



Trade Gateway FIX (FIX 5.0 SP2)

System version 1.23

Interface version 23

Document version 1.9.3

25 May 2021

Revision history

Version 1.9.3 May 4, 2021

1. Added the IgnoreDynLimits field in the [NewOrderSingle\[D\]](#) and [ExecutionReport\[8\]](#) messages.
2. Updated the list of liquidity pools in Section [3.4](#) Liquidity pool identifiers.

Version 1.9.1 December 14, 2018

1. Document structure was changed.
2. The subsections of the "Interaction with gateway", "Client requests" and "Trading system reports" sections were renamed.
3. Terminology was changed.

Version 1.9.0 03 November 2017

1. The section "Instruments of trading system" has been added.
2. The sections "Login" and "Trading system gateways" have been removed.
3. Terminology changes.
4. Error codes added.

Version 1.8.4 3 April 2017

Values 0 and X of field TimeInForce corrected in messages [NewOrderSingle](#) and [ExecutionReport](#).

Version 1.8.0 22 September 2016

1. New value X of field TimeInForce added to messages [NewOrderSingle](#) and [ExecutionReport](#).
2. New values 1030, 1031, 1032, 1033 of field ExchangeSpecialInstructions added to messages [NewOrderSingle](#) and [ExecutionReport](#).

Version 1.7.0 30 March 2016

1. New field OrdType added to message [OrderCancelReject](#).
2. Functionality of automatic order canceling in case of disconnection is available in this version (please refer to section [4.2.1.4](#)).

Version 1.6.0 24 December 2015

The order sent for execution at external price is type **OrdType=o** in system reports.

Version 1.5.0 31 August 2015

1. New field OrigClOrdID added to messages OrderCancelRequest, ExecutionReport, and OrderCancelReject.
2. Field ClOrdID changed objectives in messages OrderCancelRequest and OrderCancelReject.

Version 1.4.4 11 February 2015

1. Field BusinessRejectReason in message BusinessMessageReject corrected.
2. Interaction with trade gateway corrected at rejection of negotiated counter order by counterparty (please refer to section [2.9](#)).
3. Field structure in message DontKnowTrade changed.
4. Errors 1115, 1315, 1316, 8103, 8104, 8105, 8106, and 8201 added to error codes table.

Table of Contents

| | |
|---|----|
| 1. Trading system overview | 5 |
| 1.1. Instruments of trading system | 5 |
| 1.2. Trade modes | 5 |
| 1.2.1. Main trades mode | 5 |
| 1.2.2. Negotiated trades mode | 6 |
| 1.2.3. Negotiated repo trades mode | 6 |
| 1.2.4. Closing auction of Saint-Petersburg Exchange | 6 |
| 2. Interaction with trading gateway | 7 |
| 2.1. Order submission and rejection | 7 |
| 2.2. Order placement and routing rejection | 7 |
| 2.3. Order execution | 8 |
| 2.4. Order cancellation by the liquidity pool | 8 |
| 2.5. Order cancellation by the client | 8 |
| 2.6. Order mass cancellation | 9 |
| 2.7. Negotiated order submission and cancellation | 9 |
| 2.8. Negotiated counter order placement | 10 |
| 2.9. Negotiated order rejection by the counterparty | 10 |
| 3. Protocol overview | 12 |
| 3.1. Datatypes | 12 |
| 3.2. Message header and trailer | 12 |
| 3.3. Common components | 13 |
| 3.4. Liquidity pool identifiers | 14 |
| 4. Protocol specifications | 15 |
| 4.1. Session layer | 15 |
| 4.1.1. Session initialization | 15 |
| 4.1.2. Heartbeat messages | 16 |
| 4.1.3. Send rate limit for session messages | 16 |
| 4.1.4. Message numbers | 16 |
| 4.1.5. Message resend request | 16 |
| 4.1.6. Message numbers reset by the client | 17 |
| 4.1.7. Session termination | 17 |
| 4.1.8. Message rejection | 18 |
| 4.1.9. Disconnection | 19 |
| 4.2. Application layer | 19 |
| 4.2.1. Client requests | 19 |
| 4.2.2. Trading system reports | 24 |
| A. Error codes | 31 |
| B. Revision History | 41 |

List of Tables

| | |
|---|----|
| 2. Format of message header | 12 |
| 3. Format of message trailer | 13 |
| 4. Format of component MDInc | 13 |
| 5. Format of component Parties | 13 |
| 6. Format of message Logon[A] | 15 |
| 7. Format of message HeartBeat[0] | 16 |
| 8. Format of message TestRequest[1] | 16 |
| 9. Format of message ResendRequest[2] | 17 |
| 10. Format of message SequenceReset[4] | 17 |
| 11. Format of message Logout[5] | 18 |
| 12. Format of message Reject[3] | 18 |
| 14. Format of message NewOrderSingle[D] | 20 |
| 16. Format of message OrderCancelRequest[F] | 21 |
| 18. Format of message OrderMassCancelRequest[q] | 22 |
| 19. Format of message DontKnowTrade[Q] | 23 |
| 21. Format of message ExecutionReport[8] | 25 |
| 22. Format of message OrderCancelReject[9] | 28 |
| 23. Format of message OrderMassCancelReport[r] | 29 |
| 24. Format of message BusinessMessageReject[j] | 29 |
| 25. Format of message MarketDataIncrementalRefresh[X] | 30 |

1. Trading system overview

The trading system is designed to allow users to perform operations on financial markets. The main functions include:

1. Acceptance of orders submitted to over-the-counter and exchange markets.
2. Routing and placing of orders in available liquidity pools.
3. Registration of trades and processing of information on trades at liquidity pools.
4. Transmission of anonymous market data, collected from all liquidity pools, and non-anonymous market data as well as additional and reference data.
5. Control of clearing member's risks on operations with instruments registered in the system.
6. Other functionality for access to trading.

1.1. Instruments of trading system

The Instruments are divided into **exchange** and **over-the-counter (OTC)**. Instruments and Trade modes are listed in XML-file on [FTP-server of St Petersburg Exchange](#). OTC instruments have the following attributes:

- section of `balance_instruments` and `traded_instruments` elements has value **OTC**;
- `is_otc` of `tradeMode` element has value **1**.

Table 1. Differences in the interpretation of messages fields

| Instrument | Value of <code>OrderId</code> field | Value of <code>TrdMatchId</code> field |
|------------------|-------------------------------------|--|
| Exchange | Order ID | Trade ID |
| Over-the-counter | Tender ID | Contract ID |

All instruments of trading system are available for trades.

1.2. Trade modes

1.2.1. Main trades mode

In the main trades mode anonymous orders are executed at liquidity pools.

The Main trades mode supports five order types. The order type is determined by the set of field values in the message.

1.2.1.1. Order types

1. Market order that will execute at the best available prices until it is fully filled; any remainder will be expired.
2. Day limit order that will execute at the specified or better price; the remainder, if any, is added to the order book and will be active till the end of the trading day.
3. Extended session limit order that will execute at the specified or better price; the remainder, if any, is added to the order book and will be active till the end of the extended trading session.
4. Fill or Kill (FOK) order that will execute immediately and completely, or canceled. This is an order with specified price and volume.
5. Immediate or Cancel (IOC) order that execute immediately, completely or partially, or canceled. This is an order with specified price and volume.

The set of order types available in the trading system may differ from the set of orders supported by a specific liquidity pool.



Iceberg order is not supported in the current system version.

1.2.1.2. Execution of orders

For a group of instruments listed on the trading system, the **Main pool** is determined among several liquidity pools by the highest liquidity level. The Main liquidity pool status may influence the choice of routing strategy: by default the volume that cannot be matched against active orders in the order book will be routed to that pool.

A client order, submitted to the trading system, can be executed at liquidity pools where the indicated instrument is admitted to trading. If there is only one liquidity pool matching this criterion, the entire orders volume is routed to that pool. If there are several liquidity pools like that, the order will be executed in accordance with the best execution principles.

In the course of routing, the incoming order is consecutively matched with counter orders at each price level until the order volume is filled. If all the available price levels were checked and the incoming order has not been filled completely, the remaining volume is routed to the Main liquidity pool. After the volumes to be routed are determined, they are sent to the liquidity pools.

Routing of client order depends on the order type.

A Fill Or Kill order can be filled at one liquidity pool only, where the order initiator can get the best average weighted price; in case of several equal prices the trading system give the priority to the pool providing a lower latency.

An incoming order of other types (limit, market, Immediate Or Cancel) can be routed to several liquidity pools. For each price level consecutively, starting from the best one for the order initiator, the volume to be executed is determined on each available pool. After the volumes to be routed are determined, they are sent to the appropriate price levels to the liquidity pools.

1.2.2. Negotiated trades mode

The Negotiated trades mode supports negotiated orders with fully matching parameters. Negotiated order is an order with an indication of price, volume, initiator and counterparty. The counterparty is notified that order is submitted on its clearing account (for detail on interaction with trading gateway refer to section [2](#)).

1.2.3. Negotiated repo trades mode

Price of order for repo trades is indicated in annual interest rate. In additional price field the client can indicate the price of the first-leg instrument. If client did not indicate a price, the additional price will be settled or will be indicated by the liquidity pool.

Repo trading instrument has three legs (balance instruments):

1. Change in the obligation to deliver securities under the first part of repo trade.
2. Change in the obligation to deliver currency under the first part of repo trade.
3. Change in the obligation to deliver securities under the second part of repo trade.

Currency obligation under the second part of repo trade is changed using the price setting tool for repo trading instrument.

1.2.4. Closing auction of Saint-Petersburg Exchange

The closing auction of Saint-Petersburg Exchange supports only market order with time in force - closing auction. Trades are executed at the official closing price of the instrument of the liquidity pool, on which the security was listed. Orders, leading to cross trade, will be automatically canceled by the liquidity pool.

Trading in the closing auction:

1. During the trading day, clients submit market orders in the trading system.
2. Submission of orders is stopped according to the approved schedule of trading and orders become unavailable to cancel.
3. Closing auction is held — counter orders, sorted by ascending of the time of submission, are matched together at instrument's closing price at Main liquidity pool.
4. Remainders of orders and unfilled orders are canceled.

2. Interaction with trading gateway

2.1. Order submission and rejection

To submit an order, the client should send the `NewOrderSingle[D]` message (NOS) to the trading system gateway. The client specifies the `ClOrdID[11]` identifier, unique for each login during the trading session.

After accepting the order, the trading system will return `ExecutionReport[8]` (ER) to the client with `OrderID[37]`, and `OrdStatus[39]=0` and `ExecType[150]=0`. If the trading system rejects the order (due to invalid values or closed liquidity pool), no order identifier will be assigned and the client will receive `ExecutionReport[8]` with values `OrdStatus[39]=8` and `ExecType[150]=8`, while `OrdRejReason[103]` may explain reasons for rejection.

