Advanced trading & clearing platform

Trade Gateway FIX (FIX 5.0 SP2)

System version 1.6

Interface version 1.6

Document version 1.8.5

26 October 2017

Revision history

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. 3.2.8).

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 andOrderCancelReject.

Version 1.4.4, 11 February 2015
1. Field BusinessRejectReason in message BusinessMessageReject corrected.
2. Interaction with trade gateway corrected at rejection of negotiated counterorder by counterparty (please refer to 2.7.2).
3. Field structure in message DontKnowTrade changed.
4. Errors 1115, 1315, 1316, 8103, 8104, 8105, 8106, and 8201 added to error codes table.

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 “Trading modes.”
2. New instruction 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 3.3.1.1.1 on instruction 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

Table of Contents

List of Tables

1. Service overview

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

1. acceptance of trade instructions submitted to exchanges;
2. routing of instructions, sending orders to available trading platforms;
3. registration of trades at exchanges supported by the platform, and procession of information about trades at external exchanges, if connected to the platform;
4. transmission of anonymous and non-anonymous maret data as well as additional and reference data;
5. control of clearing member's risks on operations with instruments registered on the platform;
6. other functionality connected with providing access to trading at the market.

For a trading member with access to several exchanges the platform processes client connections on each of the exchanges.

To ensure correct usage of the trading gateway, the client should utilize up-to-date trading instruments reference data (for more information please refer to the document *Instruments reference data*).

1.1. Main trading mode

1.1.1. Instruction types

In the Main trading mode, anonymous orders are executed at trading venues. The Main trading mode supports the following types of trading instructions. The instruction type is determined by the set of field values in the message.

1. Market instruction is an instruction that will execute at the best available prices until it is fully filled; any remainder will be expired.

2. Day limit instruction is an instruction that will execute at or better than the specified price; the remainder, if any, is added to the order book and will be active till the end of the trading day.

3. Fill or Kill (FOK) is an instruction with an indication of volume and price that is to be filled immediately and completely, or canceled.

4. Immediate or Cancel (IOC) is an instruction with an indication of volume and price that requires all or part of the instruction to be executed immediately, otherwise the instruction (or any unfilled parts) will be canceled.

5. Iceberg is an instruction that contains a disclosed quantity which will be the maximum quantity displayed in the order book. Once the displayed quantity is reduced to zero, it will be replenished by the lower of the disclosed quantity and the remainder.

In the negotiation trades mode, clients can submit directed instructions for matching in the auction against opposite orders with fully matching parameters.

The set of instructions available in the trading platform may differ from the set of orders supported by a specific trading venue.

1.1.2. Execution of instructions

A client instruction submitted to the trading platform can be executed on exchanges (1) to which this particular trading member has a connection and (2) where the instrument indicated in the instruction is admitted to trading. If there is only one exchange matching these criteria, the entire instruction volume is routed to that exchange as one or several orders. If there are several exchanges like that, the instruction will be filled in accordance with the best execution principles.

For a group of instruments listed on the trading platform, the **Main exchange** is determined among several trading venues by the highest liquidity level. The Main venue 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 exchange.

The instruction volume can only be routed to a trading venue fully or partially, if the incoming instruction type matches one of the order types supported by the exchange. However, when the best execution principle is applied, procession of some instruction types implies a potential change of order type in relation to the type of incoming instruction. In the current version of the trading platform, iceberg instructions are handled this way (for more information see 1.1.2.3).

1.1.2.1. Best execution

The Best Execution service is available for the following instructions (a) submitted using a login authorized for access to several trading venues, (b) regarding instruments admitted to trading on several exchanges, (c) carrying a special routing directive.

To ensure best execution, the volume of orders to be submitted to trading venues is determined based on the aggregated order book for each instrument. An order book is generated by combining order books received by the trading member from various markets available to the trading member.

1.1.2.2. Instruction split

The procedure of splitting an instruction into orders and routing them depends on the instruction type.

A Fill Or Kill instruction can be executed on one exchange only, where the instruction initiator can get the best average weighted price; in case of several equal prices the priority is given to the exchange providing a lower latency.

An incoming instruction of other types (limit, market, Immediate Or Cancel, iceberg) can be concurrently filled on several exchanges. For each price level consecutively, starting from the best one for the instruction initiator, the volume to be executed is determined on each available exchange.

In the course of splitting, the incoming instruction is consecutively matched with counter orders at each price level until the instruction volume is filled. If all the available price levels were checked and the incoming instruction has not been filled completely, the remaining quantity is routed to the Main trading venue. After the volumes to be routed to exchanges is determined, orders are generated and sent to the trading venues.

1.1.2.3. Iceberg processing specificity

An iceberg instruction directed to all trading venues will be divided into orders in accordance with the standard procedure, i.e. based on the current order book. An order generated for submission to an exchange, where iceberg orders are supported, will be routed as an iceberg order, whereby the hidden volume equals the initial volume, if it does not exceed the order volume, or the current order volume, if the current order volume is less than, or equal to the initial hidden volume. An order generated for submission to an exchange, where iceberg orders are not available, will be routed as an IOC order. In instructions reports the initial parameters will be stated, whereas orders reports will show new parameters.

1.2. Trading platform gateways

1.2.1. Trading gateway

The client will connect via the trading gateway to submit and cancel instructions and receive reports according to the user permissions.

The client can connect to the trading platform gateway via the native or FIX protocol; the both provide interaction at the session and application levels. The session level supports realiability of message exchange. The application level allows the client to submit requests and receive reports.

1.2.2. Drop copy gateway

The client will connect via the drop copy gateway to receive reports according to the user permissions.

The client can connect to the trading platform gateway via the native or FIX protocol; the both provide interaction at the session and application levels. The session level supports realiability of message exchange. The application level enables

a unidirectional communication—the gateway transmits reports according to the login permissions, but the client is not allowed to submit transactional requests.

1.3. Login

The login is a user account to access the trading system and have specific permissions granted during the registrastion.

Table 1.1. Login permissions

Access	Description	Required
Trading member	The login belongs to trading member	required
Account	One or more accounts can be associated with the login	optional
Client identifiers	One or more client IDs can be associated with the login	optional
Gateways	A login may have an access to one or more gateway type: 1. Trade; 2. DropCopy; 3. Risk (risk management)	at least one required
Reports receipt (for trading gateways)	A login may receive a different set of reports: 1. CompleteLog: reports on both instructions and orders; 2. RestrictedLog: reports on instructions only	required

The trading member, account, and client ID, assigned to a login, restrict the scope of requests to submit and order reports to receive.

The gateway will either transmit reports on both instructions and orders or reports on instructions only. In the latter case, the client will receive a lower number of reports, but will not be aware of orders amount at trading venues.

An IP address mask is assigned to the login during the registration and defines the range of addresses authorized for the user connection.

2. Interaction with trading gateway

2.1. Reports ExecutionReport[8]

The trading system sends a report ExecutionReport[8] to the client at any change in status or volume of client's instruction or order generated by such and instruction:

1. acceptance of instruction by the trading system,
2. rejection of instruction by the trading system,
3. acceptance of order by exchange,
4. rejection of order by exchange,
5. trade,
6. partial or full execution of instruction's volume,
7. remainder cancellation after order execution,
8. partial or complete cancellation of instruction.

Each report ExecutionReport has two fields specifying the type of event that caused the report generation. They indicate the status of instruction / order and the type of report: OrdStatus[39] and ExecType[150], respectively.

