Protocol for Drop Copy Service

Future & Options Segment

Version 3.0

August 2026

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

Notice

Drop Copy Service – Future & Options Segment Revision History		
Version	Page No	Description
3.0	All Pages	Initial Draft

CONTENTS

Chapter 1 Introduction

The NSE Drop Copy (DC) service provides dedicated Trade Only and Order & Trade data. It disseminates information about members / users order and trades on a real time basis. The data is sent to users on a TCP/IP communication protocol connection. At the time of initial login, all members need to connect to the Drop Copy Gateway Router, which will assign a Drop Copy Gateway. Order Trade Drop Copy & Trade Drop Copy are two different services and can be accessed through two different gateway router connections. Members will initiate the connection to the assigned DC Gateway using the existing login credentials being used for login to the Trading System. A Corporate Manager user will get member firm level Order and trade data. However, branch manager and dealer users will get respective user order and trade data. Any changes to the login credentials during the day on the Trading System will be effective on drop copy gateway on the same day.

The order and trades data structures are now revised for the Drop Copy service.

Chapter 2 General Guidelines

Introduction

This chapter provides general guidelines for the designers and programmers who develop Drop Copy consumers. 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 drop copy system (i.e., host) or received from the drop copy system (i.e., host).

Transaction code, an important field of the message header, is a unique numeric identifier which is sent to or received from the system. This is used to identify the transaction or activity type.

Guidelines for Programmers

1. All time fields are time values with base as midnight January 1, 1980.
2. If your system uses a little-endian order, the data types such as INT, 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.

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

3. All alphabetical data must be converted to the upper case except password before sending to the host. A combination of the alphabet, numbers and special characters are allowed in the password. More details on password are explained in later chapters in this document. No NULL terminated strings should be sent to the host end. Instead, fill it with blanks before sending. The strings received from the host end are padded with blanks and are not NULL terminated.
4. 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.
   - All structures are pragma pack 2. Structures of odd size should be padded to an even number of bytes.
   - All other types of structure members are word aligned.
5. All numeric data must be set to zero (0) before sending to the host unless a value is assigned to it.
6. All reserved fields mentioned should be mapped to CHAR buffer and initialized to blank.

   Note:

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

Data Types Used

Table 2.1 DATA TYPES

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

Note: 64 Bit data type format is considered in the above table

Request Message Header

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

Table 2.2 REQUEST MESSAGE HEADER

Structure Name	REQUEST_MESSAGE_HEADER		
Packet Length	24 bytes		
Field Name	**Data Type**	**Size in Byte**	**Offset**
TransactionCode	SHORT	2	0
TraderId	INT	4	2
SequenceNumber	LONG	8	6
PartitionID	CHAR	6	14
ConcurrentLoginID	SHORT	2	20
FuncIndc	CHAR	1	22
Reserve	CHAR	1	23

The fields of Message Header are described below.

Field Name	Brief Description
TransactionCode	Transaction message number. This describes the type of message received or sent.
TraderId	This field contains the user ID.
SequenceNumber	Sequence number for the messages being sent by the User.
Partition ID	This field contains the partition ID (Stream) for which the packet is being sent by the User. Partition ID will be identified as MXXPXX; where M is market indicator followed by 2-digit market number and P is the partition indicator followed by 2-digit partition number. Eg – For partition 1 of mkt stream 2; partition ID will be M02P01
ConcurrentLoginID	This field contains Concurrent login number of the User for which User has logged in. Expected values shall be numeric value between 1 to the max user connection allowed by exchange.
FuncIndc	This field will indicate functionality. Expected value: • 'D' (Drop copy)

Response Message Header

Overall Response structure is prefaced with a Response_Message_Header (In case of packing multiple messages in one packet it will appear one time). The structure of the Response Message Header is as follows:

Table 2.3 RESPONSE MESSAGE HEADER

Structure Name	RESPONSE_MESSAGE_HEADER		
Packet Length	2 bytes		
Field Name	**Data Type**	**Size in Byte**	**Offset**
Environment Type	CHAR	1	0
Compression Indicator	CHAR	1	1

Field Name	Brief Description
EnvironmentType	Following are the possible values for Environment Type • '1' for Production Environment • '2' for Prod Parallel (Mock) Environment

Field Name	Brief Description
	• '3' for Testing Environment
Compression Indicator	Following are the possible values for Compression Indicator • '0' for No compression done • '1' for Data is compressed.

Message Header

Each response message is prefaced with a MESSAGE_HEADER. The structure of the Message Header is as follows:

Table 2.4 MESSAGE HEADER

Structure Name	MESSAGE_HEADER		
Packet Length	**14 bytes**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
TransactionCode	SHORT	2	0
ErrorCode	SHORT	2	2
SequenceNumber	LONG	8	4
Length	SHORT	2	12

Field Name	Brief Description
TransactionCode	Transaction message number. This describes the type of message received or sent.
ErrorCode	This field describes the type of error. Refer to List of Error Codes in Appendix.
SequenceNumber	Sequence number for the messages being sent by the Drop Copy Service system.
Length	Length of the data.

Order Flags

Table 2.5 ST_ORDER_FLAGS

For Big-Endian Machines:

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

For Little-Endian Machines:

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

Error Response

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

Table 2.7 ERROR_RESPONSE

Structure Name	ERROR RESPONSE		
Packet Length	142 bytes		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table 2.4)	STRUCT	14	0
Error Message [128]	CHAR	128	14

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

Book Types

Table 2.8 BOOK_TYPES

Book Type	Book ID	Market Type
Regular Lot Order	1	Normal Market
Special Terms Order	2	Normal Market
Stop Loss Order	3	Normal Market
Negotiated Order (Not used)	4	Normal Market
Odd Lot Order (Not used)	5	Odd Lot Market
Spot Order (Not used)	6	Spot Market
Auction Order (Not used)	7	Auction Market
Call Auction1 (Not used)	11	Call auction1 market
Call Auction2 (Not used)	12	Call auction2 market

Heartbeat Exchange

Member systems must exchange heartbeat signals with drop copy service system during periods of inactivity. Service Host will consider the member system as inactive after missing two heartbeats in succession and disconnect the socket connection. Heartbeats will carry request message header only. Heartbeat is to be sent only if there is inactivity for 30 seconds. The format is MESSAGE_HEADER with following detail.

Table 2.9 HEARTBEAT

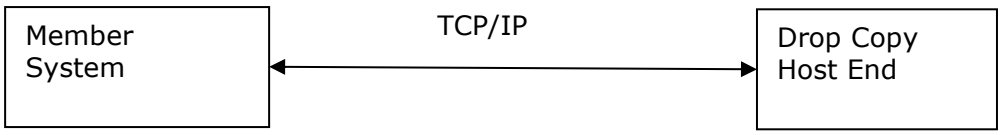
Structure Name	HEARTBEAT		
Packet Length	24 bytes		
Field Name	Data Type	Size in Byte	Offset
REQUEST_MESSAGE_HEADER (Refer Table No 2.2)	STRUCT	24	0

Field Name	Brief Description
TransactionCode	The transaction code is HEARTBEAT (23506)

Chapter 3 Drop Copy Communication

Introduction

TCP/IP communication protocol shall be used between Member System and Drop Copy Host end as per the Network setup.



Packet Format

Packet structure for communication between Member System and Host End

This structure is applicable to all messages that flow between Client and Drop Copy Host

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

- Max length will be the predefined value of 1400 bytes.
  Length = size of length field (2 bytes) +
  size of sequence number field (4 bytes) +

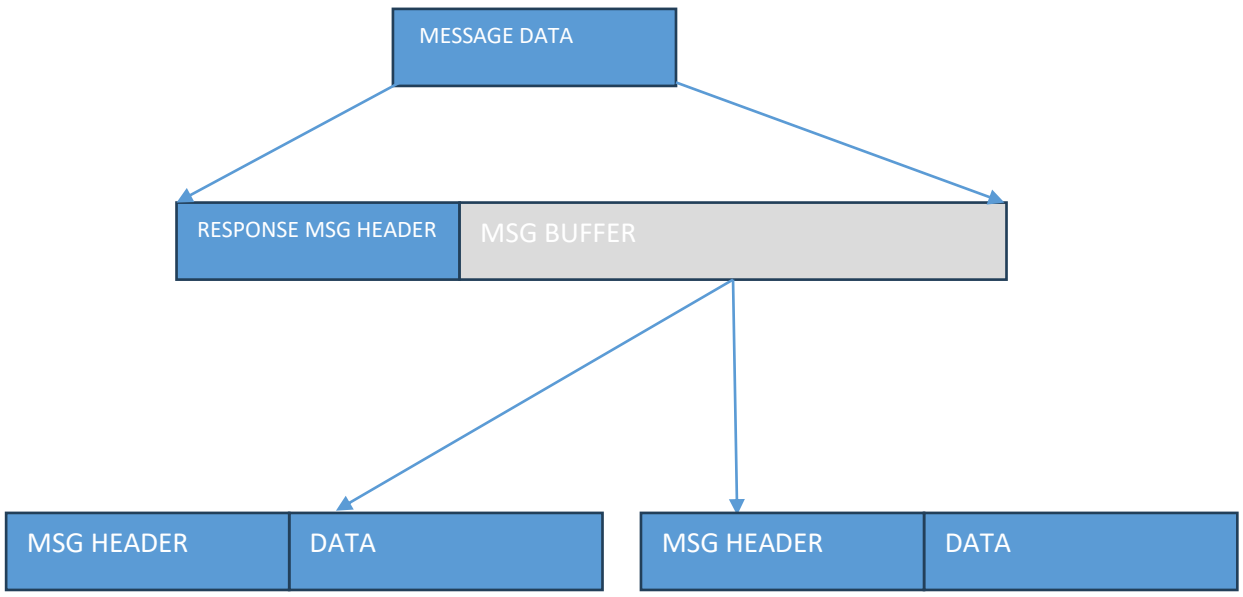
size of the checksum field (16 bytes) +

size of Message data (variable number of bytes as per the transcode).