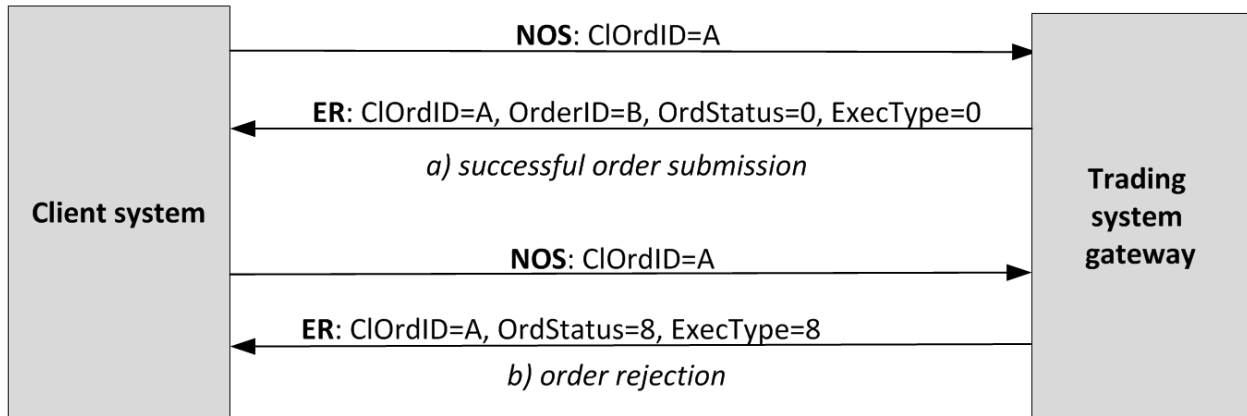


Figure 1. Adding and rejecting order

2.2. Order placement and routing rejection

To ensure best execution, the order volume is split according to the order books of the liquidity pools and splitted results are routed to liquidity pools. When a liquidity pool confirms order placement, the trading system sends report `ExecutionReport[8]` to the client containing order identifier `SecondaryOrderID` and values `OrdStatus[39]=0` and `ExecType[150]=0`.

If a liquidity pool rejects an order, the client will receive two `ExecutionReport[8]`s. One reports an unsuccessful routing (`OrdStatus[39]=8` and `ExecType[150]=8`) and another reports a partial cancel of the rejected volume (`OrdStatus[39]=4` and `ExecType[150]=4`).

A Fill Or Kill order can be routed to one liquidity pool only. If the liquidity pool can fully fill the order, the client will receive all reports in the usual way. If the order cannot be executed, the liquidity pool will reject it and the client will be notified of, first, order placement, then order rejection, and, thirdly, order cancellation.

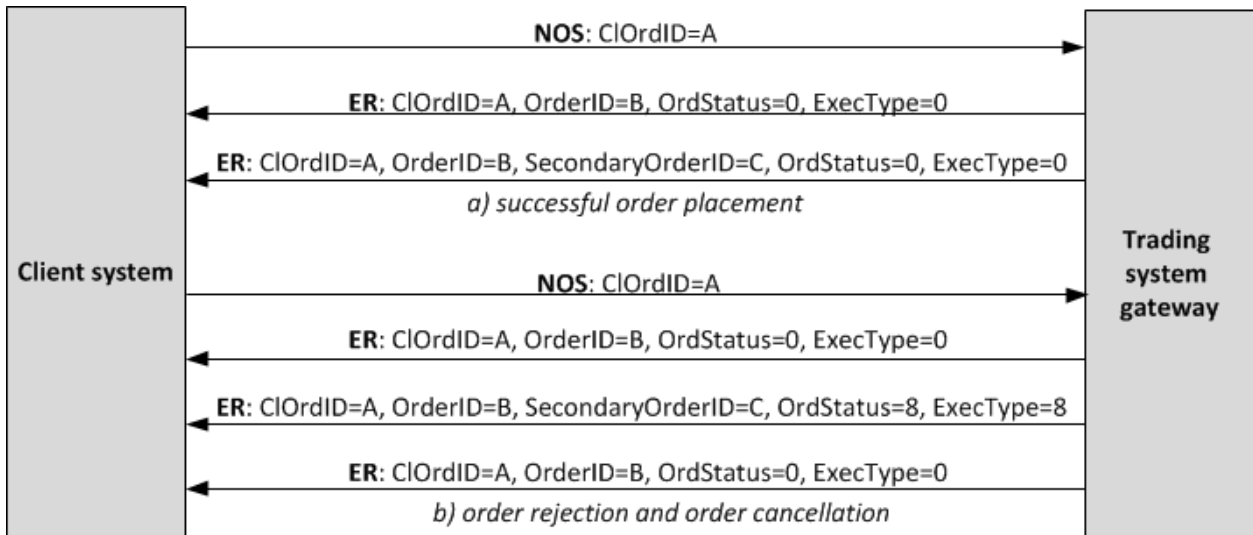


Figure 2. Order placement or rejection

2.3. Order execution

After a liquidity pool accepts an order, the client will be sent reports (`ExecType[150]=F`) about deals in liquidity pools and then on order execution in trading system. All such reports include the trade ID `TrdMatchID[880]`.

The graph below shows the submission and fully execution by one side of the trade.

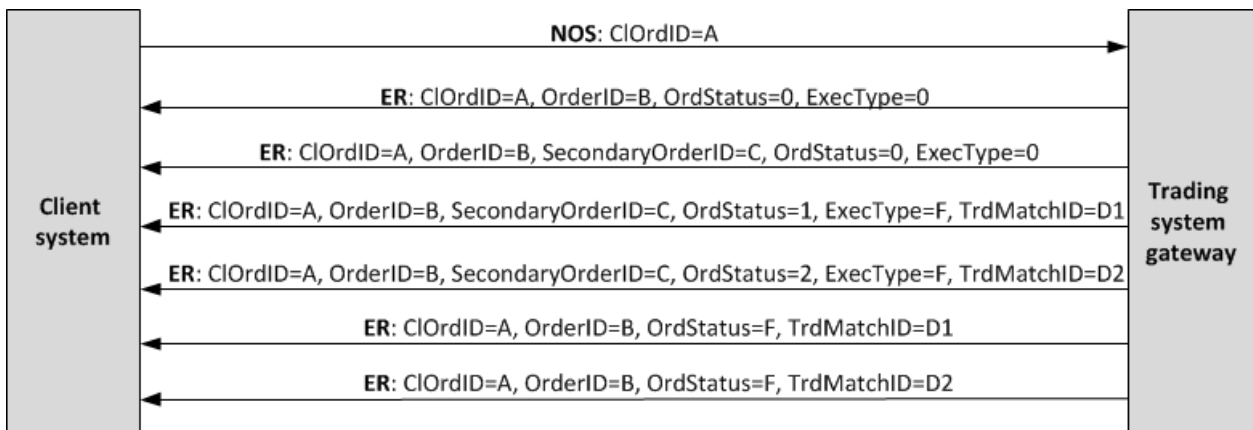


Figure 3. Submission of order and receipt of reports

2.4. Order cancellation by the liquidity pool

In some instances, the liquidity pool will cancel an order remainder, e.g. the unfilled portion of a market or IOC order, or to prevent a cross trade. So after reports on order acceptance routing and trade reports, the client should also expect `ExecutionReport[8]` (`OrdStatus[39]=4` and `ExecType[150]=4`) reports on partial or full cancellation of the order.

Moreover, to ensure best execution, the trading system may cancel an order at a liquidity pool and place it to another. In this case, the client will receive a cancellation report and a new placement report.

2.5. Order cancellation by the client



After an order has been successfully routed, a single routed volume cannot be canceled. Only the whole order can be canceled.

The client can cancel the unfilled remainder of an order. The client shall send `OrderCancelRequest[F]` (`OCRq`) to the trading system gateway and specify the identifier and certain parameters of the order.

After the order is successfully canceled, the client will receive `ExecutionReport(OrdStatus[39]=4` and `ExecType[150]=4`) reports on routed volumes cancellation and then report on order cancellation.

If an order remainder cannot be canceled or the sender has no permissions, the request will be rejected with `OrderCancelReject[9]` (OCRj).

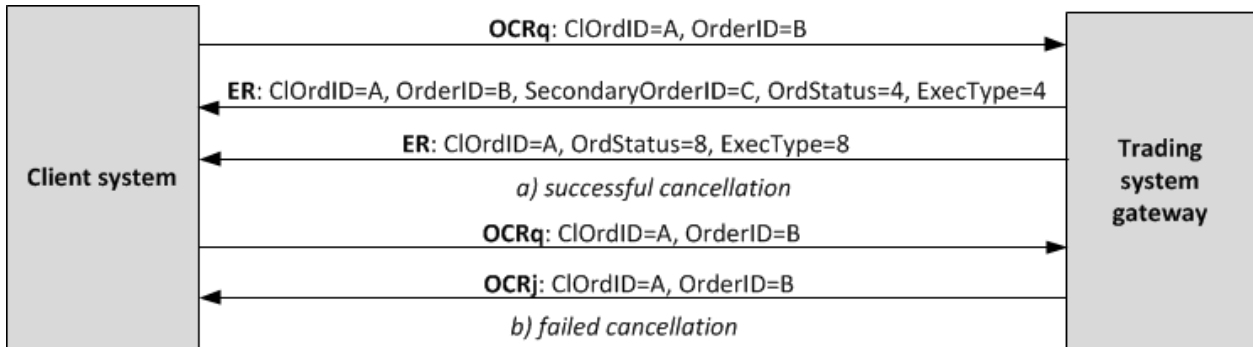


Figure 4. Order cancellation

2.6. Order mass cancellation

The client may request to cancel several orders, based on some criteria, for instance the orders referring to a certain instrument submitted from the particular login. The client shall send `OrderMassCancelRequest[q]` (MCRq) to the trading system gateway and specify the cancellation mode and, if necessary, certain parameters of orders.

The trading system receives the request and selects orders to cancel by the specified criteria, and then generates cancellation request and routes them to liquidity pools. If the orders are canceled successfully, the client will receive reports on orders cancellation `ExecutionReport(OrdStatus[39]=4` и `ExecType[150]=4`) and the report on execution of request `OrderMassCancelReport[r]` (MCRt) specifying the number of canceled orders. If no order to cancel is found, the gateway will only return `OrderMassCancelReport[r]`.

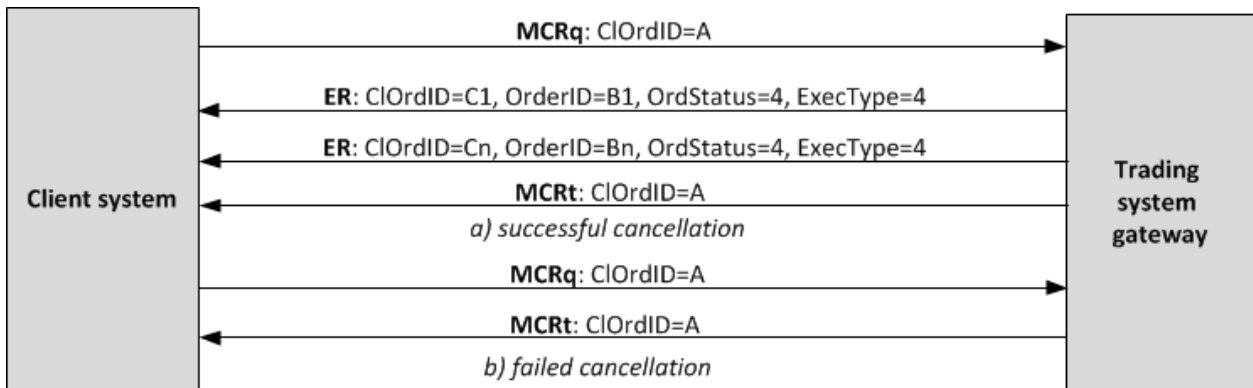


Figure 5. Orders mass cancellation

2.7. Negotiated order submission and cancellation

To submit a negotiated order, the client should send `NewOrderSingle[D]` (NOS) to the trading system gateway with unique `ClOrdID[11]` assigned.