Table 2.1. Types of reports and statuses of instruction / order

Event	Status of instruction OrdStatus[39]	Report type ExecType[150]	Volume ratio
Instruction successfully accepted by the trading system Order successfully accepted by the exchange	0	0	CumQty=0 LeavesQty=OrderQty
Instruction rejected by the trading system Order rejected by exchange	8	8	CumQty=0 LeavesQty=0
Trade: instruction volume partially executed Trade: order volume partially executed	1	F	0<CumQty<OrderQty 0<LeavesQty<OrderQty
Trade: instruction volume fully executed Trade: order volume fully executed	2	F	CumQty=OrderQty LeavesQty=0
Instruction cancellation Order cancellation	4	4	CumQty<OrderQty (may equal 0) LeavesQty=0

Each ExecutionReport contains the client's identifier of instruction ClOrdID[11]. After the instruction is accepted by the trading system, all related reports will contain identifier OrderID[37]. The exchange assigns its identifier to the accepted order and transmits it to the client in field SecondaryOrderID[198].

2.1.1. Discrimination between instruction and order reports

The ExDestination[100] value specified in the report unambiguously attribute the report to either instruction in the trading system (1001) or to order on exchange (1000).

The field SecondaryOrderID[198] is filled only in reports, sent after the order has been successfully placed. If the order has been rejected, this field is not set. In case of full rejection of an order on an exchange side, two Execution-

`Report` messages on the instruction level, about instruction placement and remainder cancellation, correpond to a single `ExecutionReport` on the order level.

The client should employ one of the following mode:

1. Instruction reports processing mode. In this mode only the instruction number on the trading system side is available. The order number on the exchange side is not available.

2. Order reports processing mode (recommended). This mode provides the whole information on orders execution on markets (the `ExecutionReport` has the instruction and order numbers). However, discrimination between instruction and order by the `SecondaryOrderID[198]` field may be ambiguous.

3. Both instruction and order reports processing mode. When using this mode, the client should ignore a report duplicating an event of previous report. Due to asynchronous report generation, the client may first receive the value `LeavesQty[151]=0` in order cancel report and then a non-zero value of `LeavesQty[151]` in instruction add report, followed by `LeavesQty[151]=0` in remainder cancellation report.

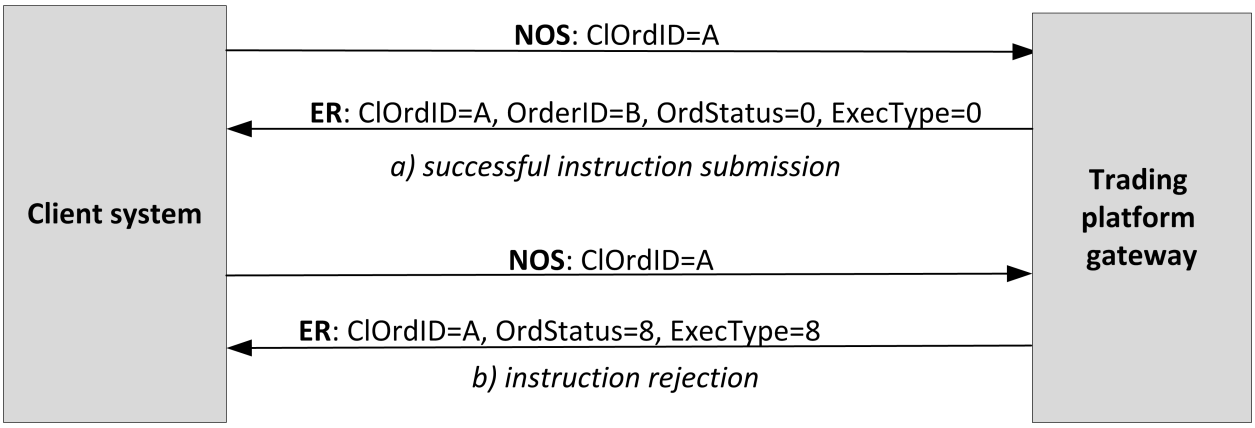
In either mode, the client shall process the Reject message.

2.2. Submission of instructions

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

After accepting the instruction, the trading platform 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 instruction (due to invalid values or closed market), no instruction 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 2.1. Submission of instructions



2.2.1. Order placement

To ensure best execution, the instruction volume is split according to the order book of the market and then generated orders are routed to trading venues. When the exchange confirms order acceptance or rejection, the trading system sends corresponding report `ExecutionReport[8]` to the client containing order identifier `SecondaryOrderID` and values `OrdStatus[39]=0` and `ExecType[150]=0`.

If the exchange rejects an order, the trading platform will return `ExecutionReport[8]` to the client (`OrdStatus[39]=8` and `ExecType[150]=8`) along with partial cancel report of the volume of the rejected order. In any cancel report, the value of the `OrderQty[38]` field will indicate rejected volume, not initial.

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

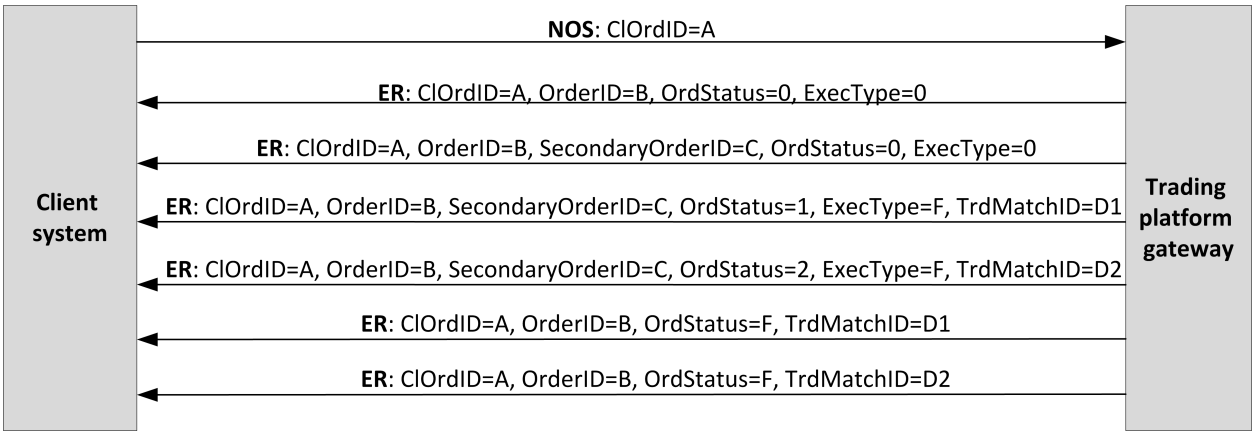
Figure 2.2. Instruction and order placement or rejection

2.3. Execution of instructions

After the trading venue accepts an order, the client will be sent reports (`ExecType[150]=F`) on order change and then on instruction change. All reports include the exchange trade ID `TrdMatchID[880]`.

The graph below shows the submission of an instruction and the following receipt of reports as seen by one side of the trade. The instruction is submitted and fully executed.

Figure 2.3. Submission of instruction and receipt of reports



2.4. Remainder cancellation after partial fill

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

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

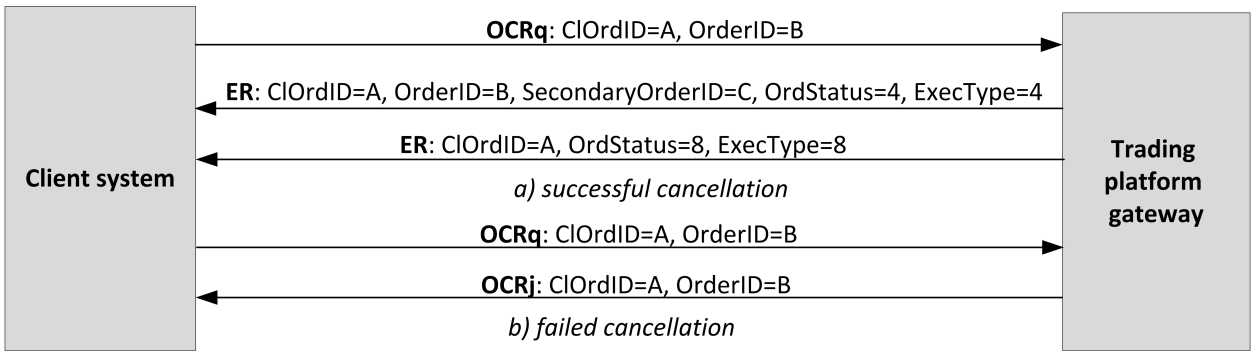
2.5. Instruction remainder cancellation

The client can cancel the unfilled remainder of an instruction. (Cancellation of an order, if several orders of one instruction are still active, is not allowed.) The client shall send `OrderCancelRequest[F]` (OCRq) to the trading platform gateway and specify the identifier and certain parameters of the instructions.