- Maximum length of message data can be 1378
- Sequence number will start from 1 and will be incremented for every packet.
- The checksum algorithm used will be MD5. Checksum is applied only on the Message data field and not on the entire packet.
- The MD5 checksum will be used only for the initial message, " GR_SECURE_USER_REGISTRATION_REQUEST ". For subsequent communications, this field will contain the authentication tag.
- The authentication tag received as part of the message header will be verified against the tag obtained after decrypting the Message Data using the new encryption method.
- If the checksum (MD5 / authentication tag) does not match, a user sign-off message with error code (19031) will be sent to the member before disconnection.

- For more details on MD5 refer: **RFC** 1321 (rfc1321) - The **MD5** Message-Digest Algorithm (http://www.faqs.org/rfcs/rfc1321.html)

- In case checksum is not matched, packet will be dropped at Exchange end.
- Message data will be of variable length and comprises of 24 bytes of request message header + variable sized data buffer as per transcode being sent in case of requests sent by Member system.
- Structure of Message Data sent by Host End is as follows



- First two bytes of message data will be response message header. Second byte of Response message header will indicate the compression indicator.
- In case message buffer is not compressed, length of message buffer will be equal to message data length – 2 bytes (Response Message header)
- In case message buffer is compressed, actual length of message will be equal to length of message buffer after decompression (Please refer Compression and Decompression section).
- Maximum length of the message buffer after decompression can be 4096 bytes
- Message buffer may have multiple data packets, each data packet will be having message header and data part. Each data packet reads to be read one by one and length of the message buffer should be reduced by the length of data packet.

Packet Validation

Validation will be done for all requests flowing between Member System and Host End. Validation will be done through the combination of Checksum, Sequence Number, and length field.

Processing by Host

Before sending the request to Host, Member System 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 fails at Host End then the disconnection of the socket connection between Member System and Host End will happen.

Processing By Member System

On receiving the response from Host, Member software is expected to validate sequence number, checksum value & length field. Also, it must decompress the data if compression indicator is set.

Sequence number must be in sequential order. For any fresh connection the number should start from 1. Checksum field and the checksum recalculated on the data field must match. Length field must be less than or equal to 1400.
If any one of these validations fails, the Member System needs to drop the connection and reestablish a fresh connection**.**

Encryption/Decryption

Exchange proposes a combination of TLS 1.3 security protocol and AES-256 bits-based symmetric encryption approach. The following is an overview.
1st Step: Member applications will connect initially to Exchange Gateway Router server using TCP with TLS 1.3 security protocol and will receive unique session key from the Exchange through the secured connection.
Gateway router will be different for Order-Trade and Trade only subscription.

2nd Step: Member applications will then connect to allocated Exchange Gateway server through TCP, and each and every message will be encrypted/decrypted using the same session key (symmetric cryptography AES 256 bits GCM encryption with authentication) at both member end and Exchange end.
With the removal of MD5 checksum and the authentication embedded in the new encryption library, improvement in performance as well as security is provided.

Below are the details of the methodology

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

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

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

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

c. A unique 32-byte session key, additional 12-byte encryption-decryption key and 16-byte IV (Initialization Vector) will be provided to member applications as part of GR response message.

(iii) Post successful communication with Gateway router server, member applications will establish a new TCP connection with the allocated gateway server of Exchange. The first message after connecting through TCP will be a non-encrypted special registration message (GR_SECURE_USER_REGISTRATION_REQUEST) to indicate that member application is using encryption. All the messages, after the first message, that are exchanged on this connection from both sides (member applications and Exchange) will be encrypted and decrypted using the unique 32-byte session key, 16-byte IV (Initialization Vector) (8-byte Static and 8-byte Dynamic) and a 12-bytes additional key value that was provided from Exchange at the time of Gateway Router handshake. GCM mode of symmetric cryptography AES 256 bits with authentication will be used by member applications and Exchange. In the member application, encryption and decryption operations are performed using a combination of static and dynamic Initialization Vectors (IVs). The static and dynamic IV is taken from GR response message received from exchange. Static IV remains unchanged, however the dynamic IV is modified for each message. The member must maintain two separate copies of the dynamic IV: ensuring that for every message the dynamic part of the IV is incremented by 1 before encryption and decremented by 1 before decryption. In the event of a box disconnection, the IVs are reset at exchange end, and a new static and dynamic IV is provided in GR response message to a fresh GR query.

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

Sample function calls which could be considered for encryption-decryption for the above proposed approaches are provided in annexure for Encryption/Decryption.

Disconnection on MD5 Checksum failure

If member is connected on encrypted channel and MD5 checksum fails then a user sign off message with error code (19031) will be sent to member before disconnection.

Compression and Decompression

This section describes the Compression and Decompression algorithm to be used for drop copy related messages.

Compression of the Data

Users can connect to Drop Copy system and seek data from the start of the day. This could result in large volume of data being served to the User depending on messages present in the system. To accommodate the increased network traffic in such cases, the exchange has come up with a compression algorithm to compress data being downloaded.

LZ4 compression algorithm is used to compress multiple data records in download response. The library function "LZ4_compress_default(const char* source, char* dest, int inputSize, int maxOutputSize)" is used for compression at host end.

To decompress the received compressed data packet, library function "LZ4_decompress_safe(const char* source, char* dest, int compressedSize, intmaxDecompressedSize)" is to be used.

Sequential Packing

To improve effective data transfer, the idea of sequential packing of messages along with the LZ4 compression algorithm has been incorporated. At the host end, sequential packing algorithm packs the messages which are meant for download. Data packets are packed in FIFO order. Post packing, compression of entire data will be done. Compression indicator field will be set to "1" in response message header to convey that message body (buffer) that follows is compressed.

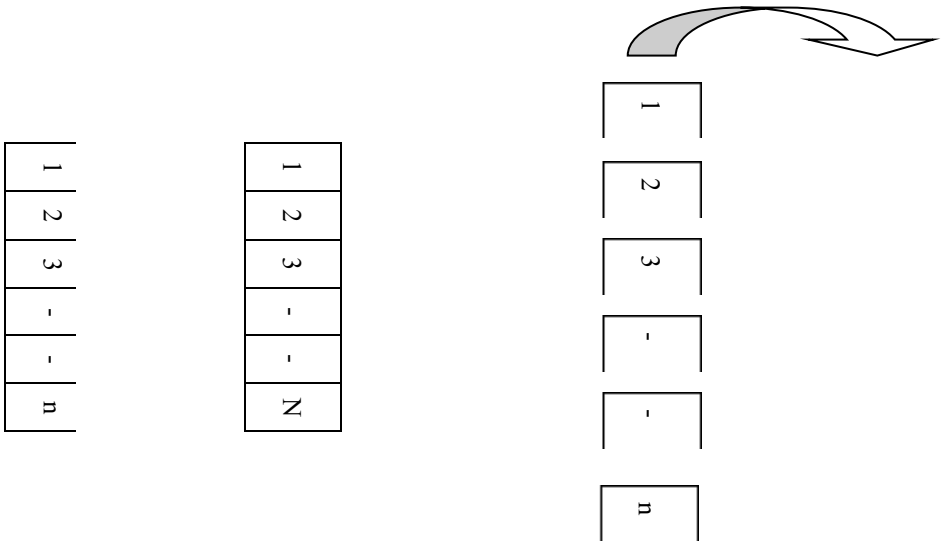
For example,
If 'n' packets are packed in a buffer, they are arranged in the following order:

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

At the front end while unpacking the buffer, the packets are to be segregated in the same order, that is, isolate each packet and process each packet as per the sequence viz- first packet first and last packet at the end.

Host End Sends Front End Receives Front end Unpacking

Process packets



Chapter 4 Logon Process

Introduction

This section describes how a user logs on to the drop copy service system for Order & Trade (O+T) or Trade(T) only. It covers the log-on request and the system responses. As mentioned earlier, Order-Trade and Trade only services will have separate Gateway Routers. For drop copy download user need to connect to respective Gateway Router to get the required data.

The client system, after issuing a sign-on request, waits for the system response. The response could be a successful logon or an error message

Users will need to connect to multiple partitions to receive data across all Market streams. Number of partitions to connect will be dynamically provided as part of Response to Logon request

Users will be allowed to login concurrently multiple times (upto max value defined by Exchange) using the same User ID.

E.g An user connecting to Drop Copy system will provide User ID and ConcurrentLogin ID to login.

Drop Copy system supports multiple logins using the same User ID. To distinguish these logins, user need to specify ConcurrentLoginID along with User ID. ConcurrentLoginID will have values like 1, 2, 3 For first login, user will specify value as 1, for next value as 2 and so on to all these Partitions. Each partition ID will have specific server IP and Port. Say GR Response sent following 5 partition ID details.