After accepting the negotiated order, the trading system will return `ExecutionReport[8]` (ER) to the client-sender with `OrderID[37]` and values `OrdStatus[39]=0` and `ExecType[150]=0`, and the client-receiver is sent `MarketDataIncrementalRefresh[X]` (MD) with identifier of update type `MDUpdateAction[279]=0`. If the trading system rejects the order (due to invalid values or closed market), no order identifier will be assigned and the client-sender will receive `ExecutionReport[8]` with values `OrdStatus[39]=8` and `ExecType[150]=8`, while field `OrdRejReason[103]` may explain reasons for rejection.

After the trading system and the liquidity pool accept the negotiated order, the client-sender may cancel it before the counterparty submits the counter order. To cancel the negotiated order, the client should send `OrderCancelRe-`

quest [F] (OCRq) to the gateway specifying the identifier and certain parameters of the order. If the negotiated order is successfully canceled, the sender will receive ExecutionReport [8] (OrdStatus [39]=4 and ExecType [150]=4) and the counterparty will get MarketDataIncrementalRefresh [X] with MDUpdateAction [279]=2.

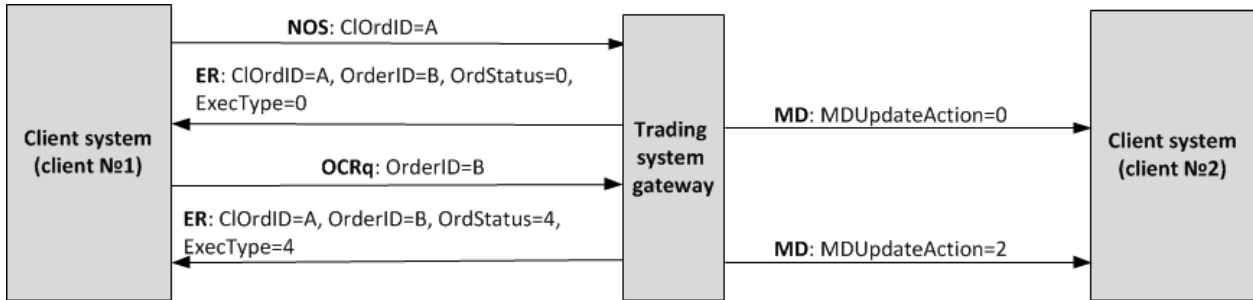


Figure 6. Negotiated order submission and cancellation

2.8. Negotiated counter order placement

To take the offer, the counterparty shall send the counter order with the same quantity of the instrument at the same price and the opposite side.

In case of mismatch in price, amount, and instrument of the order, the counter order will be placed as a new one and expect matching.

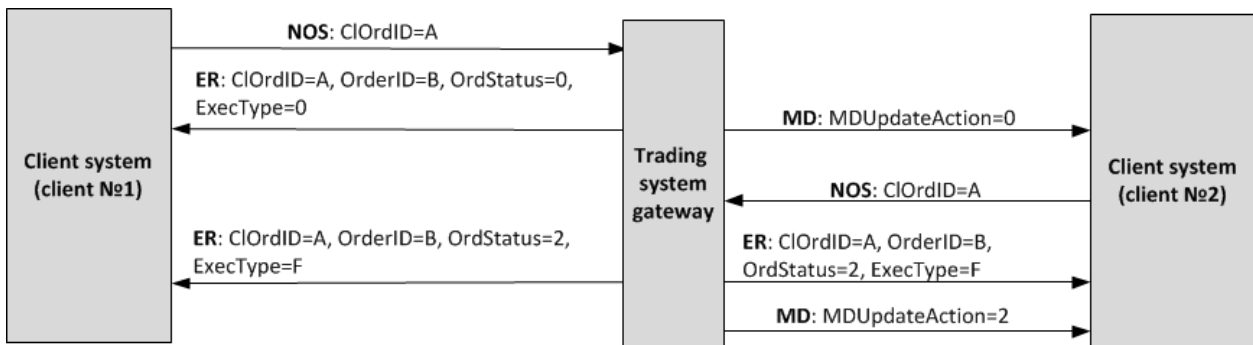


Figure 7. Successful submission of negotiated counter order placement

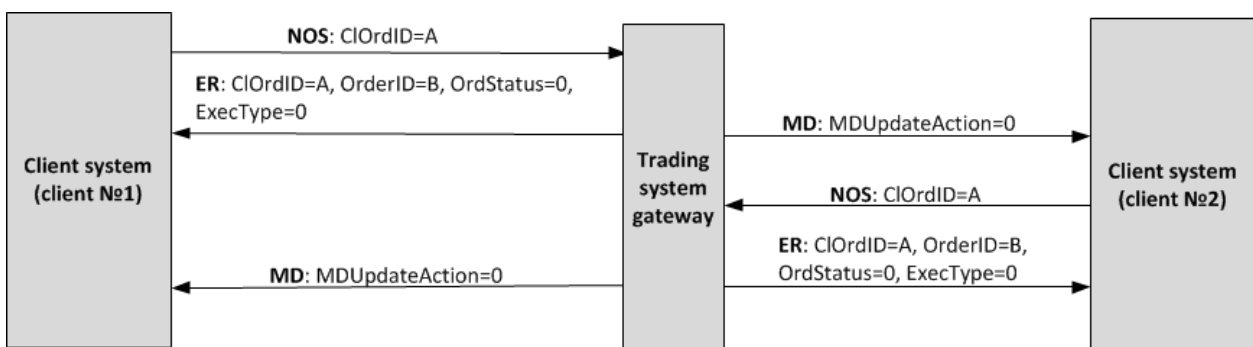


Figure 8. Failed submission of negotiated counter order placement

2.9. Negotiated order rejection by the counterparty

The counterparty can reject the negotiated order. The client should send DontKnowTrade [Q] (DKT) to the trading system gateway and specify the identifier and certain parameters of the order.

After successful rejection, the client will receive the rejection response DontKnowTrade [Q] (it will differ from the request by OrdStatus [39]=4) and MarketDataIncrementalRefresh [X] (MDUpdateAction [279]=2), while the order initiator will be sent the cancellation report ExecutionReport (OrdStatus [39]=4 and ExecType [150]=4).

Interaction with trading gateway

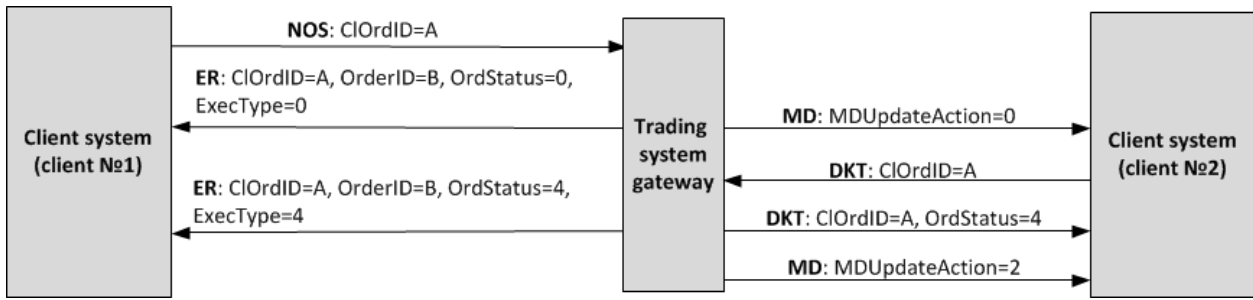


Figure 9. Negotiated order rejection

3. Protocol overview

3.1. Datatypes

The message type defined in field `MsgType[35]` of the header is specified in brackets after the message name.

Fields:

- R [required];
- N [nonrequired];
- C [conditionally required].

Datatypes

`Bool`, logical field containing one of two values: `Y` (yes) and `N` (no).

`Char`, single-character datatype. Valid values are ASCII characters: letters, numbers, and punctuation marks. Null and Start of Heading characters are invalid.

`Int` - integer.

`Length` - positive integer to indicate length in bytes.

`MultipleChar` - string of single-character values separated by spaces. For example: `18=0 z`.

`NumInGroup` - integer to indicate number of entries in a component.

`Price` - float to indicate price with point separator.

`Qty` - integer to indicate number of securities lots.

`SeqNum` - integer to indicate message sequence number.

`String` - string datatype. String can be in any encoding. Null and Start of Heading characters are invalid.

`Timestamp` - string datatype to indicate time and date of the World Time (UTC) within the accuracy of milliseconds in format `YYYYMMDD-HH:MM:SS.sss`.

3.2. Message header and trailer

Each message begins with the header and ends with the trailer.

The first three fields have fixed positions in the header, namely: `BeginString[8]=FIXT.1.1` always comes first, followed by field `BodyLength[9]` and then `MsgType[35]`. The value of `BodyLength[9]` is the message length in bytes, which is calculated starting from the tag following `BodyLength[9]` and ending with the separator before `Checksum[10]`.

Table 2. Format of message header

| Tag | Field | ✓ | Type | Description |
|-----|-----------------|---|--------------|--|
| 8 | BeginString | R | String | First fieldFIXT.1.1 |
| 9 | BodyLength | R | Length | Length of message body |
| 35 | MessageType | R | String | Message type |
| 49 | SenderCompId | R | String | Sender ID |
| 56 | TargetCompId | R | String | Receiver ID |
| 34 | MsgSeqNum | R | SeqNum | Sequence number of message |
| 43 | PosDupFlag | N | Boolean | Resending message indicator |
| 52 | SendingTime | R | UTCTimestamp | Time of message transmission |
| 122 | OrigSendingTime | N | UTCTimestamp | Time of message resending when responding to Re-sendRequest[2] |

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------|---|
| 369 | LastMsgSeqNumProcessed | N | SeqNum | Sequence number of the last processed message. Specified by the trading system gateway only |

The message trailer consists of `Checksum[10]` including a three-byte simple check sum.