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

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

Figure 2.4. Instruction cancellation

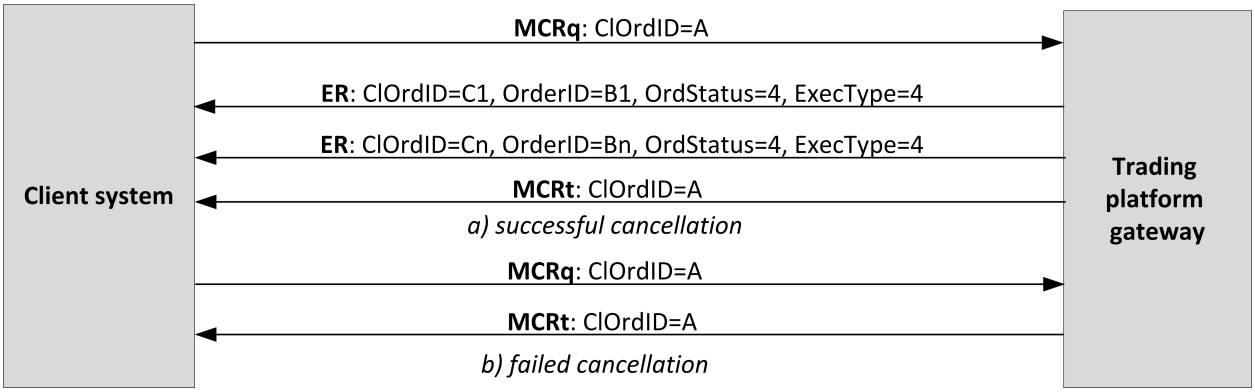


2.6. Instruction mass cancellation

The client may request to massively cancel instructions to be selected on certain grounds, for instance the instructions referring to a certain instrument submitted from the particular login. The client shall send `OrderMassCancelRequest[q]` (MCRq) to the trading platform gateway and specify the cancellation mode and, if necessary, certain parameters of instructions.

The trading platform receives the request and selects instructions to cancel by the specified criteria, and then generates cancellation request and routes them to trading venues. If the orders are canceled successfully, the client will receive reports on orders and instructions cancellation and the report on execution of order `OrderMassCancelReport[r]` (MCRt) specifying the number of canceled instructions. If no instruction to cancel is found, the gateway will only return `OrderMassCancelReport[r]`.

Figure 2.5. Instruction mass cancellation



2.7. Negotiated order submission, execution and declining

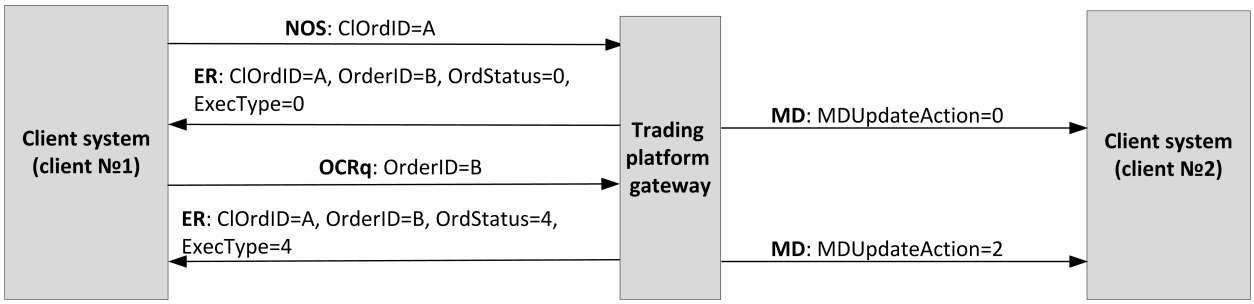
To submit a negotiated order, the client should send `NewOrderSingle[D]` (NOS) to the trading platform gateway with unique `ClOrdID[11]` assigned.

After accepting the negotiated order, the trading platform 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 instruction 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 exchange accept the negotiated order, the client-sender may cancel it before the counterparty submits the counterorder. To cancel the negotiated order, the client should send `OrderCancelRequest[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`.

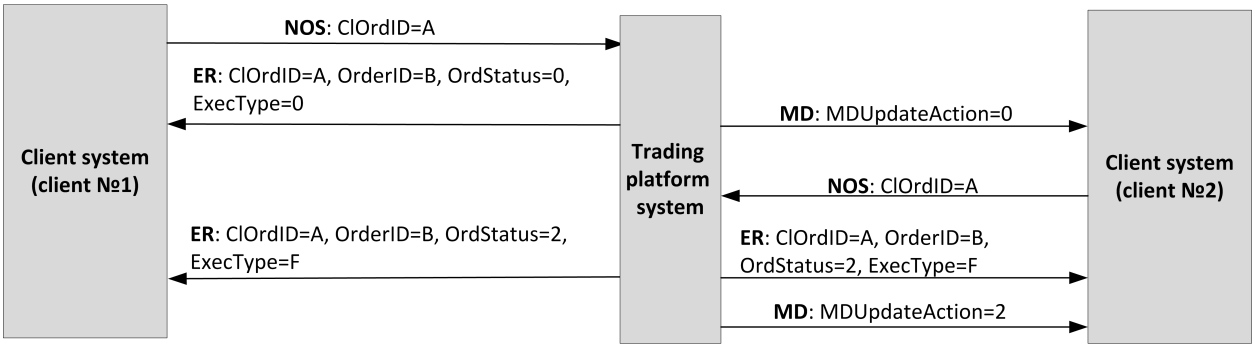
Figure 2.6. Negotiated instruction placement and cancellation



2.7.1. Negotiated counterorder placement

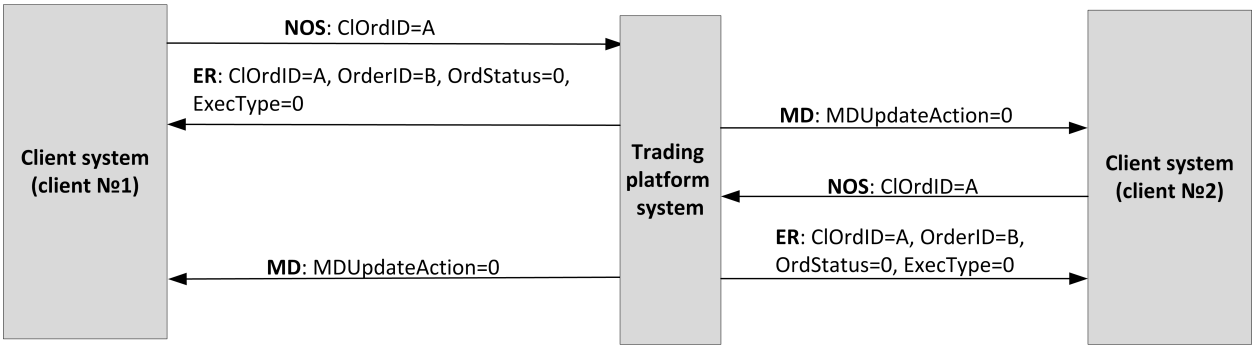
To take the offer, the counterparty shall send the counterorder with the same quantity of the instrument at the same price and the opposite side.