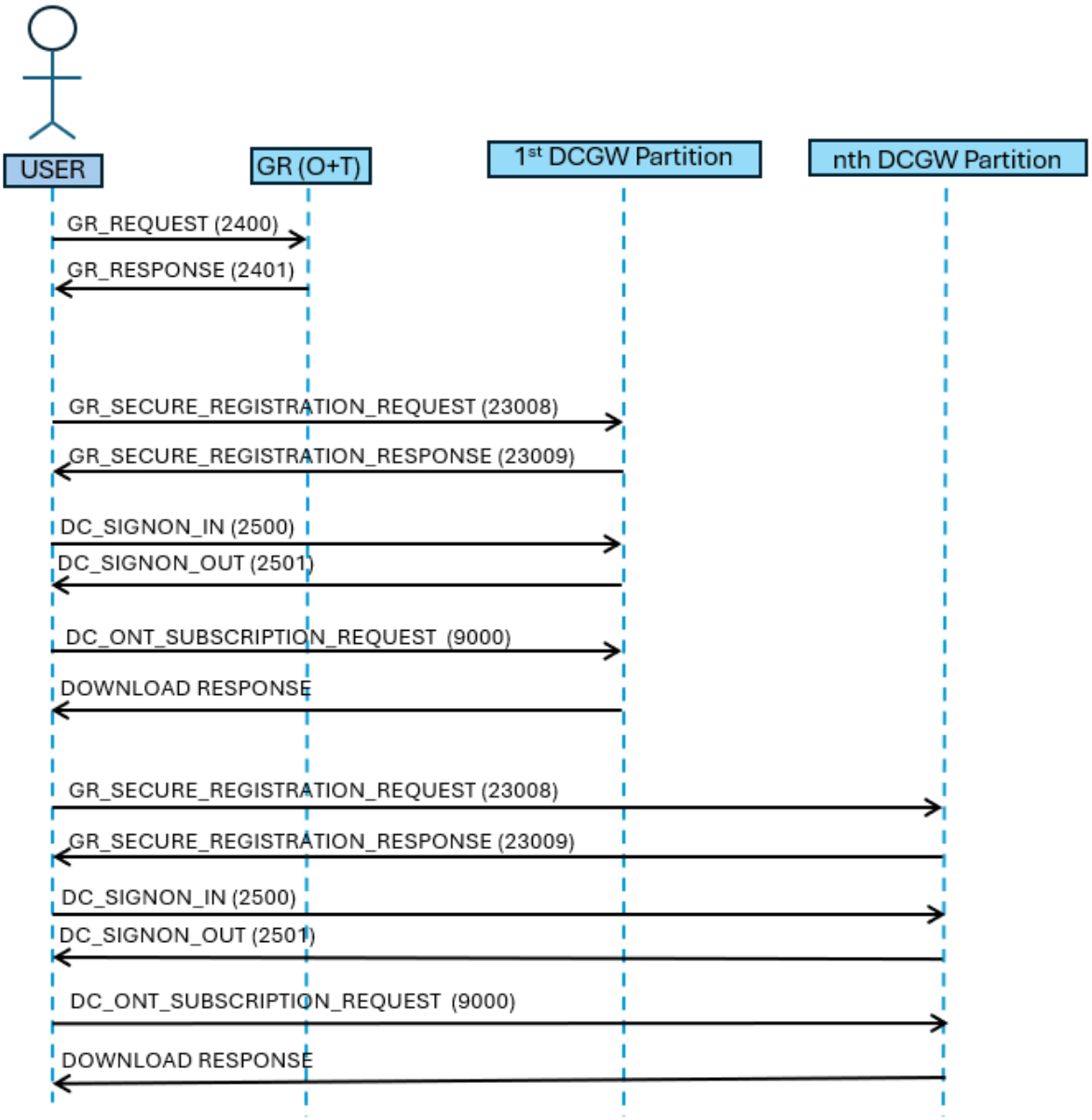
Partition ID	IP	Port
M01P01	IP1	P1
M02P03	IP2	P2
M03P05	IP3	P3
M04P04	IP4	P4
M05P01	IP5	P5

User will need to make 5 connections using IP-Port as IP1-P1, IP2-P2, IP3-P3, IP4-P4, IP5-P5 respectively. And User will need to specify values as M01P01, M02P03, M03P05 M04P04 and M05P01 for PartitionID in request header for the respective connections

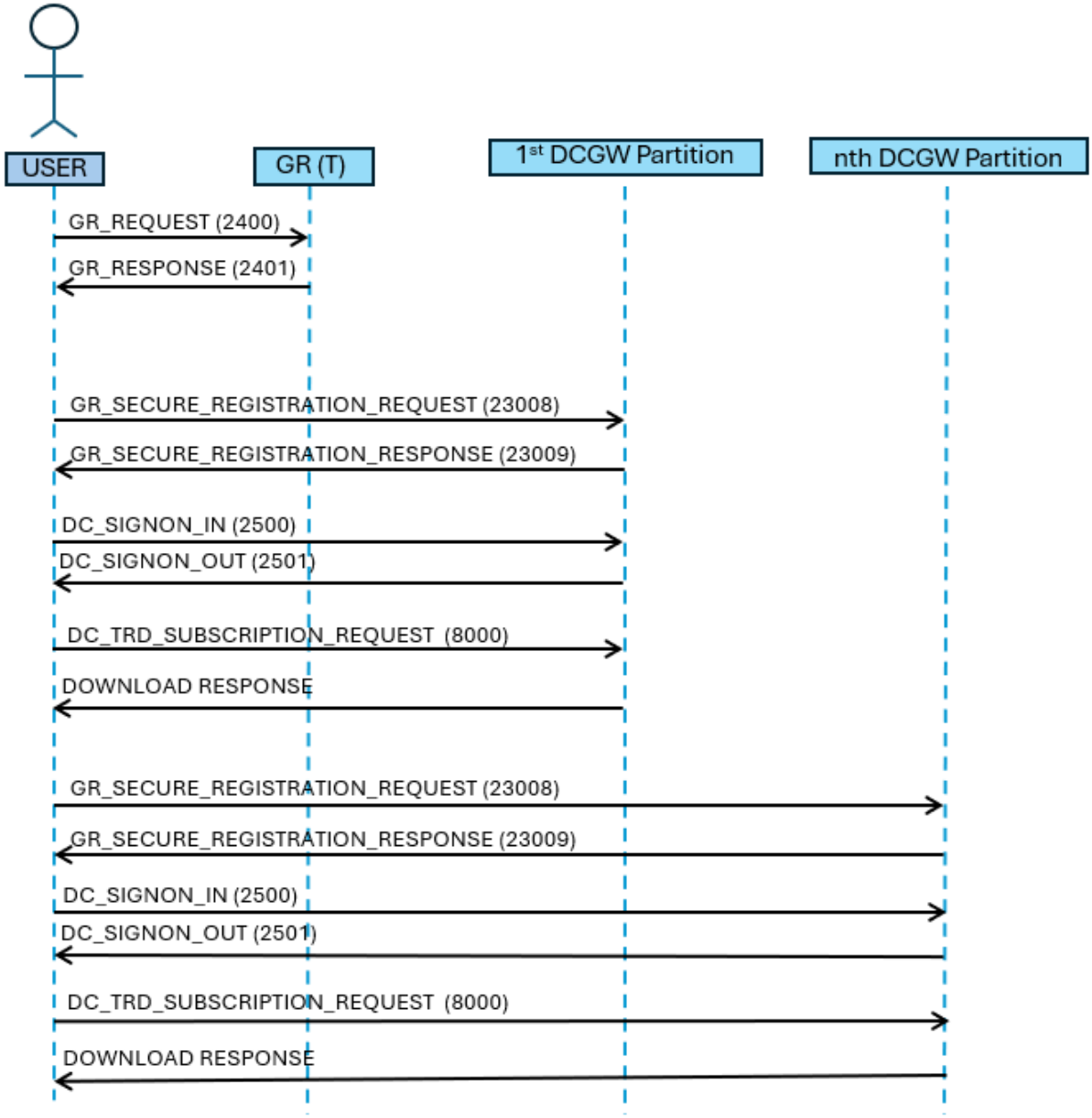
Order of Events to be followed during Logon for Drop Copy Service Service

The following sequence explains the order in which transaction codes are sent and received during the log-on process. The same can be seen in the diagram below.

Order of Events for Order and Trade Drop Copy



Order of Events for Trade Only Drop Copy



Note: Send the subscription request for the Order & Trade Drop Copy to the Gateway Servers specified in the GR Response received for the GR Request sent to the Order & Trade Gateway Router service. Similarly, Send the subscription request for the Trade Drop Copy to the Gateway servers specified in the GR Response received for the GR Request sent to the Trade Gateway Router service.

Order & Trade Drop copy Download sequence:

Seq No	Transaction Code	Sent By	Received By
1	GR_REQUEST (2400)	Client	Gateway Router (GR)
2	GR_RESPONSE (2401)	GR	Client
3	GR_SECURE_USER_REGISTRATION_REQUEST (23008)	Client	HostEnd
4	GR_SECURE_USER_REGISTRATION_RESPONSE (23009)	HostEnd	Client
5	DC_SIGNON_IN (2500)	Client	Host End
6	DC_SIGNON_OUT (2501)	Host End	Client
7	DC_TRD_SUBSCRIPTION_REQUEST (8000) / DC_ONT_SUBSCRIPTION_REQUEST (9000)	Client	Host End
8	DOWNLOAD RESPONSE (TRADE_CONFIRMATION / ORDER_RESPONSE)	Host End	Client

Logon Request & Response

1. Member to connect (TCP/IP, SSL connection) to the IP and port provided by the exchange and send the GR_REQUEST to respective GR service using OpenSSL (Version 3.4.0) library calls with TLS versions 1.3 (TLS1_3_VERSION). Refer annexure for Encryption/Decryption. Partition ID will be sent as "0" in this request
2. Exchange will send the GR_RESPONSE to the member containing array of multiple IP address and Port along with the Session key, Cryptographic key, cryptographic additional key & Cryptographic IV (Initialization Vector) on SSL connection. GR will send list of IP & Ports for total number of partitions to which Member needs to subscribe to get Order / Trade data across all Market Streams. If there is any error, then ErrorCode field in MESSAGE_HEADER will be populated with relevant error code in the GR_RESPONSE.
3. Member applications will then make a new TCP connection with the allocated Dropcopy Gateway server (IP and port provided in the GR_RESPONSE) for each partition and

send GR_SECURE_USER_REGISTRATION_REQUEST for each partition. Partition ID (received in GR_RESPONSE) along with User ID and ConcurrentLoginID (sent in GR_REQUEST) is to be populated in GR_SECURE_USER_REGISTRATION_REQUEST.

4. Exchange will send the GR_SECURE_USER_REGISTRATION_RESPONSE. If there is any error, then ErrorCode field in MESSAGE_HEADER will be populated with relevant error code in the GR_SECURE_USER_REGISTRATION_RESPONSE and the User connection will be terminated.
5. If there is no error in GR_SECURE_USER_REGISTRATION_RESPONSE, member should do encryption and decryption initialization to create encryption and decryption contexts (Please refer annexure). This initialization should be done only once for each TCP channels created for each partition. Once initialized, all further messages between member application and allocated Gateway server will be encrypted and decrypted using same encryption and decryption contexts respectively. MD5 Algorithm to be performed only on GR_SECURE_USER_REGISTRATION_RESPONSE. That means, while sending the messages to Trading system, MD5 is to be performed first and then encryption. Encrypted message length + 22 (sizeof(Header)) will have to be written in first 2 bytes of header, Sequence Number in next 4 bytes and MD5 value (of plain message) will be written in last 16 bytes of Header and the header will have to be prepended to the encrypted message. This message will be sent out to Trading System. While receiving the messages from Trading System, decryption should be done first and then MD5 is to be applied on decrypted buffer. Decryption should be done on message excluding first 22 bytes of header. For the remaining messages, Encrypted message length + 22 (sizeof(Header)) will have to be written in first 2 bytes of header, Sequence Number in next 4 bytes and authentication tag will be written in last 16 bytes of Header and the header will have to be prepended to the encrypted message. This message will be sent out to Trading System. While receiving the messages from Trading System, decryption should be done on message excluding the first 22 bytes of header and then authentication tag received as part for message header should be verified against authentication tag obtained after decryption of the message.
6. Exchange will send the DC_SIGNON_OUT. If there is any error, then ErrorCode field in MESSAGE_HEADER will be populated with relevant error code in the DC_SIGNON_OUT and the User connection will be terminated.

