

# **Protocol for Non-NEAT Front End (NNF)**

## **Initial Public Offering System**

**Version 3.0**

**Dec 2014**



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

## **Notice**

© Copyright National Stock Exchange of India Ltd (NSEIL). All rights reserved. Unpublished rights reserved under applicable copyright and trades secret laws.

The contents, ideas and concepts presented herein are proprietary and confidential. Duplication and disclosure to others in whole, or in part is prohibited.

**Changes in This Issue:**

Version 3.0		Dec 2014
Changes Added in the system - : Cut-off orders in OFS		
Pages Changed	Description	
32,33,35,37,39	Cut-off orders allowed in RS series for OFS	
15,17,53	Introduced a new series.	
32,33,34,35,36	<ul style="list-style-type: none"> <li>Introduced a new series.</li> <li>Order Value should not be greater than 2 Lakh.</li> <li>CP Code to be set to TM Id to disallow CP orders.</li> <li>Flag to be set to determine TM or CP blocking (No type A and Type B)</li> </ul>	
32,33,53	Mentioning structures for Big Endian machines	
44,45	Added a new packet for sending custodial status and order number in case of Type A and Type B orders.	
67	Added a new transaction code 2079	
8	X.25 Communication Network has been removed from Organization of this Document.	
12	X.25 Communication Network has been removed from Contents.	
12	<ul style="list-style-type: none"> <li>Description of X.25 Communication Network has been removed.</li> <li>Discription about communication network is added (TCP/IP and UDP protocol)</li> </ul>	
37	<ul style="list-style-type: none"> <li>CP order is differentiated into 2 parts <ul style="list-style-type: none"> <li>Institutional orders with 0 margin block collateral is set to NO. i.e set to '1'b.</li> <li>Institutional orders with 100 % margin block collateral are set to YES. i.e set to '0'b.</li> </ul> </li> </ul>	
44	<ul style="list-style-type: none"> <li>Added a Note regarding Order Modification criteria's for institutional order for 0% margin and 100% margin after custodial confirmation.</li> <li>Added a note regarding Order Cancellation criteria's for institutional order for 0% margin and 100% margin after custodial confirmation.</li> </ul>	
67	Message string changed for error code 16581	
68	<ul style="list-style-type: none"> <li>Message string changed for error code 16583</li> <li>Added a new error code 16635, indicating institutional orders with 0% margin cannot be cancelled.</li> </ul>	
36	CP order can only be placed when block collateral is NO i.e set to '1'b.	
43	Added a Note regarding Order Modification criteria's.	
65	Modified description for error code 16582.	
15,17	Introduced a new series	

32	Reserved1 field of Order Terms populated with DAY Bit.
33	<ul style="list-style-type: none"> <li>Introduced a new series.</li> <li>Number of orders on the same application number for a particular issue is greater than three is Not Applicable for OFS.</li> <li>Flag to be set to determine TM or CP blocking</li> </ul>
34	<ul style="list-style-type: none"> <li>Appl No Not applicable and PAN Non Mandatory for OFS.</li> <li>Benf_id to be re used for CP Code.</li> <li>Volume should not exceed the issue size</li> </ul>
35	<ul style="list-style-type: none"> <li>OFS order can only be on price points.</li> <li>PAN Not Applicable for OFS.</li> <li>CP Code cannot be NSEIL</li> <li>OFS order would be of Buy Type</li> </ul>
36	<ul style="list-style-type: none"> <li>In ORDER FLAGS, the Reserved1 bit is used for DAY order for OFS. GTC bit to be set to 0 and Reserved bit to 1 if order is for OFS.</li> <li>Margin Amount is not applicable.</li> <li>PartCategoryId, DepPartId not applicable for OFS</li> <li>RTSGNo to be used as Account Number. Added additional criteria on RTGS No</li> <li>Depository Name is not applicable. This field can be reused for ECHOBACK</li> <li>TMCP Flag to determine blocking of collaterals for TM or CP.</li> </ul>
37	<ul style="list-style-type: none"> <li>Pro_Client field would be populated as 1 for CLI and 2 for PRO Order with respect to OFS.</li> <li>Added criteria for PRO/CLI order on Account Number.</li> </ul>
38	Added an extra row in Order Terms table.
43	Added a Note indicating the fields applicable for Order Modification.
65	Added Error Code : 16567,16568,16569,16570,16571,16572,16573,16574,16575,16576,16577,16578,16579,16580,16581,16582
66	Added Error Code: 16583,16584,16585
68	<ul style="list-style-type: none"> <li>Added a row Quick Reference for Order Entry Parameters</li> <li>Added a row under order terms table</li> </ul>
27	Change for Category wise Margin Amount Calculation
31, 34	<ul style="list-style-type: none"> <li>RTGS code(16) replaced with RTGS code(25)</li> <li>Filler(20) changed to Filler(11)</li> </ul>
60,62	<ul style="list-style-type: none"> <li>Error Code 16509 description has been changed. 16527 description has been changed.</li> </ul>

31	<ul style="list-style-type: none"> <li>• ApplNumber2 replaced with PAN Number</li> <li>• Client name(25) replaced with RTGS code(16)</li> <li>• Filler(11) changed to Filler(20)</li> </ul>
60,61	Error Code 16509,16519,16520,16521,16522,16523,16524,16525,16526 16527,16528,16529,16530,16531,16532 added
59	Added error codes: 16513, 16515, 16516, 16517, 16518.
30	Added nnf field and a reserved field in order structure.
13	Message Header: The data type Reserved1 and Reserved2 field has been changed to SHORT from BYTE
14	Broadcast Message header: The data type Reserved1 and Reserved2 field has been changed to SHORT from BYTE
15	Size of 'Error message' has been increased by 2 bytes,
17,19	Sizes of the following packets have been increased by 4 bytes: logon request and logon confirmation response,
21,23,24,25,26, 27	<ul style="list-style-type: none"> <li>• Sizes of the following packets have been increased by 2 bytes: system information request, system information response, Update local Database request, Update local database header, lpo Cat Slab info, Update Local database trailer, Message download request</li> <li>• Update local database packet size changed to 80 from 76.</li> <li>• Message download packet size changed to 80 from 76</li> </ul>
30	Order entry request: The trader id field datatype has been changed from short long. To maintain the same packet size the filler has been reduced from char(16) to char(12)
41	Size of "Interactive/Broadcast message sent from control" has been increased by 4 bytes
42,43,47,48,50,52, 54,55	The packet size has been increased by 2 bytes: general message broadcast, change in system status, change in security master, Change in category status, change in security status, change in market status, MBP Update, Only MBP update, Security open message
59	Error code 16505 added
66	Size of "Invitation packet" has been increased by 2 bytes.
60	Message record packet size changed to 512
61	<ul style="list-style-type: none"> <li>• Update_localdb_data - Packet size changed to 80 from 76</li> <li>• Update_localdb_header – Packet size changed to 42.</li> <li>• Update_localdb_trailer - Packet size changed to 42.</li> </ul>
63	TAP implementation protocol added

30	<ul style="list-style-type: none"> <li>The filler field has been changed from 0 to 16.</li> <li>The volume field data type has been changed from LONG to DOUBLE.</li> <li>The order packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
34	<ul style="list-style-type: none"> <li>The “Order requested response” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Order confirmation response” packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
35	<ul style="list-style-type: none"> <li>The “Order freeze response” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Order error response” packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
36	The “Order modification request” packet size has been increased from 204 bytes to 224 bytes.
37, 38	<ul style="list-style-type: none"> <li>The “Order modification response” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Order modification confirmation response” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Order modification error response” packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
38	<ul style="list-style-type: none"> <li>The “Order cancellation request” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Order cancellation response” packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
38, 39	<ul style="list-style-type: none"> <li>The “Order cancellation confirmation response” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Order cancellation error response” packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
40	<ul style="list-style-type: none"> <li>The “Freeze approve response” packet size has been increased from 204 bytes to 224 bytes.</li> <li>The “Freeze reject response” packet size has been increased from 204 bytes to 224 bytes.</li> </ul>
25,26	New local database download has been introduced for IPO Category Slab Information to check on front end to reject the order depending on the slab download given in ipo_cat_hrchy Slab Information
30	<ul style="list-style-type: none"> <li>The filler field has been changed from 8 to 0.</li> <li>The participant category Id is reduced from 12 to 10</li> <li>The Client name is increased from 15 to 25.</li> </ul>
47	<ul style="list-style-type: none"> <li>The packet length is reduced to 79 from 81.</li> <li>The category Id is reduced from 12 to 10.</li> </ul>
59	The new error codes 16564 and 1667 are update4d in the List of Error codes.

61	The new Transaction codes 5727, 5728 are updated in the List of Transaction codes.
18,20,33	BranchId field is changed from 2 digit to 3 digit
36	Allowing the modification of category field during order modification.
31	<ul style="list-style-type: none"><li>• The Margin field has been changed from LONG to DOUBLE.</li><li>• The filler field has been changed from 20 to 8.</li><li>• A new field NnfField has been added in the order structure</li></ul>
32	The maximum margin amount that can be entered from the frontend has been increased from 2,000,000,000 to 89,99,99,99,99,99,999.
58	The description for the error code has been modified from “Invalid Margin amount. Margin amount exceeds 2,000,000,000” to “Invalid Margin amount. Margin amount exceeds 89,99,99,99,99,99,999”.
57, 58	Added error codes : 16260, 16363, 16404, 16436, 16442, 16169, 16504, 16506, 16507, 16508, 16509, 16510, 16511, 16512, 16017

## Preface

### Purpose

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

### Target Audience

The document is written for system designers and programmers of user organizations and third party software developers who are responsible for the development of software to interact with the IPO System of the National Stock Exchange.

### Organization of This Document

This document is organized as follows:

Chapters	Description
Chapter 1	Provides a brief introduction to Non-NEAT Front end (NNF).
Chapter 2	Describes the general guidelines for the designers and programmers who develop NNF. It details the data types used and also discusses the MESSAGE_HEADER that is prefaced with all the structures, the Broadcast Header and Error Message.
Chapter 3	Describes how a trader logs on to the trading system. It covers the log on request, the system responses and download of the updated information on the securities and the status of the markets.
Chapter 4	Describes entering fresh orders, modifying an existing order, and cancelling outstanding orders.
Chapter 5	Describes the messages that are received on the interactive connection. These messages are received by users not in response to any request.
Chapter 6	Describes the various broadcast messages.
Appendix	Lists the error, transaction and reason codes and also covers the various market status and market types.



**Abbreviations and Acronyms Used**

The abbreviations and acronyms used in this document are:

AGM	Annual General Meeting
ATO	At The Opening
BCID	Broadcast Circuit Id
BM	Branch Manager
CM	Corporate Manager
I/B	Interactive / Broadcast
DL	Dealer
GTC	Good Till Cancellation
MBO	Market By Order
MBP	Market By Price
NEAT	National Exchange for Automated Trading
NNF	Non Neat Front End
NSE	National Stock Exchange
OSL	Open Strata Link
RL	Regular Lot
TWS	Trader Workstation
VCID	Virtual Circuit ID
VSAT	Very Small Aperture Terminal
VV.RR.SS	Version. Release. Sub-release

## CONTENTS

Chapter 1	Introduction.....	12
Chapter 2	General Guidelines .....	13
	INTRODUCTION.....	13
	MESSAGE STRUCTURE DETAILS .....	13
	GUIDELINES FOR DESIGNERS.....	13
	GUIDELINES FOR PROGRAMMERS .....	13
	DATA TYPES USED.....	14
	MESSAGE HEADER .....	14
	BROADCAST HEADER .....	16
	ERROR MESSAGE.....	16
Chapter 3	Logon and Download Processes .....	18
	INTRODUCTION.....	18
	ORDER OF EVENTS TO BE FOLLOWED DURING LOGON AND LOGOFF.....	18
	LOGON REQUEST.....	18
	LOGON RESPONSE .....	20
	LOGON CONFIRMATION RESPONSE.....	20
	LOGON ERROR.....	22
	SYSTEM INFORMATION DOWNLOAD.....	22
	<i>System Information Request</i> .....	22
	<i>System Information Response</i> .....	23
	UPDATE LOCAL DATABASE DOWNLOAD.....	24
	<i>Update Local Database Request</i> .....	24
	<i>Update Local Database Response</i> .....	26
	<i>Partial System Information Response</i> .....	26
	<i>Update Local Database Download</i> .....	26
	<i>Update Local Database Header</i> .....	26
	<i>Update Local Database Data</i> .....	26
	NEW LOCAL DATABASE DOWNLOAD HAS BEEN INTRODUCED:.....	27
	IPO CATEGORY SLAB INFO: .....	27
	<i>Update Local Database Trailer`</i> .....	28
	MESSAGE DOWNLOAD .....	28
	<i>Message Download Request</i> .....	28
	<i>Message Download Response</i> .....	29
	<i>Message Download Header</i> .....	29
	<i>Message Download Data</i> .....	29
	<i>Message Download Trailer</i> .....	29
	LOGOFF REQUEST.....	30
	LOGOFF CONFIRMATION RESPONSE .....	30
Chapter 4	Order Management .....	31
	INTRODUCTION.....	31
	ORDER ENTRY .....	31
	<i>Order Type</i> .....	31
	<i>Rules of Order Entry</i> .....	31
	<i>Order Entry Request</i> .....	31
	<i>Order Entry Response</i> .....	38
	<i>Order Requested Response</i> .....	38
	<i>Order Confirmation Response</i> .....	38
	<i>Order Freeze Response</i> .....	39
	<i>Order Error Response</i> .....	39
	ORDER MODIFICATION .....	41
	<i>Rules of Order Modification</i> .....	41

