seven digit number. The first five digits are fixed by the

Field Name	Brief Description
	Exchange and represent the various port / switch locations. The last two digits denote the user's PC - ID. It must be any number other than '00'.
BrokerStatus	This field should be set to blank.
ShowIndex	This field should be set to blank.
BrokerEligibilityPerMarket	This field should be set to numerical zero.
MemberType	This field should be set to numerical zero.
ClearingStatus	This field should be set to blank.
BrokerName	This field should contain the broker's name.

Logon Response

The trader, after issuing a sign-on request, waits for the system response. The response will either be **Confirmation** or **Logon Error**.

Logon Confirmation Response

A successful logon results in the Logon Confirmation Response, for which the following structure is sent back:

Table 7 MS_SIGNON

Structure Name	MS_SIGNON		
Packet Length	278 bytes		
Transaction Code	SIGN_ON_REQUEST_OUT (2301)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message_Header in Chapter 2)*	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

Structure Name	MS_SIGNON		
Packet Length	278 bytes		
Transaction Code	SIGN_ON_REQUEST_OUT (2301)		
Field Name	Data Type	Size in Byte	Offset
EndTime	LONG	4	118
Reserved	CHAR	1	122
Colour	CHAR	50	123
Reserved	CHAR	1	173
UserType	SHORT	2	174
SequenceNumber	DOUBLE	8	176
Reserved	CHAR	14	184
BrokerStatus	CHAR	1	198
ShowIndex	CHAR	1	199
ST_BROKER_ELIGIBILITY_PER_MKT	STRUCT	2	200
MemberType	SHORT	2	202
ClearingStatus	CHAR	1	204
BrokerName	CHAR	25	205
Reserved	CHAR	16	230
Reserved	CHAR	16	246
Reserved	CHAR	16	262

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

Field Name	Brief Description
TransactionCode	This field is part of Message Header structure. The value should be 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 NNF workstation must be synchronised with this time.
UserId	This field contains the ID of the user or broker.
Password	This field will be set to blank
NewPassword	This filed will be set to blank
TraderName	This field contains the user name.
LastPassword ChangeDateTime	This field contains the last date and time when the password was changed.
BrokerId	This field should contain the trading member ID.
BranchId	This field should contain the Branch ID to which the broker belongs. **Note** Branch ID can be of 3 digits.
EndTime	This field contains the time when the markets last closed and it is sent as the number of seconds since midnight of January 1, 1980.

Field Name	Brief Description
	Note: If this time is different from the time sent in an earlier logon, 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 in: • '0' denotes Dealer • '4' denotes Corporate Manager • '5' denotes Branch Manager
SequenceNumber	This field contains the time when the markets closed the previous trading day.
BrokerStatus	This field contains the current status of the Broker. It can be any of the following: • 'S' for Suspended • 'A' for Active • 'D' for Deactivated • 'C' for Closeout 'V' for Voluntary CloseOut
BrokerEligibilityPerMarket	This field 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'.
MemberType	This field contains the type of member. The possible values are as follows: • '1' denotes Trading Member only • '2' denotes Trading and Clearing Member • '3' denotes Clearing Member only • '4' denotes Professional Clearing Member and Trading Member
ClearingStatus	This field contains the Clearing status of the member. The possible values are: • 'A' denotes Active • 'S' denotes Suspended • 'D' denotes Deactivated • 'V' denotes Voluntary CloseOut
BrokerName	This field contains the name of the broker.

Logon Error

In case the logon is unsuccessful an error response is generated, for which the structure returned is:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header. The transaction code is SIGN_ON_REQUEST_OUT (2301).
ErrorCode	This stores the error number. If the version number is not the same as that at the host, the version number at the host can be extracted from the Error Message. It will be located in 8 bytes from the 95th byte in the string (ERROR_RESPONSE). The format of it will be VV.RR.SS. The version number at the front end should be set to VVRRSS. Note VV – Version Number RR - Release Number SS –Sub Release Number *Refer to List of Error Codes in Appendix.*

System Information Download

System information like the current status of the markets and the values of global variables can be downloaded by the trader. For this, a *system information* request is sent. A response is returned for the request.

System Information Request

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

Table 8 MS_SYSTEM_INFO_REQ

Structure Name	MS_SYSTEM_INFO_REQ
Packet Length	44 bytes

Transaction Code	SYSTEM_INFORMATION_IN (1600)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to* *Message Header in Chapter 2)*	STRUCT	40	0
LastUpdatePortfolioTIme	LONG	4	40

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header. The transaction code is SYSTEM_INFORMATION_IN (1600).
LastUpdate PortfolioTime	If there is no Exchange defined Portfolio in the Local Database, this field should be set to zero. **Note:** Last Update Portfolio Time is the last updated time of the Exchange defined Portfolios in the LDB portfolio file.

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 9 MS_SYSTEM_INFO_DATA

Structure Name	MS_SYSTEM_INFO_DATA		
Packet Length	106 bytes		
Transaction Code	SYSTEM_INFORMATION_OUT(1601)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to* *Message Header in Chapter 2)*	STRUCT	40	0
ST_MARKET_STATUS	STRUCT	8	40
ST_EX_MARKET_STATUS	STRUCT	8	48
ST_PL_MARKET_STATUS	STRUCT	8	56
UpdatePortfolio	CHAR	1	64
MarketIndex	LONG	4	65
DefaultSettlementPeriod (Normal)	SHORT	2	69
DefaultSettlementPeriod (Spot)	SHORT	2	71
DefaultSettlementPeriod (Auction)	SHORT	2	73
CompetitorPeriod	SHORT	2	75

Structure Name	MS_SYSTEM_INFO_DATA		
Packet Length	106 bytes		
Transaction Code	SYSTEM_INFORMATION_OUT(1601)		
Field Name	Data Type	Size in Byte	Offset
SolicitorPeriod	SHORT	2	77
WarningPercent	SHORT	2	79
VolumeFreezePercent	SHORT	2	81
SnapQuoteTime	SHORT	2	83
Reserved	CHAR	2	85
BoardLotQuantity	LONG	4	87
TickSize	LONG	4	91
MaximumGtcDays	SHORT	2	95
ST_STOCK_ELIGIBLE_INDICATORS	STRUCT	2	97
DisclosedQuantityPercentAllowed	SHORT	2	99
RiskFreeInterestRate	LONG	4	101

Table 10 ST_MARKET_STATUS

Structure Name	ST_MARKET_STATUS		
Packet Length	8 bytes		
Field Name	Data Type	Size in Byte	Offset
Normal	SHORT	2	0
Oddlot	SHORT	2	2
Spot	SHORT	2	4
Auction	SHORT	2	6

Table 11 ST_EX_MARKET_STATUS

Structure Name	ST_EX_MARKET_STATUS		
Packet Length	8bytes		
Field Name	Data Type	Size in Byte	Offset
Normal	SHORT	2	0
Oddlot	SHORT	2	2
Spot	SHORT	2	4
Auction	SHORT	2	6

Table 12 ST_PL_MARKET_STATUS

Structure Name	ST_PL_MARKET_STATUS		
Packet Length	8 bytes		
Field Name	Data Type	Size in Byte	Offset
Normal	SHORT	2	0
Oddlot	SHORT	2	2
Spot	SHORT	2	4
Auction	SHORT	2	6

Table 13 ST_STOCK_ELIGIBLE_INDICATORS

Structure Name	ST_STOCK_ELIGIBLE_INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Reserved	BIT	5	0
BooksMerged	BIT	1	0
MinimumFill	BIT	1	0
AON	BIT	1	0
Reserved	Byte	1	1
For Big Endian Machines			
AON	BIT	1	0
MinimumFill	BIT	1	0
BooksMerged	BIT	1	0
Reserved	BIT	5	0
Reserved	Byte	1	1

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header. The transaction code is SYSTEM_INFORMATION_OUT (1601).
AlphaChar	This field contains the number of streams present in the host from which message will be downloaded. Note: This field is present in the Message Header. This is of two bytes. Number of streams will be populated in the first byte of Alphachar.

Field Name	Brief Description
	It should be interpreted as integer value and not as character value. Values will be numeric value 1,2,3, ….,10,11 etc. and can range from 1 to 127
MarketStatus	This field contains any of the following values: • '0' if it is Preopen (for Normal market only) • '1' if it is Open • '2' if it is Closed • '3' if it is Preopen Closed • '4' if it is Postclose 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 security when the market is closed.
UpdatePortfolio	This field contains any of the following: • 'N' if there is no change in portfolio. • 'Y' if there is any change in portfolio after Last Update Portfolio Time in the request. Note: User has to update their LDB portfolio file by sending EXCH_PORTF_IN (1775) Request *(Refer to* *EXCH_PORTF_IN* *in Chapter 3)*.
MarketIndex	This field contains the current market index.
DefaultSettlementPeriod	The default settlement period in various markets is sent in the fields DefaultSettlementPeriod (Normal), DefaultSettlementPeriod (Spot) and DefaultSettlementPeriod (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. *(Refer to Turnover Limit Exceeded or Broker Reactivated in Chapter 9)*
VolumeFreeze Percent	This field contains the volume freeze percent. *(Refer to Turnover Limit Exceeded or Broker Reactivated in Chapter 9)*

Field Name	Brief Description
SnapQuoteTime	This field is 60 seconds currently. SnapQuote is the functionality for quick view of LTP of any contract (without adding the contract on Market Watch of NEAT Front End).
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 number of days after which a Good Till Canceled order will be canceled. Currently this field contains zero.
SecurityEligibility Indicator	This field contains the Minimum Fill or All Or NON flag. If the MF flag is set, orders have the Minimum Fill attribute set. If the AON flag is set orders have the AON attribute set.
DisclosedQuantity PercentAllowed	This field contains the disclosed quantity percentage allowed. The disclosed quantity, if set, should be 100 percent of the total quantity.
RiskFreeInterestRate	This field contains the risk free interest rate.

Update Local Database Download

The list of updated securities and participants can be downloaded in response to this request. Any carried over Good Till Cancel (GTC) or Good Till Date (GTD) orders are also downloaded with this request. The following sections describe structures and fields related to Update Local Database request and response.

Update Local Database Request

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

Table 14 MS_UPDATE_LOCAL_DATABASE

Structure Name	MS_UPDATE_LOCAL_DATABASE		
Packet Length	82 bytes		
Transaction Code	UPDATE_LOCALDB_IN(7300)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to Message Header* in Chapter 2*)*	STRUCT	40	0
LastUpdateSecurityTime	LONG	4	40
LastUpdateParticipantTime	LONG	4	44
LastUpdateInstrumentTime	LONG	4	48
LastUpdateIndexTime	LONG	4	52
RequestForOpenOrders	CHAR	1	56
Reserved	CHAR	1	57
ST_MARKET_STATUS *(Refer to System Information Response in Chapter 3)*	STRUCT	8	58
ST_EX_MARKET_STATUS *(Refer to System Information Response in Chapter 3)*	STRUCT	8	66
ST_PL_MARKET_STATUS *(Refer to System Information Response in Chapter 3)*	STRUCT	8	74

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is UPDATE_LOCALDB_IN (7300).
LastUpdate SecurityTime	This field should contain the time when the security information was last updated for all security information that is downloaded. Further download requests can use the latest time to get updated information on the securities. Setting this time to zero results in download of information of all the securities present at HOST END.
LastUpdate ParticipantTime	This field should contain the time when the participant information was updated for all participant information that is downloaded. Further download requests can use the latest time to get updated information on the participants. Setting this time to zero results in download of information of all the participants present at HOST END.

Field Name	Brief Description
LastUpdate InstrumentTime	This field should contain the time when the Instruments were updated.
LastUpdateIndex Time	This field should contain the time when the index information was updated.
RequestForOpenOrders	This field should be set to 'G' if Good Till Cancellation and Good Till Date orders are to be downloaded; otherwise should be set to 'N'.
MarketStatus	This should contain the market status received in the previous response. The market status fields are accepted as input to verify if the trader has the latest system information. Any of the following are possible: • If the trader has the latest market status information, the update of the information on the securities and the participants from the specified time are downloaded. • If the status information specified is not the latest, the trader is updated on the market status alone. Upon receiving the updated market information, the trader can request for the updated security status, security, or participant information.

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 System Information Response *(Refer System Information Response Chapter 3)* is not the same at the host end or the markets are opening. In this case the market status at the host end is sent back 'wait till markets are open'. The following structure is returned:

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

The following table provides the details of the various fields present in the SYSTEM INFORMATION DATA structure.

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to* *MESSAGE_HEADER structure chapter 2)*.The transaction code sent is PARTIAL_SYSTEM_INFOMRATION (7321.
MarketStatus	This field 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.

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 15 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 to* *Message Header in Chapter 2)*	STRUCT	40	0
Reserved	CHAR	2	40

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to* *MESSAGE_HEADER structure chapter 2)*.The transaction code sent 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 packet size can be of 80 to 512 bytes and the structure is as follows:

MESSAGE_HEADER (*Refer to MESSAGE_HEADER in Chapter 2)*
INNER_MESSAGE_HEADER InnerHeader (*Refer to Inner Message Header in Chapter 2)*
CHAR Data [436]

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to Inner Message Header in Chapter 2)* The transaction code is UPDATE_LOCALDB_DATA (7304).
InnerTransaction Code	The transaction codes sent are: • BCAST_SECURITY_MSTR_CHG (7305). It is determined by NSE-Control whether to send this or not. *(Refer to Change in Security Master in Chapter 9).* • BCAST_SECURITY_STATUS_CHG (7320). 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 9).* • BCAST_PART_MSTR_CHG (7306). 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 in Participant Status in Chapter 9).* • BCAST_INSTR_MSTR_CHG (7324). If there is any change in the instrument master after the time specified by the Last Update Instrument Time, it is downloaded. *(Refer to Change in Instrument Master in Chapter 9).* • BCAST_INDEX_MSTR_CHG (7325). If there is any change in the details of the index after the time specified by the Last Index Update Time, it is downloaded *(Refer to Change in Index Master in Chapter 3).* • BCAST_INDEX_MAP_TABLE (7326). It downloads the names of different indexes. *(Refer to Index Map Table in Chapter 3)* Note All these transaction codes will be sent separately.

Change in Index Master

This structure downloads all the details of all the index which are modified after the last time the data was sent to the user. The structure is as follows:

Table 16 MS_DOWNLOAD_INDEX

Structure Name	MS_DOWNLOAD_INDEX		
Packet Length	450 bytes		
Transaction Code	BCAST_INDEX_MSTR_CHG (7325)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
NoOfRecords	SHORT	2	40
INDEX_DETAILS [17]	STRUCT ARRAY	408	42

Table 17 INDEX_DETAILS

Structure Name	INDEX_DETAILS		
Packet Length	24 bytes		
Field Name	Data Type	Size in Byte	Offset
IndexName	CHAR	15	0
Token	Long	4	15
LastUpdateDateTime	LONG	4	19

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is BCAST_INDEX_MSTR_CHG (7325).
NoOfRecords	This field contains the number of records sent for updation.
IndexName	This field contains the name of the index.
Token	This field contains the token number of the index.
LastUpdateDateTime	This field contains the time when the data has been modified.

Index Map Table

This structure downloads the names of the indices.

Table 18 MS_DOWNLOAD_INDEX_MAP

Structure Name	MS_DOWNLOAD_INDEX_MAP		
Packet Length	452 bytes		
Transaction Code	BCAST_INDEX_MAP_TABLE (7326)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
NoOfRecords	SHORT	2	40
BCAST_INDEX_MAP_DETAILS [10]	STRUCT ARRAY	410	42

Table 19 BCAST_INDEX_MAP_DETAILS

Structure Name	BCAST_INDEX_MAP_DETAILS		
Packet Length	41 bytes		
Field Name	Data Type	Size in Byte	Offset
BcastName	CHAR	26	0
ChangedName	CHAR	10	26
DeleteFlag	CHAR	1	36
LastUpdateDateTime	LONG	4	37

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2)*.The transaction code is BCAST_INDEX_MAP_TABLE (7326).
NoOfRecords	This field contains the number of index names (For example, CNX Nifty, CNX Nifty Junior, etc.) downloaded.
BcastName	This field contains the name of the index. **Note** CNX Nifty, CNX Nifty Junior, etc.

Field Name	Brief Description
ChangedName	The actual name of the index for which the information is going to be broadcast during market hours.
DeleteFlag	This field indicates whether the index name is deleted or not. The values are any of the following: • 'Y' for Yes (Deleted) • 'N' for No (Not deleted)
LastUpdateDateTime	This field contains the time when this data was modified.

Update Local Database Trailer

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

Table 20 UPDATE_LOCAL_DB_TRAILER

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is UPDATE_LOCALDB_TRAILER (7308).

Getting Exchange-defined Portfolio

The user can download the exchange-defined portfolio by sending the portfolio request structure.

Portfolio Request

The structure for portfolio request is as follows:

Table 21 EXCH_PORTFOLIO_REQ

Structure Name	EXCH_PORTFOLIO_REQ		
Packet Length	44 bytes		
Transaction Code	EXCH_PORTF_IN (1775)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
LastUpdateDtTime	LONG	4	40

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is EXCH_PORTF_IN (1775).
LastUpdateDtTime	This field should contain the time when the portfolio information was last updated for all portfolio information that is downloaded. Further download requests can use the latest time to get updated information on the portfolio. Setting this time to zero results in complete download.

Portfolio Response

This structure is sent in response to the portfolio request. The structure is as follows:

Table 22 EXCH_PORTFOLIO_RESP

Structure Name	EXCH_PORTFOLIO_RESP		
Packet Length	329 bytes		
Transaction Code	EXCH_PORTF_OUT (1776)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
NoOf Records	SHORT	2	40
MoreRecords	CHAR	1	42
Filler	CHAR	1	43
PORTFOLIO_DATA [15]	STRUCT ARRAY	19	44

Table 23 PORTFOLIO_DATA

Structure Name	PORTFOLIO_DATA		
Packet Length	19 bytes		
Field Name	Data Type	Size in Byte	Offset
Portfolio	CHAR	10	0
Token	LONG	4	10
LastUpdateDtTime	LONG	4	14
DeleteFlag	CHAR	1	18

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is EXCH_PORTF_OUT (1776).
Portfolio	This field contains the name of the portfolio.
Token	This field contains the token number of the security in the portfolio.
NoOfRecords	This field contains the number of records. Depending upon this number of records will be filled up in subsequent PORTFOLIO_DATA structure.
MoreRecords	This field is set to 'Y' if there are more records to be sent in the next pocket. If it is the last pocket, it is set to 'N'.
DeleteFlag	This field is set to 'Y' or 'N' to indicate whether the portfolio is deleted or not, where: • 'Y' means 'deleted'. • 'N' means 'not deleted'.

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. The response consists of Header and Trailer to indicate the beginning and end of download and is similar to Update Local Database Download.

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 as follows:

Table 24 MS_MESSAGE_DOWNLOAD

Structure Name	MS_MESSAGE_DOWNLOAD		
Packet Length	48 bytes		
Transaction Code	DOWNLOAD_REQUEST (7000)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to* *MESSAGE_HEADER* *in Chapter 2)*	STRUCT	40	0
SequenceNumber	Double	8	40

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to* *MESSAGE_HEADER* *structure chapter 2).*The transaction code is DOWNLOAD_REQUEST (7000).
Alpha_Char (Header)	This contains the stream number of the host to which it has to send the DOWNLOAD_REQUEST. Machine / Stream no. should be send in the first byte (AlphaChar[0]) of this field and should be of type integer value and not as character value. Values to be sent should be numeric value 1,2,3, ...,10,11 etc. and can range from 1 to 127
SequenceNumber	This field should contain the time when 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 last response message.

Message Download Response

The download response comprises of a header, data and a trailer. Each trader specific broadcast message will be sent separately.

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 Message Header in Chapter 2)*

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*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 packet size can be of 80 to 600 bytes and the structure is as follows:

MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*
The following table provides the details of the various fields present in the MESSAGE_HEADER structure.

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).* The transaction code is MESSAGE_RECORD (7021).
InnerData	Various transaction codes are received. They are as follows: 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 7.*

	• Order Entry, Modification and Cancellation responses - *Refer to Order Management, Chapter 4.* • Trade Modification and Cancellation responses - *Refer to Trade Management, Chapter 4.* • Trade Confirmation and Stop Loss Trigger - *Refer to Unsolicited Messages, Chapter 7.* 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 Messages in Chapter 9.* Contingency broadcast message. *Refer to Exception Handling in 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 Message Header in Chapter 2)*

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*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. It is a request to break the virtual circuit between the trading system host and the front end.

The structure sent is:

MS_SIGNOFF struct MESSAGE_HEADER

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to* *MESSAGE_HEADER structure chapter 2).*The transaction code is SIGN_OFF_REQUEST_IN (2320).

Logoff Confirmation Response

When the user logs on again, the user receives a packet with the details of the last user log off.

The structure sent is:

Table 25 SIGNOFF_OUT

Structure Name	SIGNOFF_OUT		
Packet Length	40 bytes		
Transaction Code	SIGN_OFF_REQUEST_OUT (2321)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to* *MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
~~UserId~~	~~LONG~~	~~4~~	~~40~~
~~Reserved~~	~~CHAR~~	~~145~~	~~44~~

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to* *MESSAGE_HEADER structure chapter 2).*The transaction code is SIGN_OFF_REQUEST_OUT (2321).

Chapter 4 Order and Trade Management

This chapter describes structures and fields for entering new orders, modifying existing orders, and canceling outstanding orders. The trader can begin entering the orders once logged on to the trading system and when the market is in pre-open or open state. The sections covered in this chapter are:

- Order Entry
- Order Modification
- Order Cancellation
- Trade Modification
- Trade Cancellation

Order Entry

Order entry allows the trader to place orders in the market. The system accepts the orders from the users and matches the orders with the orders in the order books maintained at HOST END to order matching. 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

The types of order are as follows:

- **Regular Lot:** Regular Lot Orders are orders in the normal market that have none of the following terms attached: All or None (AON), Minimum Fill (MF) 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 are attached: DQ, All or None, Minimum Fill and Trigger Price.

- **Special Terms:** Special Terms Orders are orders in the normal market which have special attributes attached to it. They must have MF or AON.
- **Negotiated Trade Orders:** Negotiated Trade Orders are regular lot orders with the Counter Party ID.
- **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.

  Market If Touched: Market If Touched Orders are orders in normal market with Trigger Price specified. They may have the Minimum Fill or the AON attribute specified. Market if touched orders are almost identical to stop orders, except that they are used when the market is currently trading on the opposite side of the order price.

Order Terms

The following terms and conditions are used during order entry and 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. The price is to be multiplied by 100.
- **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.
- **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.

- **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 if any of the following conditions is true:

- 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 that 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 unable to log into the trading system.
- 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 more 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 are more than specified days.
- GTC, GTD orders are not allowed.
- Negotiated Trade orders have GTC/GTD/IOC attribute.
- Spot orders have GTC/GTD.
- IOC and Disclosed Quantity combination is present.
- For PRO order Account Number is Broker ID or any other ID.
- For CLI order Account Number is Broker ID.
- Order attributes are not entered properly for various book types.
- 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 26 MS_OE_REQUEST

Structure Name	MS_OE_REQUEST		
Packet Length	316 bytes		
Transaction Code	BOARD_LOT_IN (2000)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to* *MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
ParticipantType	CHAR	1	40
Reserved	CHAR	1	41

Structure Name	MS_OE_REQUEST		
Packet Length	316 bytes		
Transaction Code	BOARD_LOT_IN (2000)		
Field Name	Data Type	Size in Byte	Offset
CompetitorPeriod	SHORT	2	42
SolicitorPeriod	SHORT	2	44
Modified/CancelledBy	CHAR	1	46
Reserved	CHAR	1	47
ReasonCode	SHORT	2	48
Reserved	CHAR	4	50
TokenNo	LONG	4	54
CONTRACT_DESC	STRUCT	28	58
CounterPartyBrokerId	CHAR	5	86
Reserved	CHAR	1	91
Reserved	CHAR	2	92
CloseoutFlag	CHAR	1	94
Reserved	CHAR	1	95
OrderType	SHORT	2	96
OrderNumber	DOUBLE	8	98
AccountNumber	CHAR	10	106
BookType	SHORT	2	116
Buy/SellIndicator	SHORT	2	118
DisclosedVolume	LONG	4	120
DisclosedVolumeRemaining	LONG	4	124
TotalVolumeRemaining	LONG	4	128
Volume	LONG	4	132
VolumeFilledToday	LONG	4	136
Price	LONG	4	140
TriggerPrice	LONG	4	144
GoodTillDate	LONG	4	148
EntryDateTime	LONG	4	152
MinimumFill / AONVolume	LONG	4	156
LastModified	LONG	4	160
ST_ORDER_FLAGS	STRUCT	2	164
BranchId	SHORT	2	166
TraderId	LONG	4	168
BrokerId	CHAR	5	172
cOrdFiller	CHAR	24	177

Structure Name	MS_OE_REQUEST		
Packet Length	316 bytes		
Transaction Code	BOARD_LOT_IN (2000)		
Field Name	Data Type	Size in Byte	Offset
Open/Close	CHAR	1	201
Settlor	CHAR	12	202
Pro / ClientIndicator	SHORT	2	214
SettlementPeriod	SHORT	2	216
ADDITIONAL_ORDER_FLAGS	STRUCT	1	218
Reserved	CHAR	1	219
Filler1	USHORT	1 (bit)	220
Filler2	USHORT	1 (bit)	220
Filler3	USHORT	1 (bit)	220
Filler4	USHORT	1 (bit)	220
Filler5	USHORT	1 (bit)	220
Filler6	USHORT	1 (bit)	220
Filler7	USHORT	1 (bit)	220
Filler8	USHORT	1 (bit)	220
Filler9	USHORT	1 (bit)	221
Filler10	USHORT	1 (bit)	221
Filler11	USHORT	1 (bit)	221
Filler12	USHORT	1 (bit)	221
Filler13	USHORT	1 (bit)	221
Filler14	USHORT	1 (bit)	221
Filler15	USHORT	1 (bit)	221
Filler16	USHORT	1 (bit)	221
Filler17	CHAR	1	222
Filler18	CHAR	1	223
NnfField	DOUBLE	8	224
MktReplay	LONG LONG	8	232
PAN	CHAR	10	240
Algo ID	LONG	4	250
Reserved	SHORT	2	254
LastActivityReference	LONG LONG	8	256
Reserved	CHAR	52	264

Table 27 CONTRACT_DESC

Structure Name	CONTRACT_DESC

Packet Length	28 bytes		
Field Name	Data Type	Size in Byte	Offset
InstrumentName	CHAR	6	0
Symbol	CHAR	10	6
ExpiryDate	LONG	4	16
StrikePrice	LONG	4	20
OptionType	CHAR	2	24
CALevel	SHORT	2	26

Table 28 ST_ORDER_FLAGS

Structure Name	ST_ORDER_FLAGS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
AON	BIT	1	0
IOC	BIT	1	0
GTC	BIT	1	0
Day	BIT	1	0
MIT	BIT	1	0
SL	BIT	1	0
Market	BIT	1	0
ATO	BIT	1	0
Reserved	BIT	2	1
PreOpen	BIT	1	1
Frozen	BIT	1	1
Modified	BIT	1	1
Traded	BIT	1	1
MatchedInd	BIT	1	1
MF	BIT	1	1
For Big Endian Machines			
ATO	BIT	1	0
Market	BIT	1	0
SL	BIT	1	0
MIT	BIT	1	0
Day	BIT	1	0
GTC	BIT	1	0
IOC	BIT	1	0

AON	BIT	1	0
MF	BIT	1	1
MatchedInd	BIT	1	1
Traded	BIT	1	1
Modified	BIT	1	1
Frozen	BIT	1	1
PreOpen	BIT	1	1
Reserved	BIT	2	1

Table 29 ADDITIONAL_ORDER_FLAGS

Structure Name	ADDITIONAL_ORDER_FLAGS		
Packet Length	1 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
BOC	BIT	1	0
COL	BIT	1	0
Reserved	BIT	1	0
Reserved	BIT	1	0
STPC	BIT	1	0
Reserved	BIT	3	0
For Big Endian Machines			
Reserved	BIT	3	0
STPC	BIT	1	0
Reserved	BIT	1	0
Reserved	BIT	1	0
COL	BIT	1	0
BOC	BIT	1	0

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is BOARD_LOT_IN (2000).

Field Name	Brief Description
ParticipantType	Since only exchange can initiate the auction, this field should not be set to 'I' initiator. This field 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.
Modified / CancelledBy	This field should denote who has modified or cancelled a particular order. It should take one of the following values: • 'T' for Trader • 'B' for Branch Manager • 'M' for Corporate Manager • 'C' for Exchange During order entry, this field should be blank.
ReasonCode	This field contains the reason code for a particular order request rejection or order freeze. This, along with the error code, has the details regarding the error. *Refer to Reason Codes in Appendix.* During order entry, this field should be set to zero.
TokenNumber	This is the Token Number of the contract on which order is to be placed. This field should contain a valid token number or '-1'. If the token number is set to '-1', the validations will be done only on contract descriptor. If the valid token number is sent, the validation will be done on token number as well as contract descriptor.
SecurityInformation (CONTRACT DESCRIPTOR)	This structure contains the following fields: Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA Level of the contract. This is mandatory and should be filled while sending the order entry request. CA Level should be set to zero.
CounterParty BrokerId	This field specifies the Counter Party Broker code for the Negotiated Trade Order. This field is valid only for Negotiated Trade Orders. For other books, this field should be set to blank.
CloseoutFlag	This field should be set to blank.

Field Name	Brief Description
OrderType	This field should be set to blank.
OrderNumber	This field must be sent as blank for the order entry request.
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 blank.
BookType	This field should contain the type of order. *Refer to Book Types* in *Appendix*.
Buy / SellIndicator	This field should specify whether the order is a buy or sell. The field should take one of the following values: • '1' for Buy order • '2' for Sell order
DisclosedVolume	This field should contain 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.
DisclosedVolume Remaining	This is the disclosed volume remaining from the original disclosed volume after trade(s). This is an output field. While sending order entry request to the host it should be same as disclosed volume.
TotalVolume Remaining	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry, this field should be set to Volume. For every response, the trading system will return this value.
Volume	This field should contain the order quantity. The quantity should always be in multiples of Regular Lot except for Odd Lot orders and it should 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.
VolumeFilled Today	This field specifies the total quantity traded in a day. It should be set to '0' (zero) while sending order entry request to the host.

Field Name	Brief Description
Price	This field should contain the price at which the order is placed. The price must be a multiple of the tick size. To enter a Market order, the price should be set to zero. 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. Market attribute is not allowed for Negotiated Orders. This should be multiplied by 100 before sending to 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 '0'. If it was a priced order the order gets confirmed at that price.
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 should be multiplied by 100 before sending to the trading system.
GoodTillDate	This field should contain the number of days for a GTD order. This field can be set in two ways. To specify an absolute date, this field should be set to that date in number of seconds since midnight of January 1, 1980. To specify days, this field should be set to the number of days. This can take values from two to the maximum days specified for GTC orders only. If this field is non-zero, the GTC flag must be off.
EntryDateTime	This field contains the time when the order first entered the trading system. This field should be sent as zero for the order entry request.
MinimumFill Volume	This field specifies the minimum fill quantity when the minimum fill attribute is set for an order. It should not be greater than either the volume of the order or the disclosed quantity and must be a multiple of the regular lot.
LastModified Time	In the case of order entry, this field will be same as Entry Date Time. After the order is modified it contains the time

Field Name	Brief Description
	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.
OrderTerms	This field should specify the attributes of an order. *Refer to Order Terms table in Chapter 4.*
BranchId	This field should contain the branch number to which the broker belongs. Note Branch ID can be of 3 digits.
TraderId	This field should contain the ID of the user. This field accepts only numbers.
BrokerId	This field should contain the trading member ID.
Open / Close	Open / Close order indicator. This field should contain one of the following values. • 'O' for Open • 'C' for Close
Settlor	This field should specify 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. So, this field should be set to blank for a pro order (broker's own order).
Pro-ClientOrder	This field should contain one of the following values to specify whether the order is entered on behalf of a broker or a trader. • '1' represents the client's order. • '2' represents a broker's order.
SettlementPeriod	This field should contain the number of days in a settlement cycle. Currently it is 10 days.
ADDITIONAL_ORDER_FLAGS	Refer to *Additional Order Flags* and *Order Terms Attributes* tables in Chapter 4 for the relevant description. For reserved bit kindly set the values with 0
NNFField	This field should contain a 15 digit a unique identifier for various products deployed as per Exchange

Field Name	Brief Description
	circular download ref no. 16519 dated December 14, 2010 and as updated from time to time
MktReplay	This field contains the time when the order enters the trading system. It is stamped at the host end. This field should be set to zero while sending message 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	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.
LastActivityReference	In case of order entry response, this field will contain a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified. This field should be set to zero for the order entry request.