Figure 2.7. Successful submission of negotiated counterinstructionplacement



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

Figure 2.8. Failed submission of negotiated counterinstruction placement



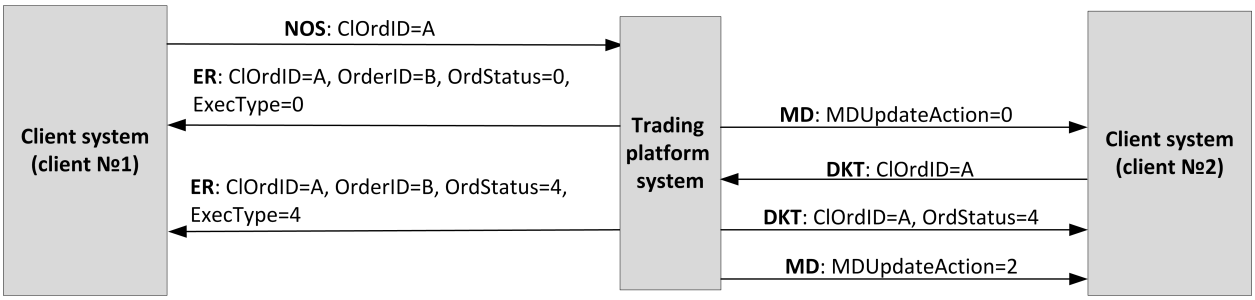
2.7.2. Negotiated counterorder declining by counterparty

The counterparty can decline the negotiated order. The client should send `DontKnowTrade[Q]` (DKT) to the trading platform 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]=2` and `ExecType[150]=F`).

Figure 2.9. Negotiated counterinstruction rejection



3. Protocol specifications

3.1. Datatypes

This section contains tables describing the message formats

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, char 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=o z`.

NumInGroup - integer to indicate number of entries in a group.

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.

UTCTimestamp - 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. Session layer

The session layer is mostly compliant with FIX Session Protocol 1.1.

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

The ID of the trading platform gateway is ECN_EQR and that of the client is the user name.

3.2.1. 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: first comes always `BeginString[8]=FIXT.1.1` 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 3.1. Format of message header

Tag	Field	✓	Type	Values	Description
8	BeginString	R	String	FIXT.1.1	First field
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 ResendRequest[2]
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 3.2. Format of message trailer

Tag	Field	✓	Type	Features
10	CheckSum	R	String	Message check sum (3 bytes)

3.2.2. Message sequence number `MsgSeqNum`

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

As reference information for the 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, the client should send `ResendRequest[2]`.

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

3.2.2.1. Request for resending

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

The client may request resending all messages sent during the current and previous trading days. If the client has forcefully 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 the client uses `BeginSeqNo[7]=0` and `EndSeqNo[16]=0`, the gateway will resend all messages starting from the lowest number available.

If the client specifies 0 only for `EndSeqNo[16]`, the server 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 requiring more messages, the client should send several consecutive requests. Any further request sent prior to resend completion will be rejected by the gateway.

Table 3.3. Format of message `ResendRequest[2]`

Tag	Field	✓	Type	Values	Description
7	BeginSeqNo	R	SeqNum		Number of first requested message
16	EndSeqNo	R	SeqNum		Number of last requested message

In response to `ResendRequest[2]`, the gateway 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 message.

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

If the client is to increase the message number expected from the server, the client should sent `SequenceReset[4]` with `GapFillFlag[123]=N` and the new expected number in the field `NewSeqNo[36]`.

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

Table 3.4. Format of message `SequenceReset[4]`

Tag	Field	✓	Type	Values	Description
36	NewSeqNo	R	SeqNum		New sequence number
123	GapFillFlag	N	Boolean	N (mode Reset ignoring field MsgSeqNum; specified by the client); Y (mode GapFill using field MsgSeqNum; specified by the server)	Indicator of gap fill

3.2.2.2. Resetting message sequence numbers

The value `ResetSeqNumFlag[141]=Y` in the `Logon[A]` message allows to reset sequence numbers. This functionality may be useful to avoid procedures for requesting and restoring missing or allegedly missing messages. It is not recommended to use this feature during the trading session when trading messages have already been sent, because after the reset these messages will not be available for request.

In response to the 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`. Thus, each party will have the next message number equal to `2`.

3.2.3. Session initialization

The `Logon[A]` initiates or confirms a FIX session. After establishing a TCP connection, the session initiator (client) sends this message and expects `Logon[A]` in response.

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 incremented.

Table 3.5. Format of message `Logon[A]`

Tag	Field	✓	Type	Values	Description
98	EncryptMethod	R	Int	0 (Encryption not supported)	Method of encryption
108	HeartBtInt	R	Int		Timeout. Value in seconds. Recommended value: from 20 to 30
95	RawDataLength	C	Length	1	The field must be present if there is RawData[96]
96	RawData	N	data	0 (do not activate automatic); 1 (activate automatic)	Automatic cancellation of all instructions submitted by this login at disconnection
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 server
554	Password	N	String		Login password
1137	DefaultApplVerId	R	String	9 (FIX50SP2)	Protocol version

3.2.4. Session termination

`Logout[5]` initiates or confirms the session termination and shall be sent after a long-term absence of messages (please refer to 3.2.5) or after receiving a message 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 3.6. Format of message `Logout[5]`

Tag	Field	✓	Type	Values	Description
1409	SessionStatus	N	Int	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)	Numeric code of the reason. To be filled by the server only
58	Text	N	String		Report on session termination reason

3.2.5. Heartbeats

To monitor the connection status, client and server exchange messages `Heartbeat[0]`. A heartbeat is to be sent by a party if it passed no messages (of the session or application layer) within the heartbeat interval. The 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 server disconnects after sending message `Logout[5]` to the client. The client is expected to act the same.

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

Table 3.7. Format of message `HeartBeat[0]`

Tag	Field	✓	Type	Values	Description
112	TestReqId	C	String		Request ID of TestRequest, to which this message is a response

Table 3.8. Format of message `TestRequest[1]`

Tag	Field	✓	Type	Values	Description
112	TestReqId	R	String		Request ID Maximum length is 32 characters. Valid characters are letters and numbers

3.2.6. 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 datatype, etc. All session level messages with invalid value of any field are also rejected by the message `Reject`.

The server 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 server 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 3.9. Format of message `Reject[3]`

Tag	Field	✓	Type	Values	Description
45	RefSeqNum	R	SeqNum		Number of rejected message
371	RefTagId	N	Int		Tag which caused the error
372	RefMsgType	N	String		Type of rejected message

Tag	Field	✓	Type	Values	Description
373	SessionRejectReason	N	Int	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 CheckSum[10] misplaced); 15 (tag from the group misplaced); 16 (invalid number of group entries)	Reason for rejection
58	Text	N	String		Error report

3.2.7. 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.

3.2.8. Automatic cancellation upon disconnection

All active instructions submitted by the login can be canceled at FIX session termination. The option should be enabled during the session initialization by the values `RawDataLength[95]=1` and `RawData[96]=1` in the `Logon[A]` message. By default, the automatic cancellation is disabled.

Instructions submitted by the user login (i.e. via the client gateway) will be canceled at FIX session termination if

1. TCP connection dropped by the client gateway,
2. after no answer received to `TestRequest[0]` during the heartbeat interval,
3. upon receipt of the `Logout[5]`.

If the automatic cancellation was enabled, all client instructions, including those submitted in previous sessions, will be canceled. The cancellation will be reported to the permitted logins. The `ExecutionReport[8]` will have the indication `Text[58]=Cancel on disconnect`.

Otherwise, the client may enable the automatic cancellation for a single instruction by specifying the value `ExecInst[18]=o` in the `NewOrderSingle[D]`. This instruction will be canceled upon disconnection even if the option was not enabled at the session initialization.