<i>Order Modification Request</i> .....	41
<i>Order Modification Response</i> .....	42
<i>Order Modification Confirmation Response</i> .....	42
<i>Order Modification Error Response</i> .....	42
ORDER CANCELLATION.....	43
<i>For Institutional orders with 0 % margin (Type B) orders cannot be cancelled at any time</i> .....	43
<i>Rules of Order Cancellation</i> .....	43
<i>Order Cancellation Request</i> .....	43
<i>Order Cancellation Response</i> .....	44
<i>Order Cancellation Confirmation Response</i> .....	44
<i>Order Cancellation Error Response</i> .....	44
Chapter 5    Unsolicited Messages .....	45
INTRODUCTION.....	45
FREEZE APPROVE RESPONSE.....	45
FREEZE REJECT RESPONSE.....	45
CUSTODIAN APPROVE/REJECT RESPONSE.....	45
INTERACTIVE/BROADCAST MESSAGES SENT FROM CONTROL.....	46
Chapter 6    Broadcast .....	48
INTRODUCTION.....	48
GENERAL MESSAGE BROADCAST.....	48
CHANGE IN SYSTEM STATUS / PARAMETERS.....	49
CHANGE IN SECURITY MASTER.....	49
CHANGE IN CATEGORY STATUS.....	53
CHANGE IN SECURITY STATUS.....	53
CHANGE OF MARKET STATUS.....	54
MARKET BY PRICE UPDATE.....	55
ONLY MARKET BY PRICE UPDATE.....	58
MARKET WATCH UPDATE.....	60
SECURITY OPEN MESSAGE.....	61
Appendix    .....	62
LIST OF ERROR CODES.....	62
REASON CODES.....	67
LIST OF TRANSACTION CODES.....	67
QUICK REFERENCE FOR ORDER ENTRY PARAMETERS.....	69
MARKET TYPE.....	70
MARKET STATUS.....	70
BOOK TYPES.....	70
SECURITY STATUS.....	70
TAP.....	71
CHANGES REQUIRED IN CTCL.....	71
TCP/IP CONNECTIVITY BETWEEN CTCL SYSTEM AND TAP BOX.....	71
PROGRAM FLOW FOR TCP/IP CONNECTIVITY.....	71
SETTING TCP KEEP-ALIVE PARAMETERS.....	72
PACKET STRUCTURE COMMUNICATING BETWEEN CTCL AND TAP BOX.....	72
CHANGES IN MESSAGE FLOW.....	73

## Chapter 1 Introduction

The National Stock Exchange of India Ltd (NSEIL) provides a fully automated screen based trading system, enabling trading members spread across the length and breadth of India to place orders directly from their offices through an extensive telecommunication network. The system is known as 'National Exchange for Automated Trading' (NEAT) system. The trading members can use NEAT Front end or Non-NEAT Front end (NNF) to establish a network connection with the IPO host system of NSE for placing orders. 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 whereas they are supported by their respective vendors and NSE is not responsible for the performance of the NNF.

### Communication Network

Connectivity to the Exchange is through WAN protocol on TCP/IP. There are two types of connectivity;

- a) Interactive connectivity – On TCP/IP for user specific messages to and from the Exchange. Trading Front can connect to the exchange through a middleware application called Trading Access Point (TAP), The TAP protocol is defined in the annexure below.
- b) Market data – The market data is sent as broadcast based on UDP connection Broadcast circuit ID (BCID) follows a unidirectional path which is from the host end to the TWS. 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

## Chapter 2 General Guidelines

### Introduction

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

### Message Structure Details

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

The message data consists of the actual data that is sent across to the host or received from the host.

Transaction code, an important field of the message header, is a unique numeric identifier which is sent to or received from the trading system. This is used to identify the transaction between the TWS and the host end.

### Guidelines for Designers

1. The order of the log-on messages should strictly be maintained as given in the following section (Chapter 3) of this document. Otherwise, the user will not be able to log on to the trading system.
2. All messages sent by the trading system will be time-stamped.
3. All time fields are number of seconds from midnight January 1 1980.
4. No host-end inquiries are 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 as the host system processes prices in paise.

### Guidelines for Programmers

1. If your system uses little-endian order, the data types such as UINT, SHORT, LONG and DOUBLE contained in a packet, which occupy more than one byte should be twiddled (byte reversed). Twiddling involves reversing a given number of bytes such that the byte in 'n' position comes to the first position; the byte in (n-1) position comes to the second position and so on. For example, if the value to be sent is 1A2B (hexadecimal), reverse the bytes to 2B1A. The same applies while receiving messages. So if the value received is 02BC, the actual value is BC02. So twiddle such data types before sending and after receiving to ensure that correct data is sent and received.

#### **Note:**

Twiddling is required because of the variety in endian order—big and little. A big-endian representation has a multi-byte integer written with its most significant byte on the left. A little-endian representation, on the other hand, places the most significant byte on the right. 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 host end uses big-endian order. Suppose your machine 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, fill it with blanks before sending. The strings received from the host end are padded with blanks and are not NULL terminated.
3. All the structures should be defined in the following manner:
  - Items of type char or unsigned char, or arrays containing items of these types, are byte aligned.
  - Structures are word aligned; structures of odd size are padded to an even number of bytes.
  - All other types of structure members are word aligned.
4. All numeric data must be set to zero (0) before sending to the host, unless a value is assigned to it.

**Note:** The values of all the constants and transaction codes given in the document are listed in Appendix.

#### Data Types Used

Data Type	Size in Bytes	Signed / Unsigned
CHAR	1	Signed
UINT	2	Unsigned
SHORT	2	Signed
LONG	4	Signed
DOUBLE	8	Signed and Floating Point

#### Message Header

This is an interactive header. Each structure is prefaced with a MESSAGE\_HEADER. Some fields in the header are fixed. The remaining fields are variable and set differently for each transaction code. The structure of the MESSAGE\_HEADER is as follows:

<b>Structure Name:</b> MESSAGE_HEADER
<b>Packet Length:</b> 40 bytes

SHORT	Reserved1
SHORT	Reserved2
LONG	LogTime
CHAR	AlphaChar [2]
SHORT	TransactionCode
SHORT	ErrorCode
CHAR	TimeStamp [8]
CHAR	TimeStamp1 [8]
CHAR	Reserved3 [8]
SHORT	MessageLength

Field Name	Description	Comment
LogTime	This field should be set to zero while sending messages. For the messages coming from the host, this field contains the time when generated by the trading system host.	
AlphaChar	This field should be set to the first two characters of Symbol if the structure contains Symbol and Series; or else it should be set to blank.	The Symbol field should contain the name of the security. The series field should have: EQ – Equity shares RV – Reverse book building shares IS – (Offer For Sale ) RS – (Offer For Sale)
TransactionCode	Transaction message number. This describes the type of message received from or sent to the host.	
ErrorCode	This field should be set to zero while sending messages to the host. In messages coming from the host, this field describes the type of error.	Refer to <i>Error Messages</i> in Appendix
TimeStamp	This field should be set to numeric zero while sending. This is used in host end.	
TimeStamp1	This field should be set to numeric zero while sending. This is the time the message arrives at the trading system host.	In TimeStamp1, current system time is sent in jiffies from host end. This is 8 bytes in host end. In front end, we typecast the first four bytes into double and store it in a variable and typecast the other four bytes into double and store in another variable. These values are used while requesting message area download.
MessageLength	Set to the length of the entire message, including the length of message header.	

Broadcast Header

The broadcast messages such as market open, market close, market in preopen are prefaced with BROADCAST\_HEADER. Some fields in the header are fixed. The remaining fields are variable and set differently for each transaction code. The structure of the BROADCAST\_HEADER is as follows:

<b>Structure Name: BROADCAST_HEADER</b>	
<b>Packet Length: 40 bytes</b>	
SHORT	Reserved1
SHORT	Reserved2
LONG	LogTime
CHAR	AlphaChar [2]
SHORT	TransCode
SHORT	ErrorCode
LONG	BCSeqNo
BYTE	Reserved
BYTE	Filler1[3]
CHAR	TimeStamp2 [8]
BYTE	Reserved3 [8]
SHORT	MessageLength

Field Name	Description	Comment
LogTime	This is the time when the message was generated by the trading system host.	
AlphaChar	This field will be set to the first two characters of Symbol if the structure contains Symbol and Series or else it will contain blank.	
TransactionCode	Transaction message number. This describes the type of message sent.	
ErrorCode	Error number. This describes the type of error. Refer to <i>Error Messages</i> in Appendix.	
BCSeqNo	BROADCAST Sequence number	
TimeStamp2	Contains the time when the message is sent from the host.	
MessageLength	Set to the length of the entire message, including the length of 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:

<b>Structure Name: ERROR_RESPONSE</b>
<b>Packet Length: 180 bytes</b>
MESSAGE_HEADER STRUCT SECURITY_INFORMATION {



```

        CHAR Symbol [10]
        CHAR Series  [2]
    }
    CHAR ErrorMessage [128]
    
```

Field Name	Description	Comment
ErrorMessage	Stores the error message. Refer to <i>List of Errors</i> in Appendix.	
Security Information	<p>This structure contains the following fields:</p> <ul style="list-style-type: none"> <li>• Symbol</li> <li>• Series.</li> </ul> <p>The Symbol field should contain the name of the security.</p> <p>The Series field should have one of the following values:</p> <ul style="list-style-type: none"> <li>• EQ – Equity shares</li> <li>• RV – Reverse book building shares</li> <li>• IS - Offer For Sale</li> <li>• RS – Offer For Sale</li> </ul>	

## Chapter 3 Logon and Download Processes

### Introduction

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

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

### Order of Events to Be Followed During Logon and Logoff

The following sequence explains the order in which transaction codes are sent and received during logon process:

Sequence No.	Transaction Code	Sent By	Received By
1	SIGN_ON_REQUEST_IN	TWS	Host End
2.	SIGN_ON_REQUEST_OUT	Host End	TWS
3.	SYSTEM_INFORMATION_IN	TWS	Host End
4.	SYSTEM_INFORMATION_OUT	Host End	TWS
5.	UPDATE_LOCALDB_IN	TWS	Host End
6.	PARTIAL_SYSTEM_INFORMATION or UPDATE_LOCALDB_HEADER, UPDATE_LOCALDB_DATA, UPDATE_LOCALDB_TRAILER	Host End	TWS
7.	DOWNLOAD_REQUEST	TWS	Host End
8.	HEADER_RECORD, MESSAGE_RECORD, TRAILER_RECORD	Host End	TWS

The following sequence explains the order in which the transcodes are sent and received during logoff process:

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

### Logon Request

When the user wants to establish an interactive circuit with the host, he sends this request. The structure sent to the host for this request is as follows:

<b>Structure Name:</b> SIGNON_REQUEST
<b>Transaction Code:</b> SIGN_ON_REQUEST_IN (2300)
<b>Packet Length:</b> 186 bytes

```

MESSAGE_HEADER (Refer to MESSAGE_HEADER in
                Chapter 2)
    LONG   UserId(changed from SHORT to LONG)
    CHAR   Password [8]
    CHAR   NewPassword [8]
    CHAR   TraderName [26]
    LONG   LastPasswordChangeDate
    CHAR   BrokerId [5]
    CHAR   Reserved1
    SHORT  BranchId
    LONG   VersionNumber
    LONG   Batch2StartTime
    CHAR   HostSwitchContext
    CHAR   Colour [50]
    CHAR   Reserved2
    SHORT  UserType
    DOUBLE SequenceNumber
    CHAR   WsClassName [14]
    CHAR   BrokerStatus
    CHAR   Reserved3
    struct BROKER_ELIGIBILITY_PER_MKT
           (Refer to Logon Confirmation Response
           in Chapter 3)
    
```

Field Name	Description	Comment
TransactionCode	SIGN_ON_REQUEST_IN (2300)	
UserId	This field should contain User ID of the user/broker. This field accepts numbers only.	
Password	This field should contain the password entered by the user. The password should be a minimum of six characters and maximum of eight characters in length. A combination of alphabets and numbers is allowed in the password.	The user should enter the password for a successful Logon. When the user logs on for the first time the default password provided by NSE must be entered and the password should be changed by entering a new password.
NewPassword	This field should contain the new password entered by the user. This field should be entered only when the user wishes to change the password or the password has expired. Otherwise this field should be blank.	The New Password should be entered along with the old password in the Password field. While logging on the system for the first time, the default password provided by NSE must be changed.
TraderName	This field when received from the host contains the user's name. This field should be sent to host as blanks.	
LastPasswordChangeDateTime	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.  This fields is increased from 2 digit to 3 digit	

Field Name	Description	Comment
VersionNumber	This field should contain the version number of the trading system. The format is VERSION.RELEASE.SUB_RELEASE. (For example, 3.05.00)	As and when these structures are changed, the version number will be changed.
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 indicates the type of user. It can take one of the following values when it is sent from the host: <ul style="list-style-type: none"> <li>• '0' denotes Dealer</li> <li>• '4' denotes Corporate Manager</li> <li>• '5' denotes Branch Manager</li> </ul> This field should be set to '0' while sending to the host.	
SequenceNumber	This field should be set to numerical zero while sending the request to host.	
WorkstationNumber	The network ID of the workstation should be provided. This is a seven digit number. The first five digits are fixed by the Exchange and represent the various ports / switch locations. The last two digits denote the user's PC - ID. It must be any number other than '00'.	
BrokerStatus	This field should be set to blank.	
BrokerEligibilityPer Market	This field should be set to numerical zero.	