Table 3. Format of message trailer

| Tag | Field | ✓ | Type | Features |
|-----|----------|---|--------|-----------------------------|
| 10 | Checksum | R | String | Message check sum (3 bytes) |

3.3. Common components

Table 4. Format of component `MDInc`

| Tag | Field | ✓ | Type | Description |
|-----|------------------|---|-------------|---|
| 268 | NoMDEntries | R | NumInGroup | Number of entries in repeated group |
| 48 | SecurityId | N | String | Numeric ID of trading instrument |
| 22 | SecurityIdSource | N | String | Liquidity pool ID of order placement. For values please refer to section 3.4 |
| 279 | MdUpdateAction | R | Char | Type of update. Possible values: <ul style="list-style-type: none"> • 0 (new order); • 2 (execution, cancellation or rejection of order) |
| 278 | MdEntryId | R | String | Order ID assigned by trading system |
| 269 | MdEntryType | R | Char | Side. Possible values: <ul style="list-style-type: none"> • 0 (buy); • 1 (sell) |
| 270 | MdEntryPx | N | Price | Price |
| 271 | MdEntrySize | N | Qty | Amount |
| 272 | MdEntryDate | R | UTCDateOnly | Date of update |
| 273 | MdEntryTime | R | UTCTimeOnly | Time of update |

Table 5. Format of component `Parties`

| Tag | Field | ✓ | Type | Description |
|-----|---------------|---|------------|--|
| 453 | NoPartyIDs | R | NumInGroup | Number of entries in repeated group |
| 448 | PartyId | R | String | Subject ID corresponding to specified PartyRole |
| 447 | PartyIdSource | R | Char | Identifies class or source of the PartyID. Possible values: <ul style="list-style-type: none"> • D |

| Tag | Field | ✓ | Type | Description |
|-----|-----------|---|------|--|
| 452 | PartyRole | R | Int | Role of the subject specified in PartyID. Possible values: <ul style="list-style-type: none"> • 1 (trading member); • 3 (client code); • 13 (initiator of negotiated order); • 17 (counterparty for negotiated order) |

3.4. Liquidity pool identifiers

Liquidity pools identifiers may be the value of fields `ExDestination[100]`, `LastMkt[30]` and `ExchangeSpecialInstructions[1139]`.

0 (DEFAULT) — liquidity pool is defined by the trading system.

1001 (TRADSYS) — all available liquidity pools

1000 — liquidity pool of Saint-Petersburg Exchange

1010 — liquidity pool of Moscow Exchange

1015 — execution at United States liquidity pools

1017 — liquidity pool of LSE

1021 — liquidity pool of german equities

4. Protocol specifications

4.1. Session layer

The session layer is compliant with FIX Session Protocol 1.1.

A FIX session is established over TCP-connection between a client gateway and the trading system gateway. Session participants are identified by fields `SenderCompID`[49] and `TargetCompID`[56].

The ID of the trading system gateway is `ECN_EQR` and that of a client is the user name.

4.1.1. Session initialization

The `Logon`[A] initiates a FIX session or confirms its start. After establishing a TCP connection, the session initiator (client) sends this message and expects `Logon`[A] in response. The `ResetSeqNumFlag`[141], `Password`[554] are filled only by the client, `NextExpectedMsgSeqNum`[789] is filled only by the trading system.

A receipt of a correct `Logon`[A] shall always result in sending response message `Logon`[A], even if `MsgSeqNum`[34] is higher than expected. Any error in `Logon`[A] shall cause a disconnection, and the number of the next expected message will not be changed.

Table 6. Format of message `Logon`[A]

| Tag | Field | ✓ | Type | Description |
|------|------------------------|---|---------|--|
| | Header | R | | Standard header |
| 98 | EncryptMethod | R | Int | Method of encryption. Possible values: <ul style="list-style-type: none"> 0 (Encryption not supported) |
| 108 | HeartBtInt | R | Int | Timeout. Value in seconds. Recommended value: from 20 to 30 |
| 95 | RawDataLength | C | Length | The field must be present if there is <code>RawData</code> [96]. Possible values: <ul style="list-style-type: none"> 1 |
| 96 | RawData | N | data | Automatic cancellation of all orders submitted by this login at logout/disconnection. Possible values: <ul style="list-style-type: none"> 0 (do not activate automatic); 1 (activate automatic) |
| 141 | ResetSeqNumFlag | N | Boolean | Reset of sequence numbers |
| 789 | NextExpectedMsgSeqNum | N | SeqNum | Number of next messages to be sent by the client. To be filled by the trading system |
| 554 | Password | N | String | Login password |
| 1137 | DefaultAppVerId | R | String | Protocol version. Possible values: <ul style="list-style-type: none"> 9 (FIX50SP2) |

4.1.2. Heartbeat messages

Client and trading system exchange messages `Heartbeat[0]` to monitor the connection status. A heartbeat is to be sent by a party if it passed no messages (of the session or application layer) within the heartbeat interval. A client specifies the timeout value `HeartBtInt[108]` in the message `Logon[A]`; the recommended value is from 20 to 30 seconds.

After the absence of messages during an interval exceeding `HeartBtInt[108]`, a party should send `TestRequest[1]` with the `TestReqID[112]` identifier. In answer the counterparty should send `Heartbeat[0]` containing the same identifier. If no response within the heartbeat interval, the system disconnects after sending message `Logout[5]` to a client. A client is expected to act the same.

If a client prefers not to send or receive heartbeats during this FIX session, 0 should be specified in `HeartBtInt[108]`.

Table 7. Format of message `HeartBeat[0]`

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------|--|
| | Header | R | | Standard header |
| 112 | TestReqId | C | String | Request ID of TestRequest, to which this message is a response |

Table 8. Format of message `TestRequest[1]`

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------|--|
| | Header | R | | Standard header |
| 112 | TestReqId | R | String | Request ID Maximum length is 32 characters. Valid characters are letters and numbers |

4.1.3. Send rate limit for session messages

The rate at which a client can send session level messages into the system is limited. When client's send rate exceeds the limit, the system terminates the user session.

4.1.4. Message numbers

All messages exchanged by the parties within a FIX session have a sequence number. The number is specified in the field `MsgSeqNum[34]` in the header of each message. The number of each subsequent message of a FIX session should be incremented, except for the cases of forced increase of the message number by request `SequenceReset[4]`.

As reference information for a client, the number of the last message processed by the trading system is indicated in the field `LastMsgSeqNumProcessed[369]`.

When receiving a message with the number higher than expected, a client should send `ResendRequest[2]`.

When the trading system receives a messages with the numbers lower than expected, a client will be sent `Logout[5]` with the value `SessionStatus[1409]=1` followed by TCP disconnection.

4.1.5. Message resend request

To request the messages previously sent by the trading system, a client can use `ResendRequest[2]`, in particular for the purpose of restoring missing messages. When receiving a message with the number higher than expected, a client should also send `ResendRequest[2]`.

A client may request resending of all messages, sent during the current and previous trading days. If a client has intentionally reset message numbering (`ResetSeqNumFlag[141]=Y` in the message `Logon[A]`), a request for resending messages, sent prior to the reset, is not possible.

The fields `BeginSeqNo[7]` and `EndSeqNo[16]` set the range of requested messages. If a client uses `BeginSeqNo[7]=0` and `EndSeqNo[16]=0`, the system will resend all messages starting from the lowest number available. If a client specifies 0 only for `EndSeqNo[16]`, the system will resend all messages of current trading session starting from `BeginSeqNo[7]`. All possible cases are as follows:

1. BeginSeqNo= n , EndSeqNo= m — request for messages from n to m ,
2. BeginSeqNo=0, EndSeqNo= n — request for messages from the lowest number available to n ,
3. BeginSeqNo= n , EndSeqNo=0 — request for messages from n to the highest number available,
4. BeginSeqNo=0, EndSeqNo=0 — request for all available messages.

Number range for requested messages is not limitless (for more details please refer to *Network Connectivity*). When a client requires more messages, a client should send several consecutive requests. If the gateway has not finished sending messages on previous request, new requests for messages will be rejected.

Table 9. Format of message ResendRequest [2]

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------|-----------------------------------|
| | Header | R | | Standard header |
| 7 | BeginSeqNo | R | SeqNum | Number of first requested message |
| 16 | EndSeqNo | R | SeqNum | Number of last requested message |

In response to ResendRequest [2], the system will return the requested messages or will modify MsgSeqNum [34] by the message SequenceReset [4]. The value PossDupFlag [43]=Y is a flag of resent messages.

After receiving ResendRequest [2], the trading system will resend messages of the application layer only and will never resend session messages. Therefore, in response to message resend request a client should expect, among others, SequenceReset [4] with GapFillFlag [123]=Y and the number of the next expected message in NewSeqNo [36].

If a client wants to increase the message number expected from the system, a client should send SequenceReset [4] with GapFillFlag [123]=N and the new expected number in the field NewSeqNo [36].

During resending, the trading system may also transmit new trading messages, so a client should also expect messages with a number exceeding the requested range. To ensure data completeness, a client is not recommended to ignore such messages with larger numbers.

Table 10. Format of message SequenceReset [4]

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|---------|---|
| | Header | R | | Standard header |
| 36 | NewSeqNo | R | SeqNum | New sequence number |
| 123 | GapFillFlag | N | Boolean | Indicator of gap fill. Possible values: <ul style="list-style-type: none"> • N (mode Reset ignoring field MsgSeqNum; specified by the client); • Y (mode GapFill using field MsgSeqNum; specified by the server) |

4.1.6. Message numbers reset by the client

The client can reset sequence numbers by the value ResetSeqNumFlag [141]=Y in the Logon [A] message. This functionality may be useful to avoid procedures for requesting and restoring missing messages. It is not recommended to use this feature during the trading session, because trading messages already sent will not be available for resend request.

In response to a client Logon [A] with ResetSeqNumFlag [141]=Y the trading system will send Logon [A] with ResetSeqNumFlag [141]=Y, MsgSeqNum [34]=1, and NextExpectedMsgSeqNum [789]=2. The next expected message number is 2.

4.1.7. Session termination

Logout [5] initiates or confirms the session termination and shall be sent after a long-term absence of messages (please refer to section 4.1.2) or after receiving a message with number lower than expected.

The reason for rejection is specified in the tag `SessionStatus` [1409]. The field `Text` [58] may contain report on the session termination reasons.

Table 11. Format of message `Logout` [5]

| Tag | Field | ✓ | Type | Description |
|------|----------------------------|---|--------|---|
| | Header | R | | Standard header |
| 1409 | <code>SessionStatus</code> | N | Int | Numeric code of the reason. To be filled by the trading system only. Possible values: <ul style="list-style-type: none"> • 5 (invalid login or password); • 5000 (violation of message exchange protocol); • 5002 (client not active); • 5003 (server stopped); • 5200 (login is already in active session) |
| 58 | <code>Text</code> | N | String | Report on session termination reason |

4.1.8. Message rejection

The message `Reject` [3] is sent in response to any invalid message (incorrectly generated or transmitted) from the other party. The reasons for rejection may be the absence of required fields, invalid message type or length, and invalid data type, etc. All session layer messages with invalid value of any field are also rejected by the message `Reject`.

The system specifies the rejected message number in the field `RefSeqNum` [45]. The value `RefSeqNum` [45]=0 means that the field `MsgSeqNum` [34] is missing in the rejected message. If the system detects an invalid value, the tag will be indicated in `RefTagID` [371]. The field `SessionRejectReason` [373] may contain the rejection reason code and `Text` [58] may have a textual description of error.