Order Terms Attributes

Order Term	Is Set To	Attribute Represented
AON	1	All Or None
IOC	1	Immediate Or Cancel
GTC	1	Good Till Cancel
Day	1	Day (This is the default attribute)
MIT	1	Market If Touched
SL	1	Stop Loss
Market	0	Market order
ATO	1	Market order in Preopen
Frozen	1	The order has gone for a freeze

Order Term	Is Set To	Attribute Represented
PreOpen	1	If set to 1, represents the order is a Preopen session order and If set to 0, represents Normal Market order
Modified	1	The order has to be modified
Traded	1	The order has been traded partially or fully
MatchedInd	1	NT order has found a matching order
MF	1	Minimum Fill
COL	1	Cancel on Logoff
STPC	0	Cancel order resulting in self trade as per default action by the exchange
STPC	1	Cancel active order resulting in self trade

Note: - STPC bit can be set for RL, 2L, 3L, Spread, SL transcodes. STPC bit in the modification transcodes should be same as set in the original order else the modification request will be rejected. In case of triggered stop loss order, bit selected during order entry will be considered.

Rules of Order Entry (when broker is in Closeout Status)

Only the following orders will be considered valid when a broker is in Closeout status:

1. Orders with the following:
   - Market Type :- NORMAL,
   - Book Type :- REGULAR (RL),

- Order Type : IMMEDIATE or CANCEL

2. Participant order entry will not be allowed if the broker is in Closeout status.
3. Value of CloseoutFlag in MS_OE_REQUEST for different transaction codes will be as follows:
   - BOARD_LOT_IN (2000): The value of the CloseoutFlag must be sent blank
   - ORDER_CONFIRMATION(2073) and ORDER_CANCEL_CONFIRMATION(2075): The CloseoutFlag will contain the value 'C', indicating that entered order is close order if the broker is in close out state
   - ORDER_ERROR (2231): CloseoutFlag may contain the value 'C', indicating that entered order is close order if the broker is in close out state.
   - For all other transcodes using the MS_OE_REQUEST structure CloseoutFlag will be ignored

Order Entry Response

The primary response is the Order Requested message which is stopped currently. The secondary response can be Order Confirmation, Order Freeze, Order Error or one of the general error responses. 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 in the Error Code.

Note: Order Requested Message (2001) is stopped to reduce the packet sent from the host end.

Market Order Response

This response is sent back to the trader when a Market order is requested while the market is in Open state. It is sent after the Order Requested response. The 'Market' flag in Order Terms is set to '0' and is priced at the prevailing price at the trading system. The message sent is:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to [MESSAGE_HEADER](#) structure chapter 2).*The transaction code is PRICE_CONFIRMATION (2012).
EntryDateTime	This field contains the date and time when the order entered the system.
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

This response does not imply that the order is confirmed, and is followed by the Order Confirmation response.

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:

MS_OE_REQUEST *(Refer to [Order Entry Request](#) in Chapter 4)*

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to [MESSAGE_HEADER](#) structure chapter 2).*The transaction code is ORDER_CONFIRMATION_OUT (2073).

EntryDateTime	This field contains the date and time when the order entered the system.
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 '0'. If it was a priced order the order gets confirmed at that price.
OrderTerms	The flags are set as discussed in *Order Entry Request in Chapter 4*

Note

The reason code in the structure can be used to differentiate orders that got freeze approval from orders that got normal confirmation.

- Reason code '17' or '18' denotes freeze approved/rejected.
- Reason code '0' denotes normal confirmation.

Order Freeze Response

Order freeze response is generated when the order placed by the trader or a modified order is awaiting approval from the exchange. 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:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is FREEZE_TO_CONTROL (2170).
OrdersTerms	*Refer to Order Entry Request in Chapter 4.*

Note:

The reason code in the structure can be used to differentiate price freeze and quantity freeze.

Reason code '18' denotes Quantity freeze and reason code '17' denotes Price freeze.

Order Error Response

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

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

The following table provides the details of the various fields present in the OREDR ENTRY REQUEST structure.

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is ORDER_ERROR_OUT (2231).
ErrorCode	This contains the error number. *Refer to List of Error Codes in Appendix.*

Order Modification

Order Modification enables the trader to modify unmatched orders.

Rules of Order Modification

The following modifications are not allowed:

- Buy to sell or vice versa.
- Modification of contract.
- Modifying Frozen orders.

- Branch Manager modifying Corporate Manager’s orders.
- Dealer modifying BM’s orders.
- DL modifying CM’s orders.
- Modifying non existing order.
- Inquiry user trying to modify orders.
- Modifying an order in such a way that it results in a branch order value to be exceeded.
- Modifying deactivated broker’s orders.
- Changing of original data.
- Modifying Open to Close and vice versa.
- Modifying existing order to stop loss limit order such that difference between trigger price and limit price is greater than permissible range.

Note: RL/ Special Terms /Stop Loss book types can be modified / switched among themselves only.

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.

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is ORDER_MOD_IN (2040).
Modified / CancelledBy	This field denotes who 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

Order_Flags	Preopen - This bit will be set to 1 for pre-open order modification response during pre-open market session. It will be set to 0 during Normal market session for normal orders and for the orders carried forward from pre-open session .
OrderNumber	Order Number is the identity of the order to be modified.
EntryDateTime	This field contains the date and time when the order entered the trading system. This is available in Order Confirmation/ Order Modification Confirmation response.
LastModified Time	In the case of order entry, this field will be same as Entry Date Time. After the order is modified it contains the time when the Order was last modified. It is the time in seconds from midnight of January 1, 1980. In case of Order Modification Request this field should contain the time when the Order was last modified
LastActivityReference	In Order modification request, this field should contain LastActivityReference value received in response of the last activity done on that order. Last activity could be order entry, order modification or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.
Note: The other fields of modification request are the same as Order entry request.	

Note: Order Modification Response (2041) is stopped to reduce the packet sent from the host end.

Price Modification Request

To modify Price of an existing regular book order (Book Type 1), following optimized structure can be used. This request is in addition to existing Modification transcode mentioned in the document. Volume will not be modified through this transcode. To modify any other properties of the order, please refer to regular Order Modification Section in Futures and Options NNF protocol.

Table 30 PRICE_ MOD

Structure Name	PRICE_ MOD		
Packet Length	106 bytes		
Transaction Code	PRICE_MOD_IN (2013) PRICE_MOD_ACK_IN (20406)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER(*Refer to [MESSAGE_HEADER](#) in Chapter 2)*	STRUCT	40	0
TokenNo	LONG	4	40
Trader ID	LONG	4	44
OrderNumber	DOUBLE	8	48
BuySell	SHORT	2	56
Price	LONG	4	58
Volume	LONG	4	62
LastModified	LONG	4	66
Reference	CHAR	4	70
LastActivityReference	LONG LONG	8	74
Reserved	CHAR	24	82

Field Name	Brief Description
TransactionCode	This field is the part of Message Header (Refer to *MESSAGE_HEADER* structure chapter 2).The transaction code is PRICE_MOD_IN (2013) PRICE_MOD_ACK_IN (20406).
TokenNo	This is the Token Number of the contract for which this order was originally placed.
Trader ID	Connected user’s Trader ID
OrderNumber	Original Order Number to be modified
BuySell	This field should contain one of the following values to specify whether the order is a buy or sell order. ‘1’ denotes Buy order ‘2’ denotes Sell order
Price	New price that will overwrite the current Limit price. If it is sent as 0 (zero) then order will be modified as Market Priced Order.
Volume	Latest image of volume should be populated in this field.
LastModified	Value of the Last modified time stamp as received on last transaction response.

Reference	The front-end may use this field at their discretion.
LastActivityReference	In Order modification request, this field should contain LastActivityReference value received in response of the last activity done on that order. Last activity could be order entry, order modification or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Note: - STPC (*Refer to Order Terms Attributes in Chapter 4*) bit set at order entry shall be considered in case of incoming 2013 transcode.

Order Modification Confirmation Response

Successful modification of the order results in Order Modification Confirmation. When the order modification is confirmed, the modified order time is filled and sent back. On modification, the order can result in a freeze. If the freeze is approved then 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.

The structure sent is as follows:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is ORDER_MOD_CONFIRM_OUT (2074).
EntryDateTime	The order time (user modified) will be populated in this field.
Modified / CancelledBy	This field denotes who 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
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	In case of order modification response, this field will contain a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Order Modification Error Response

This is sent when order modification request is rejected. The reason for rejection will be given by the Error Code in the header. The message sent is as follows:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is ORDER_MOD_REJ_OUT (2042).
Order_Flags	Preopen - This bit will be set to 1 for pre-open order modification response during pre-open market session. It will be set to 0 during Normal market session for normal orders and for the orders carried forward from pre-open session.

Effect of Modifying the Terms of an Order (on Price/Time Priority)

Field Name	Can Change	Comments
Buy/Sell	No	NA
Order Type	Yes	NA

Contract Descriptor	No	NA
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	Yes	NA
Account No.	Yes	NA
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	NA
DQ	NA	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.
Participant	Yes	

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.

Rules for order cancellation

The rules for order cancellation are as follows:

- 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 an order while the broker's status is deactivated.

Order Cancellation Request

The format of the message is as follows:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2)*.The transaction code is ORDER_CANCEL_IN (2070).
OrderNumber	This field should contain the order number which identifies the order to be cancelled.
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	In Order Cancellation request, this field should contain LastActivityReference value received in response of the last activity done on that order. Last activity could be order entry, order modification or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Note: Order Cancellation Response (2071) is stopped to reduce the packet sent from the host end.

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 same transcode will also be sent in response if normal IOC order results in partial cancellation or 2L/3L Order results in partial cancellation. The message sent is as follows:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is ORDER_CANCEL_CONFIRM_OUT (2075).
Modified / CancelledBy	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.
Order_Flags	This bit will be set to 1 for Pre-open order cancellation response and Pre-open carried forward order cancellation response. It will be set to 0 for Normal Market Open order cancellation response
LastModifiedTime	This will be the current cancellation time. (It will be same as LogTime)

Batch Order Cancellation

The following table provides the details of the various fields present in the MS_OE_REQUEST structure, for batch order cancellation.

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is BATCH_ORDER_CANCEL (9002).
LastModifiedTime	This will be the current cancellation time. (It will be same as LogTime)

Order Cancellation Error Response

The order cancellation error is sent when the cancellation 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:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2)*.The transaction code is ORDER_CXL_REJ_OUT (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. 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 contract by specifying contract information in request packet.

Kill Switch Request

The format of the message is as follows:

MS_OE_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 all orders should be cancelled.
TokenNumber	For cancellation of all orders, token number should be set to '-1'.

Field Name	Brief Description
	For cancellation of orders on particular contract, valid token number of the contract is to be sent.
SecurityInformation (CONTRACT DESCRIPTOR)	For cancellation of all orders on particular contract, this field is mandatory. This structure contains the following fields: Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA Level of the contract. CA Level should be set to zero.

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:

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

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

Trade Modification

Trade Modification is a facility provided by NSE to allow users to change client account number of executed trades.

Trade modification functionality will be available to the member irrespective of trade's Give up approval/rejection status.

Trade Modification Request

The format of the message is as follows:

Table 31 MS_TRADE_INQ_DATA

Structure Name	MS_TRADE_INQ_DATA		
Packet Length	234 bytes		
Transaction Code	TRADE_MOD_IN (5445)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to* *MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
TokenNo	LONG	4	40
CONTRACT_DESC *(Refer to Order Entry Request in Chapter 4)*	STRUCT	28	44
FillNumber	LONG	4	72
FillQuantity	LONG	4	76
FillPrice	LONG	4	80
MktType	CHAR	1	84
BuyOpenClose	CHAR	1	85
Reserved	LONG	4	86
BuyBrokerId	CHAR	5	90
SellBrokerId	CHAR	5	95
TraderId	LONG	4	100
RequestedBy	CHAR	1	104
SellOpenClose	CHAR	1	105
BuyAccountNumber	CHAR	10	106
SellAccountNumber	CHAR	10	116
Reserved	CHAR	24	126
ReservedFiller	CHAR	2	150
Reserved	CHAR	2	152
BuyPAN	CHAR	10	154
SellPAN	CHAR	10	164
Reserved	CHAR	60	174

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is TRADE_MOD_IN (5445).
TokenNumber	This field should contain the token number of the contract.
FillNumber	This field should contain the trade number of the trade to be modified.

Field Name	Brief Description
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 —Normal or Odd Lot or Spot or Auction. • ‘1‘for Normal Market. • ‘2’for Odd Lot Market • ‘3’for Spot Market • ‘4’for Auction Market
BuyOpenClose	This field should be set to ‘O‘for Open or ‘O’for Close for Buy trade.
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.
SellOpenClose	This field should contain the Open / Close indicator for the Sell trade.
BuyAccount Number	This field should contain the Account Number of the trade on Buy side.
SellAccount Number	This field should contain the Account Number of the trade on Sell side.
ReservedFiller	This filed is reserved for future use and any value in this field will be ignored.
BuyPAN	This field shall contain the PAN (Permanent Account Number/PAN_EXEMPT). This field shall be mandatory for all orders (client/participant/PRO orders).

Field Name	Brief Description
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 Error

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

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*The transaction code is TRADE_ERROR (2223).
ErrorCode	This field contains the error code. *Refer to List of Error Codes in Appendix.*

Trade Cancellation

To cancel a trade, both parties of the trade must request for trade cancellation. The process is as follows:

1. As soon as the request reaches the trading system, a 'requested message' is sent.
2. If any error was encountered in the entered data then Trade Error message is sent. Otherwise it goes as an alert to the NSE control.
3. The counter party to the trade is notified of the trade cancellation request *(Refer to Unsolicited Messages, Chapter 7).*
4. When both the parties of the trade have asked for trade cancellation, it may be approved or rejected by the Exchange *(Refer to Unsolicited Messages, Chapter 7).*

Trade Cancellation Request

The format of the message is as follows:

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

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

Field Name	Brief Description
TransactionCode	This field is the part of Message Header *(Refer to MESSAGE_HEADER structure chapter 2).*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

The format of the message is as follow

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

This is an acknowledgement signifying that the request has reached the trading system. The field details of the structure are as follows:

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:

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

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

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_ERROR (2223).
ErrorCode	This field contains the error code. Note: Refer to *List of Error Codes* in Appendix.

Chapter 5 Spread Order and Trade Management

This chapter describes structures and fields for entering new spread orders. Spread order is a combination of two normal orders on two contracts with same symbol and different expiry dates. The trader can begin entering the spread orders once the trader has logged onto the trading system and only when the market is in open state. The sections covered in this chapter are:

- Spread Order Entry

Spread Order Entry

Spread 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 immediately with the orders in the books. In case of IOC order Spread IOC orders are not allowed. If no IOC flag is mentioned then the partial traded orders or orders which are not traded are written to spread order book.

Note:

- By default all the spread orders are day orders.
- Currently Spread IOC orders are not allowed.

The other conditions not allowed are:

- Disclosed (Disclosed Quantity)
- Good Till Days (GTD)
- Good Till Cancelled (GTC)
- IOC

Order Types Allowed for Spread Order Entry

Spread order entry allows only the following order types:

- **Regular Lot:** Only Spread day orders are allowed and spread IOC orders are not allowed.
- **Special Terms:** In this case, only orders with all or None (AON) attribute are allowed. Normal Day orders and IOC orders are also allowed. AON allows the

broker to ensure that the entire order is traded or none at all. This might result in multiple trades or single trade.

The following terms and conditions **cannot** be used during spread ST order entry:

- Trigger Price (TP)
- Minimum Fill (MF)

Technical Changes in Spread

The following types of technical changes are considered:

Spread Orders and Trades

Spread day orders will be allowed only on future contracts. Valid spread combinations will be pre-defined in the Spread Combination Contract file.

Spread day orders on eligible spread combinations with price difference within the operating range, will be allowed.

Since day orders are allowed, following functionalities will be applicable for spread day orders:

- Spread order modification –
  - Change in price difference and quantity will be allowed.
  - Modification of spread day order to IOC will not be allowed.
  - All other rules for normal order modification remain same for spread order modifications.
- Spread order cancellation

Order Cancellation by System

Broker suspension – When a broker is suspended then all the outstanding spread orders of the firm will also be cancelled by the system. Respective users will get spread order cancellation message.

Batch Order Cancellation - At the end of the day, all the outstanding spread day orders will be cancelled by the system. Respective users will get spread order cancellation message.

The order cancellation due to broker suspension or batch order cancellation will be sent with new transcode BATCH_SPREAD_CXL_OUT (9004). For this transcode existing structure MS_SPD_OE_REQUEST will be used.

New Master File for Spread Combination

A new master file is introduced to maintain the valid spread combinations. The spread combination consists of two contracts. The combination will be updated on daily basis. The file will be made available every day for uploading in the front end. Structure of this file is provided in subsequent section (Refer to Spread Combination File in Chapter 5).

Broadcast for Spread Combination Master Update

Any intraday change in the spread combination master will be available as broadcast. A new structure BCAST_SPD_MSTR_CHG with transcode 7309 is defined. Structure of the new transcode 7309 is given in subsequent section *(Refer to MS_SPD_UPDATE_INFO in Chapter 5)*.

Existing Security Master Update Broadcast

The existing security master update broadcast (BCAST_SECURITY_MSTR_CHG – transcode 7305) should be used to update the information in Spread combinations for relevant contracts. The structure of the existing packet will remain unchanged.

Broadcast for Spread

Spread MBP (market by price) will be reflecting spread activities. These packets will be compressed. The existing Spread MBP (MS_SPD_MKT_INFO) structure will be changed to send broadcast for spread MBP, for price difference statistics. The changes in the structure are mentioned in *MS_SPD_MKT_INFO Chapter 9*.

Rules of Spread Order Entry

Order entry is **not allowed** if any of the following conditions is true:

- Oder is of GTC or GTD order type.
- 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 that 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 unable to log into the trading system.
- User is an inquiry user.
- User does not exist in the trading system.
- Participant is suspended.
- Participant does not exist in the trading system.
- Order price is beyond day’s minimum maximum range.
- Quantity is more than issued capital.
- Quantity is not equal to multiples of regular lot.
- Limit Price is not a multiple of Tick size.
- IOC and Disclosed Quantity combination is present.
- For PRO order, client is other than broker.
- For CLI order, Account Number is Broker ID.
- Order attributes are not entered properly for various book types.
- Both contracts of spread order having same expiry date.

Order Entry Request

The format of the order entry request is as follows:

Table 32 MS_SPD_OE_REQUEST

Structure Name	MS_SPD_OE_REQUEST		
Packet Length	480 bytes		
Transaction Code	SP_BOARD_LOT_IN (2100) SP_BOARD_LOT_ACK_IN (20408)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
ParticipantType1	CHAR	1	40
Filler1	CHAR	1	41
CompetitorPeriod1	SHORT	2	42
SolicitorPeriod1	SHORT	2	44
ModCxlBy1	CHAR	1	46
Filler9	CHAR	1	47
ReasonCode1	SHORT	2	48
StartAlpha1	CHAR	2	50
EndAlpha1	CHAR	2	52
Token1	LONG	4	54
CONTRACT_DESC *(Refer to Order Entry Request in Chapter 4)*	STRUCT	28	58
OpBrokerId1	CHAR	5	86
Fillerx1	CHAR	1	91
FillerOptions1	CHAR	3	92
Fillery1	CHAR	1	95
OrderType1	SHORT	2	96
OrderNumber1	DOUBLE	8	98
AccountNumber1	CHAR	10	106
BookType1	SHORT	2	116
BuySell1	SHORT	2	118
DisclosedVol1	LONG	4	120
DisclosedVolRemaining1	LONG	4	124
TotalVolRemaining1	LONG	4	128
Volume1	LONG	4	132
VolumeFilledToday1	LONG	4	136
Price1	LONG	4	140
TriggerPrice1	LONG	4	144
GoodTillDate1	LONG	4	148
EntryDateTime1	LONG	4	152
MinFillAon1	LONG	4	156
LastModified1	LONG	4	160

Structure Name	MS_SPD_OE_REQUEST		
Packet Length	480 bytes		
Transaction Code	SP_BOARD_LOT_IN (2100) SP_BOARD_LOT_ACK_IN (20408)		
Field Name	Data Type	Size in Byte	Offset
ST_ORDER_FLAGS *(Refer to Order Entry Request in Chapter 4)*	STRUCT	2	164
BranchId1	SHORT	2	166
TraderId1	LONG	4	168
BrokerId1	CHAR	5	172
cOrdFiller	CHAR	24	177
OpenClose1	CHAR	1	201
Settlor1	CHAR	12	202
ProClient1	SHORT	2	214
SettlementPeriod1	SHORT	2	216
ADDITIONAL_ORDER_FLAGS *(Refer to Order Entry Request in Chapter 4)*	STRUCT	1	218
Reserved	CHAR	1	219
Filler1	USHORT	1(bit)	220
Filler2	USHORT	1(bit)	220
Filler3	USHORT	1(bit)	220
Filler4	USHORT	1(bit)	220
Filler5	USHORT	1(bit)	220
Filler6	USHORT	1(bit)	220
Filler7	USHORT	1(bit)	220
Filler8	USHORT	1(bit)	220
Filler9	USHORT	1(bit)	221
Filler10	USHORT	1(bit)	221
Filler11	USHORT	1(bit)	221
Filler12	USHORT	1(bit)	221
Filler13	USHORT	1(bit)	221
Filler14	USHORT	1(bit)	221
Filler15	USHORT	1(bit)	221
Filler16	USHORT	1(bit)	221
Filler17	CHAR	1	222
Filler18	CHAR	1	223
NnfField	DOUBLE	8	224
MktReplay	LONG LONG	8	232
PAN	CHAR	10	240

Structure Name	MS_SPD_OE_REQUEST		
Packet Length	480 bytes		
Transaction Code	SP_BOARD_LOT_IN (2100) SP_BOARD_LOT_ACK_IN (20408)		
Field Name	Data Type	Size in Byte	Offset
Algo ID	LONG	4	250
Reserved	SHORT	2	254
LastActivityReference	LONG LONG	8	256
Reserved	CHAR	52	264
PriceDiff	LONG	4	316
MS_SPD_LEG_INFO (leg 2)	STRUCT	80	320
MS_SPD_LEG_INFO (leg 3)	STRUCT	80	400

Table 33 MS_SPD_LEG_INFO

Structure Name	MS_SPD_LEG_INFO		
Packet Length	80 bytes		
Field Name	Data Type	Size in Byte	Offset
Token2	LONG	4	0
CONTRACT_DESC *(Refer to Order Entry Request in Chapter 4)*	STRUCT	28	4
OpBrokerId2	CHAR	5	32
Fillerx2	CHAR	1	37
OrderType2	SHORT	2	38
BuySell2	SHORT	2	40
DisclosedVol2	LONG	4	42
DisclosedVolRemaining2	LONG	4	46
TotalVolRemaining2	LONG	4	50
Volume2	LONG	4	54
VolumeFilledToday2	LONG	4	58
Price2	LONG	4	62
TriggerPrice2	LONG	4	66
MinFillAon2	LONG	4	70
ST_ORDER_FLAGS *(Refer to Order Entry Request in Chapter 4)*	STRUCT	2	74
OpenClose2	CHAR	1	76
ADDITIONAL_ORDER_FLAGS *(Refer to Order Entry Request in Chapter 4)*	STRUCT	1	77

Reserved	CHAR	1	78
FillerY	CHAR	1	79

Note: For spread order entry leg3 is not filled.

The following table provides the details of the various fields present in the Order Entry Request structure.

Field Name	Brief Description
TransactionCode	The transaction code is SP_BOARD_LOT_IN (2100). SP_BOARD_LOT_ACK_IN (20408).
ParticipantType1	This is not used.
CompetitorPeriod1	This is not used.
SolicitorPeriod1	This is not used.
Modified / CancelledBy1	This is not used.
ReasonCode1	This is not used.
TokenNumber1	This field should contain the contract descriptor of the contract. The validations will be done only on contract descriptor.
SecurityInformation1 (contract descriptor 1)	This structure contains the following fields – Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA level of the contract. This is a mandatory field and should be filled while sending the order entry request. CA Level should be set to zero.
CounterPartyBrokerId1	This is not used.
OrderType1	This field should be set to blank.
OrderNumber1	This field should be set to blank for the order entry request.
AccountNumber1	If the order is entered on behalf of a trader, the Trader Account Number is specified in this field. For broker's own order, this field is set to blank.
BookType1	This field should contain one of the following two book types. '1' – Regular lot order '2' – Special terms order
Buy / SellIndicator1	This field should contain one of the following values to specify whether the order is a buy or sell order. • '1' denotes Buy order

Field Name	Brief Description
	- '2' denotes Sell order
DisclosedVolume1	This is not used. This must be sent as zero for the order entry request.
DisclosedVolume Remaining1	This is not used. This must be sent as zero for the order entry request.
TotalVolumeRemaining1	This field should specify the total quantity remaining from the original quantity after trade(s). For order entry, this field must be set to Volume. For every response, the trading system will return this value.
Volume1	This field should contain the quantity for which the order is placed. The quantity should always be in multiples of Regular Lot and be less than the issued capital. The order will be rejected directly if the quantity is greater than or equal to the freeze quantity determined by NSE-Control.
VolumeFilledToday1	This is not used. This must be sent as blank for the order entry request.
Price1	For spread order this is not used. This must be sent as zero in spread order entry.
TriggerPrice1	This is not used. This must be sent as zero for the order entry request.
GoodTillDate1	This is not used. This must be sent as zero for the order entry request.
EntryDateTime1	This field contains the time when the order first entered the trading system. This field must be sent as zero for the order entry request.
MinimumFillVolume1	This is not used. This must be sent as zero for the order entry request.
LastModifiedTime1	This is not used. This must be sent as zero for the order entry request.
OrderTerms1	This field specifies the attributes of an order. Only IOC and AON flags are used. And the day flag should be set to '1' as all spread orders are day orders. All other flags must be sent as zero for the order entry.

Field Name	Brief Description
	• AON, if set to '1', represents All or None attribute. • IOC, if set to '1', represents Immediate or Cancel attribute. Note – Spread IOC orders are not allowed
BranchId1	This field should contain the Branch Number to which the broker belongs.
TraderId1	This field should contain the user ID.
BrokerId1	This field should contain the trading member ID.
Open / Close1	Open / Close order indicator. This field must be set to one of the following: • 'O' denotes Open • 'C' denotes Close
Settlor1	This field should contain the ID of the participants who are responsible for settling the trades through the custodians. For 'Pro' order (brokers own order) this field should be left blank.
Pro-ClientOrder1	This field should contain one of the following values to specify whether the order is entered on behalf of the broker or a trader. • '1' represents the client's order. • '2' represents a broker's order.
SettlementPeriod1	This field should contain the number of days in a settlement cycle. Currently it is 10 days.
ADDITIONAL_ORDER_FLAGS	Refer to *Additional_Order_Flags* and *Order Terms Attributes* tables in Chapter 4 for the relevant description. For reserved bit kindly set the values with 0.
Fillers (1 – 18)	These fields are reserved for future use.
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
MktReplay	This field contains the time when the order enters the system. It is time-stamped at the host end. This should be set to zero while sending to the host.

Field Name	Brief Description
PriceDiff	This is the difference between the prices at which leg2 and leg1 should trade and it should be less than 9999999.9 Note: This is used for spread order only. It is not used for 2L/3L.
TokenNumber2	This field should contain the contract descriptor of the contract. The validations will be done only on contract descriptor.
SecurityInformation2 (contract descriptor 2)	This structure contains following fields: Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA level of the contract. This is mandatory and should be filled while sending the order entry request. CA Level should be set to zero.
CounterPartyBrokerId2	This is not used.
Order Type2	This is not used.
Buy / SellIndicator2	This field should contain one of the following values to specify if the order is a buy or sell. • '1' denotes Buy order • '2' denotes Sell order
DisclosedVolume2	This is not used. This must be sent as zero for the order entry request.
DisclosedVolume Remaining2	This is not used. This must be sent as zero for the order entry request.
TotalVolumeRemaining2	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry this field should be set to Volume. For every response the trading system will return this value.
Volume2	This field should contain the quantity of order placed. The quantity should always be in multiples of Regular Lot and be less than the issued capital. The order will be rejected directly if the quantity is greater than or equal to the freeze quantity determined by NSE-Control.
VolumeFilledToday2	This is not used. This must be sent as blank for the order entry request.

Field Name	Brief Description
Price2	This is not used. This must be sent as zero for the order entry request.
TriggerPrice2	This is not used. This must be sent as zero for the order entry request.
MinimumFillVolume2	This is not used. This must be sent as zero for the order entry request.
OrderTerms2	This field should contain the attributes of an order. Currently, only IOC and AON flags are used. And the day flag is set to '1'b as all spread orders are day orders. All other flags must be sent as zero for the order entry request. • AON, if set to '1', represents All or None attribute. • IOC, if set to '1', represents Immediate or Cancel attribute.
Open / Close2	Open / Close order indicator. This field should contain one of the following values: • 'O' denotes Open • 'C' denotes Close
ADDITIONAL_ORDER_FLAGS	Refer to *Additional_Order_Flags* and *Order Terms Attributes* tables in Chapter 4 for the relevant description. For reserved bit kindly set the values with 0
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	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.
LastActivityReference	In case of order entry response, this field will contain a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Field Name	Brief Description
	This field should be set to zero for the order entry request.

Note: - For 2L/3L/Spread orders, STPC bit from additional order flag of the first leg will be referred.

Order Entry Response

The primary response of order entry is the Order Requested Message. The secondary response includes Market Order response, Order Confirmation response, Order Freeze response, and Order Error response. Market order response is provided when the entered order is market order. Each successful order entry results in order confirmation. The order error response is provided when the entered order is rejected by the trading system. The reason for the rejection is provided by the error code.

Note: Order Requested message is stopped to reduce the packet sent from the host end.

Note: Order Request Response (2101) is stopped to reduce the packet sent from the host end.

Order Confirmation Response

Successful order entry results in Order Confirmation response when the order confirmed is returned. When the entered order goes for a freeze and that freeze is approved, a similar transaction code is sent. This can be an unsolicited message as well. The message sent is as follows:

MS_SPD_OE_REQUEST *(Refer to Order Entry Request of Spread Order in Chapter 5)*

The following table provides the details of the various fields present in the Order Confirmation Response structure.

Field Name	Brief Description
TransactionCode	The transaction code is SP_ORDER_CONFIRMATION (2124).
EntryDateTime1	The order time (original order) will be populated in this field.
OrderNumber	This field contains the order number assigned to the order.

Price	This field contains zero for both the legs.
OrderTerms	The flags are set as discussed in 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 provided by reason code and the reason string. The message sent is:

MS_SPD_OE_REQUEST *(Refer to Order Entry Request of Spread Order in Chapter 5)*

Field Name	Brief Description
TransactionCode	The transaction code is SP_ORDER_ERROR (2154).
ErrorCode	This field contains the error number. *Refer to the List of Error Codes in Appendix.*

Order Cancel Confirmation Response

In the case of Immediate or Cancel (IOC) order, the system accepts the orders from the users and tries to immediately match the orders with the orders in the books. If the order does not match, the order is cancelled by the system. Then the order cancel confirmation response is return back. The message sent is as follows:

MS_SPD_OE_REQUEST *(Refer to Order Entry Request of Spread Order in Chapter 5)*

Field Name	Brief Description
TransactionCode	The transaction code is SP_ORDER_CXL_CONFIRMATION (2130).
EntryDateTime	This field contains the date and time when the order entered the system.
OrderNumber	This field contains the order number assigned to the order.
LastModifiedTime	This will be the current cancellation time. (It will be same as LogTime)

Spread Order Modification

Order Modification enables the trader to modify unmatched orders.

Rules of Order Modification

According to the rules of Order Modification, the following modifications are not allowed:

- Buy to sell or vice versa
- Modifying Contract
- Modifying Frozen orders
- Branch Manager (BM) modifying the Corporate Manager's (CM) orders
- Dealer (DL) modifying the BM's orders
- DL modifying the 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 deactivated broker's orders
- Changing the original data

Note: RL/ST/SL book types can be modified / switched between themselves only.

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.

MS_SPD_OE_REQUEST *(Refer to Order Entry Request of Spread Order in Chapter 5)*

The following table provides the details of the various fields present in the Order Modification Request structure.