Logon Response

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

Logon Confirmation Response

A successful log-on results in the Logon Confirmation response. The following structure is sent back:

<b>Structure Name:</b> SIGNON_REQUEST
<b>Transaction Code:</b> SIGN_ON_REQUEST_OUT (2301)
<b>Packet Length:</b> 186 bytes
MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2) LONG UserId(changed from SHORT to LONG) CHAR Password [8] CHAR NewPassword [8] CHAR TraderName [26] LONG LastPasswordChangeDate CHAR BrokerId [5] CHAR Reserved1 SHORT BranchId LONG VersionNumber LONG EndTime CHAR Reserved2 [52] SHORT UserType DOUBLE SequenceNumber

```

CHAR   Reserved3[14]
CHAR   BrokerStatus
CHAR   Reserved4

Note: Use any one of following two structures:

For Small Endian Machines:
    struct  BROKER_ELIGIBILITY_PER_MKT
    {
        Reserved      - 4 bits
        Reserved      - 1 bit
        Reserved      - 1 bit
        Reserved      - 1 bit
        Normal market - 1 bit
        Reserved      - 8 bits
    }

For Big Endian Machines
    struct  BROKER_ELIGIBILITY_PER_MKT
    {
        Normal market - 1 bit
        Reserved      - 1 bit
        Reserved      - 1 bit
        Reserved      - 1 bit
        Reserved      - 4 bits
        Reserved      - 8 bits
    }
    
```

Field Name	Description	Comment
TransactionCode	The transaction code is SIGN_ON_REQUEST_OUT (2301).	
LogTime	The current time at the trading system is sent back as number of seconds since midnight of January 1, 1980. The time at the TWS must be synchronized with this.	
UserId	This field contains the ID of the user.	
TraderName	This field contains the user's name.	
LastPasswordChangeDate	This field contains the last date and time when the password was changed.	
BrokerId	This field contains the trading Member ID.	
BranchId	This field contains the ID of the branch of the particular user.  This field is changed from 2 digit to 3 digit	
EndTime	This field contains the time the markets last closed and it is sent as the number of seconds since midnight of January 1, 1980. If this time is different from the time sent in an earlier log-on, all orders, and messages for this user must be deleted from the Local	

Field Name	Description	Comment
	Database.	
UserType	This field contains the type of user who is logging on: <ul style="list-style-type: none"> <li>• '0' denotes Dealer</li> <li>• '4' denotes Corporate Manager</li> <li>• '5' denotes Branch Manager</li> </ul>	
SequenceNumber	This field contains the time when the markets closed the previous bidding day.	
BrokerStatus	This field contains the current status of the Broker. It is: <ul style="list-style-type: none"> <li>• 'S' for Suspended</li> <li>• 'A' for Active</li> <li>• 'D' for Deactivated</li> </ul>	
BrokerEligibilityPerMarket	This structure specifies the markets that are allowed for the trading member. The trading member is eligible to enter orders in the market that is set to '1'.	

### Logon Error

In case of any error, the following structure is returned.

ERROR RESPONSE (For details refer to *Error Message* in Chapter 2)

Field Name	Description	Comment
TransactionCode	The transaction code is SIGN_ON_REQUEST_OUT (2301).	
ErrorCode	This contains the error number. If the version number is not the same as at that of the host, the version number at the host can be extracted from the Error Message. It will be located in 8 bytes from the 95 <sup>th</sup> 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. Refer to Appendix for the list of error numbers.	

### System Information Download

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

#### System Information Request

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

<b>Structure Name:</b> SYSTEM_INFO_REQ
<b>Transaction Code:</b> SYSTEM_INFORMATION_IN (1600)
<b>Packet Length:</b> 40 bytes
MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2)

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

System Information Response

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

<b>Structure Name:</b> SYSTEM_INFO_DATA
<b>Transaction Code:</b> SYSTEM_INFORMATION_OUT (1601).
<b>Packet Length:</b> 86 bytes
<pre> STRUCT MESSAGE_HEADER (Refer to                 MESSAGE_HEADER in Chapter 2)  STRUCT MARKET_STATUS {     SHORT Normal     SHORT Reserved1     SHORT Reserved2     SHORT Reserved3 } LONG Reserved9 SHORT Reserved4 SHORT Reserved5 SHORT Reserved6 SHORT Reserved7 SHORT Reserved8 SHORT WarningPercent SHORT VolumeFreezePercent SHORT Reserved9 SHORT TerminalIdleTime; LONG BoardLotQuantity LONG TickSize SHORT Reserved10 <b>Note:</b> Use any one of following two structures: <b>For Small Endian Machines:</b> Struct STOCK_ELIGIBLE_INDICATORS {     Reserved - 5 bits     Reserved - 1 bit     Reserved - 1 bit     Reserved - 1 bit     Reserved - 8 bits } <b>For Big Endian Machines:</b> Struct STOCK_ELIGIBLE_INDICATORS                     </pre>

```

    {
        Reserved      - 1 bit
        Reserved      - 1 bit
        Reserved      - 1 bit
        Reserved      - 5 bits
        Reserved      - 8 bits
    }

    SHORT Reserved11
    SHORT InqTimer
    
```

Field Name	Description	Comment
TransactionCode	The transaction code is SYSTEM_INFORMATION_OUT (1601).	
MarketStatus	This field contains the following values: <ul style="list-style-type: none"> <li>• '0' if it is Preopen</li> <li>• '1' if it is Open</li> <li>• '2' if it is Closed</li> <li>• '3' if it is Suspended</li> </ul>	Orders can be entered only in the pre-open state of the market. No orders can be entered for a security when the market is closed.
WarningPercent	This field contains the percentage of the issue size. User gets a warning if the volume in the order is greater than or equal to this value.	
VolumeFreeze Percent	This field contains the percentage of the issue size. User's order goes to freeze if the volume of the order is greater than or equal to this value.	
TerminalIdleTime	This field contains the idle time of the TWS terminal.	
BoardLotQuantity	The regular lot order quantity must be a multiple of this quantity.	
TickSize	The order price must be a multiple of this tick size.	
InqTimer	This field contains the inquiry time of the terminal.	

**Update Local Database Download**

The list of updated securities and categories are downloaded in response to this request. Previous days orders are also downloaded with this request.

**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:

<b>Structure Name:</b> UPDATE_LOCAL_DATABASE
<b>Transaction Code:</b> UPDATE_LOCALDB_IN (7300)
<b>Packet Length:</b> 62 bytes
STRUCT MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2)
LONG LastUpdateSecurityTime
LONG LastUpdateParticipantTime



LONG	LastUpdateCategoryTime
CHAR	RequestForOpenOrders
CHAR	Reserved1
STRUCT	MARKET_STATUS (Refer to <i>System Information Response</i> in Chapter 3)

Field	Description	Comments
TransactionCode	The transaction code is UPDATE_LOCALDB_IN (7300).	
LastUpdate SecurityTime	This field should contain the time when the security information was last updated. This field is set for each security for which information is downloaded. Further download requests can use the latest time to get updated information on the securities. Setting this time to zero results in complete download.	
LastUpdate ParticipantTime	This field should contain the time when the participant information was updated. This field is set for each participant for whom information is downloaded. Further download requests can use the latest time to get updated information on the participants. Setting this time to zero results in complete download.	
LastUpdateCategoryTime	This field should contain the time when the category information was updated. This field is set for each category for which information is downloaded. Further download requests can use the latest time to get updated information on the participants. Setting this time to zero results in complete download.	
RequestForOpenOrders	This field should be set to 'G' if orders of the earlier days are required to be downloaded else it should be set to 'N'	
MarketStatus	This should be the market status received in the previous response. The market status fields are accepted as input to verify if the user has the latest system information. If the user has the latest market status information, the update of the information on the securities and the categories from the specified time are downloaded. If the status information specified is not the latest, the user is updated on the market status alone. Upon receiving the updated market information, the user can request for the updated security status, security, category 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 user does not have the latest market status.

#### Partial System Information Response

This is returned if the market status sent in the previous message 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 in the MARKET STATUS as 'wait till markets are open'. The following structure is returned:

SYSTEM INFORMATION DATA (For details, refer to *System Information Response* in Chapter 3)

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

#### Update Local Database Download

The download comprises of a header, data and the trailer. Each updated security status, category status and GTC order will be sent as a separate message.

#### Update Local Database Header

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

<b>Structure Name:</b> UPDATE_LDB_HEADER
<b>Transaction Code:</b> UPDATE_LOCALDB_HEADER (7307)
<b>Packet Length:</b> 42 bytes(increased from 40)
MESSAGE_HEADER (Refer to <i>MESSAGE_HEADER</i> in Chapter 2) CHAR Reserved1[2]

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

#### Update Local Database Data

The actual data is sent wrapped in another header. The outer header indicates that this message is part of the Update Local Database Data. The inner header indicates the type of data received. The packet size can be of 80 to 512 bytes and the structure is as follows:

MESSAGE\_HEADER (For details, refer to *MESSAGE\_HEADER* in Chapter 2)  
MESSAGE\_HEADER InnerHeader (For details, refer to *MESSAGE\_HEADER* in Chapter 2)  
CHAR Data [436]

Field	Description	Comments
TransactionCode	The transaction code is UPDATE_LOCALDB_DATA (7304).	

InnerTransaction Code	The transaction codes sent are: <ul style="list-style-type: none"> <li>• BCAST_STOCK_MSTR_CHG (7305). It is determined by NSE-Control whether to send this or not. (Refer to <i>Change in Security Master</i> in Chapter 6).</li> <li>• BCAST_STOCK_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 <i>Change of Security Status</i> in Chapter 6).</li> </ul>	
-----------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

New local database download has been introduced:

IPO Category Slab Info:

This download is for Check on front end to reject the order depending on the slab download given in ipo\_cat\_hrchy Slab Information Download at the time of the login

This message is sent by host end whenever there is any category change. The structure sent is:

<b>Structure Name:</b> LDB_CATGLAB_INFO
<b>Transaction Code :</b> IPO_CAT_HRCHY_UPDT_RESPONSE (5728)
<b>Packet Length:</b> 98 bytes
STRUCT MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2 char cSymbol[10]; char cSeries[2]; char cMainCatId[10]; char cSubCatId[10]; double ISharesReserved; double dMinValueSlab; double dMaxValueSlab; char cDeleteFlag; Char filler; };

Field Name	Description	Comment
TransactionCode	The transaction code is IPO_CAT_HRCHY_UPDT_RESPONSE (5728)	
Symbol	This field contains the symbol.	
Series	This field contains the series.	
MainCatId	This field contains the main category id	
SubCatId	This field contains the sub-category id	
SharesReserved	This field indicates the discount declared for the all the subcategory under each main category. The field contains the discount in paise.	This field will be used to calculate the discounted margin amount for a category, if a discount on that category has been declared.

MinValueSlab	Minimum order value for the category	
MaxValueSlab	Maximum order value for the category	
DeleteFlag	This field indicates whether the Category is deleted: <ul style="list-style-type: none"> <li>• 'Y' means deleted</li> <li>• 'N' means not deleted</li> </ul>	
Filler	Unused in frontend	

#### Update Local Database Trailer`

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

<b>Structure Name:</b> UPDATE_LDB_HEADER
<b>Transaction Code:</b> UPDATE_LOCALDB_TRAILER (7308)
<b>Packet Length:</b> 42 bytes
MESSAGE_HEADER (Refer to <i>MESSAGE_HEADER</i> in Chapter 2)
CHAR Reserved[2]

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

#### Message Download

This request is used to download the messages intended for the user from the trading system. When the user makes a request for message download, all the transactions of the user 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 Download Request

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

<b>Structure Name:</b> MESSAGE_DOWNLOAD
<b>Transaction Code:</b> DOWNLOAD_REQUEST (7000)
<b>Packet Length:</b> 48 bytes
MESSAGE_HEADER (Refer to <i>MESSAGE_HEADER</i> in Chapter 2)
DOUBLE SequenceNumber

Field	Description	Comments
TransactionCode	The transaction code is DOWNLOAD_REQUEST (7000).	
SequenceNumber	This contains 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 log-on response message.	
--	------------------------------------------------------------------------------------------------------	--

Message Download Response

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

Message Download Header

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

MESSAGE\_HEADER (For details, refer to MESSAGE\_HEADER in Chapter 2)

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

Message Download Data

The messages are similar to Update Local Database Data. The actual data is sent wrapped in another structure. The outer header indicates that this message is part of the Message Download Data. The inner header indicates the type of data received. The packet size can be of 80 to 512 bytes and the structure is as follows:

MESSAGE\_HEADER (For details, refer to MESSAGE\_HEADER in Chapter 2)  
 MESSAGE\_HEADER InnerHeader (For details, refer to MESSAGE\_HEADER in Chapter 2)  
 CHAR Data [436]

Field	Description	Comments
TransactionCode	The transaction code is MESSAGE_RECORD (7021).	
InnerData	This contains the following messages: <b>Trader Specific Messages</b> <ul style="list-style-type: none"> <li>Logon / Logoff response - Refer to <i>Logon Process</i>, Chapter 3</li> <li>Order Entry, Modification and Cancellation responses - Refer to <i>Order Management</i>, Chapter 4.</li> </ul> <b>Broadcast Messages</b> <ul style="list-style-type: none"> <li>Market Open, Market Close, Preopen Shutdown Message, Broadcast Message String, Broadcast message sent from NSE-Control. Refer to <i>Broadcast Messages</i> in Chapter 6.</li> </ul>	

Message Download Trailer

This indicates that message download is complete. The structure is as follows:

MESSAGE\_HEADER (For details, refer to MESSAGE\_HEADER in Chapter 2)

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

### Logoff Request

The process by which a user quits or signs off from the trading system is called Logoff Process..