Table 12. Format of message `Reject` [3]

| Tag | Field | ✓ | Type | Description |
|-----|----------------------------------|---|--------|--|
| | Header | R | | Standard header |
| 45 | <code>RefSeqNum</code> | R | SeqNum | Number of rejected message |
| 371 | <code>RefTagId</code> | N | Int | Tag which caused the error |
| 372 | <code>RefMsgType</code> | N | String | Type of rejected message |
| 373 | <code>SessionRejectReason</code> | N | Int | Reason for rejection. Possible values: <ul style="list-style-type: none"> • 0 (invalid tag number); • 1 (required tag missing); • 2 (invalid tag in the message); • 4 (tag with no value); • 5 (invalid value); • 6 (invalid datatype); • 11 (incorrect message type); • 13 (tag repeated in message); • 14 (tag <code>Checksum</code>[10] misplaced); • 15 (tag from the group misplaced); • 16 (invalid number of group entries) |

| Tag | Field | ✓ | Type | Description |
|-----|-------|---|--------|--------------|
| 58 | Text | N | String | Error report |

4.1.9. Disconnection

The TCP connection will be dropped if the server receives a message with an error in one of the first three fields (`BeginString[8]`, `BodyLength[9]`, and `MsgType[35]`) or the `Logon[A]` message of invalid format or containing invalid values.

4.2. Application layer

4.2.1. Client requests

4.2.1.1. Order submission

To submit a new order to the trading system, the client should send the message `NewOrderSingle[D]` with required fields.

Table 13. Required fields depending on the order types

| Order type | Required fields | |
|---|--|---|
| Market | <code>ClOrdID[11]</code> | <code>OrdType[40]=1</code> , <code>TimeInForce[59]=3</code> |
| Market order at closing auction | <code>ExDestination[100]</code> | <code>OrdType[40]=1</code> , <code>TimeInForce[59]=7</code> |
| Limit order at closing auction | <code>SecurityID[48]</code> <code>Side[54]</code> | <code>OrdType[40]=2</code> , <code>TimeInForce[59]=7</code> , <code>Price[44]</code> |
| Day active limit | <code>OrderQty[38]</code> <code>Account[1]</code> | <code>OrdType[40]=2</code> , <code>TimeInForce[59]=0</code> , <code>Price[44]</code> |
| Limit order in extended trading session | <code>PartyID[448]</code> , <code>PartyRole[452]=1</code> <code>PartyID[448]</code> , <code>PartyRole[452]=3</code> | <code>OrdType[40]=2</code> , <code>TimeInForce[59]=X</code> , <code>Price[44]</code> |
| Fill or Kill (FOK) | <code>ExchangeSpecialInstructions[1139]</code> | <code>OrdType[40]=2</code> , <code>TimeInForce[59]=4</code> , <code>Price[44]</code> |
| Immediate or Cancel (IOC) | | <code>OrdType[40]=2</code> , <code>TimeInForce[59]=3</code> , <code>Price[44]</code> |
| Negotiated | | <code>OrdType[40]=n</code> , <code>TimeInForce[59]=0</code> , <code>Price[44]</code> <code>PartyID[448]</code> , <code>PartyRole[452]=13</code> <code>PartyID[448]</code> , <code>PartyRole[452]=17</code> |

The trading system requires the client order identifier `ClOrdID[11]` to be unique during the trading session for each client gateway. It is not recommended to reuse `ClOrdID[11]` of rejected orders.

Special identifier `RefOrderID[1080]` must be assigned for a negotiated order. The counter order must contain the same ID, otherwise the orders will not match.

The Closing Auction in the Foreign Securities Market only allows market (`OrdType[40]=1`) and the Closing Auction in the Russian Securities Market allows market (`OrdType[40]=1`) and limit (`OrdType[40]=2`) orders.

The client can provide an order with a comment in the field `Text[58]` (23 bytes in UTF-8).

At the end of the trading session or extended trading session all active orders (`TimeInForce[59]=0` or `TimeInForce[59]=X`) will be canceled and the client will receive `ExecutionReport[8]` with the indicator `ExecRestatementReason[378]=EXPIRED`.

Protocol specifications

After processing a client order, the trading system will either reject it with the message `BusinessMessageReject [j]` or confirm with the message `ExecutionReport [8]` with status `ExecType [150]=0` and `OrdStatus [39]=0`.

Table 14. Format of message `NewOrderSingle [D]`

| Tag | Field | ✓ | Type | Description |
|------|------------------------|---|--------------|---|
| | Header | R | | Standard header |
| 11 | ClOrdId | R | String | Client order ID. Maximum length is 20 characters. Valid characters are Latin letters and numbers |
| 60 | TransactTime | R | UTCTimestamp | Time of order submission by user |
| 100 | ExDestination | R | Exchange | Liquidity pool ID where the order is sent to. For values please refer to section 3.4 |
| 48 | SecurityId | R | String | Numeric ID of trading instrument |
| 9303 | RoutingInstruction | N | String | Routing algorithm |
| 54 | Side | R | Char | Side of order. Possible values: <ul style="list-style-type: none"> • 1 (buy); • 2 (sell) |
| 40 | OrdType | R | Char | Type of order. Possible values: <ul style="list-style-type: none"> • 1 (market); • 2 (limit); • n (negotiated) |
| 59 | TimeInForce | R | Char | Period the order remains in effect. Possible values: <ul style="list-style-type: none"> • 0 (during the trading session); • 2 (opening auction); • 3 (immediate or cancel, IOC); • 4 (fill or kill, FOK); • 7 (closing auction); • x (during the extended trading session) |
| 44 | Price | C | Price | Price. For repo trading: annual interest yield, the value to be indicated in percentage |
| 38 | OrderQty | R | Qty | Amount of order in lots |
| 1138 | DisplayQty | N | Qty | Disclosed (visible) part of the order quantity. Required for icebergs orders: <ul style="list-style-type: none"> • $0 < \text{DisplayQty} < \text{OrderQty}$ (iceberg); • <code>DisplayQty</code> not defined (disclosed orders) |
| 1084 | DisplayMethod | N | Char | Required for icebergs orders. Possible value: <ul style="list-style-type: none"> • 1 (iceberg) |

| Tag | Field | ✓ | Type | Description |
|-------|-----------------------------------|---|--------|---|
| 1 | Account | R | String | Clearing account of the client submitting order |
| | Component Parties | R | | |
| 58 | Text | N | String | Comment. Maximum length is 23 characters |
| 1139 | ExchangeSpecialInstructions | N | String | The main liquidity pool identifier. For values please refer to Section 3.4 |
| 1080 | RefOrderId | N | String | Identifier for matching negotiated orders |
| 10104 | Price1 | N | Price | Additional price. For a repo the trade price can be specified |
| 20113 | IgnoreDynLimits | N | Char | Ignore dynamic limits. Possible values: <ul style="list-style-type: none"> • N (no); • Y (yes) |

4.2.1.2. Order cancellation

After the order has been routed to the liquidity pools, the client can cancel the order quantity that is still not filled. The client should send `OrderCancelRequest [F]` with required fields.

Table 15. Required fields depending on the cancel modes

| Cancel mode | Required fields | |
|--|---|------------------------------|
| Canceling of order by request of order originator login | OrigClOrdId ExDestination[100] | ClOrdID[11] (or OrderID[37]) |
| Canceling of order by request of login other than the order originator | SecurityId[48] Side[54] Account[1] Parties | OrderID[37] |

After processing the request, the trading system either rejects it with the message `BusinessMessageReject [j]` or confirms the cancellation with `ExecutionReport [8]`.

Table 16. Format of message `OrderCancelRequest [F]`

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------------|---|
| | Header | R | | Standard header |
| 41 | OrigClOrdId | C | String | Client ID of order to cancel. Maximum length is 20 characters. Valid characters are Latin letters and numbers |
| 11 | ClOrdId | R | String | Client ID of the command to cancel order. Maximum length is 20 characters. Valid characters are Latin letters and numbers |
| 37 | OrderId | C | String | Order ID assigned by the trading system |
| 60 | TransactTime | R | UTCTimestamp | Date and time of request generation |
| 100 | ExDestination | R | Exchange | Liquidity pool ID specified in the order. For values please refer to section 3.4 |

Protocol specifications

| Tag | Field | ✓ | Type | Description |
|-----|-----------------------------------|---|--------|---|
| 48 | SecurityId | R | String | Numeric ID of the trading instrument |
| 54 | Side | R | Char | Side of order. Possible values: <ul style="list-style-type: none"> • 1 (buy); • 2 (sell) |
| 1 | Account | R | String | Clearing account |
| | Component Parties | R | | |

4.2.1.3. Order mass cancellation

Mass cancellation of orders is available in several modes, that should be set in `OrderMassCancelRequest[q]` by the value of `MassCancelRequestType[530]`.

Table 17. Order mass cancellation modes

| Mode | Required fields |
|---|--|
| Cancellation of the orders submitted by the requesting login | <code>MassCancelRequestType[530]=7</code> |
| Cancellation of all orders of the instrument submitted by the requesting login | <code>MassCancelRequestType[530]=1, SecurityID[48]</code> |
| Cancellation of all orders of the specified instrument and the clearing account | <code>MassCancelRequestType[530]=1, SecurityID[48], Account[1]</code> |
| Cancellation of all orders of the specified instrument and the client code | <code>MassCancelRequestType[530]=1, SecurityID[48], component Parties</code> |

When setting the mode for cancellation of orders submitted by requesting login (`MassCancelRequestType[530]=7`), the client should not fill the fields `SecurityID[48]` and `ExDestination[100]`.

After processing the request, the trading system confirms cancellation of each cancelled order with a separate report `ExecutionReport[8]` with statuses `ExecType[150]=4` and `OrdStatus[39]=4`, and then sends `OrderMassCancelReport[r]`.

Table 18. Format of message `OrderMassCancelRequest[q]`

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------------|---|
| | Header | R | | Standard header |
| 11 | ClOrdId | R | String | Client ID of the command to cancel order. Maximum length is 20 characters. Valid characters are Latin letters and numbers |
| 530 | MassCancelRequestType | R | Char | Type of cancellation. Possible values: <ul style="list-style-type: none"> • 1 (for the instrument); • 7 (all orders) |
| 60 | TransactTime | R | UTCTimestamp | Date and time of request generation |

| Tag | Field | ✓ | Type | Description |
|-----|-----------------------------------|---|----------|--|
| 100 | ExDestination | N | Exchange | Liquidity pool ID specified in the order. For values please refer to section 3.4 |
| 48 | SecurityId | C | String | Numeric ID of trading instrument. Required when <code>MassCancelRequestType[530]=1</code> |
| 1 | Account | N | String | Clearing account |
| | Component Parties | N | | |

4.2.1.4. Automatic cancellation

All active orders submitted by the login can be canceled at FIX session termination:

1. TCP connection is dropped by a client gateway.
2. There is no answer received to `TestRequest[0]` during the heartbeat interval.
3. The `Logout[5]` is received.

By default, the automatic cancellation is disabled. The option should be enabled during the session initialization by the values `RawDataLength[95]=1` and `RawData[96]=1` in the `Logon[A]` message. All client orders, including those submitted in previous sessions, will be canceled. The cancellation will be reported to the logins which have access to non-anonymous market data. The `ExecutionReport[8]` will have the indication `Text[58]=Cancel` on disconnect.

4.2.1.5. Negotiated order rejection

The counterparty can reject a negotiated order. The counterparty should send `DontKnowTrade[Q]` to the trading system gateway with the order identifier `OrderID[11]`, the counterparties in the `Parties` component, and, if needed, the match identifier `RefOrderID[1080]`.