3.3. Application level

3.3.1. Client requests

3.3.1.1. Submission of instructions

To submit a new instruction to the trading system, the client should send the message `NewOrderSingle[D]` specifying the following:

- clearing account in the field `Account[1]`,
- trading instrument in the field `SecurityID[48]` (please refer to the *Instrument reference data*),
- routing algorithm in the fields `ExDestination[100]` and `ExchangeSpecialInstructions[1139]` (for more information please refer to 3.3.1.1.1),
- side of instruction in the field `Side[54]`,
- type of instruction in the field `OrdType[40]`,
- period the instruction remains if effect in the field `TimeInForce[59]`,
- quantity of instrument lots in the field `OrderQty[38]`.

For all instruction types, except for the market one (`OrdType[40]=1`), price must be set in `Price[44]`.

Table 3.10. Concordance between the instruction type and field values in messages

Type of instructions	Required fields
Market	`OrdType[40]=1` `TimeInForce[59]=3`
Market instruction at closing auction	`OrdType[40]=1` `TimeInForce[59]=7`

Limit instructions at closing auction	`OrdType[40]=2`
	`TimeInForce[59]=7`
	`Price[44]`

Day active limit	`OrdType[40]=2`
	`TimeInForce[59]=0`
	`Price[44]`

Limit instruction in extended trading session	`OrdType[40]=2`
	`TimeInForce[59]=X`
	`Price[44]`

Fill Or Kill (FOK)	`OrdType[40]=2`
	`TimeInForce[59]=4`
	`Price[44]`

Immediate Or Cancel (IOC)	`OrdType[40]=2`
	`TimeInForce[59]=3`
	`Price[44]`

Iceberg	`OrdType[40]=2`
	`TimeInForce[59]=0`
	`0<DisplayQty[1138]<OrderQty[38]`
	`DisplayWhen[1083]=2`
	`DisplayMethod[1084]=1`
	`Price[44]`

Negotiated	`OrdType[40]=n` `TimeInForce[59]=0 Price[44]`

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`) instructions.

The trading member and the client code, on whose behalf the instruction is issued, should be specified in the field PartyID[448] in the Parties group; the first group entry contains the trading member with PartyRole[452]=1 and the second entry defines the client code with PartyRole[452]=3. In a negotiated instruction, the Parties group should include two more entries - the order initiator and recipient.

The client should set the client instruction identifier in the field ClOrdID[11].The trading system requires this identifier to be unique during the trading session for each client gateway. It is not recommended to reuse ClOrdID[11] of rejected instructions.

A negotiated order can be assigned with the special identifier RefOrderID[1080] so that the counterorder must contain the same ID, otherwise the orders will not match.

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

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

The commentary, specified by the client in the Text[58] field of a NewOrderSinlge[D] message, will be reproduced in an ExecutionReport[8] message, responding to the order placement. When ExecutionReport[8] is sent in responce to initial order execution, the commentary specified by the client in the Text[58] field is not sent.

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

Table 3.11. Format of message NewOrderSingle[D]

Tag	Field	✓	Type	Values	Description
11	ClOrdId	R	String		Client instruction identifier. 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		Trading venue where the order is sent to. For values please refer to 3.3.1.1.1
48	SecurityId	R	String		Numeric ID of trading instrument
9303	RoutingInstruction	N	String		Routing algorithm
54	Side	R	Char	1 (buy); 2 (sell)	Side of instruction
40	OrdType	R	Char	1 (market); 2 (limit); n (negotiated)	Type of instruction

Tag	Field	✓	Type	Values	Description
59	TimeInForce	R	Char	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)	Period the instruction remains in effect
44	Price	C	Price		Price. For repo trading: annual interest yield, the value to be indicated in percentage
38	OrderQty	R	Qty		Volume of instruction in lots
1138	DisplayQty	N	Qty		Disclosed quantity of instruction. Required for icebergs: • `0<DisplayQty<OrderQty` (iceberg); • `DisplayQty`not defined (disclosed orders)
1084	DisplayMethod	N	Char	1 (iceberg)	Required for icebergs
1	Account	R	String		Clearing account of the client submitting instruction
	Component Parties	R			
58	Text	N	String		Comment. Maximum length is 23 characters
1139	ExchangeSpecialInstructions	N	String		The main trading venue. For values please refer to 3.3.5
1080	RefOrderId	N	String		Identifier for matching negotiated orders
10104	Price1	N	Price		Additional price. For a repo the trade price can be specified

3.3.1.1.1. Instruction routing options

The client sets the instruction routing in the two fields:

1. `ExDestination` is the trading venue, where the client order is sent to; for values please refer to 3.3.5;

2. `ExchangeSpecialInstructions` is the Main trading venue, where the instruction remainder, if any, will be sent to; for values please refer to 3.3.5. If the field is ommited, the trading system will use the default value specified in *List of Securities Available in Broker Subsystem* at NP RTS site http://nprts.ru.

3.3.1.2. Cancellation of instruction remainder

After the orders are placed on trading venues, the client can cancel the instruction quantity that is still not filled. The client should send `OrderCancelRequest[F]` to the trading system with the instruction being cancelled is identified in either of the two fields: `ClOrdID[11]` or `OrderID[37]` (allowed only for the login submitted the instruction). While cancelling an instruction submitted by another login, the user should specify `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 3.12. Format of message `OrderCancelRequest[F]`

Tag	Field	✓	Type	Values	Description
41	OrigClOrdId	C	String		Client identifier of instruction to cancel. Maximum length is 20 characters. Valid characters are Latin letters and numbers
11	ClOrdId	R	String		Client identifier of the command. Maximum length is 20 characters. Valid characters are Latin letters and numbers
37	OrderId	C	String		Instruction ID assigned by the trading system
60	TransactTime	R	UTCTimestamp		Date and time of request generation
100	ExDestination	R	Exchange		Trading venue specified in the instructions. For values please refer to 3.3.1.1.1
48	SecurityId	R	String		Numeric ID of the trading instrument
54	Side	R	Char	1 (buy); 2 (sell)	Side of instruction
1	Account	R	String		Clearing account
	Component Parties	R			

3.3.1.3. Instruction mass cancellation

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

Table 3.13. Instruction mass cancellation modes

Mode	Required fields
Cancellation of the instructions submitted by the requesting login	`MassCancelRequestType[530]=7`
Cancellation of all instructions of the instrument submitted by the requesting login	`MassCancelRequestType[530]=1` `SecurityID[48]`

Cancellation of all instructions of the instrument and the clearing account	`MassCancelRequestType[530]=1`
	`SecurityID[48]`
	`Account[1]`

Mode	Required fields

Cancellation of all instructions of the instrument and the client code	`MassCancelRequestType[530]=1`
	`SecurityID[48]`
	group `Parties`

When setting the mode for cancellation of instructions 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 instruction with a separate report `ExecutionReport[8]` with statuses `ExecType[150]=4` and `OrdStatus[39]=4`, and then sends `OrderMassCancelReport[r]`.

Table 3.14. Format of message `OrderMassCancelRequest[q]`

Tag	Field	✓	Type	Values	Description
11	ClOrdId	R	String		Client identifier of the command. Maximum length is 20 characters. Valid characters are Latin letters and numbers
530	MassCancelRequestType	R	Char	1 (for the instrument); 7 (all instructions)	Type of cancellation
60	TransactTime	R	UTCTimestamp		Date and time of request generation
100	ExDestination	N	Exchange		Trading venue specified in the instructions. For values please refer to 3.3.1.1.1
48	SecurityId	C	String		Numeric ID of trading instrument. Required when `MassCancelRequestType[530]=1`
1	Account	N	String		Clearing account
	Component Parties	N			

3.3.1.4. Negotiated counterorder declining

The client can decline a negotiated order. The client should send `DontKnowTrade[Q]` to the trading platform gateway with the instruction identifier `OrderID[11]`, the counterparties in the `Parties` group, and, if needed, the match identifier `RefOrderID[1080]`.