Field Name	**Brief** Description
TransactionCode	The transaction code is: For Order Modification: SP_ORDER_MOD_IN (2118). For Order Cancellation: SP_ORDER_CANCEL_IN (2106). For Ack Order Modification: SP_ORDER_MOD_IN (20416).

	For Ack Order Cancellation: SP_ORDER_CANCEL_IN (20414).
Modified / CancelledBy	This field should denote who 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
OrderNumber	This field should contain the Order Number of the order to be modified.
TraderId1	This field should contain the ID of the user on whose behalf order is to be modified/cancelled.
EntryDateTime	This field, while coming from the host, contains the date and time when the order entered the trading system. This field should be set to zero while sending to the host.
LastModified1	In the case of order entry, this field will be same as Entry Date Time. After the order is modified, it contains the time when the Order was last modified. It is the time in seconds from midnight of January 1, 1980. In case of Order Modification Request, this field should contain the time when the Order was last modified.
LastActivityReference	In Order modification request, this field should contain LastActivityReference value received in response of the last activity done on that order. Last activity could be order entry, order modification or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Note: Order Modification/Cancellation Response (2119/2107) message is stopped to reduce the packet from the host end.

Order Modification Confirmation Response

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

MS_SPD_OE_REQUEST *(Refer to Order Entry Request of Spread Order in Chapter 5)*

The following table provides the details of the various fields present in the Order Modification Confirmation Response structure.

Field Name	Brief Description
TransactionCode	The transaction code is SP_ORDER_MOD_CON_OUT (2136). SP_ORDER_CXL_CONFIRMATION (2130).
LastModifiedTime	This field contains the time when the order was last modified (in seconds from midnight of January 1, 1980.)
LastActivityReference	This field will contain a unique value for current activity. Currently the same shall be in nanoseconds. Changes if any shall be notified.

Order Modification Error Response

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

MS_SPD_OE_REQUEST *(Refer to Order Entry Request of Spread Order in Chapter 5)*

The following table provides the details of the various fields present in the Order Modification Error Response structure.

Field Name	Brief Description
TransactionCode	The transaction code is: For Order Modification, SP_ORDER_MOD_REJ_OUT (2133). For Order Cancellation, SP_ORDER_CXL_REJ_OUT (2127).

Spread Order Cancellation

Refer to Order Cancellation in Chapter 4.

Note: All the transaction codes used for Order Cancellation are provided along with the Order Modification transaction codes.

Spread Trade Modification

Refer to Trade Modification in Chapter 4.

Spread Trade Cancellation

Refer to *Trade Cancellation* *in Chapter 4.*

Spread Combination Master Update Broadcast

New structure of spread combination information (MS_SPD_UPDATE_INFO) is provided as follows:

Table 34 MS_SPD_UPDATE_INFO

Structure Name	MS_SPD_UPDATE_INFO		
Packet Length	132 bytes		
Transaction Code	BCAST_SPD_MSTR_CHG (7309)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER *(Refer to* *BCAST_HEADER* *in Chapter 2)*	STRUCT	40	0
Token1	LONG	4	40
Token2	LONG	4	44
SEC_INFO (SecInfo1)	STRUCT	30	48
SEC_INFO (SecInfo2)	STRUCT	30	78
ReferencePrice	LONG	4	108
DayLowPriceDiffRange	LONG	4	112
DayHighPriceDiffRange	LONG	4	116
OpLowPriceDiffRange	LONG	4	120
OpHighPriceDiffRange	LONG	4	124
ST_SPD_ELIGIBILITY	STRUCT	1	128
Reserved	CHAR	1	129
DeleteFlag	CHAR	1	130
Reserved	CHAR	1	131

Table 35 SEC_INFO

Structure Name	SEC_INFO		
Packet Length	30 bytes		
Field Name	Data Type	Size in Byte	Offset
InstrumentName	CHAR	6	0
Symbol	CHAR	10	6
Series	CHAR	2	16
ExpiryDate	LONG	4	18
StrikePrice	LONG	4	22

OptionType	CHAR	2	26
CALevel	SHORT	2	28

Table 36 ST_SPD_ELIGIBILITY

Structure Name	ST_SPD_ELIGIBILITY		
Packet Length	1 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	BIT	7 (bit)	0
Eligibility	BIT	1 (bit)	0

The following table provides the details of the various fields present in the Spread Combination Master Update Broadcast structure.

Field Name	Brief Description
Transaction Code	BCAST_SPD_MSTR_CHG (7309).
SecurityInformation1	This will contain instrument name, symbol, series, strike price, option type and corporate action level of leg 1 contract.
SecurityInformation2	This will contain instrument name, symbol, series, strike price, option type and corporate action level of leg 2 contract.
ReferencePrice	Settlement price of leg 1 contract will be the base for calculating price difference ranges.
DayLowPriceDiffRange	Day low price difference range for the combination.
DayHighPriceDiffRange	Day high price difference range for the combination.
OpLowPriceDiffRange	Operating low price difference range for the combination.
OpHighPriceDiffRange	Operating high price difference range for the combination.
Eligibility	The flag will be set to 1 if the combination is allowed to trade.

DeleteFlag	This will contain one of the following values to denote whether the spread combination is deleted or not. • 'N' – Active • 'Y' – Deleted

Periodic Broadcast for Change in Spread Combination Master

This will be periodically sent by the exchange for those spread contracts which have had any parameters changed during the day.

The structure being sent is:

Refer to MS_SPD_UPDATE_INFO in Chapter 5

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SPD_MSTR_CHG_PERIODIC (7341).

Spread Combination File

Spread combinations for the next trading day will be provided in spd_contract.txt after trading hours.

This file will have all the valid spread combinations and will be in pipe delimited format. The upload file will 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 will be separated by pipe ('|'). The fields will not be of fixed width.

The structure for spread contract file is provided as follows:

CONTROL RECORD			
S. NO	Field	Type	Max Field Length
1	Segment Indicator	CHAR	6
2	Version number	CHAR	5
DETAIL RECORD			
1	Token1	NUMBER	6
2	Token2	NUMBER	6
3	InstrumentName1	CHAR	6
4	Symbol1	CHAR	10

5	Series1	CHAR	2
6	ExpiryDate1	NUMBER	10
7	StrikePrice1	NUMBER	10
8	OptionType1	CHAR	2
9	CALevel1	NUMBER	7
10	InstrumentName2	CHAR	6
11	Symbol2	CHAR	10
12	Series2	CHAR	2
13	ExpiryDate2	NUMBER	10
14	StrikePrice2	NUMBER	10
15	OptionType2	CHAR	2
16	CALevel2	NUMBER	7
17	ReferencePrice	NUMBER	10
18	DayLowPriceDiffRange	NUMBER	10
19	DayHighPriceDiffRange	NUMBER	10
20	OpLowPriceDiffRange	NUMBER	10
21	OpHighPriceDiffRange	NUMBER	10
22	BoardLotQuantity1	NUMBER	9
23	MinimumLotQuantity1	NUMBER	9
24	TickSize1	NUMBER	9
25	BoardLotQuantity2	NUMBER	9
26	MinimumLotQuantity2	NUMBER	9
27	TickSize2	NUMBER	9
28	Eligibility	CHAR	1
29	DeleteFlag	CHAR	1

The following table provides the details of the various fields present in the Spread Combination File structure.

Field Name	Brief Description
Token1	Token number of leg 1 contract of the spread combination
Token2	Token number of leg 2 contract of the spread combination
SecurityInformation1	This will contain instrument name, symbol, series, strike price, option type and corporate action level of leg 1 contract

Field Name	Brief Description
SecurityInformation2	This will contain instrument name, symbol, series, strike price, option type and corporate action level of leg 2 contract
ReferencePrice	Settlement price of leg 1 contract will be the base for calculating price difference ranges
DayLowPriceDiffRange	Day low price difference range for the combination. It may be changed intraday. Note: The value can be a positive number/ negative number or zero.
DayHighPriceDiffRange	Day high price difference range for the combination. It may be changed intraday. Note: The value can be a positive number or zero.
OpLowPriceDiffRange	Minimum price difference at which the spread order could be placed without being rejected by the system. It may be changed intraday and can be flexed to day Low price difference. Note: The value can be a positive number/ negative number or zero.
OpHighPriceDiffRange	Maximum price difference at which the spread order could be placed without being rejected by the system It may be changed intraday and can be flexed to day high price difference. Note: The value can be a positive number or zero.
BoardLotQuantity1	Board lot quantity of leg 1 contract
MinimumLotQuantity1	Min lot quantity of leg 1 contract
TickSize1	Tick size of leg 1 contract
BoardLotQuantity2	Board lot quantity of leg 2 contract

Field Name	Brief Description
MinimumLotQuantity2	Min lot quantity of leg 2 contract
TickSize2	Tick size of leg 2 contract
Eligibility	The flag will be set to 1 if the combination is allowed to trade.
DeleteFlag	This will contain one of the following values to denote whether the spread combination is deleted or not. • ‘N‘ – Active • ‘Y’ – Deleted

Chapter 6 2L and 3L Order and Trade Management

In 2L and 3L Order and Trade Management, 2L and 3L order entry allows the trader to place orders in the market. The trading system accepts the orders from the users and tries to immediately match the orders with the orders in the books. If the order does not match, the order is cancelled by the system.

Note: By default, all 2L and 3L orders are Immediate or Cancel (IOC) orders.

Rules of 2L and 3L Order Entry

According to the 2L and 3L Order Entry, these items are not allowed in the following conditions:

- Good Till Cancellation (GTC) or Good Till Date (GTD) order
- 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 that market
- Security does not exist in the system
- Broker is suspended
- Broker does not exist in the 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 the trading system
- Participant is suspended
- Participant does not exist in trading system
- Order price is beyond day's minimum maximum range

- Quantity is more than issued capital
- Quantity is not equal to multiples of regular lot
- Limit Price is not a multiple of Tick size
- IOC and Disclosed Quantity combination
- For PRO order, client is other than broker
- For CLI order, Account Number is Broker ID
- Order attributes are not entered properly for various book types
- Contracts are same for the any 2 legs

For Multi-leg orders i.e. 2L/3L orders all tokens in the respective legs should be from the same stream.

Order Entry Request

Refer to the structure of Order Entry Request – Spread Order and Trade Management in Chapter 5.

Note: All the field names of the Leg 3 are suffixed with '3' instead of '2' as shown in the following table:

Field Name	Brief Description
TransactionCode	The transaction code for 2L order is TWOL_BOARD_LOT_IN (2102). The transaction code for 3L order is THRL_BOARD_LOT_IN (2104). The transaction code for 2L order ack is TXN_EXT_TWOL_BOARD_LOT_ACK_IN (20410). The transaction code for 3L order ack is TXN_EXT_THRL_BOARD_LOT_ACK_IN (20412).
Price1	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. This is to be multiplied by 100 before sending to the trading system. The order will be rejected directly if the price is outside the day's price range determined by NSE-Control. Note: For spread order, this is not used. This must be sent as zero for spread order entry request.

Field Name	Brief Description
Price2	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. This is to be multiplied by 100 before sending to the trading system. The order will be rejected directly if the price is outside the day's price range determined by NSE-Control. Note: The value can be a positive number or zero. For spread order, this is not used. This must be sent as zero for spread order entry request.
Note: Descriptions of other fields are same as given in Spread Order Entry Request in Chapter 4. The fields and their description given below are applicable for 3L orders only.	
ParticipantType3	This is not used.
CompetitorPeriod3	This is not used.
SolicitorPeriod3	This is not used.
Modified / CancelledBy3	This is not used.
ReasonCode3	This is not used.
TokenNumber3	This is the Token Number of the contract on which order is to be placed. This field should contain a valid token number or '-1'. If the token number is set to '-1' then validation will be done only on contract descriptor. If the valid token number is sent, validation will be done on token number as well as on the contract descriptor.
SecurityInformation3 (contract descriptor 3)	This structure contains following fields. Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA Level of the contract. This is mandatory and should be filled while sending the order entry request. CA Level should be set to zero.
CounterPartyBroker Id3	This is not used.
OrderType3	This is not used.
Buy / SellIndicator3	This field should contain one of the following values: • '1' for Buy order • '2' for Sell order

Field Name	Brief Description
DisclosedVolume3	This is not used. This must be sent as zero for the order entry request.
DisclosedVolume Remaining3	This is not used. This must be sent as zero for the order entry request.
TotalVolume Remaining3	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry this field should be set to Volume. For every response, the trading system will return this value.
Volume3	This field specifies the quantity of order placed. The quantity should always be in multiples of Regular Lot and be less than the issued capital. The order will be rejected directly if the quantity is greater than or equal to the freeze quantity determined by NSE-Control.
VolumeFilled Today3	This is not used. This must be sent as blank for the order entry request.
Price3	This field specifies 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. This is to be multiplied by 100 before sending to trading system. The order will be rejected directly if the price is outside the day's price range determined by NSE-Control.
VolumeAvailable	This is not used. This must be sent as zero for the order entry request.
MinimumFill Volume3	This is not used. This must be sent as zero for the order entry request.
Open / Close3	Open / Close order indicator. This field should contain one of the following values: • 'O' for Open • 'C' for Close
Cover / Uncover3	This field should contain one of the following values: • 'U' represents Uncovered • 'V' represents Covered

Order Entry Response

The primary response is the Order Requested Message. The secondary response can be order confirmation, order error or one of the general error responses. The order error response is given when the entered order is rejected by the trading system. The reason for the rejection is provided by error code.

Note: Order requested response (2103/2105) message is stopped to reduce the packet sent from the host end.

Market Order Response

No transaction code for price confirmation will be sent for 2L/3L orders.

Order Confirmation Response

Successful order entry results in Order Confirmation Response. The order confirmed is returned. 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 Market flag in order terms is set to '1'b for market 2L and 3L order response. The message sent is as follows:

MS_SPD_OE_REQUEST *(Refer to Spread Order Entry Request in Chapter 5)*

Field Name	Brief Description
TransactionCode	The transaction code is: For 2L order, TWOL_ORDER_CONFIRMATION (2125). For 3L order, THRL_ORDER_CONFIRMATION (2126).
EntryDateTime1	The order time (original order) will be populated in this field.
OrderNumber	This field contains the order number assigned to the order.
Price	This field contains the price of the order. If a Market order is entered when market is in Open state, the 'Market' flag in Order Terms is set and is priced at the prevailing price at the trading system. If it is a priced order, the order gets confirmed at that price.
OrderTerms	The flags are set as discussed in *Order Entry Request* discussed earlier in this chapter.

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 reason code and the reason string. The message sent is:

MS_SPD_OE_REQUEST *(Refer to Spread Order Entry Request in Chapter 5)*

Field Name	Brief Description
TransactionCode	The transaction code is TWOL_ORDER_ERROR (2155) for 2L order. The transaction code is THRL_ORDER_ERROR (2156) for 3L order.
ErrorCode	This field contains the error number. *Refer to List of Error Codes in Appendix.*

Order Cancel Confirmation Response

The system accepts the orders from the users and tries to match the orders immediately with the orders in the books. If the order does not match, the order is cancelled by the system. Then the order cancel confirmation response is sent back. The message sent is as follows:

MS_SPD_OE_REQUEST *(Refer to Spread Order Entry Request in Chapter 5)*

Field Name	Brief Description
TransactionCode	The transaction code is TWOL_ORDER_CXL_CONFIRMATION (2131) for 2L order and THRL_ORDER_CXL_CONFIRMATION (2132) for 3L order.
EntryDateTime	This field contains the date and time when the order entered the system.
OrderNumber	This field contains the order number assigned to the order.
LastModifiedTime	This will be the current cancellation time. (It will be same as LogTime)

Partial Order Cancellation Confirmation Response

Partial cancellation of order results in Order Cancellation Confirmation Response. This transcode will be sent in response if 2L/3L Order results in partial cancellation. The message sent is as follows:

MS_SPD_OE_REQUEST *(Refer to Spread Order Entry Request in Chapter 5)*

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CANCEL_CONFIRM_OUT (2075).

Field Name	Brief Description
LastModifiedTime	This will be the current cancellation time. (It will be same as LogTime)

Trade Modification

Refer to Trade Modification in Chapter 4.

Trade Cancellation

Refer to Trade Cancellation in Chapter 4.

Chapter 7 Unsolicited Messages

This section covers the messages that are received on the interactive connection. These messages are received by users not in response to any request. The sections covered in this chapter are as follows:

- Stop Loss Order Triggering
- Market If Touched Triggering
- Freeze Approve Response
- Freeze Reject Response
- Trade Confirmation
- Trade Modification
    - Trade Modification Requested Notification
    - Trade Modification Confirmation Response
    - Trade Modification Rejection Response
- Trade Cancellation
    - Trade Cancellation Requested Notification
    - Trade Cancellation Confirmation Response
    - Trade Cancellation Rejection
- Limits Updations
    - Order limit update
    - Spread Order limit update
- Interactive/Broadcast Messages Sent from Control
- Message for the change in trading status
- Identification for Market Wide Open Interest (OI) Limit Messages
- Identification for Member Violation Messages

Stop Loss Order Triggering

When any stop loss order entered is triggered, the user entering the order receives this message. The message sent is as follows:

MS_TRADE_CONFIRM *(Refer to* Trade Confirmation *discussed later in this section.)*

Field Name	Description
TransactionCode	The transaction code is ON_STOP_NOTIFICATION (2212).
LogTime (of MESSAGE_HEADER)	This field will have the trigger time

Market If Touched Triggering

~~When any Market If Touched order entered is triggered, the user entering the order receives this message. The message sent is as follows:~~

~~MS_TRADE_CONFIRM *(Refer to Trade Confirmation discussed later in this section)*~~

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

Freeze Approve Response

This message is sent when an earlier order, which had resulted in freeze, has been 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	Description
TransactionCode	The transaction code is ORDER_CONFIRMATION (2073).
LastModified DateTime	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 an earlier 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	Description
TransactionCode	The transaction code is ORDER_ERROR (2231).

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. The following structure is sent:

Note - Refer to pg.194 for bulk resulted trade

Table 37 MS_TRADE_CONFIRM

Structure Name	MS_TRADE_CONFIRM		
Packet Length	296 bytes		
Transaction Code	TRADE_CONFIRMATION (2222)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
ResponseOrderNumber	DOUBLE	8	40
BrokerId	CHAR	5	48
Reserved	CHAR	1	53
TraderNumber	LONG	4	54
AccountNumber	CHAR	10	58
Buy/SellIndicator	SHORT	2	68
OriginalVolume	LONG	4	70
DisclosedVolume	LONG	4	74
RemainingVolume	LONG	4	78
DisclosedVolumeRemaining	LONG	4	82
Price	LONG	4	86
ST_ORDER_ FLAGS *(Refer to Order Entry Request in Chapter 4)*	STRUCT	2	90
GoodTillDate	LONG	4	92
FillNumber	LONG	4	96
FillQuantity	LONG	4	100
FillPrice	LONG	4	104
VolumeFilledToday	LONG	4	108

Structure Name	MS_TRADE_CONFIRM		
Packet Length	296 bytes		
Transaction Code	TRADE_CONFIRMATION (2222)		
Field Name	Data Type	Size in Byte	Offset
ActivityType	CHAR	2	112
ActivityTime	LONG	4	114
CounterTraderOrderNumber	DOUBLE	8	118
CounterBrokerId	CHAR	5	126
Token	LONG	4	132
CONTRACT_DESC *(Refer to Order Entry Request in Chapter 4)*	STRUCT	28	136
OpenClose	CHAR	1	164
OldOpenClose	CHAR	1	165
BookType	CHAR	1	166
Reserved	LONG	4	168
OldAccountNumber	CHAR	10	172
Participant	CHAR	12	182
OldParticipant	CHAR	12	194
ADDITIONAL_ORDER_FLAGS *(Refer to Order Entry Request in Chapter 4)*	STRUCT	1	206
Reserved	CHAR	1	207
Reserved	CHAR	1	208
ReservedFiller2	CHAR	1	209
PAN	CHAR	10	210
OldPAN	CHAR	10	220
Algo ID	LONG	4	230
Reserved	SHORT	2	234
LastActivityReference	LONG LONG	8	236
Reserved	CHAR	52	244

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CONFIRMATION (2222).
ResponseOrder Number	This field contains the order number of the trader's order taking part in the trade.
BrokerId	This field contains the Trading Member ID.
TraderNumber	This field contains the trader or user ID.

Field Name	Brief Description
	Note: Data type changed from SHORT to LONG
AccountNumber	This field contains the Account Number or Client code.
Buy / SellIndicator	This field contains one of the following values. • '1' for Buy • '2' for Sell
OriginalVolume	This field contains the original traded volume.
DisclosedVolume	This field contains 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.
RemainingVolume	This field contains the volume remaining after trade(s).
DisclosedVolume Remaining	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 the Preopen Trades happening in the Opening Phase
GoodTillDate	This field contains the number of days for a GTD order. This field may be set in two ways. To specify an absolute date, this field should be set to that date in number of seconds since midnight of Jan 1, 1980. To specify days, this field should be set 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.
FillQuantity	This field contains the traded volume.
FillPrice	This field contains the price at which order has been traded.
VolumeFilled Today	This field contains the quantity traded today.
ActivityType	This field contains one of the following values to denote the activity type.

Field Name	Brief Description
	• 'B' for Buy • 'S' for Sell
ActivityTime	This field contains the time when the activity has taken place.
CounterTrader OrderNumber	This field contains same value as "ResponseOrderNumber" field.
CounterBrokerId	This field contains same value as "BrokerId" field.
SecurityInformation (Contract Descriptor)	This structure contains the following fields: Instrument Name, Symbol, Strike Price, Option Type and CA Level for the contract.
BookType	This field contains the book type—RL/ ST/ SL/ NT/ OL/ SP/ Auction.
OpenClose	This field contains either 'O' for Open or 'C' for Close.
OldOpenClose	For trade confirmation both Open Close and Old Open Close fields are same.
Participant	For trade confirmation, this field contains participant id. In the case of trade modification, this field contains the participant id same as that of original trade confirmation message.
OldParticipant	For trade confirmation, this field contains participant id. In the case of trade modification, this field contains the participant id same as that of original trade confirmation message.
~~ProCliFlag~~	~~This field contains one of the following values:~~ ~~• '1' for client's order~~ ~~• '2' for broker's order~~ ~~(same as Pro/Client/ Warehouse indicator)~~
PAN	This field shall contain the PAN
OldPAN	In case of trade modification this field shall contain the old PAN else it will be blank
Algo ID	This field shall contain the Algo ID
Reserved	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.
LastActivityReference	This field will contain a unique value for current activity. 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

Trade Modification

Trade Modification is a facility provided by NSE to allow users to change client account number of executed trades.

Trade Modification Confirmation Response

The trade modification is confirmed and the new trade data is sent.

MS_TRADE_CONFIRM *(Refer to Trade Confirmation Chapter 7)*

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 Rejection Response

The trade modification is rejected by NSE-Control.

MS_TRADE_CONFIRM *(Refer to Trade Confirmation Chapter 7)*

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_MODIFY_REJECT (2288).

Trade Cancellation

Trade Cancellation Requested Notification

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

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

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:

MS_TRADE_CONFIRM *(Refer to Trade Confirmation Chapter 7)*

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CANCEL_CONFIRM (2282).
LogTime (of MESSAGE_HEADER)	This will contain the activity Time i.e. the latest modified time.

Trade Cancellation Rejection

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

MS_TRADE_CONFIRM *(Refer to Trade Confirmation Chapter 7)*

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 *List of Error Codes*.

Limits Updations

A message is sent to the respective dealers for Updations of user order value limit and branch order value limit by the Corporate Manager.

Table 38 MS_ORDER_VAL_LIMIT_DATA

Structure Name	MS_ORDER_VAL_LIMIT_DATA		
Packet Length	206 bytes		
Transaction Code	USER_ORDER_LIMIT_UPDATE_OUT (5731)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
BrokerId	CHAR	5	40
BranchId	SHORT	2	45
UserName	CHAR	25	47
UserId	LONG	4	72
UserType	SHORT	2	76
INSTRUMENT_USER [2]	STRUCT	64	78

Table 39 INSTRUMENT_USER

Structure Name	INSTRUMENT_USER		
Packet Length	64 bytes		
Field Name	**Data Type**	**Size in Byte**	**Offset**
BranchBuyValueLimit	DOUBLE	8	0
BranchSellValueLimit	DOUBLE	8	8
BranchUsedBuyValueLimit	DOUBLE	8	16
BranchUsedSellValueLimit	DOUBLE	8	24
UserOrderBuyValueLimit	DOUBLE	8	32
UserOrderSellValueLimit	DOUBLE	8	40
UserOrderUsedBuyValueLimit	DOUBLE	8	48
UserOrderUsedSellValueLimit	DOUBLE	8	56

The following table provides the details of the various fields present in the Limits Updation structure.

Field Name	Brief Description
TransactionCode	The transaction code is: USER_ORDER_LIMIT_UPDATE_OUT (5731).
BrokerId	This field contains the Trading Member ID of the broker.
BranchId	This field contains the branch number of the trader to which he belongs
UserName	This field contains the name of user.
UserId	This field should contain the user ID of the user/broker.
INSTRUMENT_USER	Structure wherein instrument wise limit is updated. Note: INSTRUMENT_USER [0] is for Instrument type Future. INSTRUMENT_USER[1] is for Instrument type Options
BranchBuyValueLimit	This field contains the total Buy order limit for the branch to which the user belongs.
BranchSellValueLimit	This field contains the total Sell order limit for the branch to which the user belongs.
BranchUsedBuyValueLimit	This field contains the used Buy order limit for the branch to which the user belongs.
BranchUsedSellValueLimit	This field contains the used Sell order limit for the branch to which the user belongs.
UserOrderBuyValueLimit	This field contains the total Buy order limit for the user.
UserOrderSellValueLimit	This field contains the total Sell order limit for the user.
UserOrderUsedBuyValueLimit	This field contains the Used Buy order limit by the user.
UserOrderUsedSellValueLimit	This field contains the Used Sell order limit by the user.

Order Limit Update

When corporate manager updated order limit for any user DEALER_LIMIT_UPDATE_OUT transcode is being sent to all the individual users.

Table 40 DEALER_ORD_LMT

Structure Name	DEALER_ORD_LMT		
Packet Length	66 bytes		
Transaction Code	DEALER_LIMIT_UPDATE_OUT (5733)		
Field Name	Data Type	Size in Byte	Offset

MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
BrokerId	CHAR	5	40
UserId	LONG	4	45
OrdQtyBuff	DOUBLE	8	49
OrdValBuff	DOUBLE	8	57

The following table provides the details of the various fields present in the Order limit Update structure:

Field Name	Brief Description
TransactionCode	The transaction code is: USER_ORDER_LIMIT_UPDATE_OUT (5731).
BrokerId	This field contains the Trading Member ID of the broker.
UserId	This field should contain the user ID of the user/broker.
OrdQtyBuff	This field contains the maximum Order quantity which user can enter while order entry.
OrdValBuff	This field contains the maximum order value which user can enter while order entry.

Spread Order Limit Update

This happens when the Corporate Manager updates the spread order limit for any user. SPD_ORD_LIMIT_UPDATE_OUT transcode is being sent to all the individual users.

Table 41 SPD_ORD_LMT

Structure Name	SPD_ORD_LMT		
Packet Length	66 bytes		
Transaction Code	SPD_ORD_LIMIT_UPDATE_OUT (5772)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
BrokerId	CHAR	5	40
UserId	LONG	4	45
SpdOrdQtyBuff	DOUBLE	8	49
SpdOrdValBuff	DOUBLE	8	57

The following table provides the details of the various fields present in the Spread Order Limit Update structure.

Field Name	Brief Description
TransactionCode	The transaction code is: USER_ORDER_LIMIT_UPDATE_OUT (5731).
BrokerId	This field contains the Trading Member ID of the broker.
UserId	This field should contain the user ID of the user/broker.
SpdOrdQtyBuff	This field contains the maximum Order quantity which user can enter while spread order entry.
SpdOrdValBuff	This field contains the maximum order value which user can enter while spread order entry.

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 Broadcast, Chapter 9).*

Message for the Change in Trading Status

Whenever the trading status of the trading member is changed from

1. Active to suspended
2. Close out to suspended
3. Suspended to suspended (multiple suspensions)
4. Suspended to close-out

Users under the corresponding trading firm will receive the message for change in trading status. When the message is sent to a particular user, it comes as an interactive message in the following structure:

Table 42 MS_TRADER_INT_MSG

Structure Name	MS_TRADER_INT_MSG		
Packet Length	290 bytes		
Transaction Code	For interactive messages—CTRL_MSG_TO_TRADER (5295). For broadcast messages—BCAST_JRNL_VCT_MSG (6501).		
Field Name	Data Type	Size in Byte	Offset

MESSAGE_HEADER*(Refer to* *MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
TraderId	LONG	4	40
Reserved	CHAR	3	44
Reserved	CHAR	1	47
BroadCastMessageLength	SHORT	2	48
BroadCastMessage	CHAR	239	50

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

Field Name	Brief Description
TransactionCode	The transaction code is: CTRL_MSG_TO_TRADER (5295) for interactive messages and BCAST_JRNL_VCT_MSG (6501) for broadcast messages.

Identification for Market Wide Open Interest (OI) Limit Messages

Market wide OI limit messages are sent from the exchange in the general broadcast message structure BCAST_JRNL_VCT_MSG, transaction code 6501. Other general messages are also sent in this structure.

To identify the Market wide OI Limit broadcast messages, a new action code is defined with value ‘MWL’ in the field ActionCode of the current structure. There is no structural change to accommodate this change.

Structure for General Broadcast Message BCAST_JRNL_VCT_MSG (6501)

Table 43 MS_BCAST_MESSAGE

Structure Name	MS_BCAST_MESSAGE		
Packet Length	320 bytes		
Transaction Code	BCAST_JRNL_VCT_MSG (6501).		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER *(Refer to* *BCAST_HEADER in Chapter 2)*	STRUCT	40	0
BranchNumber	SHORT	2	40
BrokerNumber	CHAR	5	42
ActionCode	CHAR	3	47

ST_BCAST_DESTINATION	STRUCT	2	50
Reserved	CHAR	26	52
BroadcastMessageLength	SHORT	2	78
BroadcastMessage	CHAR	239	80

Table 44 ST_BCAST_DESTINATION

Structure Name	ST_BCAST_DESTINATION		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Reserved	BIT	4 (bit)	0
Journaling Required	BIT	1 (bit)	0
Tandem	BIT	1 (bit)	0
ControlWorkstation	BIT	1 (bit)	0
TraderWorkstation	BIT	1 (bit)	0
Reserved	CHAR	1	1
For Big Endian Machines			
Trader WorkStation	BIT	1 (bit)	0
ControlWorkStation	BIT	1 (bit)	0
Tandem	BIT	1 (bit)	0
JournallingRequired	BIT	1 (bit)	0
Reserved	BIT	4 (bit)	0
Reserved	CHAR	1	1

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

Field Name	Brief Description
TransactionCode	The transaction code is: BCAST_JRNL_VCT_MSG (6501).
BranchNumber	This field contains the branch number of the trader's branch. .
BrokerNumber	This field contains the Trading Member ID of the broker.
ActionCode	This field contains the action code which indicates the action taken. Note: For example, 'SYS' - system 'LIS' - Listing 'MWL' – Market Wide OI Limit Message

Broadcast Destination	This field specifies the destination of the message, that is, Trader Workstation or Control Workstation.
Broadcast MessageLength	This field contains the length of the broadcast message.
BroadcastMessage	This field contains the broadcast message.

Identification for Member Violation Messages

Member specific messages such as change in trading status of the member, violation messages, trade modification request notification etc., are sent from Exchange in the structure 'CTRL_MSG_TO_TRADER' (transaction code - 5295).

To identify the Member Violation messages, a new field is defined as 'ActionCode' in 'CTRL_MSG_TO_TRADER' (5295) structure. To define this new field, the reserved bytes existing in the structure are used. For Violation messages the value of 'action code' field will be "MAR".