Note :

1. Login of member will be for a given User ID and ConcurrentLoginID
2. List of Partition will be published to user in response (GR Response) to Logon Request (GR_Request).
3. Member can specify upto maximum three concurrent logins for same User ID (max value will be defined by Exchange from time to time).

Table 4.1 GR_REQUEST

Structure Name	**GR_REQUEST**		
Packet Length	**24 bytes**		
Transaction Code	**GR_REQUEST (2400)**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
REQUEST_MESSAGE_HEADER (Refer Table No 2.2)	STRUCT	24	0

Table 4.2 IP_DETAILS_PARTION_WISE

Structure Name	**IP_DETAILS_PARTITION_WISE**		
Packet Length	**26 bytes**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
IPAddress [16]	CHAR	16	0
Port	INT	4	16
PartitionID	CHAR	6	20

Field Name	Brief Description
IPAddress	This field contains IP address of DC Gateway.
Port	This field contains Port of DC Gateway.
PartitionID	This field contains Partition ID for which Data is being received.

Table 4.3 GR_RESPONSE

Structure Name	GR_RESPONSE		
Packet Length	1126 bytes		
Transaction Code	GR_RESPONSE (2401)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0
MessageIndicator	SHORT	2	14
PartitionCount	SHORT	2	16
IP_DETAILS_PARTITION_WISE [40] (Refer Table No 4.2)	CHAR	1040	18
SessionKey [8]	CHAR	8	1058
CryptographicKey [32]	CHAR	32	1066
Static Cryptographic IV	CHAR	8	1098
Dynamic Cryptographic IV	LONG LONG	8	1106
Cryptographic Additional Key	CHAR	12	1114

Field Name	Brief Description
MessageIndicator	This field will indicate whether there is more than one response expected for the request. If number of partitions exceed 40 then there will be additional response sent for next 40 partitions. The value set in this field will be following: ⇨ 0, Interim message – Meaning there is next response following this message ⇨ 1, Last Message – This is last response to the Logon Request
PartitionCount	This field will contain count of partition details that are present in this message.
IP_DETAILS_PARTITION_WISE	This will be array of IPs and Ports for number of partitions to connect. Max 40 partition details will be specified in

Field Name	Brief Description
	one message. If there are more partitions, then incremental details will be sent in subsequent response
SessionKey	This field contains Session Key for the validated session.
Cryptographic Key	Cryptographic key for both the encryption and decryption of all messages between member application and allocated Gateway Server.
Static Cryptographic IV	Static Cryptographic IV for both the encryption and decryption of all messages between member application and allocated Gateway Server.
Dynamic Cryptographic IV	Dynamic Cryptographic IV for both encryption and decryption of all messages between member application and allocated Gateway Server.
Cryptographic Additional Key	Cryptographic Additional Key for both the encryption and decryption of all messages between member application and allocated Gateway Server.

Secure User Registration Request

Table 4.4 GR_SECURE_USER_REGISTRATION_REQUEST

Structure Name	MS_ SECURE_USER_REGISTRATION_REQUEST_IN		
Packet Length	24 bytes		
Transaction Code	GR_SECURE_USER_REGISTRATION_REQUEST_IN (23008)		
Field Name	Data Type	Size in Byte	Offset
REQUEST MESSAGE HEADER (Refer Table 2.2 structure)	STRUCT	24	0

Field Name	Brief Description
Transcode	This field is the part of Message Header. The transaction code is 23008

Secure User Registration Response

Table 4.5 GR_SECURE_USER_REGISTRATION_RESPONSE

Structure Name	MS_ SECURE_USER_REGISTRATION_RESPONSE_OUT		
Packet Length	14 bytes		
Transaction Code	GR_SECURE_USER_REGISTRATION_RESPONSE_OUT (23009)		
Field Name	Data Type	Size in Byte	Offset
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0

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

Table 4.6 DC_SIGNON_IN

Structure Name	**DC_SIGNON IN**		
Packet Length	**40 bytes**		
Transaction Code	**DC_SIGNON_IN (2500)**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
REQUEST_MESSAGE_HEADER (Refer Table No 2.2)	STRUCT	24	0
Password [8]	CHAR	8	24
SessionKey [8]	CHAR	8	32

Field Name	**Brief Description**
TransactionCode	The transaction code is DC_SIGNON IN (2500)
Password	This field should contain the password entered by the user. A combination of alphabet, numbers and special characters are allowed in the password. The user should enter the valid password for a successful Logon. If password is less than 8-character length, the password should be padded by blank.

Field Name	Brief Description
SessionKey	Session Key as was received in GR_RESPONSE.

Table 4.6 DC_SIGNON_OUT

Structure Name	DC_SIGNON_OUT		
Packet Length	26 bytes		
Transaction Code	DC_SIGNON_OUT (2501)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0
UserId	INT	4	14
PartitionID	CHAR	6	18
ConcurrentLoginId	SHORT	2	24

Field Name	Brief Description
TransactionCode	The transaction code is DC_SIGNON_OUT (2501) in login response
UserId	This field contains User ID of user.
Partition ID	For messages coming from the host, this field contains the partition ID(Stream) from which the packet is being sent
ConcurrentLoginID	This field contains Concurrent login number of the User for which User has logged in.

Table 4.7 DC_SIGN_OFF

Structure Name	DC_SIGN_OFF		
Packet Length	26 bytes		
Transaction Code	DC_SIGNON_OFF (2320)		
Field Name	**Data Type**	**Field Name**	**Data Type**
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0
UserId	INT	4	14
PartitionID	CHAR	6	18

Structure Name	DC_SIGN_OFF		
Packet Length	26 bytes		
Transaction Code	DC_SIGNON_OFF (2320)		
Field Name	Data Type	Field Name	Data Type
ConcurrentLoginId	SHORT	2	24

Field Name	Brief Description
TransactionCode	The transaction code is DC_SIGN_OFF (2320)
UserId	This field contains User ID of user.
Partition ID	For messages coming from the host, this field contains the partition ID(Stream) from which the packet is being sent
ConcurrentLoginID	This field contains Concurrent login number of the User for which User has logged in.

Logon Error

In case of any error, the structure returned is:

ERROR RESPONSE (Refer to Error Response in Chapter 2)

Field Name	Brief Description
TransactionCode	The transaction code is DC_SIGNON_OUT (2501).
ErrorCode	This contains the error number. Refer to List of Error Codes in Appendix.

Chapter 5 Drop Copy Message Subscription

Introduction

NSE Drop Copy Service sends the user, intended Order & Trade download or Trade only messages. For receiving these messages, the user must send the respective subscription request on the different drop copy gateway partitions. In response to this request, the Trade only or Order & Trade messages are sent to the user.

The users must send separate subscription request for each Partition. The number of partitions and their details is sent in Logon Response from host during logon sequence.

In case of any disconnection during market hours, it is recommended to make use of the incremental download facility by sending the last received sequence number in the subsequent subscription request. It is advisable to not to initiate full download again.

In case of receipt of duplicate message sequence number the message may be dropped at user end. In case of gap in sequence number, request for Drop Copy subscription to be done by the User with last received sequence number.

In case of partition level TCP disconnection or complete disconnection across partitions, user will need to initiate from Logon Request.

Drop Copy Message Subscription Request

Table 5.1 DROP COPY MESSAGE SUBSCRIPTION

Structure Name	DC_SUBSCRIPTION_REQUEST		
Packet Length	32 bytes		
Transaction Code	DC_TRD_SUBSCRIPTION_REQUEST (8000) DC_ONT_SUBSCRIPTION_REQUEST (9000)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
REQUEST_MESSAGE_HEADER (Refer to Table 2.2)	STRUCT	24	0
SequenceNumber	LONG LONG	8	24

Field Name	Brief Description
TransactionCode	The transaction code is DC_TRD_SUBSCRIPTION_REQUEST (8000) for Trades only subscription and DC_ONT_SUBSCRIPTION_REQUEST (9000) for Order & Trade subscription and will be part of header
SequenceNumber	This contains the last sequence number as received by the user. To retrieve the messages from the beginning of the trading day, this field should be set to '0'.

Trade Subscription

In response to DC_TRD_SUBSCRIPTION_REQUEST (8000), below mentioned message packets will be sent to users. The packets having sequence number greater than the sequence number coming in DC SUBSCRIPTION_REQUEST are only downloaded

1. Trade Confirmation Response - TRADE_CONFIRMATION (2222)
2. Trade Cancellation Confirmation Response - TRADE_CANCEL_CONFIRM (2282)
3. Trade Cancellation Rejection Response - TRADE_CANCEL_REJECT (2286)
4. Trade Modification Confirmation Response - TRADE_MODIFY_CONFIRM (2287)
5. Trade Modify Reject - TRADE_MODIFY_REJECTED (2288)
6. Giveup Approval - GIVEUP_APP_CONFIRM (4506)
7. Giveup Rejection – GIVEUP_REJ_CONFIRM (4507)

The structures for the messages received on Trade Subscription can be found in the section "Message Structures"

Order & Trade (O&T) Subscription

In response to DC_ONT_SUBSCRIPTION_REQUEST (9000), below mentioned message packets will be sent to users. The packets having sequence number greater than the sequence number coming in DC SUBSCRIPTION_REQUEST are only downloaded