After processing the request, the trading platform 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 notification `MarketDataIncrementalRefresh[X]`.

Table 3.15. Format of message `DontKnowTrade[Q]`

Tag	Field	✓	Type	Values	Description
37	OrderId	R	String		Instruction ID assigned by the trading system
48	SecurityId	R	String		Numeric ID of the trading instrument

Tag	Field	✓	Type	Values	Description
54	Side	R	Char	1 (buy); 2 (sell)	Side
40	OrdType	R	Char	1 (market); 2 (limit); n (negotiated); o (out-of-auction)	Type of order
	Component Parties	R			
1080	RefOrderId	N	String		Identifier for matching negotiated orders
39	OrdStatus	R	Char	4 (canceled); 8 (rejected)	Status of order / instruction

3.3.2. Trading system reports

3.3.2.1. `BusinessMessageReject[j]`

A client request with an invalid combination of conditionally required fields, including the indication of instruction type, will be rejected with `BusinessMessageReject[j]`.

Table 3.16. Format of message `BusinessMessageReject[j]`

Tag	Field	✓	Type	Values	Description
45	RefSeqNum	R	SeqNum		Number of rejected message
372	RefMsgType	R	String		Type of rejected message
380	BusinessRejectReason	R	Int	5 (conditionally required field missing); 100 (undefined tag); 6000 (both account and parties filled)	Error code
371	RefTagId	N	Int		Tag causing the error
58	Text	N	String		Error text

3.3.2.2. `ExecutionReport[8]`

`ExecutionReport[8]` will be sent to the client in case of rejection, cancellation, modification, and expiration of a client instruction or order, as well as when placing an order in the order book (for the report types please refer to 2.1).

The cancellation report (`OrdStatus[39]=4` and `ExecType[150]=4`) usually contains the cancellation reason `ExecRestatementReason[378]`.

The trade report (`ExecType[150]=F`) includes the exchange identifier of the trade `TrdMatchID[880]` and specifies the trading venue of the trade in `LastMkt[30]`.

When instructions or orders are rejected, the report will contain rejection reasons in the field `OrdRejReason[103]`.

Each report contains the client's identifier of instruction `ClOrdID[11]`. The event causing the report can be defined by the fields `OrdStatus[39]` and `ExecType[150]`. The report containing the order identifier `SecondaryOrderID[198]` refers to the order, not the instructions.

Table 3.17. Format of message `ExecutionReport[8]`

Tag	Field	✓	Type	Values	Description
	[gate_header]	R			Standard header
1	Account	R	String		Clearing account
100	ExDestination	R	Exchange		Trading venue. For values please refer to 3.3.1.1.1
10104	Price1	N	Price		Price of the first part of repo (to be filled only for repo orders)
103	OrdRejReason	C	Int	1	Reasons for order / instruction rejection. Indicated when `ExecType(150)=8`. For values please refer to Table A.1
1080	RefOrderId	N	String		Identifier for matching negotiated orders
1083	DisplayWhen	N	Char	2	Required for iceberg
1084	DisplayMethod	N	Char	1 (iceberg)	Required for iceberg
11	ClOrdId	R	String		Client command identifier
1138	DisplayQty	N	Qty		Disclosed (visible) part of the order amount. Used for icebergs: • `0<DisplayQty<OrderQty` (iceberg); • `DisplayQty` not defined (visible instructions)
1139	ExchangeSpecialInstructions	C	String		Main trading venue. For values please refer to 3.3.5. Filled when `ExecType[150]=0 or F`, if it was indicated by the user at submission
14	CumQty	N	Qty		Executed quantity of order / instruction
150	ExecType	R	Char	0 (adding); 4 (cancellation); 8 (rejection of invalid order / instruction); F (trade)	Type of report
151	LeavesQty	R	Qty		Non-executed quantity of order / instruction

Tag	Field	✓	Type	Values	Description
18	ExecInst	N	MultipleChar-Value		Command for order handling
198	SecondaryOrderId	N	String		Order ID at exchange. If filled, the report refers to the order. Otherwise, the report refers to instruction
30	LastMkt	N	Exchange		Exchange of last trade. For values please refer to 3.3.5
31	LastPx	R	Price		Price of last trade. Filled when ExecType[150]=F
32	LastQty	R	Qty		Quantity of last trade. Filled when ExecType[150]=F
37	OrderId	N	String		Instruction ID assigned by the trading system

Tag	Field	✓	Type	Values	Description
378	ExecRestatementReason	C	Int	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 disconnection); 106 (canceled on expiration); 108 (canceled by trading platform 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 external trading venue); 116 (canceled on expiration of order at external trading venue)	The reason for cancellation of order / instruction. Indicated when `ExecType(150)=4`
38	OrderQty	R	Qty		Quantity of order / instruction in lots
388	DiscretionInst	N	Char	0	Required for a discretionary order
39	OrdStatus	R	Char	0 (active); 1 (partially executed); 2 (executed); 4 (canceled); 8 (rejected)	Status of order / instructions

Tag	Field	✓	Type	Values	Description
40	OrdType	C	Char	1 (market); 2 (limit); n (negotiated); o (out-of-auction)	Type of instruction. Not present when `ExecType[150]=4`
41	OrigClOrdId	N	String		Client identifier of instruction 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	5 (acting as market maker)	Restrictions associated with order
54	Side	R	Char	1 (buy); 2 (sell)	Side
58	Text	N	String		Comment by client
59	TimeInForce	C	Char	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)	Period the instruction remains in effect. Not present when `ExecType[150]=4`
60	TransactTime	R	UTCTimestamp		Date and time of report generation
841	DiscretionMoveType	N	Int	0	Required for discretionary order
843	DiscretionLimitType	N	Int	2	Required for discretionary order
880	TrdMatchId	R	String		Trade ID assigned by exchange. Filled when ExecType[150]=F
9303	RoutingInstruction	N	String		Routing algorithm

3.3.2.3. Report on rejection of instruction cancellation request

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

Table 3.18. Format of message `OrderCancelReject[9]`

Tag	Field	✓	Type	Values	Description
37	OrderId	R	String		Instruction ID assigned by the trading system

Tag	Field	✓	Type	Values	Description
41	OrigClOrdId	N	String		Client identifier of instruction to cancel
11	ClOrdId	R	String		Client identifier of the command
60	TransactTime	R	UTCTimestamp		Date and time of report generation
102	CxlRejReason	R	Int	1	The reason for rejection of cancellation request. For values please refer to Table A.1
40	OrdType	R	Char	1 (market); 2 (limit); n (negotiated); o (out-of-auction)	Type of instruction. Not present when `ExecType[150]=4`
39	OrdStatus	R	Char	8 (rejected)	Request status
100	ExDestination	R	Exchange		Trading venue. For values please refer to 3.3.1.1.1
48	SecurityId	R	String		Numeric ID of the trading instrument
54	Side	R	Char	1 (buy); 2 (sell)	Side
1	Account	R	String		Trading and clearing account
	Component Parties	R			
30	LastMkt	C	Exchange		Exchange of last trade. For values please refer to 3.3.5

3.3.2.4. Report on mass cancellation of instructions

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

Table 3.19. Format of message `OrderMassCancelReport[r]`

Tag	Field	✓	Type	Values	Description
11	ClOrdId	R	String		Client identifier of command
1369	MassActionReportId	R	String		Operation number
530	MassCancelRequestType	R	Char	1 (for the instrument); 7 (all instructions)	Type of cancellation

Tag	Field	✓	Type	Values	Description
531	MassCancelResponse	R	Char	0 (request rejected); 1 (instructions of the specified instrument canceled); 7 (all instructions canceled)	Status of command processing
533	TotalAffectedOrders	N	Int		Number of canceled instructions
60	TransactTime	R	UTCTimestamp		Date and time of report generation
100	ExDestination	N	Exchange		Trading venue. For values please refer to 3.3.1.1.1
48	SecurityId	N	String		Numeric ID of the trading instrument
1	Account	N	String		Trading and clearing account
	Component Parties	N			

3.3.3. Notification of negotiated counterorder placement

At submission, execution, or cancellation of a negotiated order directed to the client, the gateway will send the notification `MarketDataIncrementalRefresh[X]` containing one entry of the group `MDEntry` specifying the order parameters.