The structure sent is:

MESSAGE\_HEADER (For details, refer to *MESSAGE\_HEADER* in Chapter 2)

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

### Logoff Confirmation Response

When the user logs on again, the user receives a packet giving the details of when he logged off.

The structure sent is:

MESSAGE HEADER (For details, refer to *MESSAGE\_HEADER* in Chapter 2)

Field Name	Description	Comment
TransactionCode	The transaction code is SIGN_OFF_REQUEST_OUT (2321).	
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 workstation must be synchronized with this.	

## Chapter 4 Order Management

### Introduction

This section describes about entering new orders, modifying existing orders, and canceling existing orders. The user can begin entering the orders once he has logged on to the trading system and the normal market is in the preopen state.

### Order Entry

Order entry functionality allows the user to place orders in the market. The system accepts the orders from the users and the confirmation is sent to the corresponding users. The orders are placed in the order book with the price and time stamp.

### Order Type

#### Regular Lot

The only order type used in the IPO system is Regular Lot (RL).

### Rules of Order Entry

Order entry is not allowed in the following conditions:

- Market is closed.
- Security is suspended.
- 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 order value limit has exceeded.
- User is disabled.
- User does not exist in trading system.
- Order price is beyond 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.
- Category specified in the order doesn't exist in the Category Master
- Category is suspended.
- Number of orders on the same application number for a particular issue is greater than three.( Not Applicable for OFS )

### Order Entry Request

The format of the order entry request is as follows:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	BOARD_LOT_IN (2000)
<b>Packet Length:</b>	224 bytes
<pre> { MESSAGE_HEADER (Refer to MESSAGE_HEADER in                   Chapter 2)   CHAR    ModCxlBy   SHORT   ReasonCode   CHAR    StartAlpha[2]   CHAR    EndAlpha[2]   STRUCT  SECURITY_INFORMATION           {             CHAR Symbol [10]             CHAR Series [2]           }   DOUBLE  OrderNumber   CHAR    AppINumber [10]   CHAR    Pan [10]   CHAR    Benfld [16]   SHORT   BookType   SHORT   BuySell   DOUBLE  Volume   LONG    Price   DOUBLE  Margin   LONG    EntryDateTime   LONG    LastModified   STRUCT  ORDER_FLAGS   CHAR    PartCategoryId [10]   CHAR    DepPartId [8]   CHAR    Depository [10]   CHAR    RtgsCode [25]   CHAR    NnfAppCount   CHAR    Filler[11]   CHAR    BrokerId [5]   DOUBLE  NnfField   SHORT   BranchId   LONG    TraderId(changed from SHORT to LONG)   SHORT   ProClient } </pre>	
<b>For Small Endian Machines</b>	
<pre> STRUCT ORDER_FLAGS {   USHORT   uiFrozen      1   USHORT   uiModified    1   USHORT   uiMatchedInd  1   USHORT   uiGTC         1   USHORT   Reserved1     1   USHORT   uiFiller      1   USHORT   uiMkt         1   USHORT   uiATO         1   USHORT   uiFiller1     4   USHORT   uiTMCP        1   USHORT   uiSuspended   1   USHORT   uiFiller2     2 } </pre>	



```

For Big Endian Machines
STRUCT ORDER_FLAGS
{
    USHORT    uiATO    1
    USHORT    uiMkt    1
    USHORT    uiFiller  1
    USHORT    Reserved1 1
    USHORT    uiGTC    1
    USHORT    uiMatchedInd 1
    USHORT    uiModified 1
    USHORT    uiFrozen  1
    USHORT    uiFiller1 4
    USHORT    uiTMCP    1
    USHORT    uiSuspended 1
    USHORT    uiFiller2  2
}

```

Field Name	Description	Comment
TransactionCode	The transaction code should be BOARD_LOT_IN (2000).	
Modified / CancelledBy(Mod CxlBy)	This field should denote who has modified or cancelled a particular order. During modification/cancellation, it should have one of the following values: <ul style="list-style-type: none"> <li>• 'T' for User</li> <li>• 'B' for Branch Manager</li> <li>• 'M' for Corporate Manager</li> </ul> During order entry, this field should be blank.	Used only during order modification or cancellation.
ReasonCode	This field contains the reason code for a particular order request rejection. This, along with the error code, has the details regarding the error. During order entry, this field should be zero.	Refer to <i>List of Reason Codes</i> in Appendix.
StartAlpha	During order entry, this field should be blank.	
EndAlpha	During order entry, this field should be blank.	
Security Information	This structure contains the following fields: <ul style="list-style-type: none"> <li>• Symbol</li> <li>• Series.</li> </ul> The Symbol field should contain the name of the security. The series field should have one of the following values: <ul style="list-style-type: none"> <li>• EQ – Equity shares</li> <li>• RV – Reverse book building shares</li> <li>• IS – Offer For Sale (OFS)</li> <li>• RS – Offer For Sale(OFS)</li> </ul>	
OrderNumber	Order Number is a unique identification for an Order. During order entry this field should be zero.	The order number is assigned to the order in the order requested

Field Name	Description	Comment
		response packet.
AppNumber	This field should contain the application number of the order. The application number field is unique to a broker, that is, two brokers cannot place orders on the same application number.	N.A for Offer For Sale Security. This field should be left blank in the order packet structure.
PAN	This field should contain the PAN of the client. The PAN shall be of compulsory 10 lengths with first five places as alphabets, next four places as numbers and last place as alphabet. However, if it is not in the above format, an additional check shall be performed to ensure that the value is among the list of valid values allowed for the depository	N.A for OFS Security
Benfld	This field should contain the beneficiary ID of the user. If Depository Name is “NSDL”, it shall be of compulsory 8 length (Not Applicable for OFS) If Depository Name is “CDSL”, it shall be of compulsory 16 length.(Not Applicable for OFS)  For OFS security, this field would be used for CP Code. If PRO Order the field should be populated with trading member id. For CLI order by trading member it should be populated with its own Trading Member Code Value ‘NSEIL’, ‘NSE’ is not allowed in CP code	FOR OFS this should be of 12 lengths and remaining 4 spaces.  The CP Code cannot be left blank, cannot be NSEIL,NSE and length should be less or equal to 12 alphachar.  For RS securities, CP orders are not allowed so this field should always contain its own Trading Member id.
BookType	In IPO, only RL book type exists. Hence, this field should be made 1 for the order entry request.	
Buy / Sell Indicator	This field should specify whether the order is a buy order or a sell order. It should take one of the following values: <ul style="list-style-type: none"> <li>• ‘1’ for Buy order</li> <li>• ‘2’ for Sell order</li> </ul> FOR OFS this will be a BUY Order.	
Volume	This field should contain the order quantity. The quantity must always be in multiples of Regular Lot size and greater than the minimum lot size.  Volume greater than the issued capital cannot be entered into the system. The order will go for a freeze if the quantity exceeds the freeze range specified by NSE-Control.  For OFS Security a single order volume should not exceed the issue size.  Also, for OFS Security with RS series, a single	

Field Name	Description	Comment
	order value (order volume * order price) should not exceed Rs 2 Lakhs.	
Price	<p>This field should contain the price at which the order is placed. The price must be a multiple of the Tick Size.</p> <p>The price should be multiplied by 100 before sending the order entry request to the host.</p> <p>To enter a Market order, the price should be set to zero.</p> <p>For OFS, <b>for general category</b> market orders cannot be placed. Orders should be at price points.</p> <p><b>For RS security, cut-off orders can be placed for specific securities. The price should be set to zero for the same.</b></p>	
Margin	This field should specify the margin amount. Maximum margin amount that can be entered is 89,99,99,99,99,99,999.	N.A for OFS Security
EntryDateTime	<p>This denotes the time when the order first enters the trading system.</p> <p>This field should be set to zero for the order entry request.</p>	
LastModified Time	<p>In the case of order entry, it will be the same as EntryDateTime.</p> <p>If the order has been modified it has the time when the order was last modified. It is the time in seconds from midnight of January 1, 1980.</p> <p>This field should be set to zero for the order entry request.</p>	
ORDER_FLAGS	<p>This structure contains the following fields:</p> <ul style="list-style-type: none"> <li>• Frozen</li> <li>• Modified</li> <li>• MatchedIndicator</li> <li>• GTC</li> <li>• Reserved1</li> <li>• Filler</li> <li>• Mkt</li> <li>• ATO</li> <li>• Filler1</li> <li>• TMCP</li> <li>• Suspended</li> <li>• Filler2</li> </ul> <p>All the fields except GTC should be set to zero while sending the order entry request for IPO.</p> <p>For OFS Security, GTC bit should be set to 0 and Reserved1 should be set to 1.</p>	<p>The Reserved1 field should be set to 1 to indicate this is an OFS ORDER.</p> <p>For PRO/CLI order TMCP flag will be always set to 0 to indicate blocking collateral of TM.</p> <p>For Institutional Order with 0% Margin ( CP Orders)</p> <ul style="list-style-type: none"> <li>• TM CP Flag field would be set to 1.</li> </ul> <p>For Institutional Orders with 100% margin (CP Orders)</p>

Field Name	Description	Comment
		<ul style="list-style-type: none"> <li>TM CP Flag field would be set to 0.</li> </ul> <p>For RS series, only 100% margin is allowed so TMCP flag should be always set to 0.</p>
PartCategoryId	This field should contain the Category for which the user is placing the order.	No Category Reservation. Hence not applicable for OFS
DepPartId	This field should contain the Depository Participant ID If depository name is NSDL, Depository ID is mandatory and length should be 8 If the depository name is CDSL, Depository ID is not applicable	Not Applicable. To be left blank.
Depository	This field should contain the name of the Depository. Depository name shall be either CDSL or NSDL.	Not Applicable For OFS Security .  <b>Can be reused for echoback for OFS only.</b>
RTGS No	This field contains the RTGS No or the Cheque No .It is mandatory  For OFS this will contain the Account Number. In case of a PRO order the RTGS field would be auto populated with the Trading Member ID.  In case of CLI Order this should contain client code other than the trading member code. Value 'NSEIL' , 'NSE' is not allowed.	For EQ i.e buy orders this field should contain 16 character RTGS number and remaining 9 spaces.  For RV i.e sell orders this field should contain 25 character client names.  For IS/RS this field would be reused for Account Number with a field length of 10 and remaining 15 spaces. The Account number field cannot be left blank, cannot be NSEIL, cannot be NSE
NnfAppCount	Reserved for NNF.	
BrokerId	This field should contain the trading member ID.	
NnfField	This field should contain a 15 digit unique identifier which should be arranged in the following manner. <ul style="list-style-type: none"> <li>First six digits should contain pin code of the user</li> <li>Next three digits should contain the branch number</li> <li>Next three digits should contain the</li> </ul>	The first 12 digit should be populated to '111111111111' for internet trading.

Field Name	Description	Comment
	<p>terminal number</p> <ul style="list-style-type: none"> <li>The 13<sup>th</sup> digit should contain 0 for automated trading and 1 for non-automated trading</li> <li>The 14<sup>th</sup> digit should contain vendor code. This should be 0 for the members developing inhouse software</li> </ul> <p>The 15<sup>th</sup> digit should be 0 for use of Exchange</p>	
BranchId	<p>This field should contain the branch number to which the broker belongs.</p> <p>This field is changed from 2 digit to 3 digit</p>	
TraderId	<p>This field should contain the User ID of the user.</p>	
ProClient	<p>This field should be set to '1' during order entry. This is currently as applicable for IPO.</p> <p>In case of OFS, for PRO Order field should be set to 2 and for Client Order field should be set to 1.</p> <p>If Pro Order i.e. field set to 2, the Account number field should be of 10 length and must contain member code.</p> <p>If CLI Order i.e field set to 1, the Account Number field should be of 10 lengths and should contain client code other than member code.</p>	

**Order Terms**

Order Term	If to the term contains	Attribute Represented (in the packet received)	Should Be set to (while entering order)
Market	1	Market order	0
ATO	1	Market order in Preopen in IPO and Cut off order in OFS	0
Frozen	1	The order has gone for a freeze	0
Modified	1	The order has been modified	0
Reserved1	1	The order is an OFS Order.	1 for OFS, else 0 for Normal IPO. If Reserved1 field is set to 1 then GTC bit to be set as 0.

### Order Entry Response

The primary response to order entry is Order Requested message. 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 user has resulted in a quantity freeze and is waiting for the approval of the exchange. The order error response is given when the entered order is rejected by the host. The reason for the rejection is given in the error code and/or reason code.