After processing the request, the trading system either rejects it with the message `BusinessMessageReject[j]` or confirms the cancellation with the `DontKnowTrade[Q]` report, which differs from the request by the indicator `OrdStatus[39]=4`, and the report `MarketDataIncrementalRefresh[X]`.

Table 19. Format of message `DontKnowTrade[Q]`

| Tag | Field | ✓ | Type | Description |
|------|-----------------------------------|---|--------|--|
| | Header | R | | Standard header |
| 37 | OrderId | R | String | Order ID assigned by the trading system |
| 48 | SecurityId | R | String | Numeric ID of the trading instrument |
| 54 | Side | R | Char | Side. Possible values: <ul style="list-style-type: none"> • 1 (buy); • 2 (sell) |
| 40 | OrdType | R | Char | Type of order. Possible values: <ul style="list-style-type: none"> • 1 (market); • 2 (limit); • n (negotiated); • o (expanded liquidity pool) |
| | Component Parties | R | | |
| 1080 | RefOrderId | N | String | Identifier for matching negotiated orders |

| Tag | Field | ✓ | Type | Description |
|-----|-----------|---|------|--|
| 39 | OrdStatus | R | Char | Status of order. Possible values: <ul style="list-style-type: none"> • 4 (canceled); • 8 (rejected) |

4.2.1.6. Send rate limit for client requests

The rate at which a client can send requests into the system is limited. There are two limits:

1. When the first threshold of send rate is reached, the system starts declining the application level requests and transmits the report on rejecting request with reason "Number of messages exceeded limit".
2. When the second threshold of send rate is reached, the system terminates the user session.

4.2.2. Trading system reports

4.2.2.1. Execution reports

The trading system sends a report `ExecutionReport[8]` to the client at any change in status or volume of client's order. The client can define type of event and status of order by the `OrdStatus[39]`, `ExecType[150]` and other specific fields. The value of `ExDestination[100]` field and `SecondaryOrderID[198]` field distinguish `ExecutionReport[8]` from trading system and from liquidity pool. In any cancel and routing reports, the value of the `OrderQty` field will be indicate rejected or routed volume, not initial.

The client should be aware of the asynchronous report generation. For example, the client may first receive the value `LeavesQty[151]=0` in routing reject report and then a non-zero value of `LeavesQty[151]` in order add report, followed by `LeavesQty[151]=0` in remainder cancellation report.

Table 20. Types of `ExecutionReport[8]`

| Event | Status of order <code>OrdStatus[39]</code> | Report type <code>ExecType[150]</code> | Specific fields |
|---|---|---|--|
| Order successfully accepted by the trading system | 0 | 0 | LeavesQty=OrderQty <code>ExDestination[100]=1001, OrderID[37]</code> |
| Order routing to the liquidity pool is successful | | | LeavesQty=OrderQty, <code>ExDestination[100]=1000, OrderID[37], SecondaryOrderID[198]</code> |
| Order rejected by the trading system | 8 | 8 | LeavesQty=0 <code>ExDestination[100]=1001, OrdRejReason[103]</code> |
| Order rejected by liquidity pool | | | LeavesQty=0 <code>ExDestination[100]=1000, OrdRejReason[103]</code> |
| Trade: order volume partially executed | 1 | F | 0<LeavesQty<OrderQty <code>TrdMatchID[880], LastMkt[30]</code> |
| Trade: order volume fully executed | 2 | F | LeavesQty=0 <code>TrdMatchID[880], LastMkt[30]</code> |
| Order cancellation | 4 | 4 | LeavesQty=0 <code>ExecRestatementReason[378]</code> |

Protocol specifications

Table 21. Format of message ExecutionReport [8]

| Tag | Field | ✓ | Type | Description |
|-------|-----------------------------|---|----------|---|
| | Header | R | | Standard header |
| 1 | Account | R | String | Clearing account |
| 100 | ExDestination | R | Exchange | Liquidity pool identifier specified in the order. For values please refer to Section 3.4 |
| 10104 | Price1 | N | Price | Price of the first part of repo (to be filled only for repo orders) |
| 103 | OrdRejReason | C | Int | Reasons for order rejection. Indicated when <code>ExecType(150)=8</code> . For values please refer to Table 26 . Possible values: <ul style="list-style-type: none"> • 1 |
| 1080 | RefOrderId | N | String | Identifier for matching negotiated orders |
| 1083 | DisplayWhen | N | Char | Required for iceberg order. Possible value: <ul style="list-style-type: none"> • 2 |
| 1084 | DisplayMethod | N | Char | Required for iceberg order. Possible value: <ul style="list-style-type: none"> • 1 (iceberg) |
| 11 | ClOrdId | R | String | Client ID of the command |
| 1138 | DisplayQty | N | Qty | Disclosed (visible) part of the order quantity. Required for iceberg orders: <ul style="list-style-type: none"> • $0 < \text{DisplayQty} < \text{OrderQty}$ (iceberg); • <code>DisplayQty</code> not defined (visible order) |
| 1139 | ExchangeSpecialInstructions | C | String | The main liquidity pool identifier. For values please refer to Section 3.4 . Filled when <code>ExecType[150]=0</code> or <code>F</code> , if it was indicated by the user at submission |
| 150 | ExecType | R | Char | Type of report. Possible values: <ul style="list-style-type: none"> • 0 (adding); • 4 (cancellation); • 8 (rejection of invalid order); • <code>F</code> (trade) |
| 151 | LeavesQty | R | Qty | Unfulfilled quantity of order |
| 198 | SecondaryOrderId | N | String | Order ID at liquidity pool. If filled, the report refers to the amount, routed to a liquidity pool. Otherwise, the report refers to the client order |

Protocol specifications

| Tag | Field | ✓ | Type | Description |
|-------|-----------------------|---|----------|---|
| 20113 | IgnoreDynLimits | N | Char | Ignore dynamic limits. Possible values: <ul style="list-style-type: none"> • N (no); • Y (yes) |
| 30 | LastMkt | N | Exchange | Liquidity pool identifier of last trade. For values please refer to Section 3.4 |
| 31 | LastPx | R | Price | Price of last trade. Filled when ExecType[150]=F |
| 32 | LastQty | R | Qty | Quantity of last trade in lots. Filled when ExecType[150]=F |
| 37 | OrderId | N | String | Order ID assigned by the trading system |
| 378 | ExecRestatementReason | C | Int | The reason for cancellation of order. Indicated when ExecType(150)=4. Possible values: <ul style="list-style-type: none"> • 100 (canceled on client's OrderCancelRequest [F]); • 101 (canceled on client's OrderMassCancelRequest [q]); • 102 (canceled on broker's OrderCancelRequest [F]); • 104 (canceled on broker's OrderMassCancelRequest [q]); • 105 (canceled on logout/disconnection); • 106 (canceled on expiration); • 108 (canceled by trading system operator); • 109 (IoC remainder cancel); • 110 (canceled to prevent a cross trade); • 111 (canceled to prevent a crossed book); • 112 (canceled on counterparty's DontKnowTrade [Q]); • 114 (negotiated trade); • 115 (canceled on rejection by liquidity pool); • 116 (canceled on expiration of order at liquidity pool) |
| 38 | OrderQty | R | Qty | Quantity of order in lots |
| 388 | DiscretionInst | N | Char | Required for a hidden order with dynamic price. Possible values: <ul style="list-style-type: none"> • 0 |
| 39 | OrdStatus | R | Char | Status of order. Possible values: <ul style="list-style-type: none"> • 0 (active); • 1 (partially executed); • 2 (executed); • 4 (canceled); • 8 (rejected) |

Protocol specifications

| Tag | Field | ✓ | Type | Description |
|------|-----------------------------------|---|-------------------|---|
| 40 | OrdType | C | Char | Type of order. Not present when ExecType[150]=4. Possible values: <ul style="list-style-type: none"> • 1 (market); • 2 (limit); • n (negotiated); • o (expanded liquidity pool) |
| 41 | OrigClOrdId | N | String | Client ID of order to cancel |
| 44 | Price | C | Price | Lot price |
| 453 | Component Parties | R | | |
| 48 | SecurityId | R | String | Numeric ID of the trading instrument |
| 529 | OrderRestrictions | N | MultipleCharValue | Restrictions associated with order. Possible value: <ul style="list-style-type: none"> • 5 (acting as market maker) |
| 54 | Side | R | Char | Side of order. Possible values: <ul style="list-style-type: none"> • 1 (buy); • 2 (sell) |
| 58 | Text | N | String | Comment by client |
| 59 | TimeInForce | C | Char | Period the order remains in effect. Not present when ExecType[150]=4. Possible values: <ul style="list-style-type: none"> • 0 (during the trading session); • 2 (opening auction); • 3 (immediate or cancel, IOC); • 4 (fill or kill, FOK); • 7 (closing auction); • x (during the extended trading session) |
| 60 | TransactTime | R | UTCTimestamp | Date and time of report generation |
| 841 | DiscretionMoveType | N | Int | Required for a hidden order with dynamic price. Possible values: <ul style="list-style-type: none"> • 0 |
| 843 | DiscretionLimitType | N | Int | Required for a hidden order with dynamic price. Possible values: <ul style="list-style-type: none"> • 2 |
| 880 | TrdMatchId | R | String | Trade ID assigned by liquidity pool. Filled when ExecType[150]=F |
| 9303 | RoutingInstruction | N | String | Routing algorithm ID |

4.2.2.2. Rejection OrderCancelRequest [F] report

If the requested order cannot be canceled or the cancellation request `OrderCancelRequest [F]` contains invalid parameters, the trading system will reject the request and send `OrderCancelReject [9]` to the client.

Table 22. Format of message `OrderCancelReject [9]`

| Tag | Field | ✓ | Type | Description |
|-----|-----------------------------------|---|--------------|--|
| | Header | R | | Standard header |
| 37 | OrderId | R | String | Order ID assigned by the trading system |
| 41 | OrigClOrdId | N | String | Client ID of order to cancel |
| 11 | ClOrdId | R | String | Client ID of the command to cancel order |
| 60 | TransactTime | R | UTCTimestamp | Date and time of report generation |
| 102 | CxlRejReason | R | Int | The reason for rejection of cancellation request. For values please refer to Table 26 . Possible values: <ul style="list-style-type: none"> • 1 |
| 40 | OrdType | R | Char | Type of order. Not present when <code>ExecType[150]=4</code> . Possible values: <ul style="list-style-type: none"> • 1 (market); • 2 (limit); • n (negotiated); • o (expanded liquidity pool) |
| 39 | OrdStatus | R | Char | Request status. Possible values: <ul style="list-style-type: none"> • 8 (rejected) |
| 100 | ExDestination | R | Exchange | Liquidity pool ID specified in the order. For values please refer to section 3.4 |
| 48 | SecurityId | R | String | Numeric ID of the trading instrument |
| 54 | Side | R | Char | Side. Possible values: <ul style="list-style-type: none"> • 1 (buy); • 2 (sell) |
| 1 | Account | R | String | Trading and clearing account |
| | Component Parties | R | | |
| 30 | LastMkt | C | Exchange | Exchange of last trade. For values please refer to section 3.4 |

4.2.2.3. Order mass cancellation report

In response to `OrderMassCancelRequest [q]` the system returns the report on massive cancellation `OrderMassCancelReport [r]`. If some orders were canceled on request, the report `OrderMassCancelReport [r]` will be preceded by individual reports on cancellation of each order `ExecutionReport [8]` with `ExecType [150]=4` and `OrdStatus [39]=4`.