The `MDUpdateAction` value indicates the event: `1` at submission of a new negotiated order and `2` at execution or cancellation of negotiated order.

Table 3.20. Format of message `MarketDataIncrementalRefresh[X]`

Tag	Field	✓	Type	Values	Description
	Component MDInc	R			
	Component Parties	R			

3.3.4. Format of message components

Table 3.21. Format of component `MDInc`

Tag	Field	✓	Type	Values	Description
268	NoMDEntries	R	NumInGroup		Number of entries in repeated group
48	SecurityId	N	String		Numeric ID of trading instrument
22	SecurityIdSource	N	String		Trading venue of order placement. For values please refer to 3.3.5
279	MdUpdateAction	R	Char	0 (new order); 2 (execution, cancellation or rejection of order)	Type of update
278	MdEntryId	R	String		Instruction ID assigned by trading platform

Tag	Field	✓	Type	Values	Description
269	MdEntryType	R	Char	0 (buy); 1 (sell)	Side
270	MdEntryPx	N	Price		Price
271	MdEntrySize	N	Qty		Volume
272	MdEntryDate	R	UTCDateOnly		Date of update
273	MdEntryTime	R	UTCTimeOnly		Time of update

Table 3.22. Format of component `Parties`

Tag	Field	✓	Type	Values	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	D	Identifies class or source of the PartyID
452	PartyRole	R	Int	1 (trading member); 3 (client code); 13 (initiator of negotiated order); 17 (counterparty for negotiated order)	Role of the subject specified in PartyID

3.3.5. Trading venue identifiers

Trading venue identifiers may be in fields `ExDestination[100]`, `LastMkt[30]` and `ExchangeSpecialInstructions[1139]`.

`0` (DEFAULT) — default trading venue

`1001` (TRADSYS) — all accessible trading venues

`1000` — Saint-Petersburg Exchange, Main trading venue at Foreign Equities market

`1010` — Moscow Exchange, Main trading venue at Russian Equities market

`1015` — execution at US markets

`1016` — US market data

`1030` — NYSE

`1031` — ARCA

`1032` — NASDAQ

`1033` — BATS

Appendix A. Error codes

Table A.1. 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.
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.

Code	Description
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 declined 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	Exchange 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 decline this order.
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	Exchange 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.

Code	Description
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.
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 exchange for limited margin service.

Code	Description
1900	(internal) Таймаут по загрузке данных.
1901	(internal) Неверный параметр market_id.
1902	(internal) Неверный параметр clearing_id.
1903	(internal) Неверный параметр sess_id.
1904	(internal) Неверные параметры объекта risk_entity.
1905	(internal) Неверный параметр min_step.
1906	(internal) Неверный параметр step_price_curr.
1907	(internal) Валюта не найдена.
1908	(internal) Неверный параметр min_vol.
1909	(internal) Неверный параметр legs_num.
1910	(internal) Неверный спот-инструмент.
1911	(internal) Неверный параметр measure_instrument.
1912	(internal) Неверный кросс-курс.
1913	(internal) Неверный фьючерс-инструмент.
1914	(internal) Неверное количество балансовых инструментов.
1915	(internal) Неверный dvp-инструмент.
1916	(internal) Неверный параметр price_coeff.
1980	Invalid stages in info field.
2100	Account does not belong to member_id.
2200	No permissions to submit trading instructions.
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.
2900	(internal) Риск-модуль не готов.
2901	(internal) Не задан mainClearAccount.
2902	(internal) Не указан клиент по умолчанию для данного торгового члена.

Code	Description
2903	(internal) Нет прав отправлять команды предпоставки.
2904	(internal) Нет прав на игнорирование приостановки.
2905	(internal) Нет прав на частичное исполнение заявки.
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	Instruction not found.
3004	Instrument trading suspended.
3100	TCA of maker and that of taker have no conversion bank indicator.
3900	(internal) Аукцион не готов.
3901	(internal) Указанное поручение уже существует (внутренняя ошибка при восстановлении).
3910	(internal) Для конверсионных сделок заданы неверные параметры.
3911	Incorrect te_id.
3999	(internal) Повторение заявки с типом SPECIAL для инструмента.
4000	ECN not available or no exchange available.
4001	The specified exchange not available.
4002	Order forcedly routed to an external exchange after declined by risk management at the local exchange.
4003	Client ID not registered at all the available exchanges
4004	Client ID not registered at the local exchange.
4005	Client ID not registered at external exchange.
4006	Order cannot be routed to any exchange.
4100	Order pending cancel.
4200	Invalid client for TCA registered at external exchange.
4201	Invalid TCA for external exchange.
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 heatbeat_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.

Code	Description
5300	Invalid topic.
5301	Subscription already activated.
5302	Subscription not activated.
5303	Requested data not available.
5304	Another request processing in progress.
5400	Reset_seq indicated, but seqnums cannot be reset.
5601	Both account and parties filled.
6900	(internal) Сервис расчёта позиций не готов.
7000	Order canceled before sending to ASTS.
7001	Order canceled as no answer received.
7900	(internal) ASTS_Adapter не готов.
7901	(internal) Несоответствие лотов на биржах.
8000	Invalid message type.
8001	Invalid clearing session.
8100	Incorrect active.
8101	Incorrect settlement account.
8102	Insufficient assets.
8103	Negative amount.
8104	Incorrect clearing center account.
8105	Incorrect currency.
8106	Unknown issue.
8200	Rollback rejected.
8201	Confirm error.
8300	Incorrect settlement depositary.
8301	TCA is already registered.
8302	TCA with flag forFixedFee is already registered.
8303	Incorrect clearing member.
8304	Invalid type of TCA for this clearing member.
8305	Incorrect trade member.
8306	Incorrect depository account.
8307	Incorrect depositary account for this clearing member.
8308	Incorrect depositary account for this TCA.
8309	Incorrect account for fee.
8310	Client is not found.
8311	Incorrect extra client.
8312	Main client and extra client coincide.
8313	Incorrect main client.
8314	Incorrect combination of main client and extra client.
8315	Settlement center is not found.
8316	Incorrect depositary account for this TCA.
8317	TCA is registered without registration of external codes.

Code	Description
8318	Incorrect TCA.
8319	Incorrect TCA for withdraw.
8320	Incorrect account for this TCA.
8321	Incorrect BIC.
8322	Incorrect SwiftBIC.
8323	Incorrect bank settlement account.
8324	Incorrect settle organization.
8325	Order already executed.
8326	Error while parsing details. Too many parts.
8327	Error while parsing details. Not enough parts.
8328	Unsupported client type.
8329	Client already exist.
8330	TradeMember legalCode and TrusteeINN are not equal for client(8*).
8331	TradeMember legalCode and TrusteeINN are equal for client(9*).
8332	Unknown client type .
8333	Unknown Action.
8334	legalTypeCode is null.
8335	Incorrect Client legal type.
8336	Incorrect Client legal type(allowable types: 0-7A).
8337	Active Default ClientGroup not found.
8338	Incorrect Client Type.
8339	Client already deleted.
8340	Login is not found.
8341	OTCCode cannot be assigned to this login.
8342	OTCCode is already deleted.
8343	OTCCode cannot be deleted. There are OperationLogins.
8344	OTCCode cannot be deleted. This code is default code for TCA.
8345	OTCCode cannot be deleted. This code is default code for ClearingMember.
8346	OTCCode is already created.
8347	Wrong Trade Member for TCA .
8348	Wrong Extra Code for deletion.
8349	Member is not found.
8350	OTCCode is not found.
8351	OTCCode is assigned to another member.
8352	Invalid Exchange code.
8353	ClearAccount is not top account.
8354	ClearAccount is top account.
8355	ClearAccount is not parent account of ClearAccount .
8356	TCA for Guarantee Fund cannot be deleted.
8357	TCA holds some assets or obligations.
8358	Primary TCA is not found.