1. Price Confirmation Response - PRICE_CONFIRMATION (2012)
2. Order Modification Reject Response - ORDER_MOD_REJECT (2042)
3. Order Cancel Reject Response - ORDER_CANCEL_REJECT (2072)
4. Order Confirmation Response - ORDER_CONFIRMATION (2073)
5. Order Modification Confirmation Response - ORDER_MOD_CONFIRMATION (2074)
6. Order Cancel Confirmation Response - ORDER_CANCEL_CONFIRMATION (2075)
7. Freeze to Control - FREEZE_TO_CONTROL (2170)
8. On Stop Notification - ON_STOP_NOTIFICATION (2212)
9. Order Error Response - ORDER_ERROR (2231)
10. Batch Order Cancel - BATCH_ORDER_CANCEL (9002)
11. Spread Order Confirmation Response - SP_ORDER_CONFIRMATION (2124)
12. 2L Order Confirmation Response - TWOL_ORDER_CONFIRMATION (2125)
13. 3L Order Confirmation Response - THRL_ORDER_CONFIRMATION (2126)
14. Spread Cancel Confirmation Response - SP_ORDER_CXL_CONFIRMATION (2130)
15. 2L Cancel Confirmation Response - TWOL_ORDER_CXL_CONFIRMATION (2131)
16. 3L Cancel Confirmation Response - THRL_ORDER_CXL_CONFIRMATION (2132)
17. 2L Order Error Response - TWOL_ORDER_ERROR (2155)
18. 3L Order Error Response - THRL_ORDER_ERROR (2156)
19. Spread Batch Order Cancel - BATCH_SPREAD_CXL_OUT (9004)
20. Spread Order Modification Confirmation - SP_ORDER_MOD_CONFIRMATION (2136)
21. Spread Order Modification Reject Response - SP_ORDER_MOD_REJECT (2133)
22. Spread Order Reject Response - SP_ORDER_REJECTED (2154)
23. Spread Order Cancel Reject Response - SP_ORDER_CXL_REJECTED (2127)
24. All Trade Confirmations as listed for **"Trade Confirmations"**

Message structures for Order Confirmations can be found in the section "Message Structures". Order & Trade Subscription service is available at a member or user level.

Message Structures

Below are the structures for different types of confirmations:

Table 5.2 TRADE_CONFIRMATION

Structure Name	**TRADE_CONFIRMATION**		
Packet Length	**146 bytes**		
Transaction Code	**TRADE_CONFIRMATION (2222)** **TRADE_CANCEL_CONFIRM (2282)** **TRADE_CANCEL_REJECT (2286)** **TRADE_MODIFY_CONFIRM (2287)** **TRADE_MODIFY_REJECTD (2288)** **GIVEUP_APP_CONFIRM (4506)** **GIVEUP_REJ_CONFIRM (4507)**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0
OrderNumber	DOUBLE	8	14
TraderNumber	INT	4	22
BuySell	SHORT	2	26
OriginalVolume	INT	4	28
DisclosedVolume	INT	4	32
RemainingVolume	INT	4	38
DiclosedVolRemaining	INT	4	40
Price	INT	4	54
ST_ORDER_FLAGS	STRUCT	2	48
FillNumber	INT	4	50

Structure Name	TRADE_CONFIRMATION		
Packet Length	146 bytes		
Transaction Code	TRADE_CONFIRMATION (2222) TRADE_CANCEL_CONFIRM (2282) TRADE_CANCEL_REJECT (2286) TRADE_MODIFY_CONFIRM (2287) TRADE_MODIFY_REJECTD (2288) GIVEUP_APP_CONFIRM (4506) GIVEUP_REJ_CONFIRM (4507)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
FillQty	INT	4	54
FillPrice	INT	4	58
Token	INT	4	62
BookType	SHORT	2	66
ProClient	SHORT	2	68
Algo ID	INT	4	70
ActivityTimeInNanos	LONG	8	74
NNFField	DOUBLE	8	82
Segment	SHORT	2	90
BrokerId [5]	CHAR	5	92
Filler	CHAR	1	97
PAN [10]	CHAR	10	98
AccountNumber [10]	CHAR	10	108
Settlor	CHAR	12	118
GiveupFlag	CHAR	1	130
OpenClose	CHAR	1	131
Participant [12]	CHAR	12	132
ReasonCode	SHORT	2	144

Field Name	Brief Description
TransactionCode	The transaction code for Trade Confirmation (e.g. 2222, 2282, 2286, 2287, 2288, 4506, 4507 etc).
OrderNumber	This field contains the order number of the trader's order taking part in the trade.
BrokerId	This field contains the Trading Member ID.
TraderNumber	This field contains the trader or user ID.
AccountNumber	This field contains the Account Number or Client code.
BuySell	This field contains one of the following values based on Buy or Sell. '1' for Buy '2' for Sell.
OriginalVolume	This field contains the Original traded volume.
DisclosedVolume	This field contains the quantity to be disclosed to the market.
RemainingVol	This field contains the volume remaining after trade(s).
DisclosedVolRemaining	This field contains the disclosed volume remaining after trade(s).
Price	This field contains the order price.
OrderFlags	Refer to Table No 2.5 Note: Preopen Indicator will be set as 0 for the trades happening in Normal Market session for Normal Market orders and pre-open carried forward orders Preopen indicator will be set as 1 for trades happening in the call auction 2 market. Applicable for CM Segment only.
FillNumber	This field contains the trade number.
FillQty	This field contains the traded volume.
FillPrice	This field contains the price at which order is traded.
Token	Security identifier
BookType	This field contains the book type - RL/ ST/ SL/ NT/ OL/ SP/ AU/CA/CB.
ProClient	This field is same as Pro/Client /WHS indicator having one of the following values: '1' - client's order '2' - broker's order '4' - warehousing order Applicable for CM segment only.
PAN	This field contains the PAN (Permanent Account Number)
Algo ID	This field contains the Algo ID

Field Name	Brief Description
ActivityTimeInNanos	Activity time in nanoseconds
NNFField	This field contains a 15 digit a unique identifier for various products deployed as per Exchange circular download ref no 16519 dated December 14, 2010 and as updated from time to time Note: This field will be populated only for Trade Confirmation (2222) transcode. For the rest of the transcode this field will be null.
Segment	Represents the business segment. • '1' – Equity Derivatives (F&O) • '2' – Currency Derivatives • '3' – Capital Markets (Cash Equity) • '4' – SLBM Markets • '5' – Commodity Derivatives
Settlor	This field contains the ID of the participants who are responsible for settling the trades through the custodians. By default, all orders are treated as broker's own orders and this field defaults to the Broker Code. Note: This field will be populated only for Trade Confirmation (2222) transcode. For the rest of the transcode this field will be null.
OpenClose	This field contains either 'O' or 'C' for close.
Participant	This field contains the participant's name for trade confirmation
GiveupFlag	This field should contain Give up flag for transcodes GIVEUP_APP_CONFIRM (4506) & GIVEUP_REJ_CONFIRM (4507). For remaining transcodes it will be blank. If give up is approved, Host should send 'A'.
ReasonCode	This field contains the reason code for GiveUp packets. This has details regarding the error along with the error code. Refer to Reason Codes in FO_Trimmed_NNF_PROTOCOL document.

Table 5.4 Order Response Message

Structure Name	**ORDER_RESPONSE**		
Packet Length	**130 bytes**		
Transaction Code	**PRICE_CONFIRMATION (2012)** **ORDER_MOD_REJECT (2042)** **ORDER_CANCEL_REJECT (2072)** **ORDER_CONFIRMATION (2073)** **ORDER_MOD_CONFIRMATION (2074)** **ORDER_CANCEL_CONFIRMATION (2075)** **FREEZE_TO_CONTROL (2170)** **ON_STOP_NOTIFICATION (2212)** **ORDER_ERROR (2231)** **BATCH_ORDER_CANCEL (9002)**		
Field Name	**Data Type**	**Size in Byte**	**Offset**
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0
ReasonCode	SHORT	2	14
Token	INT	4	16
OrderNumber	DOUBLE	8	20
BookType	SHORT	2	28
BuySell	SHORT	2	30
DisclosedVolume	INT	4	32
DisclosedVolRemaining	INT	4	36
TotalVolRemaining	INT	4	40
Volume	INT	4	44
Price	INT	4	48
TriggerPrice	INT	4	52
ST_ORDER_FLAGS	STRUCT	2	56
TraderId	INT	4	58
NNFField	DOUBLE	8	62
AlgoID	INT	4	70
ActivityTimeInNanos	LONG	8	74

Structure Name	ORDER_RESPONSE		
Packet Length	130 bytes		
Transaction Code	PRICE_CONFIRMATION (2012) ORDER_MOD_REJECT (2042) ORDER_CANCEL_REJECT (2072) ORDER_CONFIRMATION (2073) ORDER_MOD_CONFIRMATION (2074) ORDER_CANCEL_CONFIRMATION (2075) FREEZE_TO_CONTROL (2170) ON_STOP_NOTIFICATION (2212) ORDER_ERROR (2231) BATCH_ORDER_CANCEL (9002)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
Reserved	SHORT	2	82
Reserved	SHORT	2	84
Reserved	SHORT	2	86
Suspended	CHAR	1	88
Open/Close	CHAR	1	89
PAN	CHAR	10	90
Segment	CHAR	1	100
ParticipantType	CHAR	1	101
AccountNumber	CHAR	10	102
ProClient	CHAR	1	112
Reserved	CHAR	1	113
Settlor	CHAR	12	114
ReferenceId	INT	4	126

Field Name	Brief Description
TransactionCode	The transaction code for O&T Confirmation (e.g., 2042, 2072, etc)

Field Name	Brief Description
ParticipantType	Since only exchange can initiate the auction, this field should be set to 'I' for initiator. This should be set to 'C' for competitor order and 'S' for solicitor order. Applicable for CM Segment only.
Reserved	This field should not be used.
ReasonCode	This field contains the reason code for a particular order request rejection or order being frozen. This has the details regarding the error along with the error code. Refer to Reason Codes in FO_Trimmed_NNF_PROTOCOL document.
Token	Instrument/token identifier
Suspended	This field specifies whether the security is suspended or not.
OrderNumber	This field contains an Order Number assigned to the order. It is a unique identification for an order. The first two digits will contain the stream number. The next fourteen digits will contain fourteen-digit sequence number.
AccountNumber [10]	If the order is entered on behalf of a trader, the trader account number should be specified in this field. For broker's own order, this field should be set to broker code.
BookType	This field contains the type of order. BOARD_LOT_IN_TR (20000) must have BookType 1 or 11 or 12.
BuySell	This field should specify whether the order is a buy or sell. It should take one of the following values: • '1' for Buy Order • '2' for Sell Order
DisclosedVolume	This field contains the quantity that has to be disclosed to the market. It is not applicable if the order has either the All Or None or the Immediate Or Cancel attribute set. It should not be greater than the volume of the order and not less than the Minimum Fill quantity if the Minimum Fill attribute is set. In either case, it cannot be less than the Minimum Disclosed Quantity allowed. It should be a multiple of the Regular lot
DisclosedVolRemaining	This field contains the disclosed volume remaining from the original disclosed volume after trade(s).
TotalVolRemaining	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry, this field should be set to Volume. Thereafter, for every response the system will return this value.
Volume	This field represents the quantity of the order placed.
Price	This field contains the price at which the order is placed. For Market orders, the price will be zero.
TriggerPrice	Applicable only for a Stop Loss order, this field provides the price at which the order is to be triggered and brought to the market. For a Stop Loss buy order, the trigger price will be less