### Order Requested Response

This response is sent back when an order is requested. This does not imply that the order has been confirmed. However, it implies that the order has reached the trading system. The message sent will be of the following format:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	BOARD_LOT_OUT (2001)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is BOARD_LOT_OUT (2001).	
OrderNumber	This field contains the order number assigned to the order.	
EntryDateTime	This field contains the date and time when the order entered the system.	

### 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 to the user. This is an unsolicited message as well. The message sent is as follows:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_CONFIRMATION_OUT (2073)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_CONFIRMATION_OUT (2073).	
EntryDateTime	This field contains the date and time when the order entered the system.	
OrderNumber	Order Number is a unique identification for an Order. This field contains the same number as the number returned in the requested	

	response.	
Price	<p>If a market order is entered, the host will set the 'ATO' bit in ORDER TERMS.</p> <p>If it is a priced order the order, it will get confirmed at that price.</p> <p>In OFS, for RS series, if cut-off order is entered, the host will set the 'ATO' bit in ORDER TERMS. Price field will have value 2147483647.</p>	
OrderTerms	The flags are set as discussed in <i>Order Entry Request</i> in Chapter 4.	

**Note:**

The reason code in the structure can be used to differentiate orders that get freeze approval from orders that get 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 is placed by the user or is modified by him and is awaiting approval from the exchange. Exchange approval of the order results in a Freeze Approval response and rejection results in a Freeze Reject response. These responses are sent as unsolicited messages.

The format sent is as follows:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	FREEZE_TO_CONTROL (2170).
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is FREEZE_TO_CONTROL (2170).	
OrderTerms	For details on ORDER_ENTRY_REQUEST, refer to <i>Order Entry Request</i> in Chapter 4	

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:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_ERROR_OUT (2231).
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_ERROR_OUT (2231).	
ErrorCode	This field contains the error number.	Refer to Appendix for a list of the error numbers.



## Order Modification

Order modification functionality enables the user to modify the orders entered by him.

### Rules of Order Modification

The following modifications are not allowed:

- Modifying Symbol and Series.
- BM modifying CM's orders.
- Dealer modifying BM's orders.
- Dealer modifying CM's orders.
- Modifying non existing order.
- Modifying an order in such a way that it results in the branch order value to be exceeded.
- Modifying an order in such a way that it results in the user limit to be exceeded.
- Modifying deactivated broker's orders.

The following modifications are allowed.

- Modifying category field.

### Order Modification Request

The user can modify the quantity, price and other attributes of an order by specifying the order number of the order to be modified.

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_MOD_IN (2040)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code should be ORDER_MOD_IN (2040).	
Modified / CancelledBy(ModCxlBy)	This field should denote who has modified or cancelled a particular order.  During modification/cancellation, it should have one of the following values: <ul style="list-style-type: none"> <li>• 'T' for User</li> <li>• 'B' for Branch Manager</li> <li>• 'M' for Corporate Manager</li> </ul>	
OrderNumber	Order Number is the identity of the order to be modified. This field should have the order number of the order to be modified.	
EntryDateTime	Entry Date Time is the date and time when the order entered the trading system. This is available in Order Confirmation/ Order Modification Confirmation response.	

Field Name	Description	Comment
<b>Note:</b> The other fields of Order Modification request are the same as Order Entry request.		

Order Modification Response

This response is sent back when an order modification is requested. This does not imply that the order modification has been confirmed. The response can be order modification confirmation, order freeze, order modification error or one of the general error responses. The order modification error response is given when the modified order is rejected by the trading system. The reason for the rejection is given in reason code and/or the error code. The message sent will be of the following format:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_MOD_OUT (2041)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_MOD_OUT (2041).	

Order Modification Confirmation Response

Successful modification of the order results in Order Modification Confirmation response. 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, order modification confirmation will be received as an “Unsolicited Message”. The structure sent is as follows:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_MOD_CONFIRM_OUT (2074)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_MOD_CONFIRM_OUT (2074).	
LastModifiedTime	It is the time when the order was last modified (in seconds from midnight of January 1, 1980).	

Order Modification Error Response

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

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_MOD_REJ_OUT (2042)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_MOD_REJ_OUT (2042).	

Note: Trading Member shall have the option to modify / cancel the orders placed for OFS during order collection period. Order modification shall be applicable only on quantity, price, client code (RTGS No field of order packet.) and CP code (benf\_id field of order packet)

Order Modification is possible only if the order is being modified from PRO to PRO, CLI to CLI or CP to CP.

For Institutional orders with 0% margin (Type B):

- Modification is allowed at any time before and after custodian confirmation till the security mod/cxl cut off time.
- Modification of an approved order would again be considered unapproved and approval for the same will be sought.
- The quantity and price can be modified upwards only.
- Downward revision of quantity and price is not allowed.
- At EOD all the unapproved orders would be retained.

For Institutional orders with 100% margin (Type A):

- Modification is allowed at any time before and after custodian confirmation till the security mod/cxl cut off time.
- Modification of an approved order would again be considered unapproved and approval for the same will be sought.
- Quantity and price can be modified upwards and downwards.
- At EOD all the unapproved orders will be cancelled.

Any other field apart from the one mentioned above will result in order modification rejection.

**Order Cancellation**

The user can cancel order entered by him by specifying the Order number.

For Institutional orders with 0 % margin (Type B) orders cannot be cancelled at any time.

For Institutional orders with 100 % margin (Type A) orders can be cancelled at any time till mod/cxl cut off time.

**Rules of Order Cancellation**

- CM can cancel BM's and User's order, but BM and User cannot cancel CM's order.
- BM can cancel User's order, but User cannot cancel BM's order.
- Deactivated broker cannot cancel his order while his status is deactivated.

**Order Cancellation Request**

The format of the message is as follows:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_CANCEL_IN (2070)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_CANCEL_IN (2070).	
OrderNumber	This field should contain the identity of the order to be cancelled.	

#### Order Cancellation Response

The response can be one of order cancellation confirmation, order cancellation error or one of the general error responses.

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_CANCEL_OUT (2071)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_CANCEL_OUT (2071).	

#### Order Cancellation Confirmation Response

Successful cancellation of order results in Order Cancellation Confirmation Response. The message sent is as follows:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_CANCEL_CONFIRM_OUT (2075)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

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

#### 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:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_CXL_REJ_OUT (2072).
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_CXL_REJ_OUT (2072).	

## Chapter 5 Unsolicited Messages

### Introduction

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

### Freeze Approve Response

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

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_CONFIRMATION (2073)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

Field Name	Description	Comment
TransactionCode	The transaction code is ORDER_CONFIRMATION (2073).	
LastModifiedTime	This field contains the time when the order was last modified.	

### 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:

<b>Structure Name:</b>	ORDER_ENTRY_REQUEST
<b>Transaction Code:</b>	ORDER_ERROR (2231)
<b>Packet Length:</b>	224 bytes

(For details on ORDER\_ENTRY\_REQUEST, refer to *Order Entry Request* in Chapter 4)

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

### Custodian Approve/Reject Response

The message is sent when custodian approves or rejects Type A or Type B order.

<b>Structure Name:</b>	TRADER_INTERACTIVE_MESSAGE_CP
<b>Transaction Code:</b>	RMS_APPROVE_REJECT(2079)
<b>Packet Size:</b>	299 bytes
	MESSAGE_HEADER (Refer to <i>MESSAGE_HEADER</i> in Chapter 2)
LONG	TraderId(changed from SHORT to LONG)
CHAR	cActionCode[ 3]
CHAR	cFiller
SHORT	BroadCastMessage Length
CHAR	BroadCastMessage [239]
CHAR	cFiller1
DOUBLE	OrderNumber

```

STRUCT ST_APPR_REJ_BIT

Note: Use any one of following two ST_APPR_REJ_BIT
structures:

For Small Endian Machines

    STRUCT ST_APPR_REJ_BIT
    {
        USHORT uiFiller           : 7 bits
        USHORT uiIs_unapproved    : 1 bit
    }

For Big Endian Machines

    STRUCT ST_APPR_REJ_BIT
    {
        USHORT uiIs_unapproved    : 1 bit
        USHORT uiFiller           : 7 bits
    }
    
```

Field Name	Description	Comment
TraderId	This is the trader id of the user.	
cActionCode	This will be used to filter messages in message area of TWS front end screen.	For example, 'SYS' - system
BroadCastMessage Length	This is the length of the bcast message to be sent to front end.	
BroadCastMessage	It contains the confirmation or rejection message.	
OrderNumber	The order_number of the order.	The order number is assigned to the order in the order requested response packet.
uils_unapproved	This bit will tell the custodial status of CP orders.	For approved Order from custodian this bit will be 0.  For rejected order, this will be 1.

Interactive/Broadcast Messages Sent From Control

A message can be sent to the user(s) from the NSE-Control Workstation. If it is sent to all the users, it comes as a broadcast in the structure BROADCAST\_MESSAGE. (Refer to *Broadcast*, Chapter 6). If the message is sent to a particular user, it comes as an interactive message in the following structure:

<b>Structure Name:</b> TRADER_INTERACTIVE_MESSAGE
<b>Transaction Code:</b> CTL_MSG_TO_TRADER (5295) BCAST_JRNL_VCT_MSG (6501)
<b>Packet Size:</b> 289 bytes
MESSAGE_HEADER (Refer to <i>MESSAGE_HEADER</i> in Chapter 2) LONG TraderId(changed from SHORT to LONG) CHAR cActionCode[ 3] Reserved SHORT BroadCastMessage Length CHAR BroadCastMessage [239]

Field Name	Description	Comment
TransactionCode	The transaction code is: CTL_MSG_TO_TRADER (5295) for interactive messages and BCAST_JRNL_VCT_MSG (6501) for broadcasting messages.	

## Chapter 6 Broadcast

### Introduction

This section describes the various Broadcast messages with their structures.

### General Message Broadcast

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

<b>Structure Name:</b> GENERAL_BROADCAST_MESSAGE	
<b>Transaction Code:</b> BCAST_JRNL_VCT_MSG (6501)	
<b>Packet Length:</b> 297 bytes	
STRUCT	MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2)
SHORT	BranchNumber
CHAR	BrokerNumber [5]
CHAR	ActionCode [3]
CHAR	Filler1 [4]
STRUCT	BROADCAST_DESTINATION
<b>Note:</b> Use any one of following two BROADCAST_DESTINATION structures:	
<b>For Small Endian Machines</b>	
STRUCT	BROADCAST_DESTINATION
{	
Reserved	: 4 bits
Journaling Required	: 1 bit
Tandem	: 1 bit
Control Workstation	: 1 bit
Trader Workstation	: 1 bit
Reserved	: 8 bit
}	
<b>For Big Endian Machines</b>	
STRUCT	BROADCAST_DESTINATION
{	
Trader Workstation	: 1 bit
Control Workstation	: 1 bit
Tandem	: 1 bit
Journaling Required	: 1 bit
Reserved	: 4 bits
Reserved	: 8 bit
}	
SHORT	BroadcastMessageLength
CHAR	BroadcastMessage [239]



Field Name	Description	Comment
TransactionCode	The transaction code is: BCAST_JRNL_VCT_MSG (6501).	
BranchNumber	This field should contain the branch number of the user to which he belongs.	
BrokerNumber	This field should contain the Trading Member ID of the broker.	
ActionCode	This field indicates action taken.	For example, 'SYS' - system
Broadcast Destination	This field contains the destination of the message, that is, the NNF TWS.	
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 by the host end when any global operating parameters are changed or status of market is changed. The structure sent is:

SYSTEM INFORMATION DATA (For details refer to *System Information Response* in Chapter 3)

<b>Structure Name:</b> SYSTEM_INFO_DATA
<b>Transaction Code:</b> BCAST_SYSTEM_INFORMATION_OUT (7206)
<b>Packet Length:</b> 86 bytes

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

Change in Security Master

This is sent by the host whenever the parameter of any security is changed or any new security is added. The structure is as follows:

<b>Structure Name:</b> SECURITY_UPDATE_INFORMATION	
<b>Transaction Code :</b> BCAST_STOCK_MSTR_CHG (7305)	
<b>Packet Length:</b> 212 bytes	
STRUCT	MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2)
SHORT	Token
STRUCT	SECURITY_INFORMATION
	{
	CHAR Symbol [10]
	CHAR Series [2]
	}
SHORT	InstrumentType
SHORT	Reserved1
DOUBLE	IssuedCapital

```

SHORT   WarningPercent
SHORT   FreezePercent
CHAR    CreditRating [12]

```

**Note:** Use any one of following two SEC\_ELIGIBILITY\_PER\_MARKET structures:

**For Small Endian Machines:**

```

STRUCT  SEC_ELIGIBILITY_PER_MARKET [ 4 ]
{
    Reserved   : 7 bits
    Eligibility : 1 bit
    SHORT     Status
}

```

**For Big Endian Machines:**

```

STRUCT  SEC_ELIGIBILITY_PER_MARKET [ 4 ]
{
    Eligibility : 1 bit
    Reserved   : 7 bits
    SHORT     Status
}

```

```

SHORT   IssueRate
LONG    IssueStartDate
LONG    Reserved2
LONG    Reserved3
LONG    BoardLotQuantity
LONG    TickSize
CHAR    Name [25]
CHAR    Reserved
LONG    Reserved4
LONG    Reserved5
LONG    Reserved6
LONG    Reserved7
LONG    Reserved8
LONG    Reserved9
LONG    Reserved10