Structure for Trader specific messages CTRL_MSG_TO_TRADER (5295)

(Existing reserved 3 bytes replaced with action code field)

Table 45 CTRL_MSG_TO_TRADER

Structure Name	CTRL_MSG_TO_TRADER		
Packet Length	290 bytes		
Transaction Code	For interactive messages— CTRL_MSG_TO_TRADER (5295).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to* *MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
TraderId	LONG	4	40
ActionCode	CHAR	3	44
Reserved	CHAR	1	47
BroadCastMessageLength	SHORT	2	48
BroadCastMessage	CHAR	239	50

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

Field Name	Brief Description

TraderId	User ID to whom the message belongs to. Note: Data type is changed from SHORT to LONG
ActionCode	This field contains the action code which indicates the action taken. Note: For example, ‘MAR’ – Margin Violation Message ‘OTH’ – Other Messages
BroadCastMessage Length	Message Length
BroadCastMessage	Message

Chapter 8 Bhavcopy

The bhavcopy is broadcasted at the end of the day. Firstly, a message is sent that the broadcast of the bhavcopy will start now. Next the header is sent indicating that actual data will follow this packet. Then the data for non-depository is sent. On completion of the data of the depository securities, a packet follows stating that the bhavcopy for the depository securities will be broadcasted after this packet. Thereafter, the data for the depository securities is broadcasted. This follows the trailer record, marking the end of bhavcopy.

Bhavcopy Broadcast

Bhavcopy is sent after closing of trading hours. However, Futures and Options trading system will have the capability to have different trading sessions for various underlying assets. As such, separate bhavcopy will be generated for each trading session.

The current trading session will be identified as ‘Regular Trading Session’. Although the trading system will be capable of having different trading sessions, this document is restricted to describe changes with respect to one such trading session. This trading session is referred as ‘Additional Trading Session’ in this document.

There are no changes in the structure of the bhavcopy broadcast, the values in Message Type field available in the current structure will be used to identify the trading session for which the bhavcopy is being broadcasted. The changes in the Message Type field value for various bhavcopy structures are presented as follows:

Message Stating the Transmission of Bhavcopy Will Start Now

This is the first message broadcasted that the bhavcopy will be started now. The structure sent is:

MS_BCAST_MESSAGE *(Refer to Broadcast, Chapter 9)*

Field Name	Description
TransactionCode	The transaction code is BCAST_JRNL_VCT_MSG (6501).

To provide co-existence for trading members, a new transcode has been provisioned that will allow the members to communicate with the exchange. This section covers the details of the new transcode as well. Members can continue to use the existing transcode and respective structures during the co-existence period. At the end of co-existence period, existing transcode and respective structures will be discontinued.

Header of Report on Market Statistics

A header precedes the actual bhavcopy that is sent to the trader.

The header for the bhavcopy is sent before actual data packet. The message structure sent is MS_RP_HDR. To identify the type of bhavcopy broadcast, the Message type field will be used.

The value for Regular Trading Session bhavcopy, Additional Trading Session bhavcopy and final bhavcopy will be “H”, “X” and “L” respectively.

The message structure sent is:

REPORT HEADER

Table 46 MS_RP_HDR

Structure Name	MS_RP_HDR		
Packet Length	108 bytes		
Transaction Code	RPRT_MARKET_STATS_OUT_RPT (1833)/ ENHNCD_RPRT_MARKET_STATS_OUT_RPT (11833)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
ReportDate	LONG	4	41
UserType	SHORT	2	45
BrokerId	CHAR	5	47
FirmName	CHAR	25	52
TraderNumber	LONG	4	77
TraderName	CHAR	26	81

The following table provides the details of the various fields present in the Report Header structure.

Field Name	Brief Description
TransactionCode	The transaction code is RPRT_MARKET_STATS_OUT_RPT (1833) or ENHNCD_RPRT_MARKET_STATS_OUT_RPT (11833).
MessageType	This field is set to ‘H’ for Regular Trading Session bhavcopy. The field is set to ‘X’ for the ‘Additional Trading Session bhavcopy. The field is set to ‘L’ for the final bhavcopy.
OrgScope	This field is reserved for future use.
ReportDate	Set to the report date.
UserType	This field specifies the type of user. It is set to ‘-1’.
BrokerId	This field specifies the Trading Member ID. It is set to blank.
BrokerName	This field specifies the name of the broker. It is set to blank.
TraderNumber	This field specifies the trader/user ID. It is set to zero. Note: Data type changed from SHORT to LONG
TraderName	This field specifies the name of the trader. It is set to blanks.

Report on Market Statistics

The actual data packet is sent after the report header. The message structure sent is MS_RP_MARKET_STATS. To identify the type of bhavcopy broadcast, the Message type field will be used. The value for Regular Trading Session bhavcopy, Additional Trading Session bhavcopy and final bhavcopy will be “R”, “Y” and “M” respectively. REPORT MARKET STATISTICS

Table 47 MS_RP_MARKET_STATS

Structure Name	MS_RP_MARKET_STATS		
Packet Length	488 bytes		
Transaction Code	RPRT_MARKET_STATS_OUT_RPT (1833).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40

Reserved	CHAR	1	41
NumberOfRecords	SHORT	2	42
MKT_STATS_DATA[6]	STRUCT	74	44

Table 47.A ENHNCD_MS_RP_MARKET_STATS

Structure Name	ENHNCD_MS_RP_MARKET_STATS		
Packet Length	372 bytes		
Transaction Code	ENHNCD_RPRT_MARKET_STATS_OUT_RPT (11833).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
Reserved	CHAR	1	41
NumberOfRecords	SHORT	2	42
ENHNCD_MKT_STATS_DATA [4]	STRUCT	82	44

Table 48 MKT_STATS_DATA

Structure Name	MKT_STATS_DATA		
Packet Length	74 bytes		
Field Name	Data Type	Size in Byte	Offset
CONTRACT_DESC *(Refer to Order Entry Request in Chapter 4)*	STRUCT	28	0
MarketType	SHORT	2	28
OpenPrice	LONG	4	30
HighPrice	LONG	4	34
LowPrice	LONG	4	38
ClosingPrice	LONG	4	42
TotalQuantityTraded	UNSIGNED LONG	4	46
TotalValueTraded	DOUBLE	8	50
PreviousClosePrice	LONG	4	58
OpenInterest	UNSIGNED LONG	4	62
ChgOpenInterest	LONG	4	66
Indicator	CHAR	4	70

Table 48.A ENHNCD_MKT_STATS_DATA

Structure Name	ENHNCD_MKT_STATS_DATA		
Packet Length	82 bytes		
Field Name	Data Type	Size in Byte	Offset
CONTRACT_DESC *(Refer to Order Entry Request* in Chapter 4*)*	STRUCT	28	0
MarketType	SHORT	2	28
OpenPrice	LONG	4	30
HighPrice	LONG	4	34
LowPrice	LONG	4	38
ClosingPrice	LONG	4	42
TotalQuantityTraded	UNSIGNED LONG	4	46
TotalValueTraded	DOUBLE	8	50
PreviousClosePrice	LONG	4	58
OpenInterest	LONG LONG	8	62
ChgOpenInterest	LONG LONG	8	70
Indicator	CHAR	4	78

The following table provides the details of the various fields present in the Report on market Statistics structure.

Field Name	Brief Description
TransactionCode	The transaction code is RPRT_MARKET_STATS_OUT_RPT (1833) or ENHNCD_RPRT_MARKET_STATS_OUT_RPT (11833).
MessageType	This field is set to 'R' for Regular Trading Session bhavcopy. This field is set to 'Y' for Additional Trading Session bhavcopy. This field is set to 'M' for final bhavcopy.
NumberOfRecords	This field contains the number of markets for which Market Statistics is being sent. In a packet, maximum 6 records can be packed.
Symbol	This field contains the Symbol of the security.
Series	This field contains the series of a security.
MarketType	This field contains one of the following values. • '1' for Normal market • '2' for Odd lot market • '3' for Spot market

	- '4' for Auction market
OpenPrice	This field contains the open price of a security.
HighPrice	This field 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 has been traded today.
TotalValueTraded	This field contains the total value of the securities trade.
PreviousClosePrice	This field contains the previous day's closing price.
OpenInterest	This field contains the open interest value.
ChgOpenInterest	This field contains the change in value of open interest.

MARKET INDEX REPORT

Table 49 MKT_IDX_RPT_DATA

Structure Name	MKT_IDX_RPT_DATA		
Packet Length	66 bytes		
Transaction Code	MKT_IDX_RPT_DATA (1836).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
IndexName	CHAR	15	41
MKT_INDEX	STRUCT	10	56

Table 50 MKT_INDEX

Structure Name	MKT_INDEX		
Packet Length	10 bytes		
Field Name	Data Type	Size in Byte	Offset
Opening	SHORT	2	0
High	SHORT	2	2
Low	SHORT	2	4
Closing	SHORT	2	6
Start	SHORT	2	8

The following table provides the details of the various fields present in the Market Index Report structure.

Field Name	Brief Description
TransactionCode	The transaction code is MKT_IDX_RPT_DATA (1836).
MessageType	This field is set to ‘R’.
Index_name	This field contains name of the index.
Opening	This field contains the opening index.
High	This field contains the high index value.
Low	This field contains the low index value.
Closing	This field contains the closing index
Start	This field contains the start index as 0.

INDUSTRY_INDEX_REPORT

Table 51 IND_IDX_RPT_DATA

Structure Name	IND_IDX_RPT_DATA		
Packet Length	394 bytes		
Transaction Code	IND_IDX_RPT_DATA (1837).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
Reserved	CHAR	1	41
NumberOf IndustryRecords	SHORT	2	42
INDUSTRY_INDEX[10]	STRUCT	35	44

Table 52 INDUSTRY_INDEX

Structure Name	INDUSTRY_INDEX		
Packet Length	35 bytes		
Field Name	Data Type	Size in Byte	Offset
IndustryName	CHAR	15	0
Opening	LONG	4	15
High	LONG	4	19
Low	LONG	4	23
Closing	LONG	4	27

Start	LONG	4	31

The following table provides the details of the various fields present in the Industry Index Report structure.

Field Name	Brief Description
TransactionCode	The transaction code is IND_IDX_RPT_DATA (1837).
MessageType	This field is set to 'R'.
Number of industry records	This field contains max number of industry index data.
Industry name	This field contains name of the index.
Opening	This field contains opening index.
High	This field contains the high index value.
Low	This field contains the low index value.
Closing	This field contains the closing index.
Start	This field contains the start index as 0.

SECTOR_INDEX_REPORT

Table 53 SECT_IDX_RPT_DATA

Structure Name	SECT_IDX_RPT_DATA		
Packet Length	248 bytes		
Transaction Code	SECT_IDX_RPT_DATA (1838).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
IndustryName	CHAR	15	41
NumberOf Industry Records	SHORT	2	56
INDEX_DATA[10]	STRUCT	19	58

Table 54 INDEX_DATA

Structure Name	INDEX_DATA
Packet Length	19 bytes

Field Name	Data Type	Size in Byte	Offset
SectorName	CHAR	15	0
IndexValue	LONG	4	15

The following table provides the details of the various fields present in the Sector Index Report structure.

Field Name	Brief Description
TransactionCode	The transaction code is SECT_IDX_RPT_DATA (1837).
MessageType	This field is set to 'R'.
Industry name	This field contains name of the index.
Number of industry records	This field contains maximum number of sector index data records.
Sector name	This field contains name of the index.
Index value	This field contains value of the index.

Trailer Record

After all the data packets are sent, trailer record is sent to indicate the end of Bhavcopy transmission. The message structure sent is MS_RP_TRAILER. To identify the type of bhavcopy, broadcast the Message type field will be used. The value for Regular Trading Session, Additional Trading Session and final bhavcopy will be "T", "Z" and "N" respectively .The structure is:

REPORT TRAILER

Table 55 MS_RP_TRAILER

Structure Name	MS_RP_TRAILER		
Packet Length	46 bytes		
Transaction Code	RPRT_MARKET_STATS_OUT_RPT (1833)/ ENHNCD_RPRT_MARKET_STATS_OUT_RPT (11833).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
NumberOfPackets	LONG	4	41
Reserved	CHAR	1	45

The following table provides the details of the various fields present in the Report Trailer structure.

Field Name	Brief Description
TransactionCode	The transaction code is: RPRT_MARKET_STATS_OUT_RPT (1833) or ENHNCD_RPRT_MARKET_STATS_OUT_RPT (11833).
MessageType	This is set to 'T' for Regular Trading Session. This is set to 'Z' for Additional Trading Session bhavcopy This is set to 'N' for final bhavcopy
NumberOfRecords	This contains the number of data packets sent in the bhavcopy.

Spread Bhavcopy

After completion of early bhavcopy broadcast, spread bhavcopy will be broadcasted. Initially a message will be sent in the broadcast message transcode BCAST_JRNL_VCT_MSG for the start of the spread bhavcopy. After the message, the header will be sent to indicate the start of spread bhavcopy broadcast, which will be followed by actual data packets. After the data packets the trailer record, marking the end of bhavcopy will be sent. Finally on completion of download, a message "Spread bhavcopy broadcasted successfully" will be sent in BCAST_JRNL_VCT_MSG

The structure and transcode of Spread bhavcopy is as follows:

The message will be sent in the existing structure

MS_BCAST_MESSAGE *(Refer to Broadcast, Chapter 9)*

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_JRNL_VCT_MSG (6501).

Following are the transcodes introduced for sending spread bhavcopy

Header of Report on Market Statistics

The header for the spread bhavcopy is sent before actual data packet. The message structure sent is MS_RP_HDR. To identify the type of spread bhavcopy, broadcast the Message type field will be used. The value for Regular Trading Session, Additional Trading Session and final bhavcopy will be "H", "X" and "L" respectively. The message structure sent is:

REPORT HEADER

Table 56 MS_RP_HDR

Structure Name	RP_HDR		
Packet Length	108 bytes		
Transaction Code	SPD_BC_JRNL_VCT_MSG(1862)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
org_scope	CHAR	1	41
report_date	LONG	4	42
user_type	SHORT	2	46
broker_number	CHAR	5	48
broker_name	CHAR	25	53
trader_number	LONG	4	78
trader_name	CHAR	26	82

The following table provides the details of the various fields present in the Report Header structure.

Field Name	Brief Description
Message_header	Same as in previous structure, No change in message header
Msg_type	Will contain a value 'H' for header packet.
Org_scope	
Report_date	Today's date
User_type	Will contain a value '-1'
Broker_number	Will contain a blank string.
Broker_name	Will contain a blank string.
Trader_number	Will contain zero value. Note : - Data type changed from SHORT to LONG
Trader_name	Will contain a blank string.

Report on Spread Market Statistics

The actual data packet is sent after the report header. The message structure sent is RP_SPD_MKT_STATS. To identify the type of spread bhavcopy broadcast, the Message type field will be used. The value for Regular Trading Session, Additional Trading Session and final bhavcopy will be “R”, “Y” and “M” respectively. REPORT SPREAD MARKET STATISTICS

Table 57 RP_SPD_MKT_STATS

Structure Name	RP_SPD_MKT_STATS		
Packet Length	278 bytes		
Transaction Code	SPD_BC_JRNL_VCT_MSG (1862).		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
Reserved	CHAR	1	41
NoOfRecords	SHORT	2	42
SPD_STATS_DATA[3]	STRUCT	78	44

Table 58 SPD_STATS_DATA

Structure Name	SPD_STATS_DATA		
Packet Length	78 bytes		
Field Name	Data Type	Size in Byte	Offset
MARKETTYPE	SHORT	2	0
INSTRUMENTNAME1	CHAR	6	2
SYMBOL1	CHAR	10	8
EXPIRYDATE1	LONG	4	18
STRIKEPRICE1	LONG	4	22
OPTIONTYPE1	CHAR	2	26
CALEVEL1	SHORT	2	28
INSTRUMENTNAME2	CHAR	6	30
SYMBOL2	CHAR	10	36
EXPIRYDATE2	LONG	4	46
STRIKEPRICE2	LONG	4	50
OPTIONTYPE2	CHAR	2	54
CALEVEL2	SHORT	2	56

Structure Name	SPD_STATS_DATA		
Packet Length	78 bytes		
Field Name	Data Type	Size in Byte	Offset
OPENPD	LONG	4	58
HIPD	LONG	4	62
LOWPD	LONG	4	66
LASTTRADEDPD	LONG	4	70
NOOFCONTRACTSTRADED	LONG	4	74

The following table provides the details of the various fields present in the Spread Market Statistics Report structure.

Field Name	Brief Description
TransactionCode	The transaction code is SPD_BC_JRNL_VCT_MSG (1862).
MessageType	This field is set to 'R' for Regular Trading Session bhavcopy. This field is set to 'Y' for Additional Trading Session bhavcopy. This field is set to 'M' for final bhavcopy.
NumberOfRecords	This field contains the number of markets for which Market Statistics is being sent. In a packet maximum 3 records can be packed.
MarketType	This field contains one of the following values. • '1' for Normal market • '2' for Odd lot market • '3' for Spot market • '4' for Auction market
Symbol1/Symbol2	This field contains the Symbol of the security of leg1 and leg2.
Instrument1/Instrument2	This field contains the instrument type of leg1 contract and leg 2 contract.
Expirydate1/ExpiryDate2	This field contains the Expiry date of leg 1 and leg2 contract respectively.
StrikePrice1/StrikePrice2	This field the strike price1 and strike price 2 of leg1 and leg2 of spread contract Note: Will not be used as spread for options are not allowed.
OptionType1/OptionType2	This field contains the Option type of leg1 and leg2 of spread contract Note: Will not be used as spread for options are not allowed.

Field Name	Brief Description
CALevel1/CAlevel2	This field contains the CAlevel value of leg1 and leg2 of spread contract. Note: Will not be used as spread for options are not allowed.
OpenPD	This field contains the Open Price difference of spread contract.
HiPD	This field contains the High Price difference of spread contract
LowPD	This field contains the Low price difference traded for spread contract
LastTradedPD	This field contains the value of last traded price difference of spread contract.
NoOfContractsTraded	This field contains number of contracts traded.

Trailer Record

After all the data packets are sent, trailer record is sent to indicate the end of Spread Bhavcopy transmission. The message structure sent is MS_RP_TRAILER. To identify the type of spread bhavcopy broadcast, the Message type field will be used. The value for Regular Trading Session, Additional Trading Session and final bhavcopy will be “T”, “Z” and “N” respectively. The structure is:

REPORT TRAILER

Table 59 MS_RP_TRAILER

Structure Name	MS_RP_TRAILER		
Packet Length	46 bytes		
Transaction Code	SPD_BC_JRNL_VCT_MSG(1862)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
MessageType	CHAR	1	40
NumberOfPackets	LONG	4	41
Reserved	CHAR	1	45

The following table provides the details of the various fields present in the Trailer Record structure.

Field Name	Brief Description
TransactionCode	The transaction code is: SPD_BC_JRNL_VCT_MSG (1862).
MessageType	This is set to 'T' for Regular Trading Session. This is set to 'Z' for Additional Trading Session bhavcopy This is set to 'N' for final bhavcopy
NumberOfPackets	This contains the number of data packets sent in the spread bhavcopy. Note: This is sent as 0 from host

Chapter 9 Broadcast

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 online 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
7200	Market By Order /MBP
7201	Mkt Watch
7202	Ticker
7208	Only MBP
7220	Limit Price Protection Rangesss
17201	Enhanced Mkt Watch
17202	Enhanced Ticker

LZO compression algorithm is used to compress the above specified broadcast transaction codes. The details of the LZO compression algorithm are described later.

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.

Sequential Packing

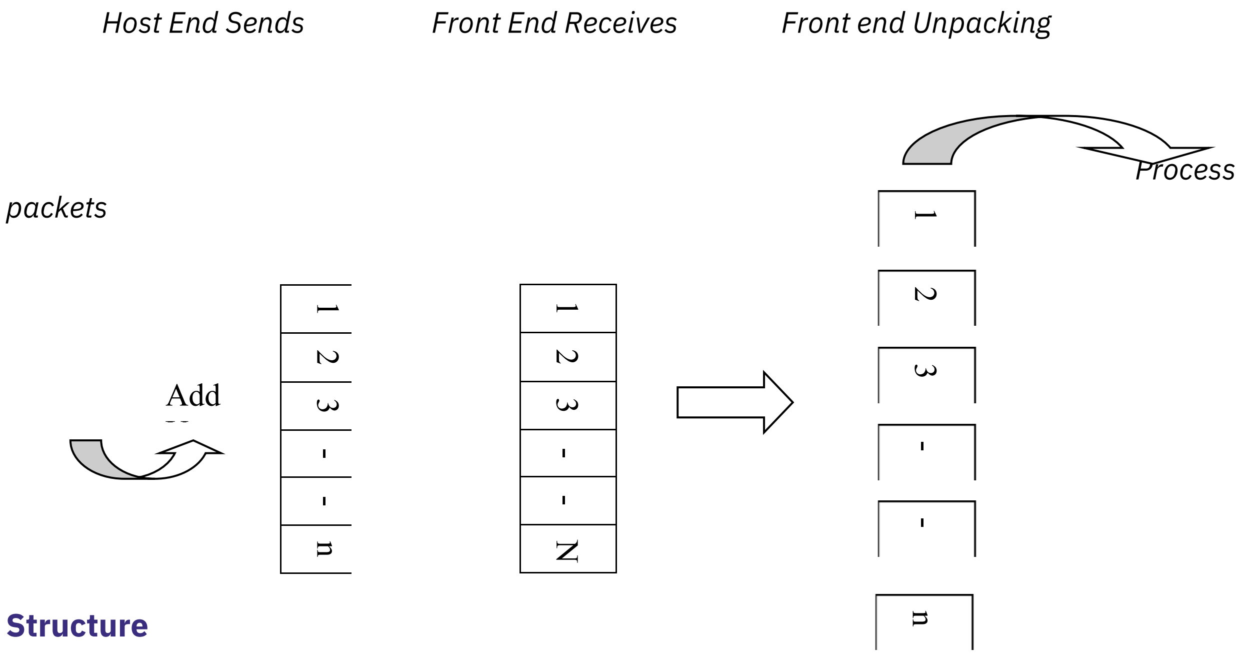
To improve the effective data transfer, the idea of sequential packing along with the LZO compression algorithm has been incorporated. At the host end, sequential packing algorithm packs the incoming data packets, which is then transmitted over the network. The data packets are packed in FIFO order.

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 unpacking 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.



Structure

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.

Pseudocode

```
struct {
SHORT iCompLen
CHAR   cCompData [MAX_MESSAGE_SIZE]
                    }BcastCmpPacket
```

Note: The above structure is currently used to interpret the incoming packets.

The iCompLen intimates us whether the packet is compressed or uncompressed. For the compressed packets (iCompLen > 0) pass the buffer to the decompression routine, else follow the uncompressed packet routing.

The packets received through the broadcast traffic have to be interpreted as follows

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

- 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*/ *broadcast header* should be
  ignored. The *message header*/ *broadcast 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.

Sample Implementation using MS-Visual Studio VC++ 6.0:

Put lzo 1.07 folder in C drive

Go to Microsoft Visual C++

Go to Tools -> Options -> Directories [ALT T O]

Set the following in the "Show directories for:"

A Include files – C:\lzo1.07

B Library files – C:\lzo1.07

C Source files – C:\lzo1.07

Go to Project->settings->Link [ALT F7]

Add lzo.lib in object / library module.

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*/ *broadcast header*. If the first byte has the value of '2', it is Futures and Options market.

The *message header*/ *broadcast header* starts from 9th byte. The remaining portion of the buffer has to be mapped as usual from the *message header*/ *broadcast header* to the following structures as specified from Chapter 4 to Chapter 11.

General Message Broadcast

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

Refer to MS_BCAST_MESSAGE in Chapter 7

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

Field Name	Description
TransactionCode	The transaction code is: BCAST_JRNL_VCT_MSG (6501).
BranchNumber	This field contains the branch number of the trader's branch. .
BrokerNumber	This field contains the Trading Member ID of the broker.

Field Name	Description
ActionCode	This field contains the action code which indicates the action taken. **Note:** For example, 'SYS' - system 'LIS' - Listing
Broadcast Destination	This field specifies the destination of the message, that is, Trader Workstation or Control Workstation.
Broadcast MessageLength	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.

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

MS_SYSTEM_INFO_DATA

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SYSTEM_INFORMATION_OUT (7206).

Change in Security Master

This is sent whenever the parameter of any security is changed. The structure is as follows:

SECURITY UPDATE INFORMATION

Table 60 MS_SECURITY_UPDATE_INFO

Structure Name	MS_SECURITY_UPDATE_INFO		
Packet Length	298 bytes		
Transaction Code	BCAST_SECURITY_MSTR_CHG (7305)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER *(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0

Structure Name	MS_SECURITY_UPDATE_INFO		
Packet Length	298 bytes		
Transaction Code	BCAST_SECURITY_MSTR_CHG (7305)		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	40
SEC_INFO *(Refer to SEC_INFO structure in Chapter 5)*	STRUCT	30	44
PermittedToTrade	SHORT	2	74
IssuedCapital	DOUBLE	8	76
WarningQuantity	LONG	4	84
FreezeQuantity	LONG	4	88
CreditRating	CHAR	12	92
ST_SEC_ELIGIBILITY_ PER_ MARKET[4]	STRUCT	3	104
IssueRate	SHORT	2	116
IssueStartDate	LONG	4	118
InterestPaymentDate	LONG	4	122
IssueMaturityDate	LONG	4	126
MarginPercentage	LONG	4	130
MinimumLotQuantity	LONG	4	134
BoardLotQuantity	LONG	4	138
TickSize	LONG	4	142
Name	CHAR	25	146
Reserved	CHAR	1	171
ListingDate	LONG	4	172
ExpulsionDate	LONG	4	176
ReAdmissionDate	LONG	4	180
RecordDate	LONG	4	184
LowPriceRange	LONG	4	188
HighPriceRange	LONG	4	192
ExpiryDate	LONG	4	196
NoDeliveryStartDate	LONG	4	200
NoDeliveryEndDate	LONG	4	204
ST_ELIGIBLITY_ INDICATORS	STRUCT	2	208
BookClosureStartDate	LONG	4	210
BookClosureEndDate	LONG	4	214
ExerciseStartDate	LONG	4	218
ExerciseEndDate	LONG	4	222
OldToken	LONG	4	226

Structure Name	MS_SECURITY_UPDATE_INFO		
Packet Length	298 bytes		
Transaction Code	BCAST_SECURITY_MSTR_CHG (7305)		
Field Name	Data Type	Size in Byte	Offset
AssetInstrument	CHAR	6	230
AssetName	CHAR	10	236
AssetToken	LONG	4	246
IntrinsicValue	LONG	4	250
ExtrinsicValue	LONG	4	254
ST_PURPOSE	STRUCT	2	258
LocalUpdateDateTime	LONG	4	260
DeleteFlag	CHAR	1	264
Remark	CHAR	25	265
BasePrice	LONG	4	290

Table 61 ST_SEC_ELIGIBILITY_PER_MARKET

Structure Name	ST_SEC_ELIGIBILITY_PER_MKT		
Packet Length	3 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Reserved	BIT	7	0
Eligibility	BIT	1	0
Status	SHORT	2	1
For Big Endian Machines			
Eligibility	BIT	1	0
Reserved	BIT	7	0
Status	SHORT	2	1

Table 62 ST_ELIGIBILITY_INDICATORS

Structure Name	ST_ELIGIBILITY_INDICATORS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Reserved	BIT	5	0
MinimumFill	BIT	1	0
AON	BIT	1	0
ParticipateInMarketIndex	BIT	1	0
Reserved	CHAR	1	1

For Big Endian Machines			
ParticipateInMarketIndex	BIT	1	0
AON	BIT	1	0
MinimumFill	BIT	1	0
Reserved	BIT	5	0
Reserved	CHAR	1	1

Table 63 ST_PURPOSE

Structure Name	ST_PURPOSE		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Exercise Style	BIT	1	0
Reserved	BIT	1	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	BIT	3	1
Is Corporate Adjusted	BIT	1	1
Is This Asset	BIT	1	1
Pl Allowed	BIT	1	1
Ex Rejection Allowed	BIT	1	1
Ex Allowed	BIT	1	1
For Big Endian Machines			
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	1	0
Exercise Style	BIT	1	0
Ex Allowed	BIT	1	1
Ex Rejection Allowed	BIT	1	1
Pl Allowed	BIT	1	1

Is This Asset	BIT	1	1
Is Corporate Adjusted	BIT	1	1
Reserved	BIT	3	1

The following table provides the details of the various fields present in the Security Master structure.

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SECURITY_MSTR_CHG (7305).
Token	This field contains the token number of the security being updated. This is unique for a particular symbol-series combination.
SecurityInformation	This contains the Symbol and Series (EQ / IL / TT) of the security.
PermittedToTrade	This field contains one of the following values. • ‘0’ - Listed but not permitted to trade. • ‘1’ - Permitted to trade.
IssuedCapital	This field contains the issue size of the security.
WarningQuantity	This field contains the warning quantity.
FreezeQuantity	This field contains the freeze quantity.
CreditRating	This field contains the credit rating of the security.
Eligibility	The flag is set to ‘1’ if the security is allowed to trade in a particular market.
Status	This field contains one of the following values. • ‘1’ - Pre-open (Only for Normal market) • ‘2’ - Open • ‘3’ - Suspended • ‘4’ - Pre-open extended • ‘5’ - Stock Open With Market • ‘6’ - Price Discovery
IssueRate	This field contains the price of the issue.
IssueStartDate	This field contains the date of issue of the security.
InterestPaymentDate	This field contains the interest payment date of the issue.

Field Name	Brief Description
IssueMaturityDate	This field contains the maturity date.
MarginPercent	This field contains the initial margin percent to be collected on a contract.
MinimumLotQuantity	This field contains the minimum lot of the order which can be placed.
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.
LowPriceRange	This field contains the lower price limit of operating ranges.
HighPriceRange	This field contains the upper price limit of operating ranges.
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	If this flag is set, the Minimum Fill attribute is allowed in orders in this security.
AON	If this flag is set, the All or None (AON) attribute is allowed in orders in this security.
ParticipantInMarket Index	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.
BookClosureEnd Date	This field contains the date when the record books in the company for shareholder names ends.
ExerciseStartDate	This field contains the starting date for Exercise.
ExerciseEndDate	This field contains the last date for Exercise.
OldToken	This field is not used.

Field Name	Brief Description
AssetInstrument	This field contains the underlying asset type, for example INDEX.
AssetName	This field contains the name of the underlying asset, for example NIFTY.
AssetToken	This field contains the token number of the asset.
IntrinsicValue	This field contains the intrinsic value of the contract.
ExtrinsicValue	This field contains the extrinsic value of the contract.
Purpose	This field contains the EX STYLE / Extraordinary General Meeting / Annual General Meeting / Interest / Bonus / Rights / Dividend flags set depending on the corporate action.
LocalUpdateDateTime	This field contains the local database update date-time.
DeleteFlag	This contains one of the following values to denote whether the security is deleted or not. • ‘N‘ – Active • ‘Y‘ – Deleted
Remark	This field contains the remarks.
BasePrice	This field contains the base price of the stock.

Periodic Broadcast for Change in Security Master

This will be periodically sent by the exchange for those securities which have had any parameters changed during the day.

The structure being sent is:

Refer to MS_SECURITY_UPDATE_INFO in Chapter 9

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SEC_MSTR_CHNG_PERIODIC (7340).

Change in Instrument Master

If a user is already logged on, and if there is any change in the data, then it is broadcast.

The structure received is as follows:

Table 64 MS_INSTRUMENT_UPDATE_INFO

Structure Name	MS_INSTRUMENT_UPDATE_INFO		
Packet Length	80 bytes		
Transaction Code	BCAST_INST_MSTR_CHG (7324)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER *(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
InstrumentId	SHORT	2	40
InstrumentName	CHAR	6	42
InstrumentDescription	CHAR	25	48
InstrumentUpdateTime	LONG	4	73
DeleteFlag	CHAR	1	77

The following table provides the details of the various fields present in the Instrument Master structure.

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_INST_MSTR_CHG (7324).
InstrumentId	This field contains the ID of the instrument.
InstrumentName	This field contains the type of the instrument. Note: For example, OPTIDX, OPTSTK,FUTIDX etc.
InstrumentDescription	This field contains the full name of the instrument. Note: For example, for Instrument Name OPTIDX, it will be OPTIONS ON INDEX.
InstrumentUpdateTime	This field contains the time when this record has been modified.
DeleteFlag	This field contains one of the following values to denote whether the instrument is deleted or not. • 'Y' for deleted • 'N' for not deleted (active)

Change Participant Status

This message is sent whenever there is any change in participants. The structure sent is as follows:

Table 65 PARTICIPANT_UPDATE_INFO

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 to Broadcast Process Header in Chapter 2)*	STRUCT	40	0
ParticipantId	CHAR	12	40
ParticipantName	CHAR	25	52
ParticipantStatus	CHAR	1	77
ParticipantUpdateDateTime	LONG	4	78
DeleteFlag	CHAR	1	82

The following table provides the details of the various fields present in the Participant Status structure.