Field Name	Brief Description
	than or equal to the limit price but greater than the last traded price. For a Stop Loss sell order, the trigger price will be greater than or equal to the limit price but less than the last traded price.
Order Flags	Refer to Table No 2.5 Note: Preopen Indicator will be set as 0 for the trades happening in Normal Market session for Normal Market orders and pre-open carried forward orders Preopen indicator will be set as 1 for trades happening in the call auction 2 market. Applicable for CM Only.
Trader Id	In Request packet, this field contains the ID of the user on whose behalf order is to be modified/cancelled. This field accepts only numbers
ProClient	This field contains one of the following values based on the order entering on behalf of the broker or a trader. • 1 represents client's order. • 2 represents broker's order. • 4 represents warehousing order
NNFField	This field contains a 15 digit a unique identifier for various products deployed as per Exchange circular download ref no 16519 dated December 14, 2010 and as updated from time to time
PAN	This field shall contain the PAN (Permanent Account Number / PAN_EXEMPT) - This field shall be mandatory for all orders (client / participant / PRO orders).
AlgoID	For Algo order this field shall contain the Algo ID issued by the exchange. For Non-Algo order, this field shall be Zero (0)
ActivityTimeInNanos	This field contains the timestamp value in nanoseconds.
Segment	Represents the business segment. • '1' – Equity Derivatives (F&O) • '2' – Currency Derivatives • '3' – Capital Markets (Cash Equity) • '4' – SLBM Markets • '5' – Commodity Derivatives
Settlor	This field contains the ID of the participants who are responsible for settling the trades through the custodians. By default, all orders are treated as broker's own orders and this field defaults to the Broker Code.
Open/Close	Open / Close order indicator. This field should contain one of the following values. • 'O' for Open • 'C' for Close

Field Name	Brief Description
LastActivityReference	In case of order entry response, this field will contain a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified. This field should be set to zero for the order entry request.
ReferenceId	This field value will be echoed back as the value received in respective order packets ReferenceId field.

Table 5.5 Spread/2L/3L Order Entry

Structure Name	SPD_2L_3L_RESPONSE		
Packet Length	146 bytes		
Transaction Code	SP_ORDER_CONFIRMATION (2124) SP_ORDER_REJECTED (2154) SP_ORDER_MOD_REJECT (2133) SP_ORDER_CXL_REJECTED (2127) SP_ORDER_MOD_CONFIRMATION (2136) BATCH_SPREAD_CXL_OUT (9004) TWOL_ORDER_CONFIRMATION (2125) TWOL_ORDER_CXL_CONFIRMATION (2131) TWOL_ORDER_ERROR (2155) THRL_ORDER_CONFIRMATION (2126) THRL_ORDER_CXL_CONFIRMATION (2132) THRL_ORDER_ERROR (2156) SP_ORDER_CXL_CONFIRMATION (2130)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
MESSAGE_HEADER (Refer Table No 2.4)	STRUCT	14	0
Token1	INT	4	14
OrderNumber1	DOUBLE	8	18
BookType1	SHORT	2	26
BuySell1	SHORT	2	28

Structure Name	SPD_2L_3L_RESPONSE		
Packet Length	146 bytes		
Transaction Code	SP_ORDER_CONFIRMATION (2124) SP_ORDER_REJECTED (2154) SP_ORDER_MOD_REJECT (2133) SP_ORDER_CXL_REJECTED (2127) SP_ORDER_MOD_CONFIRMATION (2136) BATCH_SPREAD_CXL_OUT (9004) TWOL_ORDER_CONFIRMATION (2125) TWOL_ORDER_CXL_CONFIRMATION (2131) TWOL_ORDER_ERROR (2155) THRL_ORDER_CONFIRMATION (2126) THRL_ORDER_CXL_CONFIRMATION (2132) THRL_ORDER_ERROR (2156) SP_ORDER_CXL_CONFIRMATION (2130)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
TotalVolRemaining1	INT	4	30
Volume1	INT	4	34
ST_ORDER_FLAGS	STRUCT	2	38
TraderId1	INT	4	40
NnfField	DOUBLE	8	44
Algo ID	INT	4	52
ActivityTimeInNanos	LONG	8	56
PAN	CHAR	10	64
AccountNumber1	CHAR	10	74
ProClient1	CHAR	1	84
Segment	CHAR	1	85
OpenClose1	CHAR	1	86
Reserved	CHAR	1	87
Settlor1	CHAR	12	88

Structure Name	SPD_2L_3L_RESPONSE		
Packet Length	146 bytes		
Transaction Code	SP_ORDER_CONFIRMATION (2124) SP_ORDER_REJECTED (2154) SP_ORDER_MOD_REJECT (2133) SP_ORDER_CXL_REJECTED (2127) SP_ORDER_MOD_CONFIRMATION (2136) BATCH_SPREAD_CXL_OUT (9004) TWOL_ORDER_CONFIRMATION (2125) TWOL_ORDER_CXL_CONFIRMATION (2131) TWOL_ORDER_ERROR (2155) THRL_ORDER_CONFIRMATION (2126) THRL_ORDER_CXL_CONFIRMATION (2132) THRL_ORDER_ERROR (2156) SP_ORDER_CXL_CONFIRMATION (2130)		
Field Name	**Data Type**	**Size in Byte**	**Offset**
PriceDiff	INT	4	100
ReferenceId	INT	4	104
ReasonCode	SHORT	2	108
MS_SPD_LEG_INFO (leg 2)	STRUCT	18	108
MS_SPD_LEG_INFO (leg 3)	STRUCT	18	126

Field Name	Brief Description
TransactionCode	The transaction code is SP_ORDER_CONFIRMATION (2124) SP_ORDER_REJECTED (2154) SP_ORDER_MOD_REJECT (2133)

	SP_ORDER_CXL_REJECTED (2127) SP_ORDER_MOD_CONFIRMATION (2136) BATCH_SPREAD_CXL_OUT (9004) TWOL_ORDER_CONFIRMATION (2125) TWOL_ORDER_CXL_CONFIRMATION (2131) TWOL_ORDER_ERROR (2155) THRL_ORDER_CONFIRMATION (2126) THRL_ORDER_CXL_CONFIRMATION (2132) THRL_ORDER_ERROR (2156).
ParticipantType1	This is not used.
CompetitorPeriod1	This is not used.
SolicitorPeriod1	This is not used.
ReasonCode	This field contains the reason code for a particular order request rejection or order being frozen. This has the details regarding the error along with the error code. Refer to Reason Codes in FO_Trimmed_NNF_PROTOCOL document.
Token1	This field should contain the contract descriptor of the contract. The validations will be done only on contract descriptor.
OrderNumber1	This field should be set to blank for the order entry request.
AccountNumber1	If the order is entered on behalf of a trader, the Trader Account Number is specified in this field. For broker's own order, this field is set to blank.
BookType1	This field should contain one of the following two book types. '1' – Regular lot order '2' – Special terms order
BuySell1	This field should contain one of the following values to specify whether the order is a buy or sell order. • '1' denotes Buy order • '2' denotes Sell order
DisclosedVol1	This is not used. This must be sent as zero for the order entry request.
DisclosedVolRemaining1	This is not used. This must be sent as zero for the order entry request.
TotalVolRemaining1	This field should specify the total quantity remaining from the original quantity after trade(s). For order entry, this field must be set to Volume. For every response, the trading system will return this value.
Volume1	This field should contain the quantity for which the order is placed. The quantity should always be in multiples of Regular Lot and be less than the issued capital. The order will be rejected directly if the quantity is greater than or equal to the freeze quantity determined by NSE-Control.

Price1	For spread order this is not used. This must be sent as zero in spread order entry.
TriggerPrice1	This is not used. This must be sent as zero for the order entry request.
LastModified1	This is not used. This must be sent as zero for the order entry request.
TraderId1	This field should contain the user ID.
OpenClose1	Open / Close order indicator. This field must be set to one of the following: • 'O' denotes Open • 'C' denotes Close
Settlor1	This field should contain the ID of the participants who are responsible for settling the trades through the custodians. For 'Pro' order (brokers own order) this field should be left blank.
ProClient1	This field should contain one of the following values to specify whether the order is entered on behalf of the broker or a trader. • '1' represents the client's order. • '2' represents a broker's order.
NnfField	This field should contain a 15 digit a unique identifier for various products deployed as per Exchange circular download ref no. 16519 dated December 14, 2010 and as updated from time to time
PAN	This field shall contain the PAN (Permanent Account Number/PAN_EXEMPT). This field shall be mandatory for all orders (client/participant/PRO orders).
Algo ID	For Algo order this field shall contain the Algo ID issued by the exchange. For Non-Algo order, this field shall be Zero(0)
LastActivityReference	In case of order entry response, this field will contain a unique value. Currently the same shall be in nanoseconds. Changes if any shall be notified.
PriceDiff	This is the difference between the prices at which leg2 and leg1 should trade and it should be less than 9999999.9 Note: This is used for spread order only. It is not used for 2L/3L.
ReferenceId	This field value will be echoed back as the value received in respective order packets ReferenceId field.