Code	Description
8359	Primary TCA is used for Guarantee Fund.
8360	Impossible operation for TCA guarantee fund.
8361	ExchangeDetailsCode assigned to another member.
8362	ClearingMember is not active.
8363	TCA with ClearingMember is not active.
8364	BalanceInstrument is not found.
8365	Invalid format of INN.
8366	Invalid format of non resident unique code.
8367	Invalid object of trustee manager.
8368	Invalid format of sequence describing investment portfolio.
8369	Invalid country code.
8370	details exeed allowed length of 20 symbols.
8371	Invalid format of passport of russian citizen.
8372	Invalid format of birth certificate of russian citizen.
8373	Invalid symbol in client details.
8374	Invalid country code (should be '000').
8375	Invalid country code (should be three-digit code according to Russian country code classificator).
8376	Invalid country code (should be '-').
8377	Trustee's INN should coincide with member's INN.
8378	trustee is null .
8379	Trustee's INN should not coincide with member's INN.
8380	subBroker is null .
8381	INN of trustee or subbroker in field 4 should not coincide with member's INN.
8382	Invalid trustee format.
8383	This State Registration Number of Issue exceeds allowed length of 20 symbols.
8384	Error while parsing details: Wrong client type.
8385	Incorrect assets value
8386	Settle account is blocked
8387	Asset is blocked
8388	Withdraw denied
8389	Incorrect details: Incorrect typeOfInvestmentPortfolio.
8390	Found multiple SettlOrgs with this bic.
8391	Requisites of beneficiary are missing.
8392	Found multiple SettlOrgs with this swiftBic.
8393	Can not create login, login code is exceeding maximum length of 16 symbols due to long codeRts. Use shorter codeRts.
8394	Invalid login type .
8395	Changes via EDI of login with flag IS_CH_OPERATOR are forbidden.
8396	Invalid parameters in message: .
8397	Found client has invalid type.
8398	Login is owned by another member.

Code	Description
8399	Login is deleted.
8400	Login already has this status.
8401	Deleted login cannot be updated.
8402	Can not change mainClearAccount, login is not an OperationLogin.
8403	Can't create login, login code is exceeding maximum length of 16 symbols due to long memberCode, use shorter memberCode.
8404	Tag is not found.
8405	ClientGroup is not found.
8406	OTCCode is not found in the list.
8407	OperationLogin is not found in the list.
8408	Operation is forbidden for TCA BOARD.
8409	Trust TCA requires the separate accounting type.
8410	Wrong account type for TCA with forSoleClient.
8411	Flag forSoleClient is set. Default Client is required.
8412	Exchange with flag IB_EXTERNAL is not found.
8413	Exchange with flag IB_INTERNAL is not found.
8414	Account is not connected to IB
8415	ClientStringIdentifier is not found.
8416	Invalid EntityType.
8417	Client of such type is not allowed to conclude a treaty of Individual Investment Account.
8418	This account has unfulfilled obligations.
8419	TCA is not found in the list.
8420	Invalid instanceType.
8421	Client is deleted.
8422	Application is out of date
8423	Cannot change settlement currency less then 3 day before settlement
8424	Non-settlement day
8900	(internal) Контекстные параметры определены неверно.
8901	(internal) Неподходящее для команды состояние трансфера.
8902	(internal) Трансфер в завершительной стадии.
8903	(internal) Трансфер не отправлен.
8904	(internal) Incorrect MinVol.
8905	transfer's paySum exceeds Keeping Place limit
8906	Incorrect client type code
8907	Details already exist
8908	AccountBlockStringIdentifier is not found.
8909	TradeMember's IbrokerAccountStringExternal doesn't match found one.
8910	Entity code should be empty.
8911	Provided password is invalid.
8912	Legal is not found.
8913	Wrong legal code

Code	Description
8914	Immutable object cannot be changed.
9100	Invalid application type.
9101	Incorrect number of columns.
9102	Incorrect count of rows.
9103	Invalid firm code.
9104	Currency accounts cannot use cyrillic characters in field customPurpose.
9105	Document type '%s' is not supported
9200	Incorrect number of columns in a raw (%s).
9201	Required field missing.
9202	Field length exceeded.
9203	Inadmissible characters in (%s) field.
9204	Invalid (%s) value.
9205	Invalid date format.
9300	Insufficient assets for withdrawal.
9400	Reference to message of invalid type.
9401	Invalid reference number.
9402	Reference to incoing message.
9500	No permissions to manage account.
9501	Invalid ISIN.
9502	Xml file didn't pass xsd validation: %s.
9503	Xml file encoding is not specified'.
9504	Invalid SubAccount: %s.
9600	Not a XML document.
9601	Invalid XML structure: %s.
9700	Error while parsing record: %s.
9701	Error while creating asset transfer claims: %s.
9702	Error while parsing header: %s.
13000	Warning GrossLimit_IncludeOpenOrders_At_80_Percent.
13001	Warning GrossLimit_ExcludeOpenOrders_At_80_Percent.
13002	Gross Limit By Symbol Include Open Orders Exceeded.
13003	Gross Limit By Symbol Exclude Open Orders Exceeded.
13004	Loss Limit Exceeded.
13005	Quantity > Max Order Quantity Limit.
13006	Order Money > Max Order Money Limit.
13007	Max Interval Order Count Limit exceeded.
13008	Max Interval Order Money Limit exceeded.
13009	Sell Long Check Failed.
13010	Credit Limit Issue -> Generic error.
13011	Not Connected to Credit Limit System.
13012	Market Concentration Exposure Exceeded Equity.
13013	Net Market Exposure Exceeded Equity.

Code	Description
13014	Invalid Data -> Generally if there is no price available on the symbol.
13015	Gross Limit Include Open Orders Exceeded.
13016	Gross Limit Exclude Open Orders Exceeded.
13017	Account Stopped.
13018	Gross Max Money Limits Not Set.
13019	Liquidate Only Mode Active and Order is Establishing.
13020	Max Sell Quantity Exceeded.
13021	CUSTOMER NOT PROFILED -> Unknown customer.
13022	Gross Limit By Non-US Listed Symbol Include Open Orders Exceeded ValueLimit Index 0 Exceeded.
13023	Gross Limit By Symbol ValueLimit Index 1 Exceeded.
13024	Gross Limit By Symbol ValueLimit Index 2 Exceeded.
13025	Gross Limit By Symbol ValueLimit Index 3 Exceeded.
13026	Order Money > Max Market Order Money Limit.
13027	Order Layering Detected.
13028	Potential Wash Sale Detected.
13029	Max Order Count Per Side Per Symbol Exceeded.
13030	Customer Max Interval Order Money Limit 20X CustomerSymbolSideExchange_Interval_TotalMaxMoney Limit exceeded.
13031	High Odd Lot Order Frequency.
13032	No Market Order On IPO Symbol.
13033	OTCBB MAX Money Exceeded.
13034	PreMarket MAX Money PerSymbol PerSide Limit Exceeded.
13035	Aggressive Price Check Failed.
13036	Passive Price Check Failed.
13037	Cash Account ** Unable to enter order, contact trading desk to facilitate trade if possible.
13038	Risk Reducing Only Mode Active and Order does not reduce risk.
13039	Sell short exempt not allowed.
13040	Max Order Count Per Side Per Symbol Per Exchange Exceeded.
13041	Tick size validation failed.

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

Revision history

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