Table 23. Format of message `OrderMassCancelReport [r]`

| Tag | Field | ✓ | Type | Description |
|------|-----------------------------------|---|--------------|--|
| | Header | R | | Standard header |
| 11 | ClOrdId | R | String | Client ID of the command to cancel order |
| 1369 | MassActionReportId | R | String | Operation number |
| 530 | MassCancelRequestType | R | Char | Type of cancellation. Possible values: <ul style="list-style-type: none"> • 1 (for the instrument); • 7 (all orders) |
| 531 | MassCancelResponse | R | Char | Status of command processing. Possible values: <ul style="list-style-type: none"> • 0 (request rejected); • 1 (orders of the specified instrument canceled); • 7 (all orders canceled) |
| 533 | TotalAffectedOrders | N | Int | Number of canceled orders |
| 60 | TransactTime | R | UTCTimestamp | Date and time of report generation |
| 100 | ExDestination | N | Exchange | Liquidity pool ID specified in the order. For values please refer to section 3.4 |
| 48 | SecurityId | N | String | Numeric ID of the trading instrument |
| 1 | Account | N | String | Trading and clearing account |
| | Component Parties | N | | |

4.2.2.4. Order rejection report

A client order with an invalid combination of required fields will be rejected with `BusinessMessageReject [j]`.

Table 24. Format of message `BusinessMessageReject [j]`

| Tag | Field | ✓ | Type | Description |
|-----|------------------------|---|--------|---|
| | Header | R | | Standard header |
| 45 | RefSeqNum | R | SeqNum | Number of rejected message |
| 372 | RefMsgType | R | String | Type of rejected message |
| 380 | BusinessRejectReason | R | Int | Error code. Possible values: <ul style="list-style-type: none"> • 5 (conditionally required field missing); • 100 (undefined tag); • 6000 (both account and parties filled) |
| 371 | RefTagId | N | Int | Tag causing the error |
| 58 | Text | N | String | Error text |

4.2.2.5. Negotiated order report

The system will send the report `MarketDataIncrementalRefresh[X]` to the counterparty at submission, execution, or cancellation of a negotiated order and at rejection order by the counterparty. The report contains one entry of the component `MDEntry` specifying the order parameters.

The `MDUpdateAction` value indicates the event: 0 at submission of a new negotiated order and 2 at execution or cancellation of negotiated order.

Table 25. Format of message `MarketDataIncrementalRefresh[X]`

| Tag | Field | ✓ | Type | Description |
|-----|-----------------------------------|---|------|-----------------|
| | Header | R | | Standard header |
| | Component MDInc | R | | |
| | Component Parties | R | | |

Appendix A. Error codes

Table 26. Error codes list

| Code | Description |
|------|----------------------------|
| 0 | Ok |
| 5 | Missed tag. |
| 100 | Filled excess tag. |
| 999 | Internal error. |
| 1000 | Incorrect login. |
| 1001 | Incorrect instrument. |
| 1002 | Incorrect client ID. |
| 1003 | Invalid member_id. |
| 1004 | Invalid account. |
| 1005 | Incorrect client group. |
| 1006 | Incorrect exchange. |
| 1007 | Instrument not traded. |
| 1008 | Invalid routing options. |
| 1100 | Invalid order direction. |
| 1101 | Incorrect price. |
| 1102 | Incorrect price_extra. |
| 1103 | Incorrect amount. |
| 1104 | Incorrect amount_extra. |
| 1105 | Invalid order type. |
| 1106 | Invalid time_in_force. |
| 1107 | Invalid passive_only. |
| 1108 | Invalid auto_cancel. |
| 1109 | Invalid flags. |
| 1110 | Invalid mode. |
| 1111 | Incorrect clorder_id. |
| 1112 | Incorrect orig_clorder_id. |
| 1113 | Invalid prime_exchange. |
| 1114 | Invalid date_expire. |
| 1115 | Invalid comment. |
| 1116 | Invalid level. |

Error codes

| Code | Description |
|------|---|
| 1117 | Invalid trade_mode. |
| 1200 | Invalid segment. |
| 1201 | Incorrect extra1. |
| 1202 | Incorrect OTC code for negotiated trade initiator. |
| 1203 | Incorrect OTC code for counter party. |
| 1204 | Invalid order_type for this instrument. |
| 1205 | Order_type not supported by exchange. |
| 1206 | Invalid order_type for Client ID. |
| 1207 | Incorrect price for this order_type. |
| 1208 | Incorrect amount_extra for this order_type. |
| 1209 | Invalid time_in_force for this order_type. |
| 1210 | Invalid flags for this order_type. |
| 1211 | Invalid instrument for replacement mode. |
| 1212 | Invalid member_id for replacement mode. |
| 1213 | Invalid client_id for replacement mode. |
| 1214 | Invalid account for replacement mode. |
| 1215 | Invalid parameters of rejected counter order. |
| 1216 | Invalid replacement parameters. |
| 1217 | Invalid time_in_force for this instrument. |
| 1218 | Invalid replacement mode for this login. |
| 1219 | Invalid flags for this instrument. |
| 1300 | Both orig_clorder_id and order_id filled. |
| 1301 | Duplicate clorder_id. |
| 1302 | Price exceeds limits. |
| 1303 | Order type not supported for this client ID. |
| 1304 | Order type not supported by exchange. |
| 1305 | Invalid prime_exchange for this instrument. |
| 1306 | Liquidity pool unavailable for client ID. |
| 1307 | Invalid order_type for this instrument. |
| 1308 | User has no permissions to cancel orders of account specified. |
| 1309 | User has no permissions to replace orders of account specified. |
| 1310 | User has no permissions to reject this order. |

Error codes

| Code | Description |
|------|--|
| 1311 | Order currently being replaced. |
| 1312 | Order sent before system crash, but received after recovery. |
| 1313 | Limitation not available for this instrument. |
| 1314 | User has no permissions to use this mode. |
| 1315 | This exchange is prohibited for clearing member. |
| 1316 | This exchange is prohibited for trade member. |
| 1317 | Order submission via the login is blocked. |
| 1318 | Order submission via the login is blocked for the client code. |
| 1319 | Order submission via the login is blocked for the TCA. |
| 1400 | Instrument not available for market maker. |
| 1401 | No permissions to trade this instrument. |
| 1402 | No permissions to indicate 'No matching another market maker's orders'. |
| 1403 | Client has no permissions to trade with using this account. |
| 1404 | Liquidity pool not available for this smart order router. |
| 1500 | Trade engine IDs (te_id) do not match. |
| 1501 | Incorrect te_id. |
| 1502 | Request received during the limited margin update. |
| 1700 | User has no permission for limited margin service. |
| 1701 | Client has no permissions for limited margin service. |
| 1702 | Client group has no permissions for limited margin service. |
| 1703 | Account has no permissions for limited margin service. |
| 1704 | Main account has no permissions for limited margin service. |
| 1710 | Invalid parameters for limited margin of client. |
| 1711 | Invalid parameters for limited margin of client group. |
| 1712 | Invalid parameters for limited margin of account. |
| 1713 | Invalid parameters for limited margin of main account. |
| 1714 | Request for limited margin update for client received when the previous request still processing. |
| 1715 | Request for limited margin update for client group received when the previous request still processing. |
| 1716 | Request for limited margin update for TCA received when the previous request still processing. |
| 1717 | Request for limited margin update for principal TCA received when the previous request still processing. |
| 1720 | Incorrect limit for limited margin. |
| 1721 | Incorrect instrument limit for limited margin. |

Error codes

| Code | Description |
|------|---|
| 1722 | Incorrect order limit for limited margin. |
| 1723 | Incorrect extra limit for limited margin. |
| 1750 | Insufficient limit for limited margin of client. |
| 1751 | Insufficient instrument limit for limited margin of client. |
| 1752 | Insufficient order limit for limited margin of client. |
| 1753 | Insufficient extra limit for limited margin of client. |
| 1754 | Insufficient limit for limited margin of client group. |
| 1755 | Insufficient instrument limit for limited margin of client group. |
| 1756 | Insufficient order limit for limited margin of client group. |
| 1757 | Insufficient extra limit for limited margin of client group. |
| 1758 | Insufficient limit for limited margin of account. |
| 1759 | Insufficient instrument limit for limited margin of account. |
| 1760 | Insufficient order limit for limited margin of account. |
| 1761 | Insufficient extra limit for limited margin of account. |
| 1762 | Insufficient limit for limited margin of main account. |
| 1763 | Insufficient instrument limit for limited margin of main account. |
| 1764 | Insufficient order limit for limited margin of main account. |
| 1765 | Insufficient extra limit for limited margin of main account. |
| 1766 | The client has active orders of limited margin. |
| 1767 | The client group has active orders of limited margin. |
| 1768 | The TCA has active orders of limited margin. |
| 1769 | The principal TCA has active orders of limited margin. |
| 1770 | Limited margin suspended for client. |
| 1771 | Limited margin suspended for client group. |
| 1772 | Limited margin suspended for account. |
| 1773 | Limited margin suspended for main clearing account. |
| 1780 | Invalid liquidity pool for limited margin service. |
| 1800 | Incorrect yield type specified. |
| 1801 | Incorrect yield conversion direction specified. |
| 1980 | Invalid stages in info field. |
| 2100 | Account does not belong to member_id. |
| 2200 | No permissions to submit trading instructions. |

Error codes

| Code | Description |
|------|---|
| 2201 | Client group level prohibition is set. |
| 2202 | Trade member level prohibition is set. |
| 2203 | Clearing member prohibition is set. |
| 2204 | Trade administrator level prohibition is set. |
| 2300 | No permissions to place an unsecured order. |
| 2400 | No permissions to cancel order. |
| 2600 | No permissions to set limit for clearing account. |
| 2601 | No permissions to set limits for client ID. |
| 2602 | No permissions to set limits for client group. |
| 2603 | Invalid type. |
| 2604 | Invalid value. |
| 2605 | Ambiguous type. |
| 2700 | Client ID has insufficient funds. |
| 2701 | Client ID has insufficient assets. |
| 2702 | Client group has insufficient funds. |
| 2703 | Client group has insufficient assets. |
| 2704 | Account has insufficient funds. |
| 2705 | Account has insufficient assets. |
| 2706 | Main clearing account has insufficient funds. |
| 2707 | Main clearing account has insufficient assets. |
| 2708 | Clearing member has insufficient funds. |
| 2709 | Insufficient blocked assets. |
| 3000 | Market or IOC order expired after no trades. |
| 3001 | Order canceled after no trades, to avoid a cross trade. |
| 3002 | Order canceled after no trades, to avoid a crossed book. |
| 3003 | Client order not found. |
| 3004 | Instrument trading suspended. |
| 3005 | У логина нет прав торговать данным инструментом в текущий период. |
| 3100 | TCA of maker and that of taker have no conversion bank indicator. |
| 3911 | Incorrect te_id. |
| 4000 | ECN not available or no liquidity pool available. |
| 4001 | The specified liquidity pool not available. |