Table 5.6 SPD_LEG_INFO

Structure Name	MS_SPD_LEG_INFO		
Packet Length	18 bytes		
Field Name	Data Type	Size in Byte	Offset
Token2	INT	4	0
BuySell2	SHORT	2	4
TotalVolRemaining2	INT	4	6
Volume2	INT	4	10
ST_ORDER_FLAGS	STRUCT	2	14
OpenClose2	CHAR	1	16
Reserved	CHAR	1	17

Field Name	Brief Description
Token2	This field should contain the contract descriptor of the contract. The validations will be done only on contract descriptor.
BuySell2	This field should contain one of the following values to specify if the order is a buy or sell. • '1' denotes Buy order • '2' denotes Sell order
DisclosedVol2	This is not used. This must be sent as zero for the order entry request.
DisclosedVolRemaining2	This is not used. This must be sent as zero for the order entry request.
TotalVolRemaining2	This field specifies the total quantity remaining from the original quantity after trade(s). For order entry this field should be set to Volume. For every response the trading system will return this value.
Volume2	This field should contain the quantity of order placed. The quantity should always be in multiples of Regular Lot and be less than the issued capital. The order will be rejected directly if the quantity is greater than or equal to the freeze quantity determined by NSE-Control.
VolumeFilledToday2	This is not used. This must be sent as blank for the order entry request.
Price2	This is not used. This must be sent as zero for the order entry request.
TriggerPrice2	This is not used. This must be sent as zero for the order entry request.

OpenClose2	Open / Close order indicator. This field should contain one of the following values: • 'O' denotes Open • 'C' denotes Close

Drop Copy Error Response

In case there is any error in request, the system will reject the request and send drop copy error response message to user. The reason for rejection is given in error code field in message header.

ERROR RESPONSE (Refer to Error Response in Chapter 2)

Field Name	Brief Description
TransactionCode	The transaction code is DC_ERROR_RESPONSE (8006 / 9006).
ErrorCode	This contains the error number. Refer to List of Error Codes in Appendix section.

Appendix

List of Error Codes

Error Code ID	Error Code Value	Description of Error Code
ERR_INVALID_USER_TYPE	16001	Invalid User Type
ERR_INVALID_STREAM_ID	16002	Requested download Partition ID doesn't match with logged in Partition ID.
ERR_INVALID_SIGNON	16006	Invalid sign-on, please try again.
ERR_INVALID_BROKER_OR_BRANCH	16041	Trading Member does not exist in the system.
ERR_USER_NOT_FOUND	16042	Dealer does not exist in the system.
ERR_USER_IS_DISABLED	16134	This Dealer is disabled. Please call the Exchange.
ERR_INVALID_USER_ID	16148	Invalid Dealer ID entered.
ERR_INVALID_TRADER_ID	16154	Invalid Trader ID entered.
ERR_BROKER_NOT_ACTIVE	16285	The broker is not active.
ERR_INVALID_SEQUENCE_NO	16801	Invalid sequence number in drop copy download request.
ERR_INVALID_TRANSACTION_CODE	16802	Invalid transcode in drop copy download request.
GR_ERR_RETRY	16007	Technical Error, please retry. If the issue persists, contact customer support.
GR_ERR_MULTIPLE_GR_QUERY	19029	Multiple GR_QUERY request received.
GR_ERR_MULTIPLE_GR_QUERY_FROM_SAME_IP	16089	Multiple GR_QUERY request received from same IP.
GR_ERR_INVALID_USER_ID	16148	Invalid Dealer ID entered.

GR_ERR_INVALID_MSGLENGTH	16573	Message length is invalid.
GR_ERR_INVALID_TXNCODE	16003	Erroneous transaction code received.
GR_ERR_INVALID_CONCURRENT_LOGIN_ID	16039	Concurrent login ID is out of the allowed range.
GR_ERR_USER_NOT_ALLOWED_FOR_DC	16059	User is not authorized.
GR_ERR_CONNECTION_TIMEOUT	16069	Session timeout.
ERR_INVALID_CONNECTION	17003	FuncIndc field is invalid.
ERR_MD5_CHECKSUM_FAILURE	19031	MD5 Checksum Failed.

List of Drop copy Transaction Codes

Transaction Code	Code	Structure
DC_SIGNON_IN	2500	DC_SIGNON_IN
DC_SIGNON_OUT	2501	DC_SIGNON_OUT
GR_SECURE_USER_REGISTRATION_REQUEST	23008	GR_SECURE_USER_REGISTRATION_REQUEST
GR_SECURE_USER_REGISTRATION_RESPONSE	23009	GR_SECURE_USER_REGISTRATION_RESPONSE
DC_TRD_SUBSCRIPTION_REQUEST	8000	DC_TRD_REQUEST
DC_ONT_SUBSCRIPTION_REQUEST	9000	DC_ONT_REQUEST
DC_ERROR_RESPONSE	9006	DC_ERROR_RESPONSE (ORDER)
DC_ERROR_RESPONSE	8006	DC_ERROR_RESPONSE (TRADE)
TRADE_CONFIRMATION	2222	TRADE_CONFIRMATION
TRADE_CANCEL_CONFIRM	2282	TRADE_CONFIRMATION
TRADE_CANCEL_REJECT	2286	TRADE_CONFIRMATION

Transaction Code	Code	Structure
TRADE_MODIFY_CONFIRM	2287	TRADE_CONFIRMATION
ORDER_CONFIRMATION	2073	ORDER_RESPONSE
ORDER_ERROR	2231	ORDER_RESPONSE
ORDER_MOD_CONFIRMATION	2074	ORDER_RESPONSE
ORDER_MOD_REJECT	2042	ORDER_RESPONSE
ORDER_CANCEL_CONFIRMATION	2075	ORDER_RESPONSE
ORDER_CANCEL_REJECT	2072	ORDER_RESPONSE
BATCH_ORDER_CANCEL	9002	ORDER_RESPONSE
ON_STOP_NOTIFICATION	2212	ORDER_RESPONSE
PRICE_CONFIRMATION	2012	ORDER_RESPONSE
FREEZE_TO_CONTROL	2170	ORDER_RESPONSE
SP_ORDER_CONFIRMATION	2124	SPD_2L_3L_RESPONSE
TWOL_ORDER_CONFIRMATION	2125	SPD_2L_3L_RESPONSE
THRL_ORDER_CONFIRMATION	2126	SPD_2L_3L_RESPONSE
SP_ORDER_CXL_CONFIRMATION	2130	SPD_2L_3L_RESPONSE
TWOL_ORDER_CXL_CONFIRMATION	2131	SPD_2L_3L_RESPONSE
THRL_ORDER_CXL_CONFIRMATION	2132	SPD_2L_3L_RESPONSE
SP_ORDER_ERROR	2154	SPD_2L_3L_RESPONSE
TWOL_ORDER_ERROR	2155	SPD_2L_3L_RESPONSE

Transaction Code	Code	Structure
THRL_ORDER_ERROR	2156	SPD_2L_3L_RESPONSE
BATCH_SPREAD_CXL_OUT	9004	SPD_2L_3L_RESPONSE
DC_SIGN_OFF	2320	DC_SIGN_OFF
SP_ORDER_MOD_CONFIRMATION	2136	SPD_2L_3L_RESPONSE
SP_ORDER_MOD_REJECT	2133	SPD_2L_3L_RESPONSE
SP_ORDER_CXL_REJECTED	2127	SPD_2L_3L_RESPONSE
TRADE_MODIFY_REJECTED	2288	TRADE_CONFIRMATION
GIVEUP_APP_CONFIRM	4506	TRADE_CONFIRMATION
GIVEUP_REJ_CONFIRM	4507	TRADE_CONFIRMATION

Note: If any transcode is not present in NNF document, then end user needs to drop that transcode.

List of Transaction Codes Containing Timestamp in Nanoseconds

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

Transaction Code	Code
PRICE_CONFIRMATION	2012
ORDER_MOD_REJECT	2042
ORDER_CANCEL_REJECT	2072
ORDER_CONFIRMATION	2073
ORDER_MOD_CONFIRMATION	2074
ORDER_CANCEL_CONFIRMATION	2075

FREEZE_TO_CONTROL	2170
ON_STOP_NOTIFICATION	2212
ORDER_ERROR	2231
BATCH_ORDER_CANCEL	9002
TRADE_CONFIRMATION	2222
TRADE_CANCEL_CONFIRM	2282
TRADE_CANCEL_REJECT	2286
TRADE_MODIFY_CONFIRM	2287
SP_ORDER_CONFIRMATION	2124
TWOL_ORDER_CONFIRMATION	2125
THRL_ORDER_CONFIRMATION	2126
SP_ORDER_MOD_CONFIRMATION	2136
SP_ORDER_MOD_REJECT	2133
SP_ORDER_CXL_REJECTED	2127
TRADE_MODIFY_REJECTED	2288
GIVEUP_APP_CONFIRM	4506
GIVEUP_REJ_CONFIRM	4507

Note: If any transcode is not present in NNF document, then end user needs to drop that transcode.

Annexure for Encryption/Decryption

Sr. No.	The following are sample function calls of OpenSSL library in Linux (for reference)
1	***Note –*** • Openssl Library version used is OpenSSL3.4.0. • TLS protocol version has been set to 1.3 (TLS1_3_VERSION). Following are the system library calls for TLS1.3- **SSL/TLS library initialization →** 1. **SSL_library_init**() - Initialize SSL library by registering algorithms. 2. **OpenSSL_add_all_algorithms**() - Adds all algorithms to the table (digests and ciphers) 3. **SSL_load_error_strings**() - Registers the error strings for all libcrypto and libssl error strings. 4. **SSL_CTX_new**(**TLS_client_method**()) - Create a new SSL_CTX object as framework for TLS/SSL enabled functions. 5. **SSL_CTX_set_min_proto_version**(SSL_CTX *ctx, int version) - Set the minimum protocol versions to TLS1_3_VERSION. 6. **SSL_CTX_set_max_proto_version**(SSL_CTX *ctx, int version) - Set the maximum protocol versions to TLS1_3_VERSION. **Establishing the SSL/TLS connection→** 1. **socket**(PF_INET, SOCK_STREAM, 0) - Create TCP socket. 2. **connect**(int sockfd, const struct sockaddr *addr, socklen_t addrlen) - Initiate the TCP/IP connection with server. 3. **SSL_new**(SSL_CTX *ctx) - Create new SSL connection state. 4. **SSL_set_fd**(SSL *ssl, int fd) - Attach the socket descriptor. 5. **SSL_connect**(SSL *ssl) - Perform the SSL connection. **Validating the Gateway Router server certificate →** 1. **SSL_get_peer_certificate**(const SSL *ssl) - Get the server's certificate. 2. **X509_STORE_new**() - This function returns a new X509_STORE. 3. **X509_STORE_CTX_new**() - This function returns a newly initialised X509_STORE_CTX. 4. **X509_STORE_load_locations**(X509_STORE *ctx, const char *file, const char *dir) - Configure files and directories used by a certificate store. The path of CA certificate

	(gr_ca_cert1.pem) will be used in this function. The CA certificate (gr_ca_cert1.pem) will be provided by the Exchange for validation of Gateway Router certificate. 5. **X509_STORE_CTX_init**(X509_STORE_CTX *ctx, X509_STORE *trust_store, X509 *target, STACK_OF(X509) *untrusted) - This function returns a newly initialised X509_STORE_CTX structure. 6. **X509_verify_cert**(X509_STORE_CTX *ctx) - This function builds and verify X509 certificate chain. **Send and Receive messages on SSL/TLS connection →** 1. **SSL_write**(SSL *ssl, const void *buf, int num) - Send message on SSL. 2. **SSL_read**(SSL *ssl, void *buf, int num) - Receive message from SSL.