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 has been changed.
ParticipantStatus	This field contains one of the following values to denote the status of the participant that has been changed: • ‘S‘ – Suspended • ‘A’ – Active
ParticipantUpdateDateTime	This field contains the time when the participant information was changed. It is in number of seconds from January 1, 1980
DeleteFlag	This field contains one of the following values to indicate whether the participant is deleted or not: • ‘Y’ for ‘deleted’ • ‘N’ for ‘not deleted’ Note: This field is not populated by HostEnd and it will sent as Blank

Change of Security Status

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

SECURITY STATUS UPDATE INFORMATION

Table 66 MS_SECURITY_STATUS_UPDATE_INFO

Structure Name	MS_SECURITY_STATUS_UPDATE_INFO		
Packet Length	462 bytes		
Transaction Code	BCAST_STOCK_STATUS_CHG (7320) and BCAST_STOCK_STATUS_CHG_PREOPEN (7210)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER *(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
NumberOfRecords	SHORT	2	40
TOKEN_AND_ELIGIBILITY[35]	STRUCT	420	42

Table 67 TOKEN_AND_ELIGIBILITY

Structure Name	TOKEN_AND_ELIGIBILITY		
Packet Length	12 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
ST_SEC_STATUS_PER_MARKET[4]	STRUCT	8	4

Table 68 ST_SEC_STATUS_PER_MARKET

Structure Name	ST_SEC_STATUS_PER_MARKET		
Packet Length	2 bytes		
Field Name	Data Type	Size in Byte	Offset
Status	SHORT	2	0

The following table provides the details of the various fields present in the Security Status structure.

Field Name	Brief Description
TransactionCode	The transaction codes are: BCAST_STOCK_STATUS_CHG (7320) and BCAST_STOCK_STATUS_CHG_PREOPEN (7210).
NumberOfRecords	This field contains the number of times the structure TOKEN AND ELIGIBILITY is repeated.

Field Name	Brief Description
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 any of the following values: • '1' - Pre-open • '2' - Open • '3' - Suspended • '4' - Pre-open extended • '6' - Price Discovery

Turnover Limit Exceeded or Broker Reactivated

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

Table 69 MS_BROADCAST_LIMIT_EXCEEDED

Structure Name	MS_BROADCAST_LIMIT_EXCEEDED		
Packet Length	98 bytes		
Transaction Code	BCAST_TURNOVER_EXCEEDED (9010) and BROADCAST_BROKER_REACTIVATED (9011)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
BrokerCode	CHAR	5	40
CounterBrokerCode	CHAR	5	45
WarningType	SHORT	2	50
Token	LONG	4	52
InstrumentName	CHAR	6	56
Symbol	CHAR	10	62
ExpiryDate	LONG	4	72
StrikePrice	LONG	4	76
OptionType	CHAR	2	80
CA Level	SHORT	2	82
TradeNumber	LONG	4	84

TradePrice	LONG	4	88
TradeVolume	LONG	4	92
Final	CHAR	1	96
Filler	CHAR	1	97

The following table provides the details of the various fields present in the <XYZ> structure.

Field Name	Brief Description
TransactionCode	The transaction codes are: • 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 code of broker who is about to exceed or has already exceeded his turnover limit.
CounterBrokerCode	This field is not in use.
WarningType	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. The value is set to ‘1’ if the turnover limit is about to exceed, and ‘2’ if turnover limit has been exceeded. In the latter case the broker has been deactivated.
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
InstrumentName	This field contains the instrument name
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.
ExpiryDate	This field contains the Expiry date.
StrikePrice	The field contains the strike price
OptionType	This field contains the option type.
CA Level	This field contains Corporate action level.
TradeNumber	This field is applicable only if the transaction code is BCAST_TURNOVER_EXCEEDED. This contains the trade number in which the broker has last traded.

Field Name	Brief Description
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.
Filler	This field is reserved for future use.

Change of Market Status

Sequence of the Market open messages:
Following message codes will be sent as a part of regular (first) market opening

- BC_OPEN_MSG (6511). This is sent when the market is opened.

Following message codes will still be sent, in case of Market opening for the second time during the day e.g. during sun outage, circuit hit

- BC_PRE_OR_POST_DAY_MSG (6531). This is sent when the market is preopened.
- BC_PRE_OPEN_ENDED (6571). This is sent when the pre-open period ends.
- BC_OPEN_MSG (6511). This is sent when the market is opened.

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

BCAST_VCT_MESSAGES

Table 70 MS_BCAST_VCT_MSGS

Structure Name	MS_BCAST_VCT_MSGS		
Packet Length	320 bytes		
Transaction Code	BCAST_TURNOVER_EXCEEDED (9010) and BROADCAST_BROKER_REACTIVATED (9011)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
Token	LONG	4	40

SEC_INFO *(Refer to SEC_INFO in Chapter 5)*	STRUCT	30	44
MarketType	SHORT	2	74
ST_BCAST_DESTINATION	STRUCT	2	76
BroadcastMessageLength	SHORT	2	78
BroadcastMessage	CHAR	239	80

The following table provides the details of the various fields present in the Market Status structure.

Field Name	Brief Description
TransactionCode	• BC_OPEN_MSG (6511). This is sent when the market is opened. • BC_CLOSE_MSG (6521). This is sent when the market is closed. • BC_PRE_OR_POST_DAY_MSG (6531). This is sent when the market is preopened. • BC_PRE_OPEN_ENDED (6571). This is sent when the pre-open period ends. • EQUAL BC_POSTCLOSE_MSG (6522). This is sent when the Market is in Postclose session.
SecurityInformation	This field contains the symbol and series of a security.
MarketType	This field contains the value to indicate the type of market. • '1' for Normal • '2' for Odd Lot • '3' for Spot • '4' for Auction
BroadcastDestination	This field, if set to '1', specifies that the message is for the TWS.
BroadcastMessage Length	This field contains the length of the broadcast message.
BroadcastMessage	This field contains the contents of the broadcast message.

In addition : To identify the category of the market in the message, the existing field 'AlphaChar' in the broadcast message header (BCAST_HEADER) of the message structure MS_BCAST_VCT_MSG, will contain values as indicated below

Field Name	Brief Description
AlphaChar	This field in will have information to indicate the market category Note: “TD”: Normal Market ‘EX” : Exercise “S1”: Regular category “S2”: Extended category

Ticker and Market Index

To provide co-existence for trading members, a new transcode has been provisioned that will allow the members to communicate with the exchange. This section covers the details of the new transcode as well. Members can continue to use the existing transcode and respective structures during the co-existence period. At the end of co-existence period, existing transcode and respective structures will be discontinued.

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

Table 71 MS_TICKER_TRADE_DATA

Structure Name	MS_TICKER_TRADE_DATA		
Packet Length	484 bytes		
Transaction Code	BCAST_TICKER_AND_MKT_INDEX (7202)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
Number of Records	SHORT	2	40
ST_TICKER_INDEX_INFO[17]	STRUCT	26	42

Table 71_A MS_ENHNCD_TICKER_TRADE_DATA

Structure Name	MS_ENHNCD_TICKER_TRADE_DATA		
Packet Length	498 bytes		
Transaction Code	BCAST_ENHNCD_TICKER_AND_MKT_INDEX (17202)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
Number of Records	SHORT	2	40

ST_ENHNCD_TICKER_INDEX_INFO [12]	STRUCT	38	42

Table 72 ST_TICKER_INDEX_INFO

Structure Name	ST_TICKET_INDEX_INFO		
Packet Length	26 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
OpenInterest	UNSIGNED LONG	4	14
DayHiOI	UNSIGNED LONG	4	18
DayLoOI	UNSIGNED LONG	4	22

Table 72_A ST_ENHNCD_TICKER_INDEX_INFO

Structure Name	ST_ENHNCD_TICKER_INDEX_INFO		
Packet Length	38 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
OpenInterest	LONG LONG	8	14
DayHiOI	LONG LONG	8	22
DayLoOI	LONG LONG	8	30

The following table provides the details of the various fields present in the Ticker and Market Index structure.

Field Name	Brief Description
TransactionCode	The transaction code sent is BCAST_TICKER_AND_MKT_INDEX (7202)/ BCAST_ENHNCD_TICKER_AND_MKT_INDEX (17202).
NumberOfRecords	This field contains the number of times (maximum 17 in transcode 7202 and maximum 12 in transcode 17202) the structure TICKER INDEX INFORMATION is repeated.

Field Name	Brief Description
Token	This field contains the token number, which is a unique number given to a particular symbol-series combination.
MarketType	This field contains the type of market.
FillPrice	This field contains the price at which the order has been traded.
FillVolume	This field contains the quantity of security traded.
Openinterest	This field contains the value of open interest.
DayHiOi	This field contains the feed of highest open interest value of the day.
DayLoOi	This field contains the feed of lowest open interest value of the day.

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.

BROADCAST MBO MBP

Table 73 MS_BCAST_MBO_MBP

Structure Name	MS_BCAST_MBO_MBP		
Packet Length	410 bytes		
Transaction Code	BCAST_MBO_MBP_UPDATE (7200)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
ST_INTERACTIVE_MBO_DATA	STRUCT	235	40
Record Buffer	CHAR	Sizeof(ST_MBP_INFO)*10	275
Total Buy Quantity	DOUBLE	8	375
Total Sell Quantity	DOUBLE	8	383
ST_INDICATOR	STRUCT	2	391
ClosingPrice	LONG	4	393
OpenPrice	LONG	4	397
HighPrice	LONG	4	401
LowPrice	LONG	4	405

Table 74 ST_INTERACTIVE_MBO_DATA

Structure Name	ST_INERACTIVE_MBO_DATA		
Packet Length	235 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
BookType	SHORT	2	4
TradingStatus	SHORT	2	6
VolumeTradedToday	UNSIGNED LONG	4	8
LastTradedPrice	LONG	4	12
NetChangeIndicator	CHAR	1	16
NetPriceChangeFromClosingPrice	LONG	4	17
LastTradeQuantity	LONG	4	21
LastTradeTime	LONG	4	25
AverageTradePrice	LONG	4	29
AuctionNumber	SHORT	2	33
AuctionStatus	SHORT	2	35
InitiatorType	SHORT	2	37
InitiatorPrice	LONG	4	39
InitiatorQuantity	LONG	4	43
AuctionPrice	LONG	4	47
AuctionQuantity	LONG	4	51
RecordBuffer	CHAR	Sizeof(ST_MBO_INFO)*10	55

Table 75 ST_MBO_INFO

Structure Name	ST_MBO_INFO		
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	STRUCT	2	12
Min Fill Qty	LONG	4	14

Table 76 ST_MBP_INFO

Structure Name	ST_MBP_INFO		
Packet Length	10 bytes		
Field Name	Data Type	Size in Byte	Offset
Qty	LONG	4	0
Price	LONG	4	4
NoOfOrders	SHORT	2	8

Table 77 ST_INDICATOR

Structure Name	ST_INDICATOR		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Reserved	BIT	4	0
Sell	BIT	1	0
Buy	BIT	1	0
Last Trade Less	BIT	1	0
Last Trade More	BIT	1	0
Reserved	CHAR	1	1
For Big Endian Machines			
Last Trade More	BIT	1	0
Last Trade Less	BIT	1	0
Buy	BIT	1	0
Sell	BIT	1	0
Reserved	BIT	4	0
Reserved	CHAR	1	1

Table 78 ST_MBO_MBP_TERMS

Structure Name	ST_MBO_MBP_TERMS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
For Small Endian Machines			
Reserved	BIT	6	0
AON	BIT	1	0

Structure Name	ST_MBO_MBP_TERMS		
Packet Length	2 bytes		
Field Name	Data Type	Size	Offset
MF	BIT	1	0
Reserved	CHAR	1	1
For Big Endian Machines			
MF	BIT	1	0
AON	BIT	1	0
Reserved	BIT	6	0
Reserved	CHAR	1	1

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

Field Name	Brief Description
TransactionCode	The transaction code set for the purpose is BCAST_MBO_MBP_UPDATE (7200).
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
BookType	This field contains the book type—RL / ST / OL/ SP / AU Book Type Market '1' RL '2' ST '5' Odd Lot '6' SP '7' AU Note: Process the message only if book type is '1' or '2'. Skip the message in other cases.
TradingStatus	This field contains the trading status of the security. It can be one of the following: '1' - Preopen '2' - Open '3' - Suspended '4' - Preopen Extended '6' – Price Discovery

Field Name	Brief Description
VolumeTradedToday	This field contains the total quantity of a security traded on the current day.
LastTradedPrice	s the price at which the latest trade in a security has
NetChangeIndicator	This is a flag which indicates any change of the order price from the Last Traded Price. • '+' for increase • '-' for decrease
NetPriceChangeFromthe ClosingPrice	s the net change between the closing price and the LTP. ains the closing price same as that of the Closing Price ubsequently. Since in this MBO/MBP packet, both LTP is being sent, it is for the front end to calculate the value ge from the Closing Price by the formula: ((closing price – LTP)/closing price)*100.
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	s the auction number. The maximum value this field can set to zero other than auction.
AuctionStatus	*Refer to market status in Appendix.*
InitiatorType	This field contains the initiator type—control or trader. Presently, initiator type is set to 'control', since only the Exchange can initiate an Auction. Otherwise Default value is set to Blank.
InitiatorPrice	s the price of the security of the initiator's auction order. t to zero.
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.

Field Name	Brief Description
AuctionQuantity	This field contains the quantity at which auction in a security takes place. Otherwise it is set to zero.
RecordBuffer (MBO INFORMATION)	s the five best Buy orders and five best Sell orders from
RecordBuffer (MBP INFORMATION)	This field contains the five best Buy prices and five best Sell prices from the order book.
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 are 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.

Only Market by Price Update

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

BROADCAST ONLY MBP

Table 79 MS_BCAST_ONLY_MBP

Structure Name	MS_BCAST_ONLY_MBP		
Packet Length	470 bytes		
Transaction Code	BCAST_ONLY_MBP (7208)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
NoOfRecords	SHORT	2	40
INTERACTIVE_ONLY_MBP_DATA[2]	STRUCT	214	42

Table 80 INTERACTIVE_ONLY_MBP_DATA

Structure Name	INTERACTIVE_ONLY_MBP_DATA		
Packet Length	214 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
BookType	SHORT	2	4
TradingStatus	SHORT	2	6
VolumeTradedToday	UNSIGNED LONG	4	8
LastTradedPrice	LONG	4	12
NetChangeIndicator	CHAR	1	16
VolTrdTodayExcdIndc	CHAR	1	17
NetPriceChangeFromClosingPrice	LONG	4	18
LastTradeQuantity	LONG	4	22
LastTradeTime	LONG	4	26
AverageTradePrice	LONG	4	30
AuctionNumber	SHORT	2	34
AuctionStatus	SHORT	2	36
InitiatorType	SHORT	2	38
InitiatorPrice	LONG	4	40
InitiatorQuantity	LONG	4	44
AuctionPrice	LONG	4	48
AuctionQuantity	LONG	4	52
RecordBuffer	CHAR	Sizeof(MBP_INFORMATION) *10	56
BbTotalBuyFlag	SHORT	2	176
BbTotalSellFlag	SHORT	2	178
TotalBuyQuantity	DOUBLE	8	180
TotalSellQuantity	DOUBLE	8	188
ST_INDICATOR *(Refer to ST_INDICATOR structure in Chapter 9)*	STRUCT	2	196
ClosingPrice	LONG	4	198
OpenPrice	LONG	4	202
HighPrice	LONG	4	206
LowPrice	LONG	4	210

Table 81 MBP_INFORMATION

Structure Name	MBP_INFORMATION		
Packet Length	12 bytes		
Field Name	Data Type	Size in Byte	Offset
Quantity	LONG	4	0
Price	LONG	4	4
NumberOfOrders	SHORT	2	8
BbBuySellFlag	SHORT	2	10

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

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_ONLY_MBP (7208).
NoOfRecords	This field contains the number of securities sent.
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
BookType	This field contains the book type—RL / ST / SL / NT / OL/ SP *Refer to Book Types in Appendix*
TradingStatus	This field contains the trading status of the security. It can be one of the following: '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.

Field Name	Brief Description
	Kindly refer to VolTrdTodayExcdIndc flag description for handling data greater than unsigned long max value (i.e. 4294967295).
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 is a flag which indicates any change of the order price from the LTP: • '+' 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.
VolTrdTodayExcdIndc	This is a flag which indicates whether the 'VolumeTradedToday' is exceeding unsigned long max value (i.e. 4294967295). In case this flag is set as '1', it is for the front end to calculate the value of total traded volume from the VolumeTradedToday' by the formula: VolumeTradedToday + 1 + unsigned long max value (i.e. 4294967295) In case value of flag is any value other than '1', it is to be considered that total traded volume has not exceeded the unsigned long max value (4294967295)
NetPriceChange	This field contains the net change between the closing price and the LTP. Presently, it contains the closing price same as that of the Closing Price field mentioned subsequently. Since in this MBP packet, both LTP and closing price are being sent, it is for the front end to calculate the value of Net Price Change from the Closing Price by the formula: ((closing price – LTP)/closing price)*100.

Field Name	Brief Description
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 PreOpen. Once matching starts it will contain the Average Trade Price
AuctionNumber	This field contains the auction number. Currently it is not in use.
AuctionStatus	*Refer to market status Appendix.*
InitiatorType	This field contains the initiator type— control or trader. Presently initiator type is set to 'control', since only the Exchange can initiate an auction. Otherwise, it is set to blank.
InitiatorPrice	This field contains the price of the security of the initiator's auction order. Otherwise, it is set to zero.
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 (MBP INFORMATION)	This field contains the five best Buy prices and five best Sell prices from the order book. 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.

Field Name	Brief Description
	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	This field, currently, contains a value of zero, since buy back concept is not implemented.
BbTotalsell Flag	This field, currently, contains a value of zero, since buy back concept is not implemented.
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 are 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 Indicative Open Price received. If received opening 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 Trade Price received. 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. 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.

Field Name	Brief Description
LowPrice	This field contains the lowest trade price.
MBPInformation	This field 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.

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.

To provide co-existence for trading members, a new transcode has been provisioned that will allow the members to communicate with the exchange. This section covers the details of the new transcode as well. Members can continue to use the existing transcode and respective structures during the co-existence period. At the end of co-existence period, existing transcode and respective structures will be discontinued.

The structure sent for the purpose is:

Table 82 MS_BCAST_INQ_RESP_2

Structure Name	MS_BCAST_INQ_RESP_2		
Packet Length	472 bytes		
Transaction Code	BCAST_MW_ROUND_ROBIN (7201)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
NoOfRecords	SHORT	2	40
ST_MARKET_WATCH_BCAST[5]	STRUCT	86	42

Table 82_A MS_ ENHNCD_BCAST_INQ_RESP_2

Structure Name	MS_ENHNCD_BCAST_INQ_RESP_2		
Packet Length	492 bytes		
Transaction Code	BCAST_ ENHNCD_MW_ROUND_ROBIN (17201)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
NoOfRecords	SHORT	2	40
ST_ENHNCD_MARKET_WATCH_BCAST[5]	STRUCT	90	42

Table 83 ST_MARKET_WATCH_BCAST

Structure Name	ST_MARKET_WATCH_BCAST		
Packet Length	86 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
ST_MKT_WISE_INFO [3]	STRUCT	26	4
OpenInterest	UNSIGNED LONG	4	82

Table 83_A ST_ENHNCD_MARKET_WATCH_BCAST

Structure Name	ST_ENHNCD_MARKET_WATCH_BCAST		
Packet Length	90 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
ST_MKT_WISE_INFO [3]	STRUCT	26	4
OpenInterest	LONG LONG	8	82

Table 84 ST_MKT_WISE_INFO

Structure Name	ST_MKT_WISE_INFO		
Packet Length	26 bytes		
Field Name	Data Type	Size in Byte	Offset
ST_INDICATOR *(Refer to ST_INDICATOR structure in Chapter 9)*	STRUCT	2	0
BuyVolume	LONG	4	2
BuyPrice	LONG	4	6
SellVolume	LONG	4	10
SellPrice	LONG	4	14

Structure Name	ST_MKT_WISE_INFO		
Packet Length	26 bytes		
Field Name	Data Type	Size in Byte	Offset
LastTradePrice	LONG	4	18
LastTradeTime	LONG	4	22

The following table provides the details of the various fields present in the MS_BCAST_INQ_RESP_2 / **MS_ ENHNCD_BCAST_INQ_RESP_2** structure.

Field Name	Brief Description
TransactionCode	The transaction code sent is BCAST_MW_ROUND_ROBIN (7201)/BCAST_ ENHNCD_MW_ROUND_ROBIN (17201)
NumberofRecords	This field contains the number of times the structure MARKET WATCH BROADCAST is repeated.
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
Indicator	This field contains flags which are 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.
OpenInterest	This field contains the feed of Open Interest.

Security Open Message

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

Table 85 MS_SEC_OPEN_MSGS

Structure Name	MS_SEC_OPEN_MSGS		
Packet Length	62 bytes		
Transaction Code	SECURITY_OPEN_PRICE (6013)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
Symbol	CHAR	10	40
Series	CHAR	2	50
Token	LONG	4	52
OpeningPrice	LONG	4	56
Reserved	BIT	4 (bit)	60

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

Field Name	Description
TransactionCode	The transaction code is SECURITY_OPEN_PRICE (6013).
Security Information	This field contains the symbol and series for a particular security.
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
OpeningPrice	This field contains the 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 now is 9 sec but it can be changed by the NSE control. This is just 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	Description
TransactionCode	The transaction code is BCAST_CIRCUIT_MSG (6541).

Multiple Index Broadcast

This is a multiple index broadcast. It will be coming through Cash broadcast circuit. It sends the broadcast structure as follows:

Table 86 MS_BCAST_INDICES

Structure Name	MS_BCAST_INDICES		
Packet Length	468 bytes		
Transaction Code	BCAST_INDICES (7207)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER (Refer to Broadcast Process Header in Chapter 2)	STRUCT	40	0
NumberOfRecords	SHORT	2	40
MS_INDICES [6]	STRUCT	71	42

Table 87 MS_INDICES

Structure Name	MS_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
Market Capitalisation	DOUBLE	8	61
NetChangeIndicator	CHAR	1	69
Reserved	CHAR	1	70

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

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 on this number, there will be records filled up in subsequent INDICES structure.
Indices	This is an array of structure. Number of records field shows how many records this structure will contain. The attributes of the structure are described subsequently.
IndexName	This field contains the name of the index. For example: Nifty
IndexValue	This field contains online market index value at that instance of broadcast.
HighIndexValue	This field contains the day's highest index value.
LowIndexValue	This field contains the day's lowest index value.
OpeningIndex	This field contains the opening index value when market opens.
ClosingIndex	This field, if market is open, contains the previous day's closing index. After day's batch processing is over this field contains today's closing index.
PercentChange	This field contains percentage 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 Capitalisation of securities participating in that index.
NetChangeIndicator	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

Industry Index Broadcast

It will be coming through Cash (Capital Market) broadcast circuit. It sends the Index structure as follows:

Table 88 MS_BCAST_INDUSTRY_INDICES

Structure Name	MS_BCAST_INDUSTRY_INDICES		
Packet Length	442 bytes		
Transaction Code	BCAST_INDUSTRY_INDEX_UPDATE (7203)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
NoOfRecs	SHORT	2	40
INDUSTRY_INDICES [20]	STRUCT	20	42

Table 89 INDUSTRY_INDICES

Structure Name	INDUSTRY_INDICES		
Packet Length	20 bytes		
Field Name	Data Type	Size in Byte	Offset
IndustryName	CHAR	15	0
IndexValue	LONG	4	15

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

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_INDUSTRY_INDEX_UPDATE (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 INDUSTRY_INDICES structure.
Industry	This is an array of structure. Number of records field shows how many records this structure will contain. This structure has the attributes of Industry Name and Index Value.
IndustryName	This field contains the name of the index. For example, Nifty
IndexValue	This field contains the on line market index value at the time of log-on.

Global Indices and Contracts Broadcast

The new transcode MS_GLOBAL_INDICES will be sent for the broadcast of Global Indices. The Structure for same is given as follows:

Table 90 MS_GLOBAL_INDICES

Structure Name	MS_GLOBAL_INDICES		
Packet Length	138 bytes		
Transaction Code	GI_INDICES_ASSETS (7732)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
INDEX_DETAILS	STRUCT	98	40

Table 91 INDEX_DETAILS

Structure Name	INDEX_DETAILS		
Packet Length	98 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
Name	CHAR	50	4
Open	LONG	4	54
High	LONG	4	58
Low	LONG	4	62
Last	LONG	4	66
Close	LONG	4	70
PrevClose	LONG	4	74
LifeHigh	LONG	4	78
LifeLow	LONG	4	82
filler1	LONG	4	86
filler2	LONG	4	90
filler3	LONG	4	94

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

Field Name	Brief Description
TransactionCode	The transaction code sent is GI_INDICES_ASSETS (7732).

Field Name	Brief Description
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
Name	This field contains the name of the index.
Open	This field contains the opening index value when market opens.
High	This field contains the day's highest index value.
Low	This field contains the day's lowest index value.
Last	This field contains the day's lowest index value.
Close	This field contains the day's closing index value.
PrevClose	This field contains the previous day's closing index value.
LifeHigh	This field contains the highest index in the year.
LifeLow	This field contains the lowest index in the year.

The new transcode MS_GLOBAL_CONTRACTS will be sent for the broadcast of Global Indices contracts. The Structure for same is given as follows:

Table 92 MS_GLOBAL_CONTRACTS

Structure Name	MS_GLOBAL_CONTRACTS		
Packet Length	162 bytes		
Transaction Code	GI_CONTRACT_ASSETS (7733)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
CONTRACT_DETAILS	STRUCT	122	40

Table 93 CONTRACTS_DETAILS

Structure Name	CONTRACTS_DETAILS		
Packet Length	122 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
NseSymbol	CHAR	16	4
InstrumentName	CHAR	6	20
ExpDay	SHORT	2	26
ExpMonth	SHORT	2	28
ExpYear	SHORT	2	30
OptionType	CHAR	2	32

Structure Name	CONTRACTS_DETAILS		
Packet Length	122 bytes		
Field Name	Data Type	Size in Byte	Offset
StrikePrice	LONG	4	34
BidPrice	LONG	4	38
AskPrice	LONG	4	42
BidSize	DOUBLE	8	46
AskSize	DOUBLE	8	54
Open	LONG	4	62
High	LONG	4	66
Low	LONG	4	70
Last	LONG	4	74
Close	LONG	4	78
PrevClose	LONG	4	82
LimitHigh	LONG	4	86
LimitLow	LONG	4	90
TotalTrades	DOUBLE	8	94
OpenInterest	DOUBLE	8	102
filler1	LONG	4	110
filler2	LONG	4	114
filler3	LONG	4	118

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

Field Name	Brief Description
TransactionCode	The transaction code sent is GI_CONTRACT_ASSETS (7733).
Token	This field contains the token number which is a unique number given to a particular symbol-series combination.
BidPrice	This field contains the bid price of the contract.
AskPrice	This field contains the ask price of the contract.
NseSymbol	This field contains the Symbol of the security.

Field Name	Brief Description
InstrumentName	This field contains the Instrument Type. For example – FUTIDX
ExpDay	This field contains the Expiry Day in number (1-31).
ExpMonth	This field contains the Expiry Month in number (1-12).
ExpYear	This field contains the Expiry Year in number (YYYY).
OptionType	This field contains the Option Type for the Contract.
StrikePrice	This field contains the Strike Price for the Contract.
BidSize	This field contains the bid size of the contract
AskSize	This field contains the ask size of the contract
Open	This field contains the opening traded value when market opens.
High	This field contains the day's highest traded value.
Low	This field contains the day's lowest traded value.
Last	This field contains the last traded value of the contract.
Close	This field contains the day's closing value of the contract.
PrevClose	This field contains the previous day's closing value of the contract.
LimitHigh	This field contains the high Price limit of the contract.
LimitLow	This field contains the Low Price limit of the contract.
OpenInterest	This field contains the Open interest of the contract.
TotalTrades	This field contains the total trades for the contract.

Spread Market by Price

It comes through the broadcast circuit and if broadcast is not available it comes through the interactive circuit. This is broadcast for every activity occurring. The structure is as follows:

Table 94 MS_SPD_MKT_INFO

Structure Name	MS_SPD_MKT_INFO		
Packet Length	204 bytes		
Transaction Code	BCAST_SPD_MBP_DELTA (7211)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER*(Refer to BCAST_HEADER in Chapter 2)*	STRUCT	40	0
Token1	LONG	4	40

Structure Name	MS_SPD_MKT_INFO		
Packet Length	204 bytes		
Transaction Code	BCAST_SPD_MBP_DELTA (7211)		
Field Name	Data Type	Size in Byte	Offset
Token2	LONG	4	44
MbpBuy	SHORT	2	48
MbpSell	SHORT	2	50
LastActiveTime	LONG	4	52
TradedVolume	UNSIGNED LONG	4	56
TotalTradedValue	DOUBLE	8	60
MbpBuys [5]	STRUCT	10	68
MbpSells[5]	STRUCT	10	118
TotalOrderVolume	STRUCT	16	168
OpenPriceDifference	LONG	4	184
DayHighPriceDifference	LONG	4	188
DayLowPriceDifference	LONG	4	192
LastTradedPriceDifference	LONG	4	196
LastUpdateTime	LONG	4	200

Table 95 MbpBuys

Structure Name	MbpBuys		
Packet Length	10 bytes		
Field Name	Data Type	Size in Byte	Offset
NoOrders	SHORT	2	0
Volume	LONG	4	2
Price	LONG	4	6

Table 96 MbpSells

Structure Name	MbpSells		
Packet Length	10 bytes		
Field Name	Data Type	Size in Byte	Offset
NoOrders	SHORT	2	0
Volume	LONG	4	2
Price	LONG	4	6

Table 97 TotalOrderVolume

Structure Name	TotalOrderVolume
Packet Length	16bytes

Field Name	Data Type	Size in Byte	Offset
Buy	DOUBLE	8	0
Sell	DOUBLE	8	8

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

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_SPD_MBP_DELTA (7211).
Token1	This field contains the token number of the security with early expiry date.
Token2	This field contains the token number of the security with later expiry date.
MBPbuy	This field contains the total number of buys for that particular combination.
MBPsell	This field contains the total number of sells for that particular combination.
LastActiveTime	This field contains the time stamp at which the last activity was done.
Tradedvolume	This field contains the total traded quantity of trades today.
TotalTradedValue	This field contains the total value of trades happened on that particular combination
MBPSells	This is an array of five, consisting of five best sell orders for the particular combination. It has the following fields: • NoOrders which contains the number of orders with the same price. • Volume which contains the total volume ordered with the same price. • Price which contains the price of the orders.
MBPbuys	This is an array of five, consisting of five best buy orders for the particular combination. It has the following fields: • NoOrders which contains the number of orders with the same price. • Volume which contains the total volume ordered with the same price. • Price which contains the price of the orders.
TotalOrderVolume	This structure is made of the following fields: • Buy which contains the total buy volume ordered for the particular combination.

Field Name	Brief Description
	• Sell which contains the total sell volume ordered for the particular combination.
OpenPriceDifference	This field will contain price difference of the first spread-spread trade of the day.
DayHighPriceDifference	This field will contain maximum of the price difference of spread-spread trades during the day.
DayLowPriceDifference	This field will contain minimum of the price difference of spread-spread trades during the day.
LastTradedPriceDifference	This field will contain price difference of the latest spread-spread trade.
LastUpdateTime	This field contains the time stamp at which the last activity was done. This is same as LastActiveTime.

Underlying Open Interest

This information is sent for the open interest of the underlying asset.

To provide co-existence for trading members, a new transcode has been provisioned that will allow the members to communicate with the exchange. This section covers the details of the new transcode as well. Members can continue to use the existing transcode and respective structures during the co-existence period. At the end of co-existence period, existing transcode and respective structures will be discontinued.