Error codes

| Code | Description |
|------|--|
| 4002 | Order forcedly routed to a liquidity pool after rejected by risk management at the trading system. |
| 4003 | Client ID not registered at all the available liquidity pools. |
| 4004 | Client ID not registered at the trading system. |
| 4005 | Client ID not registered at liquidity pool. |
| 4006 | Order cannot be routed to any liquidity pool. |
| 4100 | Order pending cancel. |
| 4200 | Invalid client for TCA registered at liquidity pool. |
| 4201 | Invalid TCA for liquidity pool. |
| 5000 | Invalid application message type. |
| 5001 | Invalid routing_dest. |
| 5002 | Invalid message type for this login. |
| 5003 | Login has no permissions to submit such instruction. |
| 5200 | User already logged in. |
| 5201 | Discovery service settings timeout. |
| 5202 | Incorrect heartbeat_ms. |
| 5203 | Incorrect user ID / password. |
| 5204 | Incorrect message sequence number. |
| 5205 | Invalid session message type. |
| 5206 | User not logged in. |
| 5207 | Another resend request processing in progress. |
| 5208 | Incorrect range limit. |
| 5209 | Invalid reset_seq. |
| 5210 | Requested messages range excess. |
| 5211 | Invalid session message size. |
| 5212 | Disconnected by the operator. |
| 5300 | Invalid topic. |
| 5301 | Snapshot with updates has already been requested. |
| 5302 | Snapshot with updates has not been requested. |
| 5303 | Requested data not available. |
| 5304 | Another request processing in progress. |
| 5400 | Reset_seq indicated, but seqnums cannot be reset. |
| 5401 | Number of messages exceeded limit. |

Error codes

| Code | Description |
|-------|--|
| 5601 | Both account and parties filled. |
| 7000 | Order canceled before sending to ASTS. |
| 7001 | Order canceled as no answer received. |
| 13300 | Ошибка в уникальном идентификаторе запроса REQUEST_NO |
| 13301 | Неправильный отправитель |
| 13302 | Депозитарный субсчет %r или клиринговый счет %s не найден для брокера %t |
| 13303 | Несоответствие полей корпоративного события |
| 13304 | Контрольная сумма не сходится |
| 13305 | Неправильный код УТ |
| 13306 | Несовпадение контрольных остатков |
| 13307 | Истекло время приёма раскрытия |
| 13308 | Маркировка уже произведена |
| 13309 | У переданной группы клиентов повторяется значение в одном из полей ClientCode, ClientRegulator-Code, ExtClientId |
| 13310 | Нельзя подавать налоговое раскрытие по неспецдеп. счетам другого участника |
| 13311 | Не найден клиент |
| 13312 | По счету не предоставлено налоговое раскрытие |
| 13313 | Неправильный тип заявки. |
| 13314 | Номер заявки не совпадает. |
| 13315 | Брокер не найден. |
| 13316 | Клиент уже удалён ранее. |
| 13317 | Неправильный тип клиента. |
| 13318 | Отсутствует налоговый номер. |
| 13319 | Отсутствует страна. |
| 13320 | Неизвестная страна. |
| 13321 | Отсутствует основной адрес. |
| 13322 | Отсутствует город в основном адресе. |
| 13323 | Отсутствует страна в основном адресе. |
| 13324 | Неизвестная страна в основном адресе. |
| 13325 | Отсутствует почтовый адрес. |
| 13326 | Отсутствует город в почтовом адресе. |
| 13327 | Отсутствует страна в почтовом адресе. |
| 13328 | Неизвестная страна в почтовом адресе. |

Error codes

| Code | Description |
|-------|---|
| 13329 | Отсутствует имя (first name) на английском. |
| 13330 | Отсутствует фамилия (second name) на английском. |
| 13331 | Отсутствует имя (first name) на русском. |
| 13332 | Отсутствует фамилия (second name) на русском. |
| 13333 | Отсутствует название на английском. |
| 13334 | Отсутствует GIIN. |
| 13335 | Отсутствует статус Chapter 3. |
| 13336 | Неправильный статус Chapter 3. |
| 13337 | Отсутствует статус Chapter 3 Imy. |
| 13338 | Неправильный статус Chapter 3 Imy. |
| 13339 | Отсутствует статус Chapter 4. |
| 13340 | Неправильный статус Chapter 4. |
| 13341 | Отсутствует LOB. |
| 13342 | Неправильный LOB. |
| 13343 | Отсутствует уточнение для LOB=OTHER. |
| 13344 | Отсутствует налоговая ставка. |
| 13345 | Неправильная налоговая ставка. |
| 13346 | Отсутствует признак отправки отчета 1042-S через брокера. |
| 13347 | Несколько посредников имеют одинаковый level. |
| 13348 | Отсутствует название посредника на английском. |
| 13349 | Отсутствует GIIN у посредника. |
| 13350 | Отсутствует статус Chapter 3 у посредника. |
| 13351 | Неправильный статус Chapter 3 у посредника. |
| 13352 | Отсутствует статус Chapter 3 Imy у посредника. |
| 13353 | Неправильный статус Chapter 3 Imy у посредника. |
| 13354 | Отсутствует статус Chapter 4 у посредника. |
| 13355 | Неправильный статус Chapter 4 у посредника. |
| 13356 | Отсутствует LOB у посредника. |
| 13357 | Неправильный LOB у посредника. |
| 13358 | Отсутствует уточнение для LOB=OTHER у посредника. |
| 13359 | Отсутствует страна у посредника. |
| 13360 | Неизвестная страна у посредника. |

Error codes

| Code | Description |
|-------|---|
| 13361 | Не допускается указание значения N в поле NotificationViaBroker. |
| 13362 | Необходимо корректно подписать анкету. |
| 13363 | Невалидные данные клиента. |
| 13365 | Отсутствуют данные о файле W8. |
| 13366 | У клиента истекает срок действия анкеты, анкету необходимо обновить. |
| 13367 | У клиента истёк срок действия анкеты. |
| 13368 | Клиент удален из учета, используется только для архивных данных. |
| 13369 | Анкета %s распознана и принята. |
| 13370 | Анкета %s заполнена с ошибками, необходимо направить исправленную анкету. |
| 13371 | Анкета %s не совпадает с заявлением. |
| 13372 | Файл анкеты %s не найден. |
| 13373 | Анкета %s проверена вручную. |
| 13374 | Файл %s неизвестного формата. |
| 13375 | Отсутствует основной адрес, но почтовый указан как SameAs. |
| 13376 | Отсутствует город в основном адресе у посредника. |
| 13377 | Отсутствует страна в основном адресе у посредника. |
| 13378 | Неизвестная страна в основном адресе у посредника. |
| 13379 | У посредника отсутствует основной адрес, но почтовый указан как SameAs. |
| 13380 | Отсутствует город в почтовом адресе у посредника. |
| 13381 | Отсутствует страна в почтовом адресе у посредника. |
| 13382 | Неизвестная страна в почтовом адресе у посредника. |
| 13383 | Не указаны реквизиты посредника |
| 13384 | Не допускается одновременное указание элементов IntermediaryIdentification и IntermediaryLegal у посредника |
| 13385 | Неправильный брокер в IntermediaryIdentification |
| 13386 | Клиент не найден по IntermediaryIdentification |
| 13387 | Клиент, найденный по IntermediaryIdentification, имеет тип PERSON |
| 13388 | Для клиента не допускается упрощенная идентификация |
| 13389 | Более одного юридического лица найдено по заданным параметрам %s |
| 13390 | Для юрлица с ИНН %s ошибка: список GIIN пересекается со списком GIIN другого юрлица, при этом ИНН этих юрлиц не совпадают |
| 13391 | Для юрлица с ИНН %s ошибка: ИНН совпадает с ИНН другого юрлица, а списки GIIN этих юрлиц не пересекаются |

Error codes

| Code | Description |
|-------|--|
| 13392 | Список клиентских кодов не соответствует данному клиенту %s |
| 13393 | Файл %s успешно найден |
| 13394 | В данных клиента не заполнен или некорректно заполнен адрес. |
| 13395 | В данных клиента не заполнен или некорректно заполнен почтовый адрес. |
| 13396 | В данных клиента не заполнен или некорректно заполнен документ физ.лица. |
| 13397 | В данных клиента не заполнен или некорректно заполнен FATCA-статус. |
| 13398 | В данных клиента не заполнено или некорректно заполнено имя или наименование кириллицей. |
| 13399 | В данных клиента не заполнено или некорректно заполнено имя или наименование латиницей. |
| 13400 | В данных клиента не заполнен или некорректно заполнен налоговый номер. |
| 13401 | В данных клиента не заполнена или некорректно заполнена дата рождения физ.лица. |
| 13402 | В данных клиента не заполнена или некорректно заполнена подпись в анкете. |
| 13403 | Необходимо прислать анкету W8BEN. |
| 13404 | В данных клиента CLIENT повторяются счета ASSET |
| 13405 | В пачке сообщений найдено более одной анкеты для одного и того же клиента |
| 13406 | Повторяющиеся остатки |
| 13407 | Поле request_no ссылается не на request типа payment, а на другой тип request-a |
| 13408 | Необходимо указать корректную страну налогового резидентства |
| 13409 | Неизвестный код страны ОКСМ %с |

Also you can get errors come in range —11000-11999. These are the error codes returned by the trading system of the Moscow stock exchange (ASTS). To get the ASTS error id , you need to subtract 11000 from the internal error id. The description of these errors, a client can get from the ASTS documentation.

Appendix B. Revision History

Version 1.4.3 15 December 2014

Requirement to specify primary stock exchange in the order corrected.

Version 1.4.2 28 November 2014

Errors 9103, 9205, 9300, 9400, 9401, 9402, 9500, 9600, and 9601 added to error codes table.

Version 1.4.1 21 November 2014

1. Sections "Mode of negotiated repo transactions" and "Closing auction" added to section "Trade modes."
2. New order types added.
3. New error codes added.
4. Necessity of fields OrdType and ExchangeSpecialInstructions for message ExecutionReport corrected.
5. Field BusinessRejectReason in message BusinessMessageReject corrected.
6. Field ExecRestatementReason in message ExecutionReport corrected.

Version 1.3.0 29 October 2014

1. New field Price1 added and description of field Price changed in messages NewOrderSingle and ExecutionReport.
2. Field DiscretionPrice added to ExecutionReport.

Version 1.2.3 16 October 2014

Necessity of field OrderQty for message ExecutionReport corrected.

Version 1.2.2 10 October 2014

1. Field ExchangeSpecialInstructions added to messages NewOrderSingle and ExecutionReport.
2. Section on order routing added.
3. Field OrdType for negotiated order corrected.
4. New values of field BusinessRejectReason in message BusinessMessageReject corrected.
5. Field ExecRestatementReason in message ExecutionReport corrected.

Version 1.2.1 2 October 2014

New values of field TimeInForce added.

Version 1.1.0 9 June 2014

Functionality of canceling active orders on Moscow Stock Exchange by request MassCancel not available in this version.

Version 1.0 6 June 2014

Functionality of automatic order canceling in case of disconnection is not available in this version.

Version 0.3 June 2, 2014

Fields RefOrderID[1080] and ExecInst[18] added to message format NewOrderSingle[D] and ExecutionReport[8].

Version 0.2 May 8, 2014

Negotiated trading support added.