2	**For symmetric encryption/decryption methodology –**
	
	GCM_IV_LEN 16
	aad_len 12
	GCM_TAG_LEN 16
	
	Encryption Block:
	
	Initialization→
	User receives list of partition from GR.
	For each partition, user need to create encryption-context which can be achieved by calling below function.
	For each partition user will have to use separate encryption-context.
	
	void encrypt_init(EVP_CIPHER_CTX **enc_ctx,
	const unsigned char *key,
	const unsigned char *iv)
	{
	int retv = 0;
	if (!*enc_ctx) *enc_ctx = **EVP_CIPHER_CTX_new()**;
	retv = **EVP_EncryptInit** (*enc_ctx, **EVP_aes_256_gcm**(), NULL, NULL);
	retv = **EVP_CIPHER_CTX_ctrl** (*enc_ctx, EVP_CTRL_GCM_SET_IVLEN, GCM_IV_LEN, NULL);
	retv = **EVP_EncryptInit**(*enc_ctx, NULL, key, NULL);
	}
	
	Encryption→

To encrypt data, use partition specific encryption-context which was created in above initialization steps.

```
void encrypt_data(EVP_CIPHER_CTX *enc_ctx,
                  const unsigned char *iv,
                  const unsigned char *plaintext,
                  int plaintext_len,
                  const unsigned char *aad,
                  int aad_len,
                  unsigned char *ciphertext,
                  unsigned char *tag)
{
    int len = 0;
    int len2=0;
    int retv = 0;
    retv = EVP_EncryptInit(enc_ctx, NULL, NULL, iv);
    retv = EVP_EncryptUpdate(enc_ctx, NULL, &len, aad, aad_len);
    retv = EVP_EncryptUpdate(enc_ctx, ciphertext, &len, plaintext, plaintext_len);
OSSL_PARAM params[2] = { OSSL_PARAM_END, OSSL_PARAM_END };
    retv = EVP_EncryptFinal_ex(enc_ctx, ciphertext, &len2);
    params[0] =
OSSL_PARAM_construct_octet_string(OSSL_CIPHER_PARAM_AEAD_TAG, tag,
GCM_TAG_LEN);
    retv = EVP_CIPHER_CTX_get_params(enc_ctx, params);
}
```

Decryption Block:

Initialization→

Similar to encryption-context user need to create decryption-context or each partition, which can be achieved by calling below function.
For each partition user will have to use separate decryption-context.

```
void decrypt_init(EVP_CIPHER_CTX **dec_ctx,
                  const unsigned char *key,
                  const unsigned char *iv)
{
    int retv = 0;
    if (!*dec_ctx) *dec_ctx = EVP_CIPHER_CTX_new();
    retv = EVP_DecryptInit(*dec_ctx, EVP_aes_256_gcm(), NULL, NULL);
```

```
    retv = EVP_CIPHER_CTX_ctrl(*dec_ctx, EVP_CTRL_GCM_SET_IVLEN,
GCM_IV_LEN, NULL);
    retv = EVP_DecryptInit(*dec_ctx, NULL,key, NULL);
}
```

Decryption→

To decrypt data, use partition specific decryption-context which was created in above initialization steps.

```
void decrypt_data(EVP_CIPHER_CTX *dec_ctx,
                  const unsigned char *iv,
                  const unsigned char *ciphertext,
                  int ciphertext_len,
                  const unsigned char *aad,
                  int aad_len,
                  const unsigned char *tag,
                  unsigned char *plaintext)
{
    int len = 0; int len2 = 0;
    int retv;
    retv = EVP_DecryptInit(dec_ctx, NULL, NULL, iv);
    retv = EVP_DecryptUpdate(dec_ctx, NULL, &len, aad, aad_len);
    retv = EVP_DecryptUpdate(dec_ctx, plaintext, &len, ciphertext, ciphertext_len);
    OSSL_PARAM params[2] = { OSSL_PARAM_END, OSSL_PARAM_END };
    params[0] =
OSSL_PARAM_construct_octet_string(OSSL_CIPHER_PARAM_AEAD_TAG, tag,
GCM_TAG_LEN);
    retv = EVP_CIPHER_CTX_set_params(dec_ctx, params);
    retv = EVP_DecryptFinal_ex(dec_ctx,plaintext, &len2);
    if (retv <= 0)
        printf("!!!Decryption Failed!!!\n");
    else
        printf("!!!Decryption Successful!!!\n"); }
```

Note –

- The ones highlighted in bold are OpenSSL library functions.
- plaintext is the actual message buffer.
- ciphertext is the encrypted message buffer.

	//=====Pseudocode Dynamic IV changes=====
	// Define the IV structure
	typedef struct { char caStaticIv[8]; // Static IV (8 bytes)
	long long lDynamicIv; // Dynamic IV (64-bit integer)
	} CRYPTOGRAPHIC_IV_KEY;
	
	// Original IV received from GR response
	CRYPTOGRAPHIC_IV_KEY sIv;
	
	// Separate copies for encryption and decryption
	CRYPTOGRAPHIC_IV_KEY sEncCryptoGraphicIv;
	CRYPTOGRAPHIC_IV_KEY sDecCryptoGraphicIv;
	
	// Step 1: Initialize from GR response
	sIv = get_iv_from_gr_response(); // sIv is populated with the static and dynamic IV values
	
	// Step 2: Create two copies - One for encryption and One for decryption.
	sEncCryptoGraphicIv = sIv;
	sDecCryptoGraphicIv = sIv;
	
	// Step 3: Before Encryption - The dynamic IV is incremented by 1.
	sEncCryptoGraphicIv.lDynamicIv += 1;
	encrypted_data = encrypt(data, &sEncCryptoGraphicIv);
	
	// Step 4: Before Decryption - The dynamic IV is decremented by 1.
	sDecCryptoGraphicIv.lDynamicIv -= 1;
	decrypted_data = decrypt(encrypted_data, &sDecCryptoGraphicIv);

FAQs

Q – What do I need to do before I try connecting directly to Drop Copy Service system?

Kindly contact Member Services Team before initiating the connectivity with Drop Copy Service or any other Exchange provided system.

Q – Where to connect?

Exchange shall provide a list of addresses, IP address and Port number(s). to connect to the Drop Copy service.

Q – How to connect?

Member's application must initiate a TCP socket connection to the address given by the Exchange and follow the login process as mentioned in the document.

Q – How to Logoff?

Member's application must shut down the established TCP connection(s) gracefully to log-off from Exchange Drop Copy service.

Q – What User Ids / Passwords to be used for login to drop copy?

Member should use existing NNF User ID and password, as used for login to Exchange Trading system.

Q – How to reset the password through drop copy?

Through drop copy user can't reset the password but any password change/reset done via the Exchange Trading System will be reflected in Drop Copy system. New login to drop copy service, after password reset via Trading System, should be done with the new password.

Q – With the same user id can we take simultaneously login on Interactive channel for order entry and on Drop Copy channel?

Yes. Drop copy channel is independent of the Interactive channel.

Q – What information shall be provided in the drop copy?

If only trade data API is implemented as per section 1 then following information will be sent through Drop Copy system

- Trade confirmation
- Trade modification confirmation
- Trade modification reject
- Trade cancel confirmation
- Trade cancel reject
- Giveup approval confirmation
- Giveup reject confirmation

If order and trade data API is implemented as per section 2 then following information will be sent through Drop Copy system

- Trade confirmation
- Trade modification confirmation
- Trade cancel confirmation
- Trade cancel reject
- Giveup approval confirmation
- Giveup reject confirmation
- Order confirmation
- Order modification reject
- Order modification confirmation
- Order cancel reject
- Order cancel confirmation
- Price confirmation
- Freeze to Control
- On Stop Notification
- Order error
- Batch Order cancel
- Spread order confirmation
- Spread order cancel reject
- Spread order cancel confirmation
- Spread order modification reject
- Spread order modification confirmation

- Spread order reject
- Spread batch cancel out
- 2L order confirmation
- 2L order cancel confirmation
- 2L order error
- 3L order confirmation
- 3L order cancel confirmation
- 3L order error

Q – Will clearing member also get trade data?

Yes. All the trades related to Clearing Member will be available.

Q – How shall we know that we have received all the trades (End of Day)?

No explicit message will be sent to indicate end of messages.

Q – What happens if I login late or miss receiving some trade in the drop copy channel?

During download request user needs to specify the last received sequence number from where the messages download should start.

Q – For order and trade data how will sequence number field value in Message header be provided?

Sequence number is user wise and streamwise unique value maintained at host end. For different user types i.e. Dealer, Branch manager, Corporate manager and clearing member, sequence number is uniquely maintained for each user. Also, for different streams available at host end, sequence number value is uniquely maintained. For download request, user must send sequence number value which was received in last message from drop copy service.

Q – Time from which login available to the system?

Details shall be clarified through a circular.

Q – Can I connect to the drop copy channel after close of market?

Details shall be clarified through a circular.

Q – Till when I can connect to the drop copy channel?

Details shall be clarified through a circular.