The structure sent is as follows:

Table 98 CM_ASSSET_OI

Structure Name	CM_ASSET_OI		
Packet Length	504 bytes		
Transaction Code	MKT_MVMT_CM_OI_IN (7130)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	2	0
Reserved	CHAR	2	2
LogTime	LONG	4	4
MarketType	CHAR	2	8
TransactionCode	SHORT	2	10
NoOfRecords	SHORT	2	12

Reserved	CHAR	8	14
TimeStamp	LONG LONG	8	22
Reserved	CHAR	8	30
MessageLength	SHORT	2	38
OPEN_INTEREST [58]	STRUCT	8	40

Table 98_A ENHNCD_CM_ASSSET_OI

Structure Name	ENHNCD_CM_ASSET_OI		
Packet Length	508 bytes		
Transaction Code	ENHNCD_MKT_MVMT_CM_OI_IN (17130)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	2	0
Reserved	CHAR	2	2
LogTime	LONG	4	4
MarketType	CHAR	2	8
TransactionCode	SHORT	2	10
NoOfRecords	SHORT	2	12
Reserved	CHAR	8	14
TimeStamp	LONG LONG	8	22
Reserved	CHAR	8	30
MessageLength	SHORT	2	38
OPEN_INTEREST [39]	STRUCT	12	40

Table 99 OPEN_INTEREST

Structure Name	OPEN_INTEREST		
Packet Length	8 bytes		
Field Name	Data Type	Size in Byte	Offset
TokenNo	LONG	4	0
CurrentOI	UNSIGNED LONG	4	4

Table 99_A ENHNCD_OPEN_INTEREST

Structure Name	ENHNCD_OPEN_INTEREST		
Packet Length	12 bytes		
Field Name	Data Type	Size in Byte	Offset
TokenNo	LONG	4	0

Structure Name	ENHNCD_OPEN_INTEREST		
Packet Length	12 bytes		
Field Name	Data Type	Size in Byte	Offset
CurrentOI	LONG LONG	8	4

Field Name	Brief Description
TransactionCode	The transaction code is MKT_MVMT_CM_OI_IN (7130)/ ENHNCD_MKT_MVMT_CM_OI_IN (17130).
LogTime	This field should be set to zero while sending messages to the host end. For messages coming from the host, this contains the time the message was generated by the trading system.
MarketType	It contains the Market Type for the transaction code MKT_MVMT_CM_OI_IN/ ENHNCD_MKT_MVMT_CM_OI_IN.
NoOfRecords	It contains the number of times (maximum 58 in transcode 7130 and maximum 39 in transcode 17130) the OPEN INTEREST is repeated.
TimeStamp	This field contains the time when the message (reply) is sent from the host.
TokenNumber	This field contains the token number of the underlying asset.
CurrentOI	This field contains the Current Open Interest of the underlying asset.

Limit Price Protection Ranges

This structure contains the Limit Price Protection range broadcast data.

Table 100
MS_BCAST_LIMIT_PRICE_PROTECTION_RANGE

Structure Name	MS_BCAST_LIMIT_PRICE_PROTECTION_RANGE		
Packet Length	344 bytes		
Transaction Code	BCAST_LIMIT_PRICE_PROTECTION_RANGE(7220)		
Field Name	Data Type	Size in Byte	Offset
BCAST_HEADER(*Refer to BCAST_HEADER in Chapter 2*))	STRUCT	40	0

LIMIT_PRICE_PROTECTION_RANGE_DATA	STRUCT	304	40

Table 101
LIMIT_PRICE_PROTECTION_RANGE_DATA

Structure Name	LIMIT_PRICE_PROTECTION_RANGE_DATA		
Packet Length	304 bytes		
Field Name	Data Type	Size in Byte	Offset
MsgCount	LONG	4	0
LIMIT_PRICE_PROTECTION_RANGE_DETAILS[25]	STRUCT	300	4

Table 102
LIMIT_PRICE_PROTECTION_RANGE_DETAILS

Structure Name	LIMIT_PRICE_PROTECTION_RANGE_DETAILS		
Packet Length	12 bytes		
Field Name	Data Type	Size in Byte	Offset
TokenNumber	LONG	4	0
HighExecBand	LONG	4	4
LowExecBand	LONG	4	8

Field Name	Brief Description
TransactionCode	The transaction code is BCAST_LIMIT_PRICE_PROTECTION_RANGE (7220)
MsgCount	This field contains the number of tokens present in that packet
TokenNumber	This field contains the token number of the contract.
HighExecBand	This field contains the high LPP band
LowExecBand	This field contains the low LPP band

Chapter 10 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.

With the removal of MD5 checksum and the authentication embedded in the new encryption library, improvement in performance as well as security is provided.

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 and 16-byte IV (Initialization Vector) value 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 11 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’

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 connection/IP, SSL) 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 and 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 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 members 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 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 Error Code 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 box connection termination.
7. Once a connection for a particular BoxID is established, all users linked with this BoxID can login using MS_SIGNON structure. Refer Chapter 3 for login request and response using MS_SIGNON structure.
8. For further flow refer to existing protocol defined in Chapter 3 of Protocol Document

Gateway Router Request

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	STRUCT	40	0

Structure Name	MS_GR_REQUEST		
Packet Length	48 bytes		
Transaction Code	GR_REQUEST (2400)		
Field Name	Data Type	Size in Byte	Offset
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

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	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

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.
Cryptographic IV (Initialization Vector)	Cryptographic IV (Initialization Vector) for both the encryption and decryption of all messages between member application and allocated Gateway Server.

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
Error Code	This field is part of Message Header. Error Code will be set if the query is unsuccessful. *Refer to List of Error Codes in Appendix*

Field Name	Brief Description
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	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_RESPONSE_OUT (23009)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
MESSAGE_HEADER	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

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	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 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

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	STRUCT	40	0
BoxId	SHORT	2	40
Reserved	CHAR	10	42

Field Name	Brief Description
Transaction Code	This field is part of Message Header. The transaction code is 23001
Error Code	This field is 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

User Log on Request

A few fields in the Logon message (transaction code 2300) have to be populated differently for direct connection:

Field Name	Description
TransactionCode	The transaction code is MS_SIGNON (2300).
Colour	All should be spaces
ShowIndex	Should be = 'T' for Trim-NNF protocol. Note: Only Trim-NNF protocol is supported by Direct Interface

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

If authentication information is correct, member systems will receive a successful MS_SIGNON structure 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 the 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.

Heart Beat

Structure Name	HEARTBEAT		
Packet Length	40 bytes		
Transaction Code	23506		
Field Name	Da ta Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to Chp 1)*	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 MSG_HEADER described in this document wherever applicable in front of business messages.

Connection Termination

When the connection is terminated by exchange, exchnage will send Sign off packet (transaction code: 20322) with appropriate error code (*Refer to List of Error Codes in Appendix*).

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	STRUCT	40	0
BoxId	SHORT	2	40

Field Name	Brief Description
Transaction Code	This field is the part of Message Header.The transaction code is BOX_SIGN_OFF (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 12 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 dissiminated 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 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 and interactive 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 incase they have got disconnected in step 1.

Message structure

Message structure is as follows:

MS_BCAST_CONT_MESSAGE

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	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 13 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

Table 110 BRANCH_ORD_VAL_LIMIT_UPDATE_REQ

Structure Name	BRANCH_ORD_VAL_LIMIT_UPDATE_REQ		
Packet Length	136 bytes		
Transaction Code	BRANCH_ORD_VAL_LIMIT_UPDATE_IN (5716)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
BrokerId	CHAR	5	40
Reserved	CHAR	25	45
BranchId	SHORT	2	70
BRANCH_LIMITS[2]	STRUCT	32	72

Table 111 BRANCH_LIMITS

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

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

Field Name	Brief Description
TransactionCode	The transaction code is

Field Name	Brief Description
	BRANCH_ORD_VAL_LIMIT_UPDATE_IN (5716)
BrokerId	This field should contain the Trading Member ID
BranchId	This field should contain the Branch number for which limit to be set
BRANCH_LIMITS	Branch limits struct is used for both Futures and Options • BRANCH_LIMITS [0] is for FUTURES • BRANCH_LIMITS [1] is for OPTIONS
BranchBuyValueLimit	This field should contain branch buy limit to be set (in lakhs) Valid values: 0 to 99999999999.99. This should be multiplied by (100000*100) before sending to the trading system.
BranchSellValueLimit	This field should contain branch sell limit to be set (in lakhs) Valid values: 0 to 99999999999.99. This should be multiplied by (100000*100) before sending to the trading system.

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 message sent will be of the following format: BRANCH_ORD_VAL_LIMIT_UPDATE_REQ *(Refer to Branch Order Value Limit Update Request in Chapter 12)*

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

Field Name	Brief Description
TransactionCode	The transaction code is BRANCH_ORD_VAL_LIMIT_UPDATE_OUT (5717)
ErrorCode	This field contains error code. If error code field value is zero (0) then reset password for user is done successfully.

If branch order value limit update request is rejected by trading system then Error response packet will be sent to user who has sent limit update request. Reason for rejection will be given by ErrorCode in the header and ErrorMessage in the packet.

The message sent will be of the following format:
MS_ERROR_RESPONSE *(Refer to Error Message in Chapter 2)*

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

Field Name	Brief Description
TransactionCode	The transaction code is BRANCH_ORD_VAL_LIMIT_UPDATE_OUT (5717)
ErrorCode	This field contains error code. Refer to List of Error Codes in Appendix.
ErrorMessage	This field contains the error message.

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

Table 112 USER_ORD_VAL_LIMIT_UPDATE_REQ

Structure Name	USER_ORD_VAL_LIMIT_UPDATE_REQ		
Packet Length	208 bytes		
Transaction Code	USER_ORD_VAL_LIMIT_UPDATE_IN (5730)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
BrokerId	CHAR	5	40
Reserved	CHAR	1	45
BranchId	SHORT	2	46
Reserved	CHAR	26	48
UserId	LONG	4	74
Reserved	CHAR	2	78

USER_LIMITS[2]	STRUCT	64	80

Table 113 USER_LIMITS

Structure Name	USER_LIMITS		
Packet Length	64 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	32	0
UserOrderBuyValueLimit	DOUBLE	8	32
UserOrderSellValueLimit	DOUBLE	8	40
Reserved	CHAR	16	48

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ORD_VAL_LIMIT_UPDATE_IN (5730)
BrokerId	This field should contain the Trading Member ID
BranchId	This field should contain the Branch ID of the user for which limit to be set.
UserId	This field should contain the User ID of the user for which limit to be set
USER_LIMITS	User limits struct is used for both Futures and Options • USER_LIMITS [0] is for FUTURES • USER_LIMITS [1] is for OPTIONS
UserOrderBuyValueLimit	This field should contain user buy limit to be set (in lakhs) Valid values: 0 to 99999999999.99 This should be multiplied by (100000*100) before sending to the trading system.
UserOrderSellValueLimit	This field should contain user sell limit to be set (in lakhs) Valid values: 0 to 99999999999.99 This should be multiplied by (100000*100) before sending to the 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 message sent will be of the following format:
USER_ORD_VAL_LIMIT_UPDATE_REQ (*Refer to User Order Value Limit Update Request in Chapter 12*)

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ORD_VAL_LIMIT_UPDATE_OUT (5731)
ErrorCode	This field contains error code. If error code field value is zero (0) then reset password for user is done successfully.

If user order value limit update request is rejected by trading system then error response packet will be sent to user who has sent limit update request. Reason for rejection will be given by Errorcode in the header ErrorMessage in the packet.

The message sent will be of the following format:
MS_ERROR_RESPONSE *(Refer to Error Message in Chapter 2)*

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ORD_VAL_LIMIT_UPDATE_OUT (5731)
ErrorCode	This field contains error code. Refer to List of Error Codes in Appendix.
ErrorMessage	This field contains the error message.

Order Limit

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

Normal Order Limit Update Request

Table 115 NORMAL_ORD_LIMIT_UPDATE_REQ

Structure Name	NORMAL_ORD_LIMIT_UPDATE_REQ		
Packet Length	66 bytes		
Transaction Code	NORMAL_ORD_LIMIT_UPDATE_IN (5732)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER*(Refer to MESSAGE_HEADER in Chapter 2)*	STRUCT	40	0
BrokerId	CHAR	5	40
Reserved	CHAR	1	45
UserId	LONG	4	46
OrderQtyLimit	DOUBLE	8	50
OrderValLimit	DOUBLE	8	58

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

Field Name	Brief Description
TransactionCode	The transaction code is NORMAL_ORD_LIMIT_UPDATE_IN (5732)
BrokerId	This field should contain the Trading Member ID
UserId	This field should contain the User ID for which limit to be set
OrderQtyLimit	This field should contain Order Quantity limit to be Set Valid values : 1 to 9999999999
OrderValLimit	This field should contain Order Value Limit to be Set (in lakhs) Valid values: 0 to 99999999.99 This should be multiplied by (100000*100) before sending to the trading system.

Normal Order Limit Update Response

On successful normal order limit updates, exchange will send Normal 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).

If normal order limit update request is rejected by trading system then Normal Order Limit Update Response packet will be sent to user who has sent limit update request. Reason for rejection will be given by ErrorCode in the header.

The message sent will be of the following format:
NORMAL_ORD_LIMIT_UPDATE_REQ (*Refer to Normal Order Limit Update Request in Chapter 12*)

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

Field Name	Brief Description
TransactionCode	The transaction code is NORMAL_ORD_LIMIT_UPDATE_OUT (5733)
ErrorCode	This field contains error code. If error code field value is zero (0) then order limit update is done successfully. If error code field value is non-zero then request for order limit update is rejected. Refer to List of Error Codes in Appendix.

Spread Order Limit Update Request

Similar to Normal order limit update, spread order limits can also be updated with below mentioned request.
NORMAL_ORD_LIMIT_UPDATE_REQ (*Refer to Normal Order Limit Update Request in Chapter 12*)

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

Field Name	Brief Description
TransactionCode	The transaction code is

Field Name	Brief Description
	SPREAD_ORD_LIMIT_UPDATE_IN (5771)
OrderQtyLimit	This field should contain Spread Order Quantity limit to be Set Valid values : 1 to 9999999999
OrderValLimit	This field should contain Spread Order Value Limit to be Set (in lakhs) Valid values: 0 to 99999999.99 This should be multiplied by (100000*100) before sending to the trading system.

Spread Order Limit Update Response

On successful spread order limit updates, exchange will send Spread 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).

If spread order limit update request is rejected by trading system then Spread Order Limit Update Response packet will be sent to user who has sent limit update request. Reason for rejection will be given by ErrorCode in the header.

The message sent will be of the following format:
NORMAL_ORD_LIMIT_UPDATE_REQ (*Refer to Normal Order Limit Update Request in Chapter 12*)

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

Field Name	Brief Description
TransactionCode	The transaction code is SPREAD_ORD_LIMIT_UPDATE_OUT (5772)
ErrorCode	This field contains error code. If error code field value is zero (0) then order limit update is done successfully. If error code field value is non-zero then request for order limit update is rejected. Refer to List of Error Codes in Appendix.

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

Request structure is mentioned as below:
MS_SIGNON (*refer to* *MS_SIGNON* *chapter 3*)

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

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_OFF_TRADER_IN (5584)
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 message sent will be of the following format:
MS_SIGNON (*refer to* *MS_SIGNON* *chapter 3*)

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

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_OFF_TRADER_OUT (5585).
ErrorCode	This field contains error code. If error code field value is zero (0) then reset password for user is done successfully.

If User Reset request is rejected by trading system then Error Response packet will be sent to user who has sent user reset request. Reason for rejection will be given by ErrorCode in the header and ErrorMessage in the packet.

The message sent will be of the following format:
MS_ERROR_RESPONSE *(Refer to Error Message in Chapter 2)*

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

Field Name	Brief Description
TransactionCode	The transaction code is SIGN_OFF_TRADER_OUT (5585).
ErrorCode	This field contains error code. Refer to List of Error Codes in Appendix.
ErrorMessage	This field contains the error message.

Reset Password

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

- The user's password will reset to "Neat@FO1" 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 password for other corporate manager.

User Password Reset Request

Table 116 RESET_USER_PASSWORD_IN_FO

Structure Name	RESET_USER_PASSWORD_IN_FO		
Packet Length	58 bytes		
Transaction Code	RESET_USER_PASSWORD_IN (5740)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	14	44

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

Field Name	Brief Description
TransactionCode	The transaction code is RESET_USER_PASSWORD_IN (5740)
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 message sent will be of the following format:
RESET_USER_PASSWORD_IN_FO (*refer to User Password Reset Request_User_Password_Reset in Chapter 12)*

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

Field Name	Brief Description
TransactionCode	The transaction code is RESET_USER_PASSWORD_OUT (5741)
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 logoff from exchange trading system.
Corporate Manager can enable/disable COL status for the users within trading member's firm.

User COL Status Update Request

Table 117 COL_USER_STATUS_CHANGE_REQ

Structure Name	COL_ USER_STATUS_CHANGE_REQ

Packet Length	52 bytes		
Transaction Code	COL_USER_STATUS_CHANGE _IN (5744)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
ColUserBit	CHAR	1	44
Reserved	CHAR	7	45

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

Field Name	Brief Description
TransactionCode	The transaction code is COL_USER_STATUS_CHANGE_IN (5744)
UserId	This field should contain user id for which COL status to be set
ColUserBit	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.)

Table 118 COL_USER_STATUS_CHANGE_RESP

Structure Name	COL_USER_STATUS_CHANGE_RESP		
Packet Length	46 bytes		
Transaction Code	COL_USER_STATUS_CHANGE _OUT (5745)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
ColUserBit	CHAR	1	44

Reserved	CHAR	1	45

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

Field Name	Brief Description
TransactionCode	The transaction code is COL_USER_STATUS_CHANGE_OUT (5745)
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.
ColUserBit	This field will contain user's COL status. 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 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_Trade_Cancellation_Request to exchange trading system.

User TRD-MOD Status Update Request

Table 119 USER_TRD_MOD/CXL_STATUS_CHG_REQ

Structure Name	USER_ TRD_MOD/CXL_STATUS_CHG_REQ		
Packet Length	52 bytes		
Transaction Code	USER_TRD_MOD/CXL_STATUS_CHG _IN (5738)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
TrdModCxlBit	CHAR	1	44
Reserved	CHAR	7	45

The following table provides the details of the various fields present in the USER_TRD_MOD/CXL_STATUS_CHG_REQ structure.

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD/CXL_STATUS_CHG_IN (5738)
UserId	This field should contain user id for which trade mod status to be set.
TrdModCxlBit	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 update, 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 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.

Table 120 USER_TRD_MOD/CXL_STATUS_CHG_RESP

Structure Name	USER_TRD_MOD/CXL_STATUS_CHG_RESP		
Packet Length	46 bytes		
Transaction Code	USER_TRD_MOD/CXL_STATUS_CHG_OUT (5739)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
TrdModCxlBit	CHAR	1	44
Reserved	CHAR	1	45

The following table provides the details of the various fields present in the USER_TRD_MOD/CXL_STATUS_CHG_RESP structure.

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD/CXL_STATUS_CHG_OUT (5739)
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 mod 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 Trade 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

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_Trade_Cancellation_Request to exchange trading system.

User TRD-CXL Status Update Request

The message sent will be of the following format:
USER_TRD_MOD/CXL_STATUS_CHG_REQ *(refer to User TRD-MOD Status Update Request chapter 12)*

The following table provides the details of the various fields present in the USER_TRD_MOD/CXL_STATUS_CHG_REQ structure.

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD/CXL_STATUS_CHG_IN (5738)
AlphaChar	To identify status change for Trade Cancellation, AlphaChar values to be set as below

	- AlphaChar[0] = 'T' - AlphaChar[1] = 'X'
UserId	This field should contain user id for which trade cxl 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 update, 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 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_STATUS_CHG_RESP *(refer to User TRD-MOD Status Update Response chapter 12)*

The following table provides the details of the various fields present in the USER_TRD_MOD/CXL_STATUS_CHG_RESP structure.

Field Name	Brief Description
TransactionCode	The transaction code is USER_TRD_MOD/CXL_STATUS_CHG_OUT (5739)
AlphaChar	To identify status change for Trade Cancellation, AlphaChar values populated will be as below. - AlphaChar[0] = 'T' - AlphaChar[1] = 'X'
ErrorCode	This field contains error code. 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.
TrdCxlBit	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 Trade 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

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

Table 121 USER_ADDR_UNLOCK_REQ_FO

Structure Name	USER_ADDR_UNLOCK_REQ_FO

Packet Length	114 bytes		
Transaction Code	USER_ADDR_UNLOCK_IN (5427)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	70	44

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_IN (5427)
UserId	This field should contain user id for which unlock request to be made.

User Unlock Request Confirmation

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.

Table 122 USER_ADDR_UNLOCK_CONFIRM_FO

Structure Name	USER_ADDR_UNLOCK_CONFIRM_FO		
Packet Length	80 bytes		
Transaction Code	USER_ADDR_UNLOCK_CONFIRM_OUT (5428)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	36	44

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_CONFIRM_OUT (5428)
ErrorCode	This field contains error code.

	If error code field value is zero (0) then unlock request for user is made to exchange successfully. If error code field value is non-zero then unlock request for user is not initiated. Refer to List of Error Codes in Appendix.

User Unlock Approve Response

On Approving the User unlock alert, trading system will send user unlock Approve Response to user who has sent user unlock request.

Table 123 USER_ADDR_UNLOCK_APPROVE_FO

Structure Name	USER_ADDR_UNLOCK_APPROVE_FO		
Packet Length	80 bytes		
Transaction Code	USER_ADDR_UNLOCK_APPROVE_OUT (5483)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
UserId	LONG	4	40
Reserved	CHAR	36	44

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_APPROVE_OUT (5483)

User Unlock Reject Response

On Rejecting the User unlock alert, trading system will send User Unlock Reject Response to user who has sent User Unlock Request.
The message sent will be of the following format:
USER_ADDR_UNLOCK_APPROVE_FO *(refer to User Unlock Approve Response chapter 12)*

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

Field Name	Brief Description
TransactionCode	The transaction code is USER_ADDR_UNLOCK_REJECT_OUT (5484)

Trading Member Level Kill Switch

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

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

Member Level Kill Switch Request

The format of the message is as follows:

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

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

Field Name	Brief Description
TransactionCode	The transaction code is KILL_SWITCH_IN (2062).
TraderId	This field should contain 0 for Trading Member level kill switch request.
TokenNumber	For cancellation of all orders, token number should be set to '-1'. For cancellation of orders on particular contract, valid token number of the contract is to be sent.
SecurityInformation (CONTRACT DESCRIPTOR)	For cancellation of orders on particular contract, this field is mandatory. This structure contains the following fields: Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA Level of the contract. CA Level should be set to zero.

Member Level Kill Switch Error Response

The Member level 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:

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

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

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

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 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 all orders should be cancelled.
TokenNumber	For cancellation of all orders, token number should be set to '-1'. For cancellation of orders on particular contract, valid token number of the contract is to be sent.
SecurityInformation (CONTRACT DESCRIPTOR)	For cancellation of all orders on particular contract, this field is mandatory. This structure contains the following fields: Instrument Name, Symbol, Expiry Date, Strike Price, Option Type and CA Level of the contract. CA Level should be set to zero.

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:

MS_OE_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 Corporate Manager and Branch Manager to place orders in the market. For Order Entry request, please refer Trimmed Order Entry Request Structure from Trimmed Structures section.

For Order Entry response, please refer Trimmed Order Entry/Mod/Cxl Response Structure from Trimmed Structures section.

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. 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.

For Order Modification request, please refer Trimmed Order Mod/Cxl Request Structure from Trimmed Structures section.

For Order Modification response, please refer Trimmed Order Entry/Mod/Cxl Response Structure from Trimmed Structures section.

Order Cancellation

The functionality enables the Corporate Manager and Branch Manager to cancel their unmatched/partially matched orders by specifying the order number. Corporate Manager can cancel his own order and also for his Branch Managers and Dealers/Traders. Branch Manager can cancel his own order and also for his Dealers/Traders.

For Order Cancellation request, please refer Trimmed Order Mod/Cxl Request Structure from Trimmed Structures section.

For Order Cancellation response, please refer Trimmed Order Entry/Mod/Cxl Response Structure from Trimmed Structures section.

Trade Modification

This functionality enables 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 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.

For 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 should be zero.
BrokerId	This field should contain the trading member ID on whose behalf the order is being placed

For Closeout order entry response, please refer Trimmed Order Entry/Mod/Cxl Response Structure from Trimmed Structures section.

Spread Order Entry

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

Please refer Spread Order Entry section (in Chapter 5) for further details.

Spread Order Modification

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

Please refer Spread Order Modification section (in Chapter 5) for further details.

Spread Order Cancellation

The functionality enables the Corporate Manager and Branch Manager to cancel their unmatched/partially matched spread orders by specifying the order number. Corporate Manager can cancel his own spread order and also for this Branch Managers and Dealers/Traders. Branch Manager can cancel his own spread order and also for his Dealers/Traders.

Please refer Spread Order Cancellation section (in Chapter 5) for further details.

2L and 3L Order Entry

This functionality enables Corporate Manager and Branch Manager to place 2L and 3L orders in the market.

For Order entry request, please refer Order Entry Request section (in Chapter 6).

For Order entry response, please refer Order Entry Response section (in Chapter 6).

Chapter 14 Give Up Trade Confirmation Messages

The Give up Approve/Reject Confirmation message is sent to NNF users when the Clearing Member of the Participant approves/rejects the participant trade.

Give up trade confirmation messages shall sent to the member till the availability of connectivity between CCs & Exchange.

The sections covered in this chapter are:

- Give Up Approve Confirmation Response to Trading Member
- Give Up Reject Confirmation Response to Trading Member

Give Up Approve Confirmation Response to Trading Member

Successful Give up Approval Confirmation is sent to the terminal of trading member who had put the participant order (buy/sell). The message sent is as follows:

Table 105 GIVEUP_RESPONSE

Structure Name	GIVEUP_RESPONSE		
Packet Length	122 bytes		
Transaction Code	GIVEUP_APP_CONFIRM_TM (4506)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER *(Refer to Message Header in Chapter 2)*	STRUCT	40	0
ReasonCode	SHORT	2	40
GIVEUP	STRUCT	79	42

Table 106 GIVEUP

Structure Name	GIVEUP		
Packet Length	79 bytes		
Field Name	Data Type	Size in Byte	Offset
OrderNumber	DOUBLE	8	0
FillNumber	LONG	4	8
InstrumentName	CHAR	6	12
Symbol	CHAR	10	18
ExpiryDate	LONG	4	28
StrikePrice	LONG	4	32
OptionType	CHAR	2	36
CALevel	SHORT	2	38

Structure Name	GIVEUP		
Packet Length	79 bytes		
Field Name	Data Type	Size in Byte	Offset
FillVolume	LONG	4	40
FillPrice	LONG	4	44
BrokerId	CHAR	5	48
Filler	CHAR	1	53
BuySell	SHORT	2	54
BookType	SHORT	2	56
LastModifiedDateTime	LONG	4	58
InitiatedByControl	CHAR	1	62
OpenClose	CHAR	1	63
ReservedFiller	CHAR	1	64
Participant	CHAR	12	65
GiveupFlag	CHAR	1	77
Deleted	CHAR	1	78

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

Field Name	Brief Description
TransactionCode	The transaction code is GIVEUP_APP_CONFIRM_TM (4506).
OrderNumber	This field will contain the Order Number for the approved Individual order.
FillNumber	This field contains the trade number.
InstrumentName	This field contains the Instrument Name identifier. Valid values are: ‘A’ -- FUTIDX ‘B’ -- FUTSTK ‘C’ -- OPTIDX ‘D’ -- OPTSTK
Symbol	This field should contain a valid Security Name. For example: “ABB”
ExpiryDate	This should contain valid Expiry Date of the contract.
StrikePrice	This field will contain a valid strike for Options Contract and for Futures Contract it will be -1.
OptionType	This field contains the OptionType identifier. Valid values are:

Field Name	Brief Description
	CE -- CALL OPTION PE -- PUT OPTION XX -- FUTURES Contract
CALevel	This field should contain the Corporate Action Level. It should be zero.
FillVolume	This field contains the quantity of security traded.
FillPrice	This field contains the price at which order has been traded.
BrokerId	This field contains the Trading Member ID.
BuySell	This field should contain one of the following values to specify whether the order is a buy or sell order: '1' denotes Buy order '2' denotes Sell order
BookType	This field contains the book type *Refer to Book Types in Appendix.*
LastModifiedDateTime	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.
InitiatedByControl	This field should contain the value Y/N based on approval initiated by Control or not. Host should send N in this field.
OpenClose	This field contains either 'O' for Open or 'C' for Close.
Participant	This field contains the participant name. For trade confirmation
GiveupFlag	This field should contain Give up flag. If giveup is approved, Host should send 'A'.
Deleted	Host should send N is this field.

Give Up Reject Confirmation Response to Trading Member

Successful Give up Reject Confirmation is also sent to the terminal of trading member who had put the participant order (buy/sell). The message sent is as follows:

Refer to GIVEUP_RESPONSE in Chapter 13

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

Field Name	Brief Description
TransactionCode	The transaction code is GIVEUP_REJ_CONFIRM_TM (4507).
GiveupFlag	This field should contain Give up flag. If giveup is rejected, Host should send ‘R’.

Chapter 15 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 MS_OE_REQUEST_TR in Appendix)*
TRIMMED_ORDER_MOD_ACK_IN TRIMMED_ORDER_CANCEL_ACK_IN	20402 20404	*Refer to MS_OM_REQUEST_TR in Appendix)*
PRICE_MOD_ACK_IN	20406	*Refer to PRICE_MOD in Appendix)*
SP_BOARD_LOT_ACK_IN TWOL_BOARD_LOT_ACK_IN THRL_BOARD_LOT_ACK_IN SP_ORDER_CANCEL_ACK_IN SP_ORDER_MOD_ACK_IN	20408 20410 20412 20414 20416	*Refer to MS_SPD_OE_REQUEST in Chapter 6)*

Immediate Ack Response

Table 102 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)/ QUICK_ACK_PM_RESP (20407)/ QUICK_ACK_SP_RESP (20409)/ QUICK_ACK_TWOL_RESP (20411)/ QUICK_ACK_THRL_RESP (20413)/ QUICK_ACK_SP_CANCEL_RESP (20415)/ QUICK_ACK_SP_MOD_RESP (20417)		
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

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)/ QUICK_ACK_PM_RESP (20407)/ QUICK_ACK_SP_RESP (20409)/ QUICK_ACK_TWOL_RESP (20411)/ QUICK_ACK_THRL_RESP (20413)/ QUICK_ACK_SP_CANCEL_RESP (20415)/ QUICK_ACK_SP_MOD_RESP (20417)		
Field Name	Data Type	Size in Byte	Offset
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), QUICK_ACK_PM_RESP (20407), QUICK_ACK_SP_RESP (20409), QUICK_ACK_TWOL_RESP (20411), QUICK_ACK_THRL_RESP (20413), QUICK_ACK_SP_CANCEL_RESP (20415), QUICK_ACK_SP_MOD_RESP (20417)
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

List of Error Codes

The error codes along with their corresponding value and description are listed in the following table.

Error Code ID	Error Code Value	Description of Error Code
INVALID_INSTRUMENT_TYPE	293	Invalid instrument type.
ORDER_NUMBER_INVALID	509	Order does not exist.
ORD_CXL_INITIATOR_AUC_NOT_ALLOWED	8049	Initiator is not allowed to cancel auction order.
AUCTION_NUMBER_INVALID	8485	Auction number does not exist
MARKET_CLOSED	16000	The trading system is not available for trading.
e$invalid_user	16001	Header user ID is not equal to user ID in the order packet.
ERROR_BAD_TRANS_CODE	16003	Invalid Transcode
E$user_already_signed_on	16004	The user is already signed on.
E$invalid_signoff	16005	System error while trying to sign-off. Please call the Exchange.
E$invalid_signon	16006	Invalid Box/User sign-on. Please try again.
e$signon_not_possible	16007	Signing onto the trading system is restricted. Please try later on.
ERR_INVALID_SYMBOL	16012	Invalid Symbol.
ERR_INVALID_ORDER_NUMBER	16013	Invalid order number
e$not_your_order	16014	This order is not yours.
E$not_your_fill	16015	This trade is not yours.
E$invalid_fill_number	16016	Invalid trade number.
E$stock_not_found	16019	Stock not found.