```

**Note:** Use any one of following two ELIGIBILITY\_INDICATORS structures:

**For Small Endian Machines:**

```

STRUCT  ELIGIBILITY_INDICATORS
{
    Reserved       :      5 bits
    Reserved22    :      1 bit
    Reserved23    :      1 bit
    Reserved11    :      1 bit
    Reserved      :      8 bits
}

```

**For Big Endian Machines:**

```

STRUCT  ELIGIBILITY_INDICATORS
{
    Reserved11:    1 bit
    Reserved23    : 1 bit
    Reserved22    : 1 bit
}

```

```

        Reserved      : 5 bits
        Reserved      : 8 bits
    }

    LONG      Reserved12
    LONG      Reserved13

Note: Use any one of following two PURPOSE structures:
For Small Endian Machines:
    STRUCT  PURPOSE
    {
        Reserved14      : 2 bit
        Reserved15      : 1 bit
        Reserved16      : 1 bit
        Reserved17      : 1 bit
        Reserved18      : 1 bit
        Reserved19      : 1 bit
        Reserved20      : 1 bit
        Reserved21      : 8 bit
    }

For Big Endian Machines:
    STRUCT  PURPOSE
    {
        Reserved20      : 1 bit
        Reserved19      : 1 bit
        Reserved18      : 1 bit
        Reserved17      : 1 bit
        Reserved16      : 1 bit
        Reserved15      : 1 bit
        Reserved14      : 2 bit
        Reserved21      : 8 bit
    }

    LONG  LocalUpdateDateTime
    CHAR  DeleteFlag
    CHAR  Remark [25]
    
```

Field Name	Description	Comment
TransactionCode	The transaction code is BCAST_STOCK_MSTR_CHG (7305).	
Token	Token number of the security being updated. This is unique for a particular symbol-series combination.	
Security Information	This structure contains the following fields: <ul style="list-style-type: none"> <li>• Symbol</li> <li>• Series</li> </ul>	

Field Name	Description	Comment
	The Symbol field should contain the name of the security. The Series field should have one of the following values: <ul style="list-style-type: none"> <li>• EQ – Equity shares</li> <li>• RV – Reverse book building shares</li> <li>• IS – Offer For Sale</li> <li>• RS – Offer For Sale</li> </ul>	
InstrumentType	Instrument type of the security will be set to '0' (Equities)	
IssuedCapital	This field contains the issue size of the security.	
WarningQuantity	This field contains the Warning Quantity percentage.	
FreezeQuantity	This field contains the Freeze Quantity percentage.	
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 i.e. it will be set to 1 for normal market.	
Status	<ul style="list-style-type: none"> <li>• '1' – Pre-open</li> <li>• '2' – Open</li> <li>• '3' – Suspended</li> </ul>	
IssueRate	This field contains the Minimum quantity for an order in the issue.	
IssueStartDate	This field contains the Date of issue of the security.	
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.	
LocalUpdateDateTime	This field contains the update date-time of the local database.	
DeleteFlag	This indicates whether the security is deleted or not. <ul style="list-style-type: none"> <li>• 'N' – Active</li> <li>• 'Y' – Deleted</li> </ul>	
Remark	Remarks.	

Change in Category Status

This message is sent by host end whenever there is any category change. The structure sent is:

<b>Structure Name:</b> CATEGORY_UPDATE_INFORMATION	
<b>Transaction Code :</b> BCAST_CATEG_MSTR_CHG (7309)	
<b>Packet Length:</b> 81 bytes	
STRUCT	MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2)
CHAR	CategoryId [10]
CHAR	CategoryName [25]
CHAR	CategoryStatus
LONG	CategoryUpdateDateTime
CHAR	DeleteFlag

Field Name	Description	Comment
TransactionCode	The transaction code is BCAST_CATEG_MSTR_CHG (7309).	
CategoryId	This field contains the category ID.	
CategoryName	This field contains the name of the category which is changed.	
CategoryStatus	This field contains the status of the category which has changed: <ul style="list-style-type: none"> <li>• 'S' – Suspended</li> <li>• 'A' – Active</li> </ul>	
CategoryUpdateDateTime	This field contains the time when the category information was changed. It is number of seconds from January 1, 1980	
DeleteFlag	This field indicates whether the Category is deleted: <ul style="list-style-type: none"> <li>• 'Y' means deleted</li> <li>• 'N' means not deleted</li> </ul>	

Change in Security Status

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

<b>Structure Name:</b> SECURITY_STATUS_UPDATE_INFORMATION	
<b>Transaction Code:</b> BCAST_STOCK_STATUS_CHG (7320) BCAST_STOCK_STATUS_CHG_PREOPEN (7210)	
<b>Packet Length:</b> 472 bytes	

```

STRUCT    MESSAGE_HEADER (Refer to
          MESSAGE_HEADER in Chapter 2)
SHORT    NumberOfRecords
STRUCT    TOKEN_AND_ELIGIBILITY [43]
        {
          SHORT    Token
          SEC_STATUS_PER_MARKET [4]
                {
                  SHORT    Status
                }
        }
    
```

Field Name	Description	Comment
TransactionCode	The transaction codes are: BCAST_STOCK_STATUS_CHG (7320). BCAST_STOCK_STATUS_CHG_PREOPEN (7210).	
NumberOfRecords	This field contains the number of tokens and their eligibility.	The structure TOKEN AND ELIGIBILITY is repeated these many times.
Token	This field contains the token number of the security which is changed.	
Status	This field contains the new status of the security. This can take the following values: <ul style="list-style-type: none"> <li>• '1' - Pre-open</li> <li>• '2' - Open</li> <li>• '3' - Suspended</li> </ul>	

Change of Market Status

Whenever the status of the market changes, the following structure is changed and sent by host:

<b>Structure Name:</b> STRUCT BROADCAST_VCT_MSGS
<b>Transaction Code:</b> BC_CLOSE_MSG (6521) / BC_PRE_OR_POST_DAY_MSG (6531) / BC_PRE_OPEN_ENDED (6571)
<b>Packet Length:</b> 297 bytes
<pre> STRUCT    BROADCAST_HEADER STRUCT    SECURITY_INFORMATION         {           CHAR    Symbol    [10]           CHAR    Series    [2]         } SHORT    Market Type <b>Note:</b> Use any one of following two BROADCAST_DESTINATION structures:  <b>For Small Endian Machines:</b> STRUCT    BROADCAST_DESTINATION     </pre>

```

        {
            Reserved          : 4 bits
            Journaling Required : 1 bit
            Tandem             : 1 bit
            Control Workstation : 1 bit
            Trader Workstation : 1 bit
            Reserved          : 8 bits
        }

For Big Endian Machines:
    STRUCT BROADCAST_DESTINATION
    {
        Trader Workstation : 1 bit
        Control Workstation : 1 bit
        Tandem             : 1 bit
        Journaling Required : 1 bit
        Reserved          : 4 bits
        Reserved          : 8 bits
    }

    SHORT BroadcastMessageLength
    CHAR  BroadcastMessage [239]
    
```

Field Name	Description	Comment
TransactionCode	The transaction codes are: <ul style="list-style-type: none"> <li>• BC_CLOSE_MSG (6521). This is sent when the market is closed.</li> <li>• BC_PRE_OR_POST_DAY_MSG (6531). This is sent when the market is preopen.</li> <li>• BC_PRE_OPEN_ENDED (6571). This is sent when the preopen period ends.</li> </ul>	
SecurityInformation	This contains the symbol and series of a security.	
MarketType	This contains the type of market; for Normal market – ‘1’.	
Broadcast Destination	This field is set to ‘1’ and signifies the message is for the TWS.	
BroadcastMessage Length	This contains the length of the broadcast message.	
BroadcastMessage	This contains the contents of the broadcast message.	

**Market By Price Update**

The information regarding the best buy orders and the best sell orders is sent by the host in the following format.

<b>Structure Name:</b> BROADCAST_MBO_MBP_INFORMATION
<b>Transaction Code:</b> BCAST_MBO_MBP_UPDATE (7200)
<b>Packet Length:</b> 491 bytes

STRUCT MESSAGE\_HEADER (Refer to MESSAGE\_HEADER in Chapter 2)

STRUCT INTERACTIVE\_MBO\_DATA

```
{
    SHORT    Token
    SHORT    BookType
    SHORT    TradingStatus
    LONG     Reserved1
    LONG     Reserved2
    CHAR     Reserved17
    LONG     Reserved18
    LONG     Reserved3
    LONG     Reserved4
    LONG     Reserved5
    SHORT    Reserved6
    SHORT    Reserved7
    SHORT    Reserved8
    LONG     Reserved9
    LONG     Reserved10
    LONG     Reserved11
    LONG     Reserved12
    CHAR     RecordBuffer [ sizeof ( MBO_INFO ) * 10 ]
}
```

```
    SHORT    TraderId
    LONG     Qty
    LONG     Price
```

**Note:** Use any one of following two  
MBO\_MBP\_TERMS structures

**For Small Endian Machines:**

STRUCT MBO\_MBP\_TERMS

```
{
    Reserved    : 6 bits
    Reserved    : 1 bit
    Reserved    : 1 bit
    Reserved    : 8 bits
}
```

**For Endian Machines:**

STRUCT MBO\_MBP\_TERMS

```
{
    Reserved    : 6
    Reserved    : 1
    Reserved    : 1
    Reserved    : 8
}
```

```
    LONG Min Fill Qty
}
```

```
    CHAR     Record Buffer [ sizeof ( MBP_INFO ) * 10 ]
}
```

```
    DOUBLE Qty
    LONG Price
    LONG NoOfOrders
```



```

        SHORT BbFlag
        Reserved 2bytes
    }
    SHORT    TotalBuyFlag
    SHORT    TotalSellFlag
    DOUBLE   TotalBuyQty
    DOUBLE   TotalSellQty
Note: Use any one of following two INDICATOR structures:

For Small Endian Machines:
    STRUCT  INDICATOR
    {
        Reserved      : 4 bits
        Reserved      : 1 bit
        Reserved      : 1 bit
        Reserved      : 1 bit
        Reserved      : 1 bit
        Reserved      : 8 bits
    }

For Big Endian Machines:
    STRUCT  INDICATOR
    {
        Reserved      : 1 bit
        Reserved      : 1 bit
        Reserved      : 1 bit
        Reserved      : 1 bit
        Reserved      : 4 bits
        Reserved      : 8 bits
    }

    LONG    Reserved13
    LONG    Reserved14
    LONG    Reserved15
    LONG    Reserved16
}

//END OF BROADCAST_MBO_MBP_INFORMATION
    
```

Field Name	Description	Comment
TransactionCode	The transaction code is BCAST_MBO_MBP_UPDATE (7200).	
Token	This is a unique number given to a particular symbol-series combination.	
BookType	This contains the book type. In IPO, the book type is '1'	
TradingStatus	This contains trading status of the security. It can be: <ul style="list-style-type: none"> <li>• Preopen - 1</li> <li>• Open - 2</li> </ul>	

Field Name	Description	Comment
	<ul style="list-style-type: none"> <li>Suspended - 3</li> </ul>	
RecordBuffer (MBO INFORMATION )	This field contains the 5 best Buy orders and 5 best Sell orders from the order book.	Reserved for future use.
RecordBuffer (MBP INFORMATION )	This field contains the 5 best Buy prices and 5 best Sell prices from the order book.	
TotalBuyFlag	This field contains the value zero.	
TotalSellFlag	This field contains the value zero.	
TotalBuyQuantity	This field contains the total quantity of buy orders for the security.	
TotalSellQuantity	This field contains the total quantity of sell orders for the security.	

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

<b>Structure Name:</b> BROADCAST_ONLY_MBP
<b>Transaction Code:</b> BCAST_ONLY_MBP (7208)
<b>Packet Length:</b> 333bytes
<pre> MESSAGE_HEADER (Refer to MESSAGE_HEADER in Chapter 2) SHORT NoOfRecords INTERACTIVE_ONLY_MBP_DATA [2] {     SHORT Token     SHORT BookType     SHORT TradingStatus     LONG Reserved1     LONG Reserved2     CHAR Reserved17     LONG Reserved18     LONG LastTradeQuantity     LONG Reserved4     LONG Reserved5     SHORT Reserved6     SHORT Reserved7     SHORT Reserved8     LONG Reserved9     LONG Reserved10     LONG Reserved11     LONG Reserved12     CHAR RecordBuffer [ sizeof ( MBP_INFO ) * 10 ]     SHORT BbTotalBuyFlag     SHORT BbTotalSellFlag         </pre>

```

DOUBLE TotalBuyQuantity
DOUBLE TotalSellQuantity

Note: Use any one of following two MBP_INDICATOR structures:

For Small Endian Machines:
STRUCT INDICATOR
{
    Reserved      : 4 bits
    Reserved      : 1 bit
    Reserved      : 1 bit
    Reserved      : 1 bit
    Reserved      : 1 bit
    Reserved      : 8 bits
}

For Big Endian Machines:
STRUCT INDICATOR
{
    Reserved      : 1 bit
    Reserved      : 1 bit
    Reserved      : 1 bit
    Reserved      : 1 bit
    Reserved      : 4 bits
    Reserved      : 8 bits
}

LONG Reserved13
LONG Reserved14
LONG Reserved15
LONG Reserved16
}
} End of BROADCAST_ONLY_MBP
    
```

<b>Structure Name: MBP_INFO</b>
DOUBLE Qty
LONG Price
LONG NoOfOrders
SHORT BbFlag
Reserved 2bytes

Field Name	Description	Comment
TransactionCode	The transaction code is BCAST_ONLY_MBP (7208).	
NoOfRecords	This number contains the number of securities sent.	
Token	This is a unique number given to a particular symbol-series combination.	
BookType	This field contains the book type 1 (Regular Lot)	
TradingStatus	This contains trading status of the security.	