Error Code ID	Error Code Value	Description of Error Code
e$order_price_out_of_revised_price_range	16020	Order price is outside the revised price range
SECURITY_NOT_AVAILABLE	16035	Security is unavailable for trading at this time. Please try later.
BROKER_NOT_FOUND	16041	Trading member does not exist in the system.
USER_NOT_FOUND	16042	Dealer does not exist in the system.
DUPLICATE_RECORD	16043	This record already exists on the NEAT system.
e$order_modified	16044	Order has been modified. Please try again.
STOCK_SUSPENDED	16049	Stock is suspended.
ERR_FUNCTION_NOT_AVAILABLE	16052	Function Not Available When Preopen trade cancel request is rejected
e$change_password	16053	Your password has expired, must be changed.
ERR_INVALID_BRANCH	16054	Invalid branch for trading member.
OE_PROGRAM_ERROR	16056	Program error.
ERR_INVALID_STATUS	16063	Requested user status is active.
ERR_DATA_NOT_CHANGED	16070	If Data in the incoming packet is same as the existing data.
e$dup_trd_cxl_request	16086	Duplicate trade cancel request.
ERR_INVALID_BUYER_USER_ID	16098	Invalid trader ID for buyer.
ERR_INVALID_SELLER_USER_ID	16099	Invalid trader ID for buyer.
e$invalid_version	16100	Your system version has not been updated.
OE_SYSTEM_ERROR	16104	The system could not complete your transaction - Admin notified.

Error Code ID	Error Code Value	Description of Error Code
ERR_USER_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.
OE_ATO_IN_OPEN	16169	Order priced ATO cannot be entered when a security is open.
e$dup_request	16198	Duplicate modification or cancellation request for the same trade has been encountered.
e$only_cp_allowed	16227	Only market orders are allowed in postclose.
e$sl_mit_nt_not_allowed_pclose	16228	SL, MIT or NT orders are not allowed during Post Close.
e$gtc_gtd_ord_not_allowed_pclose	16229	GTC or GTD orders are not allowed during Post Close.
OE_CONT_MOD_NOT_ALLOWED	16230	Continuous session orders cannot be modified.
TRD_CONT_MOD_NOT_ALLOWED	16231	Continuous session trades cannot be changed.
STR_PRO_PARTIVIPANT_INVALID	16233	Proprietary requests cannot be made for participant.
ERROR_INVALID_PRICE	16247	“Invalid Price”
OE_DIFF_TRD_MOD_VOL	16251	Trade modification with different quantities is received.
ERROR_USER_NOT_EXISTS_IN_SYSTEM	16260	User does not exists in system
ERR_ALREADY_DELETED	16264	User or Branch is deleted.
RECORD_NOT_FOUND	16273	Record does not exist.
OE_MARKETS_CLOSED	16278	The markets have not been opened for trading.

Error Code ID	Error Code Value	Description of Error Code
OE_SECURITY_NOT_ADMITTED	16279	The contract has not yet been admitted for trading.
OE_SECURITY_MATURED	16280	The contract has matured.
OE_SECURITY_EXPELLED	16281	The security has been expelled.
OE_ISSUED_CAP_EXCEEDS	16282	The order quantity is greater than the issued capital.
OE_PRICE_NOT_MULT	16283	The order price is not multiple of the tick size.
OE_PRICE_EXCEEDS_DAY_MIN_MAX	16284	The order price is out of the day's price range.
OE_IS_NOT_ACTIVE	16285	The broker is not active.
e$system_wrong_state	16300	The system is in a wrong state to make the requested change.
OE_AUCTION_PENDING	16303	The auction is pending.
OE_QTY_FREEZE_CAN	16307	The order has been cancelled due to quantity freeze.
OE_PRICE_FREEZE_CAN	16308	The order has been cancelled due to price freeze.
OE_SOL_PERIOD_OVER	16311	The Solicitor period for the Auction is over.
OE_COMP_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_LIMIT_TRIGGER	16315	The limit price is worse than the trigger price.
OE_TRIGGER_PRICE_NOT_MULT	16316	The trigger price is not a multiple of tick size.
OE_NO_AON_ATTRIB	16317	AON attribute not allowed.
OE_NO_MF_ATTRIB	16318	MF attribute not allowed.

Error Code ID	Error Code Value	Description of Error Code
OE_NO_AON_IN_ATTRIB1	16319	AON attribute not allowed at Security level.
OE_NO_MF_ATTRIB1	16320	MF attribute not allowed at security level.
OE_MF_GREATER_DISC	16321	MF quantity is greater than disclosed quantity.
OE_MF_NOT_MULT	16322	MF quantity is not a multiple of regular lot.
OE_MF_GREATER_ORIGINAL	16323	MF quantity is greater than Original quantity.
OE_DISC_GREATER_ORIGINAL	16324	Disclosed quantity is greater than original quantity.
OE_DISC_NOT_MULT	16325	Disclosed quantity is not a multiple of regular lot.
OE_GTD_GREATER	16326	GTD is greater than that specified at the trading system.
OE_QUANTITY_GERATER_RL	16327	Odd lot quantity cannot be greater than or equal to regular lot size.
OE_QUANTITY_NOT_MULT_RL	16328	Quantity is not a multiple of regular lot.
OE_BROKER_NOT_PERMITTED	16329	Trading member not permitted in the market.
OE_IS_SUSPENDED	16330	Security is suspended.
OE_BRANCH_LI MIT_EXCEEDED	16333	Branch order value limit has been 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.
ERR_TRADING_NOT_ALLOWED	16348	Trading is not allowed in this market.

Error Code ID	Error Code Value	Description of Error Code
OE_NT_REJECTED	16357	Control has rejected the Negotiated Trade.
CHG_ST_EXISTS	16363	Status is in the required state.
OE_SECURITY_IN_PREOPEN	16369	Contract is in preopen.
OE_INQ_NOT_ALLOWED	16372	Order entry not allowed for the user as it is of inquiry type.
OE_SECURITY_INELIGIBLE	16387	Contract not allowed to trader in.
e$fok_order_cancelled	16388	“Order Cancelled By System” When Preopen unmatched orders are cancelled by the system after preopen session ends. When normal market unmatched orders are cancelled by the system if order collection phase is planned after circuit hit. When IOC unmatched orders are cancelled by the system.
TURNOVER_LIMIT_NOT_PROVIDED	16392	Turnover limit not provided. Please contact Exchange.
ERR_CANNOT_MOD_AUC_ORDER	16397	Cannot modify Auction orders
OE_MAX_DQ_ALLOWED	16400	DQ is less than minimum quantity allowed.
OE_ADMIN_SUSP_CAN	16404	Order has been cancelled due to freeze admin suspension.
e$invalid_buy_sell_type	16405	BUY – SELL type entered is invalid.
e$invalid_book_type	16406	BOOK type entered is invalid.
e$invalid_trigger_price	16408	trigger_price entered has invalid characters.
e$invalid_pro_client	16414	Pro/Client should be either 1 (client) or 2 (broker).

Error Code ID	Error Code Value	Description of Error Code
e$invalid_instructions	16415	Invalid combination of book type and instructions (order_type).
e$invalid_order_parameters	16416	Invalid combination of mf/aon/disclosed volume.
e$nnf_req_exceeded	16418	Number of NNF requests exceeded.
INVALID_ORDER	16419	This error code will be returned for invalid data in the order packet.
ERR_BOX_RATE_EXCEEDED_AT_MILLISECOND_LEVEL	16420	Box Rate has been exceeded by the Member at Millisecond level
e$gtd_gt_maturity	16440	GTD is greater than Maturity date.
DQ_NOT_ALLOWED_IN_PREOPEN	16441	DQ Orders are not allowed in preopen.
ST_ORD_NOT_ALLOWED_POPEN	16442	ST orders are not allowed in preopen.
e$ord_lim_exceeds_ord_val_lim	16443	Order value exceeds the order limit value.
ERR_USR_ORD_VALUE_LIMIT_EXCEEDED	16444	User Order value limit exceeded.
SL_NOT_ALLOWED	16445	Stop Loss (SL) orders are not allowed.
MIT_NOT_ALLOWED	16446	Market If Touched (MIT) orders are not allowed.
E$ord_not_allowed_in_preopen	16447	Order entry not allowed in Pre-open.
ERROR_SL_LMT_RSNBLTY_CHECK	16448	Difference between limit price and trigger price is beyond permissible range
e$not_modifiable	16514	Not modifiable.
e$tm_cm_does_not_exist	16518	Clearing member, trading member link not found.
e$not_clg_mem	16521	Not a clearing member.
e$user_not_corp_mgr	16523	The user in not a corporate manager.
e$pm_cm_invalid	16532	Clearing member participant link not found.

Error Code ID	Error Code Value	Description of Error Code
e$corp_mgr_vu_mod	16533	Enter either Trading Member or participant.
e$invalid_participant	16541	Participant is invalid.
e$trade_approved_by_cm	16550	Trade cannot be modified /cancelled. It has already been approved by CM.
e$cm_stock_suspended	16552	Stock has been suspended.
e$broker_not_permitted_in_fut	16554	Trading member not permitted in futures.
e$broker_not_permitted_in_opt	16555	Trading member not permitted in options.
e$qty_less_than_min_lot	16556	Quantity less than the minimum lot size.
e$disc_qty_less_than_min_lot	16557	Disclose quantity less than the minimum lot size.
e$mf_qty_less_than_min_lot	16558	Minimum fill is less than the minimum lot size.
e$already_rejected	16560	The give up trade has already been rejected.
e$nt_orders_not_allowed	16561	Negotiated orders not allowed.
e$nt_trade_not_allowed	16562	Negotiated trade not allowed.
e$inconsistent_broker_branch	16566	User does not belong to broker or branch.
M$post_close_start	16570	The market is in post-close.
M$post_close_ended	16571	The closing session has ended.
M$post_close_trades	16572	Closing session trades have been generated.
e$invalid_msg_length	16573	Message length is invalid.
e$invalid_open_close_type	16574	Open - Close type entered is invalid.
e$nnf_inq_req_exceeded	16576	No. of NNF inquiry requests exceeded.
e$participant_and_volume_changed	16577	Both participant and volume changed.

Error Code ID	Error Code Value	Description of Error Code
e$invalid_cover_uncover_type	16578	Cover - Uncover type entered is invalid.
e$illegal_participant	16580	Order does not belong to the given participant.
e$invalid_fill_price	16581	Invalid trade price.
e$pro_no_participant	16583	For Pro order participant entry not allowed.
e$invalid_account_no	16585	Not a valid account number.
e$allow_no_participant_order	16586	Participant order entry not allowed.
M$delete_all_orders	16589	All continuous session orders are being deleted now.
e$cum_ur_ord_val_limit_exceeded	16597	Branch limit should be greater than sum of user limits.
e$branch_ord_val_limit_exceeded	16598	Branch limit should be greater than used limit.
ERR_ORD_VAL_EXCEEDED	16600	The order value has exceeded maximum permissible limit.
ERR_PREOPEN_ORDER_REJECT	16601	Request Rejected by the exchange
e$dealer_value_limit_exceeds	16602	Dealer value limit exceeds the set limit.
e$participant_not_found	16604	Participant not found.
e$either_leg_failed	16605	One leg of spread/2L failed.
e$qty_greater_than_freeze_qty	16606	Quantity greater than Freeze quantity.
e$spread_not_allowed	16607	Spread not allowed.
e$spread_allowed_if_stock_open	16609	Spread allowed only when stock is open.
e$qty_should_be_same	16610	Both legs should have same quantity.
e$ord_mod_qty_frz_not_allowed	16611	Modified order quantity freeze not allowed.
e$trade_rec_modified	16612	The trade record has been modified.

Error Code ID	Error Code Value	Description of Error Code
e$tm_order_cant_be_modified	16615	Order cannot be modified.
e$tm_order_cant_be_cancelled	16616	Order cannot be cancelled.
e$tm_trade_cant_be_manipulated	16617	Trade cannot be manipulated.
e$cm_of_tm_suspended	16625	Clearing member is suspended.
e$expdate_not_in_ascending_ord	16626	Expiry date not in ascending order.
e$invalid_contract_comb	16627	Invalid contract combination.
e$bm_cannot_cancel_cm_orders	16628	Branch manager cannot cancel corporate manager’s order.
e$bm_cannot_cancel_bm_orders	16629	Branch manager cannot cancel other branch manager’s order.
e$cm_cannot_cancel_cm_orders	16630	Corporate manager cannot cancel other corporate manager’s order.
e$spread_in_different_underlying	16631	Spread not allowed for different underlying.
e$invalid_cli_ac	16632	Cli A/c number cannot be modified as trading member ID.
e$br_ord_limit_fut_buy_exceeded	16636	Futures buy branch order value limit has been exceeded.
e$br_ord_limit_fut_sell_exceeded	16637	Futures sell branch order value limit has been exceeded.
e$br_ord_limit_opt_buy_exceeded	16638	Options buy branch Order Value Limit has been exceeded.
e$br_ord_limit_opt_sell_exceeded	16639	Options sell branch order value limit has been exceeded.
e$ur_ord_limit_fut_buy_exceeded	16640	Futures buy used limit exceeded the user limit.
e$ur_ord_limit_fut_sell_exceeded	16641	Futures sell used limit exceeded the user limit.
e$ur_ord_limit_opt_buy_exceeded	16642	Options buy used limit exceeded the user limit.

Error Code ID	Error Code Value	Description of Error Code
e$ur_ord_limit_opt_sell_exceeded	16643	Options sell used limit exceeded the user limit.
e$cant_appr_bhav_copy_generated	16645	Cannot approve. Bhavcopy generated.
e$Collateral_Lmt_Chk	16646	Cannot modify.
e$address_not_found	16656	No address in the database.
e$stk_in_popen	16662	Contract is opening. Please wait for the contract to open.
e$invalid_nnf_field	16666	Invalid NNF field.
e$gtcgtd_not_allowed	16667	GTC GTD orders not allowed.
ERR_USER_ALREADY_SIGNED_OFF	16683	User has already signed off.
ERR_NO_PRIVILEGE	16684	User has no authority to request for change of mentioned User in actual packet.
CLOSEOUT_ORDER_REJECT	16686	This error code will be returned if Close out order rejected by the system.
CLOSEOUT_FRZ_REJECT	16687	This error code will be returned if the close out order entered is going into freeze. (Since freeze is not allowed for close out orders)
CLOSEOUT_NOT_ALLOWED	16688	This error code will be returned if the close out order is not allowed in the system.
CLOSEOUT_TRDMOD_REJECT	16690	This error code will be returned when a Trade MOD request is placed by a broker in Close-out.
PARTIAL_ORDER_REJECT	16706	Cancelled by the system.
PARTIAL_QUICK_ORDER_CXL_REJ	16708	System Error. Orders not completely cancelled by the system. Please request quick CXL again.
ERROR_INVALID_SPRD_COMBINATION	16711	Spread order entered has invalid combination

Error Code ID	Error Code Value	Description of Error Code
e$price_diff_out_of_range	16713	Price difference is beyond operating range.
RMS_REJECTED_IN_PREOPEN	16725	Order entry / Modification rejected by the Exchange.
ERROR_ALGOID_NNFID_MISMATCH_1	16730	NNF id & Algo id mismatch - Algo ID entered is 0 in order request.
ERROR_ALGOID_NNFID_MISMATCH_2	16731	NNF id & Algo id mismatch - For Non-Algo orders Algo id should be 0 (zero) in order request.
ERROR_ALGO_MKT_NOT_ALLOWED	16732	Market order not allowed for Algo order.
ERROR_INVALID_NNF_ID	16733	Invalid NNF Id
ERR_USR_NOT_FOUND_IN_NNF_FILE	16778	User is not NNF user
e$vc_order_rejected	16793	Order entered has invalid data.
e$ssd_order_rejected	16794	Order entered has invalid data.
e$order_cancelled_for_vc	16795	Order cancelled due to voluntary close out.
e$order_cancelled_for_ssd	16796	Order cancelled due to OI violation.
MSG_CODE_VOLUNTARY_CLOSE_OUT_STATUS	16797	Broker is in Voluntary Closeout.
MSG_CODE_SUSPENDED_STATUS	16798	Broker is Suspended.
e$bo_price_out_of_range	16803	Bulk order rejected due to price freeze.
e$bo_excess_quantity	16804	Bulk order rejected due to quantity Freeze.
e$user_ineligible_for_bulk_orders	16805	Trader not eligible for bulk order.
e$user_not_allowed_for_regular	16806	Trader allowed to enter only bulk order.
e$account_debarred	16807	The account is disabled from trading as per directions of SEBI/Statutory Authority.
e$account_debarred_by_pit	16816	Account is disabled for trading in the scrip during the Trading Window

Error Code ID	Error Code Value	Description of Error Code
		closure period (SEBI PIT Reg). Please contact the company for more details.
ERR_USR_ALREADY_UNLCKED	16810	User is already unlocked.
ERR_DUPLICATE_UNLCK_ALRT	16811	User unlock request is already present for requested user.
ERR_ACTV_NUM_OF_USRS_IN_BRNCH_EXCEEDED	17022	Active number of users in branch exceeded
EC_TRD_MOD_REJ_CLI_CP_MOD_NOT_ALLOWED	17039	Client code/Participant modification not allowed
ERROR_QUANTITY_LIM_EXCEEDS_QTY_VAL_LIM	17045	Order quantity exceeds quantity value limit for the user.
USER_TRD_MOD_DISABLED	17046	Trade modification not allowed for the user.
PREOPEN_TRADE_CANCELLATION_NOT_ALLOWED	16055	Trade executed during pre-open session not allowed to cancel.
ERR_DEPNDENT_SESSN_NOT_ACTIVE	17063	Dependent session is not active
~~e$trd_price_out_of_stock_tpp~~ e$trd_price_out_of_stock_lpp	17070	The Price is out of the current ~~execution~~ LPP range
e$order_cancelled_for_self_trade	17071	The order could have resulted in self trade
e$invalid_packet	17101	The packet has invalid data
ssssse$hearbeat_not_received	17102	Heartbeat not received
e$Invalid_box_id	17104	Invalid box id
e$seq_no_mismatch	17105	Sequence number mismatch
e$box_rate_exceeded	17106	Box Rate has been exceeded by the Member
ERROR_HB_RATE_EXCEEDED	17107	Heart beat rate exceeded by the member
e$max_user_count_exceeded	17142	Maximum user login allowed per box has been exceeded
e$invalid_box_ip_combination	16403	Login from invalid IP

Error Code ID	Error Code Value	Description of Error Code
ERR_INVALID_PAN_ID	17177	Invalid PAN Id
ERR_INVALID_ALGO_ID	17179	Invalid Algo Id
ERR_INVALID_VALUE_IN_RESERVED	17180	Invalid value in the Reserved Field
ERR_ ALGO_ID_DISABLED	17185	Order rejected as Algo ID is disabled by the Exchange
ERR_ORDER_CANCELLED_ ALGOID_DISABLED	17186	Order cancelled as Algo ID is disabled by the Exchange
ERR_CHECKSUM_FAILED_GR	19028	Checksum verification failed at Gateway Router
ERR_MULTIPLE_GR_QUERY_RCV	19029	Multiple GR_QUERY request received
ERR_MKT_ORDER_NOT_ALLOWED	17181	Contract not traded. Market order not allowed
ERR_TRADE_BEYOND_MARKUP_PRICE	17182	Order could have resulted in trade beyond mark-up price
ERR_ENCRYPTION_FLAG_MISMATCH	19030	Encryption Flag Mismatch
ERR_MD5_CHECKSUM_FAILURE	19031	MD5 Checksum Failed
ERR_USER_HAVING_NULL_RIGHTS	17184	Order Rejected as user has NO trading rights

Reason Codes

The reason codes and the corresponding values are listed in the following table.

Reason Code	Value
Exercise	2
Position liquidation	3
Security	5
Broker	6
Branch	7

User	8
Participant	9
Counter Party	10
Order Number	11
Market Status	12
Auction Number	15
Order Type	16
Price Freeze	17
Quantity Freeze	18
Contract	20
Exercise Mode Mismatch	30

List of Transaction Codes

The transaction codes and the corresponding structure are listed in the following table.

Transaction Code	Code	Structure	Size	I/B*
SYSTEM_INFORMATION_IN	1600	MS_SYSTEM_INFO_REQ	44	I
SYSTEM_INFORMATION_OUT	1601	MS_SYSTEM_INFO_DATA	106	I
EXCH_PORTF_IN	1775	EXCH_PORTFOLIO_REQ	44	I
EXCH_PORTF_OUT	1776	EXCH_PORTFOLIO_RESP	344	I
RPRT_MARKET_STATS_OUT_RPT	1833	MS_RP_MARKET_STATS REPORT_TRAILER REPORT_HEADER	488 48 108	B
ENHNCD_RPRT_MARKET_STATS_OUT_RPT	11833	ENHNCD_MS_RP_MARKET_STATS REPORT_TRAILER REPORT_HEADER	372 48 108	B
SPD_MKT_STATS_RPT_DATA	1862	RP_SPD_MKT_STATS	104	B
BOARD_LOT_IN	2000	MS_OE_REQUEST	316	I
NEG_ORDER_TO_BL	2008	MS_OE_REQUEST	316	I

Transaction Code	Code	Structure	Size	I/B*
NEG_ORDER_BY_CPID	2009	MS_OE_REQUEST	316	B
ORDER_MOD_IN	2040	MS_OE_REQUEST	316	I
ORDER_MOD_REJECT	2042	MS_OE_REQUEST	316	I
ORDER_CANCEL_IN	2070	MS_OE_REQUEST	316	I
ORDER_CANCEL_REJECT	2072	MS_OE_REQUEST	316	I
ORDER_CONFIRMATION	2073	MS_OE_REQUEST	316	I
ORDER_MOD_CONFIRMATION	2074	MS_OE_REQUEST	316	I
ORDER_CANCEL_CONFIRMATION	2075	MS_OE_REQUEST	316	I
PRICE_MOD_IN PRICE_MOD_ACK_IN	2013 20406	PRICE_MOD	106	I
CANCEL_NEG_ORDER	2076	MS_OE_REQUEST	316	I
SP_BOARD_LOT_IN SP_BOARD_LOT_ACK_IN	2100 20408	MS_SPD_OE_REQUEST	480	I
TWOL_BOARD_LOT_IN TWOL_BOARD_LOT_ACK_IN	2102 20410	MS_SPD_OE_REQUEST	480	I
THRL_BOARD_LOT_IN THRL_BOARD_LOT_ACK_IN	2104 20412	MS_SPD_OE_REQUEST	480	I
SP_ORDER_CANCEL_IN SP_ORDER_CANCEL_ACK_IN	2106 20414	MS_SPD_OE_REQUEST	480	I
SP_ORDER_MOD_IN SP_ORDER_MOD_ACK_IN	2118 20416	MS_SPD_OE_REQUEST	480	I
SP_ORDER_CONFIRMATION	2124	MS_SPD_OE_REQUEST	480	I
TWOL_ORDER_CONFIRMATION	2125	MS_SPD_OE_REQUEST	480	I
THRL_ORDER_CONFIRMATION	2126	MS_SPD_OE_REQUEST	480	I
SP_ORDER_CXL_REJ_OUT	2127	MS_SPD_OE_REQUEST	480	I
SP_ORDER_CXL_CONFIRMATION	2130	MS_SPD_OE_REQUEST	480	I
TWOL_ORDER_CXL_CONFIRMATION	2131	MS_SPD_OE_REQUEST	480	I

Transaction Code	Code	Structure	Size	I/B*
THRL_ORDER_CXL_CONFIRMATION	2132	MS_SPD_OE_REQUEST	480	I
SP_ORDER_MOD_REJ_OUT	2133	MS_SPD_OE_REQUEST	480	I
SP_ORDER_MOD_CON_OUT	2136	MS_SPD_OE_REQUEST	480	I
TWOL_ORDER_ERROR	2155	MS_SPD_OE_REQUEST	480	I
THRL_ORDER_ERROR	2156	MS_SPD_OE_REQUEST	480	I
FREEZE_TO_CONTROL	2170	MS_OE_REQUEST	316	I
ON_STOP_NOTIFICATION	2212	MS_TRADE_CONFIRM	296	I
TRADE_CONFIRMATION	2222	MS_TRADE_CONFIRM	296	I
TRADE_ERROR	2223	MS_TRADE_INQ_DATA	234	I
ORDER_ERROR	2231	MS_OE_REQUEST	316	I
TRADE_CANCEL_CONFIRM	2282	MS_TRADE_CONFIRM	296	I
TRADE_CANCEL_REJECT	2286	MS_TRADE_CONFIRM	296	I
TRADE_MODIFY_CONFIRM	2287	MS_TRADE_MODIFY_CONFIRM	296	I
TRADE_MODIFY_REJECT	2288	MS_TRADE_CONFIRM	296	I
SIGN_ON_REQUEST_IN	2300	MS_SIGNON	278	I
SIGN_ON_REQUEST_OUT	2301	MS_SIGNON MS_ERROR_RESPONSE	278 182	I
ERROR_RESPONSE_OUT	2302	MS_ERROR_RESPONSE	182	I
SIGN_OFF_REQUEST_OUT	2321	SIGNOFF OUT	190	I
GR_REQUEST	2400	MS_GR_REQUEST	48	I
GR_RESPONSE	2401	MS_GR_RESPONSE	124 136	I
GIVEUP_APP_CONFIRM_TM	4506	GIVEUP_RESPONSE	122	I
GIVEUP_REJ_CONFIRM_TM	4507	GIVEUP_RESPONSE	122	I
BCAST_CONT_MSG	5294	MS_BCAST_CONT_MESSAGE	244	B
CTRL_MSG_TO_TRADER	5295	MS_TRADER_INT_MSG	290	B
USER_ADDR_UNLOCK_IN	5427	USER_ADDR_UNLOCK_REQ_FO	114	I

Transaction Code	Code	Structure	Size	I/B*
USER_ADDR_UNLOCK_CONFIRM_OUT	5428	USER_ADDR_UNLOCK_CONFIRM_FO	322	I
TRADE_CANCEL_IN	5440	MS_TRADE_INQ_DATA	234	I
TRADE_CANCEL_OUT	5441	MS_TRADE_INQ_DATA	234	I
TRADE_MOD_IN	5445	MS_TRADE_INQ_DATA	234	I
USER_ADDR_UNLOCK_APPROVE_OUT	5483	USER_ADDR_UNLOCK_APPROVE_FO	76	I
USER_ADDR_UNLOCK_REJECT_OUT	5484	USER_ADDR_UNLOCK_APPROVE_FO	76	I
SIGN_OFF_TRADER_IN	5584	MS_SIGNON	278	I
SIGN_OFF_TRADER_OUT	5585	MS_SIGNON MS_ERROR_RESPONSE	278 182	I
BRANCH_ORD_VAL_LIMIT_UPDATE_IN	5716	BRANCH_ORD_VAL_LIMIT_UPDATE_REQ	136	I
BRANCH_ORD_VAL_LIMIT_UPDATE_OUT	5717	BRANCH_ORD_VAL_LIMIT_UPDATE_REQ MS_ERROR_RESPONSE	136 182	I
USER_ORD_VAL_LIMIT_UPDATE_IN	5730	USER_ORD_VAL_LIMIT_UPDATE_REQ	208	I
USER_ORD_VAL_LIMIT_UPDATE_OUT	5731	USER_ORD_VAL_LIMIT_UPDATE_REQ MS_ERROR_RESPONSE	208 182	I
NORMAL_ORD_LIMIT_UPDATE_IN	5732	NORMAL_ORD_LIMIT_UPDATE_REQ	66	I
NORMAL_ORD_LIMIT_UPDATE_OUT	5733	NORMAL_ORD_LIMIT_UPDATE_REQ	66	I
USER_TRD_MOD/CXL_STATUS_CHG_IN	5738	USER_TRD_MOD/CXL_STATUS_CHG_REQ	52	I
USER_TRD_MOD/CXL_STATUS_CHG_OUT	5739	USER_TRD_MOD/CXL_STATUS_CHG_RESP	46	I

Transaction Code	Code	Structure	Size	I/B*
RESET_USER_PASSWORD_IN	5740	RESET_USER_PASSWORD_IN_FO	58	I
RESET_USER_PASSWORD_OUT	5741	RESET_USER_PASSWORD_IN_FO	58	I
COL_USER_STATUS_CHANGE_IN	5744	COL_USER_STATUS_CHANGE_REQ	52	I
COL_USER_STATUS_CHANGE_OUT	5745	COL_USER_STATUS_CHANGE_RESP	46	I
SPREAD_ORD_LIMIT_UPDATE_IN	5771	NORMAL_ORD_LIMIT_UPDATE_REQ	66	I
SPREAD_ORD_LIMIT_UPDATE_OUT	5772	NORMAL_ORD_LIMIT_UPDATE_REQ	66	I
SECURITY_OPEN_PRICE	6013	MS_SEC_OPEN_MSGS	62	B
BCAST_JRNL_VCT_MSG	6501	MS_TRADER_INT_MSG	290	B
BC_OPEN_MESSAGE	6511	MS_BCAST_VCT_MSGS	320	B
BC_CLOSE_MESSAGE	6521	MS_BCAST_VCT_MSGS	320	B
BC_PREOPEN_SHUTDOWN_MSG	6531	MS_BCAST_VCT_MSGS	320	B
BC_CIRCUIT_CHECK	6541	MESSAGE_HEADER	40	B
BC_NORMAL_MKT_PREOPEN_ENDED	6571	MS_BCAST_VCT_MSGS	320	B
DOWNLOAD_REQUEST	7000	MS_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
MKT_MVMT_CM_OI_IN	7130	CM_ASSET_OI	504	B
ENHNCD_MKT_MVMT_CM_OI_IN	17130	ENHNCD_CM_ASSET_OI	508	B
BCAST_MBO_MBP_UPDATE	7200	MS_BCAST_MBO_MBP	410	B
BCAST_MW_ROUND_ROBIN	7201	MS_FO_BCAST_INQ_RESP_2	472	B
BCAST_ENHNCD_MW_ROUND_ROBIN	17201	MS_ENHNCD_BCAST_INQ_RESP_2	492	B

Transaction Code	Code	Structure	Size	I/B*
BCAST_TICKER_AND_MKT_INDEX	7202	MS_FO_TICKER_TRADE_DATA	484	B
BCAST_ENHNCD_TICKER_AND_MKT_INDEX	17202	MS_ENHNCD_TICKER_TRADE_DATA	492	B
BCAST_INDUSTRY_INDEX_UPDATE	7203	MS_BCAST_INDUSTRY_INDICES	442	B
BCAST_SYSTEM_INFORMATION_OUT	7206	MS_SYSTEM_INFO_DATA	106	B
BCAST_ONLY_MBP	7208	MS_BCAST_ONLY_MBP	470	B
BCAST_SECURITY_STATUS_CHG_PREOPEN	7210	MS_SECURITY_STATUS_UPDATE_INFO	462	B
BCAST_SPD_MBP_DELTA	7211	MS_SPD_MKT_INFO	204	B
BCAST_LIMIT_PRICE_PROTECTION_RANGE	7220	MS_BCAST_LIMIT_PRICE_PROTECTION_RANGE	344	B
UPDATE_LOCALDB_IN	7300	MS_UPDATE_LOCAL_DATABASE	82	I
UPDATE_LOCALDB_DATA	7304	Packet of size >40 and <=548	80-548	I
BCAST_SECURITY_MSTR_CHG	7305	MS_SECURITY_UPDATE_INFO	298	B/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_HEADER	42	I
BCAST_SECURITY_STATUS_CHG	7320	MS_SECURITY_STATUS_UPDATE_INFO	462	B
PARTIAL_SYSTEM_INFORMATION	7321	MS_SYSTEM_INFO_DATA	106	I
BCAST_INSTR_MSTR_CHG	7324	MS_INSTRUMENT_UPDATE_INFO	80	I/B
BCAST_INDEX_MSTR_CHG	7325	MS_DOWNLOAD_INDEX	450	I
BCAST_INDEX_MAP_TABLE	7326	MS_DOWNLOAD_INDEX_MAP	462	I
BCAST_SEC_MSTR_CHNG_PERIODIC	7340	MS_SECURITY_UPDATE_INFO	298	B

Transaction Code	Code	Structure	Size	I/B*
BCAST_SPD_MSTR_CHG_PERIODIC	7341	MS_SPD_UPDATE_INFO	132	B
BATCH_ORDER_CANCEL	9002	MS_OE_REQUEST	316	I
BCAST_TURNOVER_EXCEEDED	9010	MS_BROADCAST_TLIMIT_EXCEEDED	98	B
BROADCAST_BROKER_REACTIVATED	9011	MS_BROADCAST_TLIMIT_EXCEEDED	98	B
BOARD_LOT_IN_TR TRIMMED_BOARD_LOT_ACK_IN	20000 20400	MS_OE_REQUEST_TR	158	I
ORDER_MOD_IN_TR TRIMMED_ORDER_MOD_ACK_IN	20040 20402	MS_OM_REQUEST_TR	186	I
ORDER_CANCEL_IN_TR TRIMMED_ORDER_CANCEL_ACK_IN	20070 20404	MS_OM_REQUEST_TR	186	I
~~ORDER_QUICK_CANCEL_IN_TR~~	~~20060~~	~~MS_OM_REQUEST_TR~~	~~186~~	~~I~~
ORDER_CONFIRMATION_TR	20073	MS_OE_RESPONSE_TR	240	I
TXN_EXT_QUICK_ACK_OE_RESP TXN_EXT_QUICK_ACK_OM_RESP TXN_EXT_QUICK_ACK_OC_RESP TXN_EXT_QUICK_ACK_PM_RESP TXN_EXT_QUICK_ACK_SP_RESP TXN_EXT_QUICK_ACK_TWOL_RESP TXN_EXT_QUICK_ACK_THRL_RESP TXN_EXT_QUICK_ACK_SP_CANCEL_RESP TXN_EXT_QUICK_ACK_SP_MOD_RESP	20401 20403 20405 20407 20409 20411 20413 20415 20417	MS_ACK_RESPONSE	22	I
ORDER_MOD_CONFIRMATION_TR	20074	MS_OE_RESPONSE_TR	240	I

Transaction Code	Code	Structure	Size	I/B*
ORDER_CXL_CONFIRMATION_TR	20075	MS_OE_RESPONSE_TR	240	I
TRADE_CONFIRMATION_TR	20222	MS_TRADE_CONFIRM_TR	230	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	54	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

*** I/B - 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
SP_ORDER_CONFIRMATION	2124
TWOL_ORDER_CONFIRMATION	2125
THRL_ORDER_CONFIRMATION	2126
SP_ORDER_CXL_REJ_OUT	2127
SP_ORDER_CXL_CONFIRMATION	2130
TWOL_ORDER_CXL_CONFIRMATION	2131
THRL_ORDER_CXL_CONFIRMATION	2132

Transaction Code	Code
SP_ORDER_MOD_REJ_OUT	2133
SP_ORDER_MOD_CON_OUT	2136
SP_ORDER_ERROR	2154
TWOL_ORDER_ERROR	2155
THRL_ORDER_ERROR	2156
FREEZE_TO_CONTROL	2170
ON_STOP_NOTIFICATION	2212
TRADE_CONFIRMATION	2222
ORDER_ERROR	2231
BATCH_ORDER_CANCEL	9002
BATCH_SPREAD_CXL_OUT	9004
ORDER_CONFIRMATION_TR	20073
ORDER_MOD_CONFIRMATION_TR	20074
ORDER_CXL_CONFIRMATION_TR	20075
TRADE_CONFIRMATION_TR	20222

Quick Reference for Order Entry Parameters

The order flags are as follows:

Order Terms:

Order Flags	Input/Output
MF	Input, to be set when the min fill quantity is given
AON	Input
IOC	Input
GTC	Input
Day	Input
MIT	Input
SL	Input
Market	Output
ATO	Output

Frozen	Output
Modified	Output
Traded	Output
MatchedInd	Output

Status	Market	Book Type	Order Terms and Other Characteristic Fields
Preopen	Normal Market	RL**	(non-zero value of GoodTillDate)/DAY/GTC mandatory, mutually exclusive, input Market order is placed then ATO bit is set to ‘1’b
Open	Normal Market	RL**	(non-zero value of GoodTillDate)/DAY/ GTC/ IOC mandatory, mutually exclusive, input MKT output, set if Market order
Open	Normal Market	SL**	SL mandatory, input (non-zero value of GoodTillDate) /DAY/ GTC/ IOC mandatory, mutually exclusive, input MF/ AON mutually exclusive, input MKT output, set if Market order Trigger Price is mandatory
Open	Normal Market	ST**	(non-zero value of GoodTillDate)/DAY/ GTC/ IOC mandatory, mutually exclusive, input MF/ AON mandatory, mutually exclusive, input MKT output, set if Market order
Postclose	Normal Market	RL/ST	DAY / IOC mandatory, mutually exclusive, input DQ / MF / AON mutually exclusive, input Market order is mandatory
Close			Order entry is not allowed

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

Note: Order requested message is stopped for the following transcodes (Both interactive and journal download).