Field Name	Description	Comment
	It can be: <ul style="list-style-type: none"> <li>• Preopen - 1</li> <li>• Open - 2</li> <li>• Suspended - 3</li> </ul>	
RecordBuffer (MBP INFORMATION )	This field contains 5 best Buy prices and 5 best Sell prices from the order book.	
BbTotalbuyFlag	This field contains numeric zero.	
BbTotalsell Flag	This field contains numeric zero.	
TotalBuyQuantity	This field contains the total quantity of buy orders in a security.	
TotalSellQuantity	This field contains the total quantity of sell orders in a security.	
MBP_INFO	This field contains quantity, price and number of orders for a maximum of five best prices.	

Market Watch Update

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

<b>Structure Name:</b> BROADCAST_INQ_RESPONSE
<b>Transaction Code:</b> BCAST_MW_ROUND_ROBIN (7201)
<b>Packet Length:</b> 562 bytes
<pre> STRUCT BROADCAST_HEADER (Refer to BROADCAST_                         HEADER in Chapter 2)     SHORT   Number of Records     STRUCT  MARKET_WATCH_BCAST [5]     {         SHORT   Token         STRUCT  MARKET_WISE_INFO [3]         {             STRUCT INDICATOR             DOUBLE  BuyVolume             LONG    BuyPrice             DOUBLE  SellVolume             LONG    SellPrice             LONG    Reserved1             LONG    Reserved2         }     }         </pre>

Field Name	Description	Comment
TransactionCode	The transaction code sent is BCAST_MW_ROUND_ROBIN (7201).	
NumberofRecords	This field contains the number of tokens and their attributes.	The structure MARKET WATCH BROADCAST is repeated these many times.
Token	This is a unique number given to a particular symbol-series combination.	

Field Name	Description	Comment
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.	

Security Open Message

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

<b>Structure Name:</b> SECURITY_OPEN_MSGS
<b>Transaction Code:</b> SECURITY_OPEN_PRICE (6013).
<b>Packet Length:</b> 58 bytes
<pre> STRUCT MESSAGE_HEADER (Refer to                         MESSAGE_HEADER in Chapter 2)   STRUCT SECURITY_INFORMATION     {       CHAR Symbol [10]       CHAR Series [2]     }   SHORT Token   LONG Reserved1                     </pre>

Field Name	Description	Comment
TransactionCode	The transaction code is SECURITY_OPEN_PRICE (6013).	
Security Information	This contains the symbol and series for a particular security.	
Token	This is a unique number given to a particular symbol-series combination.	

## Appendix

### List of Error Codes

Error Code ID	Error Code Value	Description of Error Numbers
ERR_MARKET_NOT_OPEN	16000	The trading system is not available for trading.
ERR_INVALID_USER	16001	Erroneous transaction code received.
ERR_BAD_TRANSACTION_CODE	16003	
ERR_USER_ALREADY_SIGNED_ON	16004	User already signed on.
ERR_INVALID_SIGNON	16006	Invalid signon, Please try again.
ERR_SIGNON_NOT_POSSIBLE	16007	Signing on to the trading system is restricted. Please try later on.
ERR_INVALID_SYMBOL	16012	Invalid symbol/series.
ERR_SECURITY_NOT_AVAILABLE	16035	Security is unavailable for trading at this time. Please try later.
ERR_INVALID_BROKER_OR_BRANCH	16041	Trading Member does not exist in the system.
ERR_USER_NOT_FOUND	16042	User does not exist in the system.
ERR_PASSWORD_HAS_EXPIRED	16053	Your password has expired, must be changed.
ERR_INVALID_BRANCH	16054	Branch does not exist in the system.
ERR_PROGRAM_ERROR	16056	Program error.
ERR_SYSTEM_ERROR	16104	System could not complete your transaction - ADMIN notified.
ERR_CANT_COMPLETE_YOUR_REQUEST	16123	System not able to complete your request. Please try again.
ERR_USER_IS_DISABLED	16134	This user is disabled. Please call the Exchange.
ERR_INVALID_USER_ID	16148	Invalid User Id entered.
ERR_INVALID_TRADER_ID	16154	Invalid trading member id entered.
OE_DELETED_BUT_EXISTS	16260	Has been deleted but exists in the file
ERR_NOT_FOUND	16273	Does not exist.
ERR_MARKETS_CLOSED	16278	The markets have not been opened for trading.
OE_SECURITY_NOT_ADMITTED	16279	The security has not yet been admitted in the system.
ERR_QUANTITY_EXCEEDS_ISSUED_CAPI	16282	The order quantity is greater than the

Error Code ID	Error Code Value	Description of Error Numbers
TAL		issued capital.
ERR_PRICE_NOT_MULT_TICK_SIZE	16283	The order price is not multiple of the tick size.
ERR_PRICE_EXCEEDS_DAY_MIN_MAX	16284	The order price is out of the day's price range.
ERR_BROKER_NOT_ACTIVE	16285	The broker is not active.
ERR_QUANTITY_FREEZE_CANCELLED	16307	The order is canceled due to quantity freeze.
OE_QUANTITY_GREATER_RL	16327	Quantity is greater than Regular lot size.
ERR_QUANTITY_NOT_MULT_BOARD_LOT	16328	Quantity is not a multiple of regular lot.
ERR_BROKER_NOT_PERMITTED_IN_MKT	16329	Trading Member not permitted in the market.
ERR_SECURITY_IS_SUSPENDED	16330	Security is suspended.
ERR_BRANCH_LIMIT_EXCEEDED	16333	Branch Order Value Limit is exceeded.
OE_ORD_CAN_CHANGED	16343	The order to be cancelled has changed.
OE_ORD_CANNOT_CANCEL	16344	The order cannot be cancelled.
OE_ORD_CANNOT_MODIFY	16346	The order cannot be modified.
CHG_ST_EXISTS	16363	New status requested should not be same as existing one.
OE_SECURITY_IN_PREOPEN	16369	The security status is preopen.
ERR_WRONG_LOGIN_ADDRESS	16403	You are trying to sign on from a different location. Sign on is not allowed.
OE_ADMIN_SUSP_CAN	16404	The Order has been cancelled due to admin suspension of security.
ERR_INVALID_BUY_SELL	16413	Requested by can be either BUY or SELL & not both.
ERR_NNF_REQ_EXCEEDED	16417	No. of NNF requests exceeded.
ERR_INVALID_ORDER	16418	Order entered has invalid data.
ERR_INVALID_ALPHA_CHAR	16420	Alpha char must be the same as first two chars of symbol.
ERR_INVALID_BOOK_TYPE	16422	Book type should be 1(RL).
ERR_INVALID_MSG_LENGTH	16424	Message length is invalid.
OE_USER_LIMIT_EXCEEDED	16436	Used limit cannot exceed the user order value limit.
ORD_VAL_EXCEEDS_ORD_LIM_VAL	16442	Order value exceeds the order limit value

Error Code ID	Error Code Value	Description of Error Numbers
ORD_NOT_ALLOWED_IN_PREOPEN	16197	Order Entry or Modification not allowed0020in preopen.
SERIES_NOT_ALLOWED_IN_PREOPEN	16440	Order Entry is not allowed in preopen for the series - xx.” (where xx stands for the series)
ERR_INVALID_ORDER_NUMBER	16103	Invalid Order Number
ERR_INVALID_PRICE	16247	Invalid Price
ERR_BRANCH_NOT_FOUND	16262	Branch not found.
ERR_ORDER_NOT_FOUND	16060	Order not found.
ERR_ORDER_NOT_ON_BOOK	16277	Order not on Book.
ERR_ORDER_PRICE_OUT_OF_RANGE	16179	Order price for the security out of the specified range.
OE_ATO_IN_OPEN	16169	Order Priced ATO cannot be entered when stock status is open.
ERR_NO_OF_BIDS_PER_APPLN_EXCEEDED	16501	No. of bids per application number has exceeded. A Maximum of 3 bids per application is allowed.
ERR_INVALID_CATEGORY_FOR_APPLN	16502	Invalid category for the application number.
ERR_INVALID_APPLN_NUM_FOR_TRADER	16503	Invalid application number for the trader.
OE_DISALLOW_MARKET_ORDERS	16504	ATO orders on this security are disallowed.
ERR_INVALID_QTY	16506	Invalid Quantity or exceeds maximum quantity allowed.
ERR_INVALID_FLAG	16507	One of the parameters in the <i>order terms</i> is incorrect.
ERR_INVALID_APPLN_NUMBER	16508	Invalid application number or appln 1 and appln 2 does not match.
ERR_INVALID_RTGS_CODE	16509	For EQ the error code indicates RTGS Code or Cheque Number has invalid data. For RV the error code indicates Client Name has invalid data.
ERR_INVALID_MARGIN	16510	Invalid Margin amount. Margin amount exceeds 89,99,99,99,99,99,999.
ERR_INVALID_CATEGORY	16511	Category not entered.
ERR_INVALID_NNFFIELD	16512	NnfField is invalid.



Error Code ID	Error Code Value	Description of Error Numbers
ERR_INVALID_PAN	16519	PAN has invalid data.
ERR_DEP_NSDDL_CDSL	16520	Depository name should be either CDSL or NSDL.
ERR_INVALID_DPID	16521	Depository Participant ID has invalid data.
ERR_INVALID_BENFID	16522	Beneficiary ID has invalid data.
ERR_INVALID_DEPOSITORY	16523	Depository Participant ID for CDSL should be blank.
ERR_STOCK_ALREADY_SUSPENDED	16171	Stock already suspended.
ERR_STOCK_DELETED	16256	Stock deleted.
ERR_STOCK_SUSPENDED	16049	Stock Suspended.
ERR_INTERN_FILE_CHANGE	16017	Intermediate file change. Check values and resubmit.
ERR_STOCK_NOT_FOUND	16019	Stock not found.
ERR_SUSPENDED	16020	Suspended.
ERR_INVALID_ORDER_PARAMS	16415	Invalid Order Parameters.
ERR_MODCXL	16444	Order entry not allowed in this session
ERR_ALLOW_SELL	16445	Sell orders not allowed in this session
ERR_CATEGORY_BLANK	16446	Application Number Must Be Entered.
ERR_INVALID_ISSUE	16447	Issue is Invalid.
ERR_QTY_LT_MIN_QTY	16448	Quantity is less than minimum quantity.
OE_ORD_VAL_OUT_OF_BOUNDS	16564	Order Value is out of days permissible value
STR_VALUE_OUT_OF_RANGE	1667	Order Value is not within permissible limits for the Category
OE_DISALLOW_ATO_MARKET_ORDERS	16505	ATO orders are not allowed for this category under this security
OE_LMT_ORD_NOT_ALLOW_CATG	16513	Limit Orders Not Allowed for this Category.
OE_QTY_MOD_NOT_ALLOWED_UP	16515	Quantity cannot be Modified Upwards.
OE_QTY_MOD_NOT_ALLOWED_DOWN	16516	Quantity cannot be Modified Downwards.
OE_PRICE_MOD_NOT_ALLOWED_UP	16517	Price cannot be Modified Upwards.
OE_PRICE_MOD_NOT_ALLOWED_DOWN	16518	Price cannot be Modified Downwards.
OE_MARGIN_MOD_NOT_ALLOWED_UP	16524	Modification of Margin Amt Upwards not allowed.
OE_MARGIN_MOD_NOT_ALLOWED_DOW	16525	Modification of Margin Amt Downwards

Error Code ID	Error Code Value	Description of Error Numbers
N		not allowed.
MOD_APPL_NUM_NOT_ALLOWED	16526	Application Number Modification Not Allowed.
MOD_RTGS_NOT_ALLOWED	16527	For EQ the error code indicates RTGS No. Modification Not Allowed. For RV the error code indicates CLIENT NAME Modification Not Allowed.
MOD_DEPOSITORY_NOT_ALLOWED	16528	Depository Modification Not Allowed.
MOD_DEP_ID_NOT_ALLOWED	16529	Depository Participant Modification Not Allowed.
MOD_BENF_ID_NOT_ALLOWED	16530	Beneficiary ID Modification Not Allowed.
MOD_CATEGORY_NOT_ALLOWED	16531	Category Modification Not Allowed.
MOD_PAN_NOT_ALLOWED	16532	PAN Modification Not Allowed.
ORDER_NO_IN_STOCK_TIME	16567	The order time does not falls in open and close time of the stock.
MOD_CLOSED_FOR_MODIFICATION	16568	Stock closed for Modification
RMS_LIMIT_EXPIRED	16569	The collateral limits expired on RMS.
RMS_MASTER_VALIDATION_FAILURE	16570	Master validation failure at RMS end.
RMS_TIMEOUT	16571	Order timed out on RMS
INVALID_CP_CODE	16572	Invalid CP Code
INVALID_ACC_NO	16573	Invalid Account No
PRO_CLI_MOD_NOT_ALLOWED	16574	Pro to Cli and vice versa modification not allowed
ORDER_MOD_ALRDY_CONF_RMS	16575	Modification after confirmation by Custodian.
ORDER_CXL_ALRDY_CONF_RMS	16576	Cancellation after confirmation by Custodian.
INVALID_PRO_CLI	16577	Value entered is Invalid
ACC_NO_MOD_NOT_ALLOWED	16578	Account No Modification Not Allowed
CP_CODE_MOD_NOT_ALLOWED	16579	CP Code Modification Not Allowed
CONF_REJ_COLLAT_FUND	16580	Custodian confirmation failed due to non-availability of funds. Order retained with trading member
ORDER_CAN_CUST_UNDO	16581	Custodian has unconfirmed previously confirmed order.
ORD_CAN_COLLAT_FUND	16582	Custodian confirmation failed due to non-availability of funds.
ORD_CAN_CUST_REJ	16583	Order rejection by custodian

Error Code ID	Error Code Value	Description of Error Numbers
ORD_BATCH_CAN	16584	Order cancelled, too late to confirm
ORD_CUST_REJ	16585	Custodian rejection, order remains with Trading Member
INVALID_TYPE_TMCP	16586	Order cancelled due to TM/CP not set
ORDER_CAN_CUST	16635	Order cancellation not allowed (in case Institutional orders with 0% margin.)