- BOARD_LOT_OUT
- SPOT_OUT
- NEGOTIATED_OUT
- STO_OUT
- ODD_LOT_OUT
- ON_STOP_OUT
- SP_BOARD_LOT_OUT
- TWOL_BOARD_LOT_OUT
- THRL_BOARD_LOT_OUT
- ORDER_MOD_OUT
- ORDER_CANCEL_OUT
- SP_ORDER_CANCEL_OUT
- SP_ORDER_MOD_OUT
- TWOL_ORDER_CANCEL_OUT
- TWOL_ORDER_MOD_OUT
- THRL_ORDER_CANCEL_OUT
- THRL_ORDER_MOD_OUT

Market Types

The market types are listed in the following table.

Market Type ID	Status
1	Normal Market
2	Odd Lot Market (Not used)
3	Spot Market (Not used)
4	Auction Market (Not used)

Market Status

The market can be one of the statuses listed in the following table.

Market Status ID	Status
0	PreOpen (Only for Normal Market)
1	Open
2	Closed
3	PreOpen Ended
4	Postclose

Book Types

There are seven books listed in the following table and these books fall under any one of the four market types.

Book ID	Book Type	Market Type
1	Regular lot order	Normal Market
2	Special terms order	Normal Market
3	Stop loss / MIT order	Normal Market
4	Negotiated order (Not used)	Normal Market
5	Odd lot order (Not used)	Odd Lot Market
6	Spot order (Not used)	Spot Market
7	Auction order (Not used)	Auction Market

Security Status

The security status is listed in the following table.

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

Activity Types

The activity types that are sent in the reports along with their description and code are listed in the following table.

Activity Type	Description	Code
ORIGINAL_ORDER	When the order is entered it is taken as original order. GTC/GTD orders still	1

	in the book also come with this activity type.	
ACTIVITY_TRADE	The trade done.	2
ACTIVITY_ORDER_CXL	The order is cancelled.	3
ACTIVITY_ORDER_MOD	The order is modified.	4
ACTIVITY_TRADE_MOD	The trade is modified.	5
ACTIVITY_TRADE_CXL_1	The trade cancellation was requested.	6
ACTIVITY_TRADE_CXL_2	Action has been taken on this request.	7
ACTIVITY_BATCH_ORDER_CXL	At the end of the day all un traded Day orders are cancelled. GTC/GTD orders due for cancellation are also cancelled.	8
ACTIVITY_ORDER_MOD_REJECT	When the order modification is rejected.	9
ACTIVITY_TRADE_MOD_REJECT	When the trade modification is rejected.	10
ACTIVITY_TRADE_CXL_REJECT	When the trade cancellation is rejected.	11
ACTIVITY_ORDER_REJECTED	When the order entry is rejected.	12
ACTIVITY_ORDER_IN_BOOK		13
ACTIVITY_ORDER_CXL_REJECT	When order cancel requested, gets rejected.	14
ACTIVITY_PRICE_FREEZE_IN	Order entered, caused price freeze.	15
ACTIVITY_PRICE_FREEZE_CXLD	Order in price freeze is cancelled from CWS.	16
ACTIVITY_FREEZE_ADMIN_SUSP	Order is rejected through admin suspension when quantity is freezed.	17
ACTIVITY_QTY_FREEZE_IN	Order entered, caused quantity freeze.	18
ACTIVITY_QTY_FREEZE_CXLD	Order in quantity freeze is cancelled from CWS.	19
ACTIVITY_ORD_BROKER_SUSP	When order is cancelled because of broker suspension.	20

ACTIVITY_SPREAD_TRADE_CXL	When spread trade is cancelled.	43

Pipe Delimited File Structures

A new field category indicator has been introduced in contract.txt which will specify the category of market hours in which the contract is available to trade. Previously Reserved 3 byte (after OptionType field) was used to add this field. The changes for same are highlighted in yellow.

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.

Contract File Structure

HEADER

Table 126 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 127 STOCK_STRUCTURE

Structure Name	STOCK_STRUCTURE		
Packet Length	304 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

Structure Name	STOCK_STRUCTURE		
Packet Length	304 bytes		
Field Name	Data Type	Size in Byte	Offset
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
Reserved	CHAR	1	44
Category	CHAR	1	45
Reserved	CHAR	1	46
CALevel	SHORT	2	47
Reserved	CHAR	1	49
ReservedIdentifier	CHAR	1	50
Reserved	CHAR	1	51
PermittedToTrade	SHORT	2	52
Reserved	CHAR	1	54
IssueRate	SHORT	2	55
Reserved	CHAR	1	57
ST_SEC_ELIGIBILITY_ PER_ MARKET [4]	STRUCT	6	58
IssueStartDate	LONG	4	64
Reserved	CHAR	1	68
InterestPaymentDate	LONG	4	69
Reserved	CHAR	1	73
Issue Maturity Date	LONG	4	74
Reserved	CHAR	1	78
MarginPercentage	LONG	4	79
Reserved	CHAR	1	83
MinimumLotQuantity	LONG	4	84
Reserved	CHAR	1	88
BoardLotQuantity	LONG	4	89
Reserved	CHAR	1	93
TickSize	LONG	4	94

Structure Name	STOCK_STRUCTURE		
Packet Length	304 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	98
IssuedCapital	DOUBLE	8	99
Reserved	CHAR	1	107
FreezeQuantity	LONG	4	108
Reserved	CHAR	1	112
WarningQuantity	LONG	4	113
Reserved	CHAR	1	117
ListingDate	LONG	4	118
Reserved	CHAR	1	122
ExpulsionDate	LONG	4	123
Reserved	CHAR	1	127
ReadmissionDate	LONG	4	128
Reserved	CHAR	1	132
RecordDate	LONG	4	133
Reserved	CHAR	1	137
NoDeliveryStartDate	LONG	4	138
Reserved	CHAR	1	142
NoDeliveryEndDate	LONG	4	143
Reserved	CHAR	1	147
LowPriceRange	LONG	4	148
Reserved	CHAR	1	152
HighPriceRange	LONG	4	153
Reserved	CHAR	1	157
ExDate	LONG	4	158
Reserved	CHAR	1	162
BookClosureStartDate	LONG	4	163
Reserved	CHAR	1	167
BookClosureEndDate	LONG	4	168
Reserved	CHAR	1	172
LocalLDBUpdateDateTime	LONG	4	173
Reserved	CHAR	1	177
ExerciseStartDate	LONG	4	178
Reserved	CHAR	1	182
ExerciseEndDate	LONG	4	183
Reserved	CHAR	1	187

Structure Name	STOCK_STRUCTURE		
Packet Length	304 bytes		
Field Name	Data Type	Size in Byte	Offset
TickerSelection	SHORT	2	188
Reserved	CHAR	1	190
OldTokenNumber	LONG	4	191
Reserved	CHAR	1	195
CreditRating	CHAR	12	196
Reserved	CHAR	1	208
Name	CHAR	25	209
Reserved	CHAR	1	234
EGMAGM	CHAR	1	235
Reserved	CHAR	1	236
InterestDividend	CHAR	1	237
Reserved	CHAR	1	238
RightsBonus	CHAR	1	239
Reserved	CHAR	1	240
MFAON	CHAR	1	241
Reserved	CHAR	1	242
Remarks	CHAR	24	243
Reserved	CHAR	1	267
ExStyle	CHAR	1	268
Reserved	CHAR	1	269
ExAllowed	CHAR	1	270
Reserved	CHAR	1	271
ExRejectionAllowed	CHAR	1	272
Reserved	CHAR	1	273
PlAllowed	CHAR	1	274
Reserved	CHAR	1	275
Settlement Indicator	CHAR	1	276
Reserved	CHAR	1	277
IsCorporateAdjusted	CHAR	1	278
Reserved	CHAR	1	279
SymbolForAsset	CHAR	10	280
Reserved	CHAR	1	290
InstrumentOfAsset	CHAR	6	291
Reserved	CHAR	1	297
BasePrice	LONG	4	298

Structure Name	STOCK_STRUCTURE		
Packet Length	304 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	302
DeleteFlag	CHAR	1	303

Table 128 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

Field Name	Brief Description
	• '4' - Preopen extended • '5' - Stock Open With Market • '6' - Price Discovery
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.
CreditRating	Credit rating of the security.

Field Name	Brief Description
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.
Settlement Indicator	Indicates whether the contract is cash settled or physical settled Value will be C = Cash Settled, P= Physical Settled
IsCorporateAdusted	This field shows whether this contract is corporate adjusted.

Field Name	Brief Description
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

The following table provides the description of the newly added field.

Field Name	Brief Description
Category	The market hours in which the contract is available to trade. Note: The following will be the values: '1': Represents Regular market hours. '2': Represents Extended market hours.

Participant Structure

Header

Table 131 PARTICIPANT_FILE_HEADER

Structure Name	PARTICIPANT_FILE_HEADER		
Packet Length	14 bytes		
Field Name	Data Type	Size in Byte	Offset
NSEFO	CHAR	6	0
Reserved	CHAR	1	6
VersionNumber	CHAR	5	7
Reserved	CHAR	2	12

Structure

Table 132 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.

Security File Structure

Header

Table 133 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
CreationTime	LONG	4	15

Security Structure

Table 134 SECURITY_STRUCTURE

Structure Name	SECURITY_STRUCTURE		
Packet Length	230 bytes		
Field Name	Data Type	Size in Byte	Offset
Token	LONG	4	0
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	17	34
Reserved	CHAR	1	51
ST_SEC_ELIGIBILITY_ PER_ MARKET [6] (Refer)	STRUCT	5	52
BoardLotQty	LONG	4	57
Reserved	CHAR	1	61
TickSize	LONG	4	62
Reserved	CHAR	1	66
Name	CHAR	25	67
Reserved	CHAR	1	92
IssueRate	SHORT	2	93
Reserved	CHAR	1	95
IssueStartDate	LONG	4	96
Reserved	CHAR	1	100
IssueIPDate	LONG	4	101
Reserved	CHAR	1	105

Structure Name	SECURITY_STRUCTURE		
Packet Length	230 bytes		
Field Name	Data Type	Size in Byte	Offset
Issue Maturity Date	LONG	4	106
Reserved	CHAR	1	110
FreezePercent	SHORT	2	111
Reserved	CHAR	1	113
ListingDate	LONG	4	114
Reserved	CHAR	1	118
ExpulsionDate	LONG	4	119
Reserved	CHAR	1	123
ReAdmissionDate	LONG	4	124
Reserved	CHAR	1	128
ExDate	LONG	4	129
Reserved	CHAR	1	133
RecordDate	LONG	4	134
Reserved	CHAR	1	138
NoDeliveryStartDate	LONG	4	139
Reserved	CHAR	1	143
NoDeliveryEndDate	LONG	4	144
Reserved	CHAR	1	148
ParticipateInIndex	CHAR	1	149
Reserved	CHAR	1	150
AON	CHAR	1	151
Reserved	CHAR	1	152
MinFill	CHAR	1	153
Reserved	CHAR	1	154
WarningPercent	SHORT	2	155
Reserved	CHAR	1	157
BookClosureStartDate	LONG	4	158
Reserved	CHAR	1	162
BookClosureEndDate	LONG	4	163
Reserved	CHAR	1	167
Dividend	CHAR	1	168
Reserved	CHAR	1	169
Rights	CHAR	1	170
Reserved	CHAR	1	171
Bonus	CHAR	1	172

Structure Name	SECURITY_STRUCTURE		
Packet Length	230 bytes		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	1	173
Interest	CHAR	1	174
Reserved	CHAR	1	175
AGM	CHAR	1	176
Reserved	CHAR	1	177
EGM	CHAR	1	178
Reserved	CHAR	1	179
Remark	CHAR	25	180
Reserved	CHAR	1	205
LocalDBUpdateDateTime	LONG	4	206
Reserved	CHAR	1	210
DeleteFlag	CHAR	1	211
Reserved	CHAR	1	212
FaceValue	LONG	4	213
Reserved	CHAR	1	217
ISIN	CHAR	12	218

Table 135 ST_SEC_ELIGIBILITY_PER_MARKET

Structure Name	ST_SEC_ELIGIBILITY_PER_MAKRET		
Packet Length	5 bytes		
Field Name	Data Type	Size in Byte	Offset
Security Status	SHORT	2	0
Reserved	CHAR	1	2
Eligibility	CHAR	1	3
Reserved	CHAR	1	4

Field Name	Brief Description
Token	Token number of the security being updated. This is unique for a particular symbol-series combination.
SecurityInformation	This contains the Symbol, Series (EQ / IL / TT) and Instrument type.
IssuedCapital	Issue size of the security.

Field Name	Brief Description
PermittedToTrade	This field can have any one of the following value: • '0' - Listed but not permitted to trade • '1' - Permitted to trade
CreditRating	Credit rating of the security.
SecurityStatus	This field can have any one of the following value: • '1' - Preopen (Only for Normal market) • '2' - Open • '3' - Suspended • '4' - Preopen extended • '5' - Stock Open With Market
Eligibility	The flag is set to 1 if the security is allowed to trade in a particular market.
BoardLotQuantity	Regular lot size.
TickSize	Tick size/ Min spread size.
Name	Security name.
IssueRate	Price of the issue.
IssueStartDate	Date of issue of the security.
InterestPaymentDate	Interest payment date
IssueMaturityDate	Maturity date.
FreezePercent	Freeze percent for the security.
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.
ParticipateInIndex	This field can have any one of the following value: • '0' - Not Participate In Index • '1' - Participate In Index
AON	This field can have any one of the following value: • '0' - AON not allowed • '1' - AON allowed
MF	• '0' - MF not allowed

Field Name	Brief Description
	- ‘1’ - MF allowed
Warning Percent	Warning percent.
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.
Dividend	This field can have any one of the following value: - ‘0’ - No Dividend - ‘1’ - Dividend
Rights	This field can have any one of the following value: - ‘0’ - No Rights - ‘1’ - Rights
Bonus	This field can have any one of the following value: - ‘0’ - No Bonus - ‘1’ - Bonus
Interest	This field can have any one of the following value: - ‘0’ - No Interest - ‘1’ - Interest
EGM	This field can have any one of the following value: - ‘0’ - No EGM - ‘1’ - EGM
AGM	This field can have any one of the following value: - ‘0’ - No AGM - ‘1’ - AGM
Remark	Remarks
LocalLDBUpdateDateTime	This is the local database update date-time.
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
Face value	Face value of security Note: Already present in the security.txt but was not used. To correctly use the ISIN field face value should be considered.

Field Name	Brief Description
ISIN	ISIN number of security

Trimmed Structures

Trimmed Order Entry Request Structure

Table 136 MS_OE_REQUEST_TR

Structure Name	MS_OE_REQUEST_TR		
Packet Length	158 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	BOARD_LOT_IN_TR (20000) TRIMMED_BOARD_LOT_ACK_IN (20400)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
UserID	LONG	4	2
ReasonCode	SHORT	2	6
TokenNo	LONG	4	8
CONTRACT_DESC_TR	STRUCT	26	12
AccountNumber	CHAR	10	38
BookType	SHORT	2	48
Buy / SellIndicator	SHORT	2	50
DisclosedVolume	LONG	4	52
Volume	LONG	4	56
Price	LONG	4	60
GoodTillDate	LONG	4	64
ST_ORDER_FLAGS	STRUCT	2	68
BranchId	SHORT	2	70
TraderId	LONG	4	72
BrokerId	CHAR	5	76
Open/Close	CHAR	1	81
Settlor	CHAR	12	82
Pro / ClientIndicator	SHORT	2	94

ADDITIONAL_ORDER_FLAGS	STRUCT	1	96
Filler	LONG	4	98
NnfField	DOUBLE	8	102
PAN	CHAR	10	110
Algo ID	LONG	4	120
Reserved	SHORT	2	124
Reserved	CHAR	32	126

Table 137 CONTRACT_DESC_TR

Structure Name	CONTRACT_DESC_TR		
Packet Length	26 bytes		
Field Name	Data Type	Size in Byte	Offset
InstrumentName	CHAR	6	0
Symbol	CHAR	10	6
ExpiryDate	LONG	4	16
StrikePrice	LONG	4	20
OptionType	CHAR	2	24

Field Name	Brief Description
TransactionCode	The transaction code is BOARD_LOT_IN_TR (20000)/ TRIMMED_BOARD_LOT_ACK_IN (20400).
ReasonCode	This field contains the reason code for a particular order request rejection or order freeze. This, along with the error code, has the details regarding the error. Refer to Reason Codes in Appendix. During order entry, this field should be set to zero
TokenNumber	This is the Token Number of the contract on which order is to be placed. This field should contain a valid token number or '-1'. If the token number is set to '-1', the validations will be done only on contract descriptor. If the valid token number is sent, the validation will be done on token number as well as contract descriptor
SecurityInformation (CONTRACT_DESCRIPTOR_TR)	This structure contains the following fields:

Field Name	Brief Description
	Instrument Name, Symbol, Expiry Date, Strike Price and Option Type of the contract. This is mandatory and should be filled while sending the order entry request.
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 blank.
BookType	This field should contain the type of order. Refer to Book Types in Appendix.
Buy / SellIndicator	This field should specify whether the order is a buy or sell. The field should take one of the following values: • '1' for Buy order • '2' for Sell order
DisclosedVolume	This field should contain 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.
Volume	This field should contain the order quantity. The quantity should always be in multiples of Regular Lot except for Odd Lot orders and it should 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. The price must be a multiple of the tick size. To enter a Market order, the price should be set to zero. 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. Market attribute is not allowed for Negotiated

Field Name	Brief Description
	Orders. This should be multiplied by 100 before sending to the trading system.
GoodTillDate	This field should contain the number of days for a GTD order. This field can be set in two ways. To specify an absolute date, this field should be set to that date in number of seconds since midnight of January 1, 1980. To specify days, this field should be set to the number of days. This can take values from two to the maximum days specified for GTC orders only. If this field is non-zero, the GTC flag must be off.
OrderTerms	This field should specify the attributes of an order.
BranchId	This field should contain the branch number to which the broker belongs.
TraderId	This field should contain the ID of the user. This field accepts only numbers.
BrokerId	This field should contain the trading member ID.
Open / Close	Open / Close order indicator. This field should contain one of the following values. • 'O' for Open • 'C' for Close
Settlor	This field should specify 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. So, this field should be set to blank for a pro order (broker's own order).
Pro-ClientOrder	This field should contain one of the following values to specify whether the order is entered on behalf of a broker or a trader. • '1' represents the client's order. • '2' represents a broker's order.
ADDITIONAL_ORDER_FLAGS	Refer to *Additional Order Flags* and *Order Terms Attributes* tables in Chapter 4 for the relevant description. For reserved bit kindly set the values with 0
NNFField	This field should contain a 15 digit a unique identifier for various products deployed as

Field Name	Brief Description
	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	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.

Trimmed Order Mod/Cxl Request Structure

Table 138 MS_OM_REQUEST_TR

Structure Name	MS_OM_REQUEST_TR		
Packet Length	186 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	ORDER_MOD_IN_TR (20040) ORDER_CANCEL_IN_TR (20070) TRIMMED_ORDER_MOD_ACK_IN (20402) TRIMMED_ORDER_CANCEL_ACK_IN (20404) ~~ORDER_QUICK_CANCEL_IN_TR (20060)~~		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
UserID	LONG	4	2
Modified / CancelledBy	CHAR	1	6
TokenNo	LONG	4	8
CONTRACT_DESC_TR	STRUCT	26	12
OrderNumber	DOUBLE	8	38
AccountNumber	CHAR	10	46
BookType	SHORT	2	56
Buy / SellIndicator	SHORT	2	58

Structure Name	MS_OM_REQUEST_TR		
Packet Length	186 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	ORDER_MOD_IN_TR (20040) ORDER_CANCEL_IN_TR (20070) TRIMMED_ORDER_MOD_ACK_IN (20402) TRIMMED_ORDER_CANCEL_ACK_IN (20404) ~~ORDER_QUICK_CANCEL_IN_TR (20060)~~		
Field Name	Data Type	Size in Byte	Offset
DisclosedVolume	LONG	4	60
DisclosedVolumeRemaining	LONG	4	64
TotalVolumeRemaining	LONG	4	68
Volume	LONG	4	72
VolumeFilledToday	LONG	4	76
Price	LONG	4	80
GoodTillDate	LONG	4	84
EntryDateTime	LONG	4	88
LastModified	LONG	4	92
ST_ORDER_FLAGS	STRUCT	2	96
BranchId	SHORT	2	98
TraderId	LONG	4	100
BrokerId	CHAR	5	104
Open/Close	CHAR	1	109
Settlor	CHAR	12	110
Pro / ClientIndicator	SHORT	2	122
ADDITIONAL_ORDER_FLAGS	STRUCT	1	124
Filler	LONG	4	126
NnfField	DOUBLE	8	130
PAN	CHAR	10	138
Algo ID	LONG	4	148
Reserved	SHORT	2	152
LastActivityReference	LONG LONG	8	154
Reserved	CHAR	24	162

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_MOD_IN_TR (20040), ORDER_CANCEL_IN_TR (20070), TRIMMED_ORDER_MOD_ACK_IN (20402), TRIMMED_ORDER_CANCEL_ACK_IN (20404), ~~ORDER_QUICK_CANCEL_IN_TR (20060).~~
Modified / CancelledBy	This field denotes who 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
OrderNumber	Order Number is the identity of the order to be modified.
EntryDateTime	This field contains the date and time when the order entered the trading system. This is available in Order Confirmation/ Order Modification Confirmation response.
LastModified Time	In the case of order entry, this field will be same as Entry Date Time. After the order is modified it contains the time when the Order was last modified. It is the time in seconds from midnight of January 1, 1980. In case of Order Modification Request This field should contains the time when the Order was last modified
TraderId	This field should contain the ID of the user on whose behalf order is to be modified/cancelled.
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)

Field Name	Brief Description
Reserved	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.
LastActivityReference	In Order modification/cancellation request for an order, this field should contain LastActivityReference value received in response of the last activity done on that order. Last activity could be order entry, order modification or last trade of that order. Currently the same shall be in nanoseconds. Changes if any shall be notified.
Note: The other fields of modification request are the same as MS_OE_REQUEST.	

Trimmed Order Entry/Mod/Cxl Response Structure

Table 139 MS_OE_RESPONSE_TR

Structure Name	MS_OE_RESPONSE_TR		
Packet Length	240 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
LogTime	LONG	4	2
UserId	LONG	4	6
ErrorCode	SHORT	2	10
TimeStamp1	LONG LONG	8	12
TimeStamp2	CHAR	1	20
Modified / CancelledBy	CHAR	1	21
ReasonCode	SHORT	2	22
TokenNo	LONG	4	24

Structure Name	MS_OE_RESPONSE_TR		
Packet Length	240 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075)		
Field Name	Data Type	Size in Byte	Offset
CONTRACT_DESC_TR	STRUCT	26	28
CloseoutFlag	CHAR	1	54
OrderNumber	DOUBLE	8	56
AccountNumber	CHAR	10	64
BookType	SHORT	2	74
Buy / SellIndicator	SHORT	2	76
DisclosedVolume	LONG	4	78
DisclosedVolumeRemaining	LONG	4	82
TotalVolumeRemaining	LONG	4	86
Volume	LONG	4	90
VolumeFilledToday	LONG	4	94
Price	LONG	4	98
GoodTillDate	LONG	4	102
EntryDateTime	LONG	4	106
LastModified	LONG	4	110
ST_ORDER_FLAGS	STRUCT	2	114
BranchId	SHORT	2	116
TraderId	LONG	4	118
BrokerId	CHAR	5	122
Open/Close	CHAR	1	127
Settlor	CHAR	12	128
Pro / ClientIndicator	SHORT	2	140
ADDITIONAL_ORDER_FLAGS	STRUCT	1	142
Filler	LONG	4	144
NnfField	DOUBLE	8	148
TimeStamp	LONG LONG	8	156
PAN	CHAR	10	164
Algo ID	LONG	4	174
Reserved	SHORT	2	178
LastActivityReference	LONG LONG	8	180

Structure Name	MS_OE_RESPONSE_TR		
Packet Length	240 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	ORDER_CONFIRMATION_TR (20073) ORDER_MOD_CONFIRMATION_TR (20074) ORDER_CXL_CONFIRMATION_TR (20075)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	52	188

Field Name	Brief Description
TransactionCode	The transaction code is ORDER_CONFIRMATION_TR (20073), ORDER_MOD_CONFIRMATION_TR (20074), ORDER_CXL_CONFIRMATION_TR (20075),
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. Machine / Stream no. should be interpreted as integer value and not as character value. Values will be numeric value 1,2,3, ….,10,11 etc. and can range from 1 to 127
Modified / CancelledBy	This field denotes who 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
OrderNumber	Order Number is the identity of the order to be modified.
EntryDateTime	This field contains the date and time when the order entered the trading system. This is available in Order Confirmation/ Order Modification Confirmation response.

Field Name	Brief Description
LastModified Time	In the case of order entry, this field will be same as Entry Date Time. After the order is modified it contains the time when the Order was last modified. It is the time in seconds from midnight of January 1, 1980. In case of Order Modification Request This field should contains the time when the Order was last modified
TimeStamp	In this field Time will be sent in nanoseceonds (from 01-Jan-1980 00:00:00).
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	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.
LastActivityReference	This field will contain a unique value for current activity. Currently the same shall be in nanoseconds. Changes if any shall be notified.
Note: The other fields of modification request are the same as MS_OE_REQUEST.	

Trimmed Trade Confirmation Response

Table 140 MS_TRADE_CONFIRM_TR

Structure Name	MS_TRADE_CONFIRM_TR		
Packet Length	230 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	TRADE_CONFIRMATION_TR (20222)		
Field Name	Data Type	Size in Byte	Offset
TransactionCode	SHORT	2	0
LogTime	LONG	4	2
TraderId	LONG	4	6

Structure Name	MS_TRADE_CONFIRM_TR		
Packet Length	230 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	TRADE_CONFIRMATION_TR (20222)		
Field Name	Data Type	Size in Byte	Offset
Timestamp	LONG LONG	8	10
Timestamp1	DOUBLE	8	18
Timestamp2	DOUBLE	8	26
ResponseOrderNumber	DOUBLE	8	34
BrokerId	CHAR	5	42
Reserved	CHAR	1	47
AccountNumber	CHAR	10	48
Buy/SellIndicator	SHORT	2	58
OriginalVolume	LONG	4	60
DisclosedVolume	LONG	4	64
RemainingVolume	LONG	4	68
DisclosedVolume Remaining	LONG	4	72
Price	LONG	4	76
ST_ORDER_ FLAGS	STRUCT	2	80
GoodTillDate	LONG	4	82
FillNumber	LONG	4	86
FillQuantity	LONG	4	90
FillPrice	LONG	4	94
VolumeFilledToday	LONG	4	98
ActivityType	CHAR	2	102
ActivityTime	LONG	4	104
Token	LONG	4	108
CONTRACT_DESC_TR	STRUCT	26	112
OpenClose	CHAR	1	138
BookType	CHAR	1	139
Participant	CHAR	12	140
ADDITIONAL_ORDER_FLAGS	STRUCT	1	152
PAN	CHAR	10	153
Algo ID	LONG	4	164
Reserved	SHORT	2	168
LastActivityReference	LONG LONG	8	170

Structure Name	MS_TRADE_CONFIRM_TR		
Packet Length	230 bytes		
Usage	Use pragma pack(2) Use pragma pack(1) for ADDITIONAL_ORDER_FLAGS		
Transaction Code	TRADE_CONFIRMATION_TR (20222)		
Field Name	Data Type	Size in Byte	Offset
Reserved	CHAR	52	178

Field Name	Brief Description
TransactionCode	The transaction code is TRADE_CONFIRMATION_TR (20222).
PAN	This field shall contain the PAN
Algo ID	This field shall contain the Algo ID
Reserved	This field is reserved for future use. This should be populated as 0 for the message to be accepted by exchange host.
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 à** 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 connectionà** 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 à** 1. **SSL_get_peer_certificate**(const SSL *ssl) - Get the GR 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 à** 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 -** **Encryption:**

Initialization→

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

    if(1 != EVP_EncryptInit_ex(*ctx, EVP_aes_256_gcm(), NULL, key, iv))
        handleErrors();
}
```

Encryption→

```
void encrypt(EVP_CIPHER_CTX *ctx, unsigned char *plaintext, int
plaintext_len, unsigned char *ciphertext, int *ciphertext_len)
{
    int len;

    if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
        handleErrors();
    *ciphertext_len = len;
}
```

Decryption:

Initialization→

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

    if(1 != EVP_DecryptInit_ex(*ctx, EVP_aes_256_gcm(), NULL, key, iv))
        handleErrors();
}
```

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);
}
```

<u>Decryption Block:</u>

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);
```