Reason Codes

The reason codes for rejecting the order and the corresponding values are given below.

Reason Code	Value
Security	5
Broker	6
Branch	7
User	8
Participant	9
Order Number	11
Normal Market	12
Order	16
Quantity Freeze	18
Category	29

List of Transaction Codes

Transaction Code	Code	Structure	Size	I/B*
SYSTEM_INFORMATION_IN	1600	MESSAGE_HEADER	40	I
SYSTEM_INFORMATION_OUT	1601	SYSTEM_INFO_DATA	86	I
BOARD_LOT_IN	2000	ORDER_ENTRY_REQUEST	224	I
BOARD_LOT_OUT	2001	ORDER_ENTRY_REQUEST	224	I
ORDER_MOD_IN	2040	ORDER_ENTRY_REQUEST	224	I
ORDER_MOD_OUT	2041	ORDER_ENTRY_REQUEST	224	I
ORDER_MOD_REJECT	2042	ORDER_ENTRY_REQUEST	224	I
ORDER_CANCEL_IN	2070	ORDER_ENTRY_REQUEST	224	I
ORDER_CANCEL_OUT	2071	ORDER_ENTRY_REQUEST	224	I

Transaction Code	Code	Structure	Size	I/B*
ORDER_CANCEL_REJECT_OUT	2072	ORDER_ENTRY_REQUEST	224	I
ORDER_CONFIRMATION_OUT	2073	ORDER_ENTRY_REQUEST	224	I
ORDER_MOD_CONFIRMATION_OUT	2074	ORDER_ENTRY_REQUEST	224	I
ORDER_CANCEL_CONFIRMATION_OUT	2075	ORDER_ENTRY_REQUEST	224	I
RMS_APPROVE_REJECT	2079	TRADER_INTERACTIVE_MESSAGE_CP	299	I
FREEZE_TO_CONTROL	2170	ORDER_ENTRY_REQUEST	224	I
ORDER_ERROR_OUT	2231	ORDER_ENTRY_REQUEST	224	I
SIGN_ON_REQUEST_IN	2300	SIGNON_REQUEST	186	I
SIGN_ON_REQUEST_OUT	2301	SIGNON_REQUEST ERROR_RESPONSE	186 180	I
SIGN_OFF_REQUEST_IN	2320	MESSAGE_HEADER	40	I
SIGN_OFF_REQUEST_OUT	2321	MESSAGE_HEADER	40	I
SECURITY_OPEN_PRICE	6013	SECURITY_OPEN_MSGS	58	B
BCAST_JRNL_VCT_MSG	6501	TRADER_INTERACTIVE_MESSAGE BROADCAST_MESSAGE	289	B
BC_CLOSE_MESSAGE	6521	BROADCAST_VCT_MSGS	297	B
BC_NORMAL_MKT_PREOPEN_ENDED	6571	BROADCAST_VCT_MSGS	297	B
DOWNLOAD_REQUEST	7000	MESSAGE_DOWNLOAD	48	I
HEADER_RECORD	7011	MESSAGE_HEADER	40	I
MESSAGE_RECORD	7021	Packet of size >80 and <=512	512	I
TRAILER_RECORD	7031	MESSAGE_HEADER	40	I
BROADCAST_MBO_MBP_UPDATE	7200	BROADCAST_MBO_MBP	491	B
BROADCAST_ONLY_MBP	7208	BROADCAST_MBO_MBP	333	B
BROADCAST_MW_ROUND_ROBIN	7201	BROADCAST_INQ_RESPONSE	562	B
BROADCAST_SYSTEM_INFORMATION_OUT	7206	SYSTEM_INFO_DATA	86	B
BCAST_STOCK_STATUS_CHG_PREOPEN	7210	SECURITY_STATUS_UPDATE_INFO	472	B
UPDATE_LOCALDB_IN	7300	UPDATE_LOCAL_DATABASE	62	I
UPDATE_LOCALDB_DATA	7304	Packet of size >80 and <=512	512	I
BCAST_STOCK_MSTR_CHG	7305	SECURITY_UPDATE_INFO	212	B
UPDATE_LOCALDB_HEADER	7307	UPDATE_LDB_HEADER	42	I
UPDATE_LOCALDB_TRAILER	7308	UPDATE_LDB_HEADER	42	I

Transaction Code	Code	Structure	Size	I/B*
BCAST_CATEG_MSTR_CHG	7309	CATEGORY_UPDATE_INFO	81	B
BCAST_STOCK_STATUS_CHG	7320	SECURITY_STATUS_UPDAT E_INFO	472	B
PARTIAL_SYSTEM_INFORMATION	7321	SYSTEM INFORMATION DATA	86	I
IPO_CAT_HRCHY_UPDT_REQUEST	5727	MESSAGE_HEADER	40	I
IPO_CAT_HRCHY_UPDT_RESPONSE	5728	SYSTEM_INFO_DATA	98	B

\* Interactive/Broadcast

Quick Reference for Order Entry Parameters

The order flags are given below.

**Order Terms:**

- GTC : input (for IPO)
- Market : output
- ATO : output
- Frozen : output
- Modified : input
- MatchedInd : output
- Reserved1 : input (for OFS only)

Status	Market	Book Type	Order Terms and Other Characteristic Fields
Preopen	Normal Market	RL	Only GTC bit should be set. (For IPO) Only Reserved1 bit should be set (FOR OFS Only)
Open	Normal Market	RL	Order entry is not allowed.
Close	Normal Market	RL	Order entry is not allowed.

**Market Type**

The market type in case of IPO is:

Status	Market Status Id
Normal Market	1

**Market Status**

The market can be in one of the following statuses:

Status	Market Status Id
Preopen (only for Normal Market)	0
Open	1
Closed	2

**Book Types**

There is only one book & the book is for Normal Market.

Book Type	Book ID	Market Type
Regular Lot Order	1	Normal Market

**Security Status**

Status	Status ID
Pre open	1
Open	2
Suspended	3

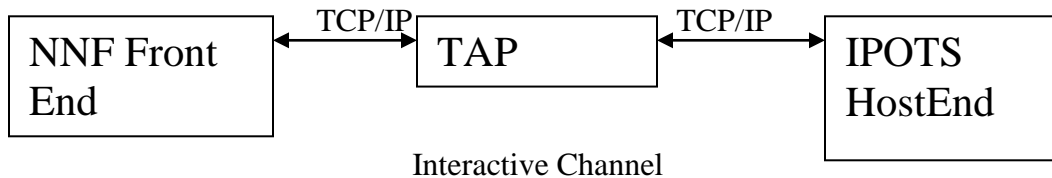
## TAP

The National Stock Exchange of India has enhanced IPO Market Trading System by changing existing 2-tier system to 3-tier system with introduction of a middle-ware between Front-End and Host End.

The introduced middle-ware is named as **TAP - Trading Access Point** and all the communication between front-end and host-end will be through this **TAP** box.

For **TAP** deployment, certain changes have been introduced in IPO trading system such as Network protocol changes, Message flow changes which will be explained in details in following sections.

This document will explain in detail the changes that have to be made to Front-end so that it will be compliant with the changed IPO trading system.



With the changes, TCP/IP communication protocol will be used between the CTCL (NNF) Front-End and TAP box also TCP-IP communication protocol will be used between TAP box and IPO Trading System host end.

Please note that there are no changes for Broadcast.

### Changes required in CTCL

#### TCP/IP Connectivity between CTCL system and TAP Box

Steps to follow for TCP/IP connectivity (Windows based) between Front-End and TAP Box are as follows

- Type of the socket to be used is SOCK\_STREAM
- Address family for the socket is PF\_INET
- Front-End establishes socket connection to TAP Box.
- TAP Box accepts front-end socket connection
- Recv () function will be called to receive data packets from TAP Box.
- Send () function will be called to send data packets to TAP Box.
- Pragma pack (2) to be used for all structures flowing between Front-End and TAP Box.
- It is recommended to disable Nagle's algorithm.

#### Program Flow for TCP/IP Connectivity

##### START

```
SET socket type = SOCK_STREAM
```

```

SET socket address family = PF_INET
CREATE socket using above options
SET no. of bytes to be received and no. of bytes to be sent at SOL_SOCKET    level
SET socket option to TCP_NODELAY at IPPROTO_TCP
SET remote address of host in AF_INET address family
SET remote port id for connection with TAP Box
WAIT for events of the type FD_READ or FD_CLOSE
IF the event is FD_CLOSE then
    CONNECTION is lost
ELSEIF the event is FD_READ then
    RECEIVE the packet and process it
    WAIT for invitation packet
    
```

For non Windows based systems, use equivalent parameters in the above steps.

Setting TCP Keep-alive parameters

In order to detect abrupt network disconnection between TAP and NNF, it is required to enable TCP Keep-alive option in NNF application. This can be done using following steps:

1. After TCP socket is created, NNF application has to set SO\_KEEPAKIVE option of the socket at SOL\_SOCKET level.
2. Also, following Keep-alive parameters should be set in the Operating System on which NNF is installed
  - a. Keep Alive Time : 20 seconds
  - b. Number of Keep Alive Retries: 5
  - c. Keep Alive Interval : 2 seconds

Packet structure communicating between CTCL and TAP Box

**Packet Format**

Length (2 bytes)	Sequence number (4 bytes)	Checksum(MD5) for business data (16 bytes)	Business 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 business data (variable number of bytes).
- Sequence number will start from 1 and will be incremented for every packet.
- Business data will be of variable length and comprises of 40 bytes (increased by 2 bytes) of header + variable sized data buffer.
- The checksum algorithm used will be MD5. Checksum is applied only on the Business data field and not on the entire packet.
- For more details on MD5 refer: **RFC 1321 (rfc1321) - The MD5 Message-Digest Algorithm** (<http://www.faqs.org/rfcs/rfc1321.html>)

**Packet Validation**



Validation will be done for all requests flowing between TAP Box and Front-End. Validation will be done through the combination of Checksum, Sequence Number and length field.

**Request Processing by TAP Box:**

Before sending the request to TAP box, Front-End will have to generate a sequence number and checksum value. All the requests being sent from Front-End will be sent in the format described above.

If validation of sequence number, checksum value & length will fail at TAP Box end, TAP Box will disconnect the socket connection.

**Response Processing by CTCL software:**

On receiving the response from TAP Box, CTCL software is expected to validate sequence number, checksum value & length field.

Sequence number must be in sequential order. For any fresh connection the number should start from 1. On reaching the maximum limit of the data type the numbers should be reset to 1. Checksum field and the checksum recalculated on the data field must match. Length field must be less than or equal to 1024.

If any one of these validations fails, the Front-End needs to drop the connection and reestablish a fresh connection.

Changes in Message Flow

**Flow Control Mechanism**

Packet flow between TAP Box and Front-End will be controlled using Flow Control Mechanism.

TAP Box will send an '**Invitation Packet**' to Front-End on establishing a fresh connection.

Number of requests specified in Invitation Packet will be the maximum limit for requests which will be allowed to be sent to TAP Box from Front-End.

After the front-end has utilized the invitation count fully, the TAP box will send the next invitation packet. Only on receiving the next invitation, the front-end is expected to send subsequent requests.

**Invitation Packet Structure**

Invitation Packet

The format of the invitation packet is as follows:

<b>Structure Name:</b>	INVITATION_PACKET
<b>Transaction Code:</b>	INVITATION_TRANSCODE(15000)
<b>Packet Length:</b>	42 bytes

```

    { MESSAGE_HEADER (Refer to MESSAGE_HEADER in
      SHORT Invitation_count
    }

```

Field Name	Description	Comment
TransactionCode	The transaction code should be INVITATION_TRANSCODE(15000)	
Invitation_count	Maximum number of requests which will be allowed to be sent to TAP Box from Front-End. This is an output field from the TAP box.	

**Note:**

In the invitation packet, the following fields are to be used

1. Transaction Code field
2. Message length field
3. Invitation Count Field

Rests of the fields of this packet are reserved and are to be ignored while processing.

**Exception Handling:**

In case of any exception while processing the 'invitation packet' the connection should be dropped. A new connection needs to be established by the front end with the TAP box.