

04 OCTOBER 2022

**TWIME protocol specification for MOEX
Equities and Currency markets**



Table of contents

A. History of changes	4
1. Introduction.....	4
1.1. Document purpose	4
1.2. Targeted audience	4
1.3. High level description of SBE-GW	4
2. Presentation level	6
2.1. Data Types	6
2.1.1. Integer types.....	6
2.1.2. Decimal types.....	6
2.1.3. Timestamps	7
2.1.4. String types	7
2.1.5. Enums	7
2.2. Message Header.....	10
2.3. Message Schema.....	10
3. Session level.....	11
3.1. Supported message types	11
3.1.1. Establish (message id=6)	11
3.1.2. EstablishmentAck (message id=7).....	12
3.1.3. EstablishmentReject (message id=8)	12
3.1.4. Terminate (message id=4)	12
3.1.5. RetransmitRequest (message id=2)	13
3.1.6. Retransmission (message id=3)	13
3.1.7. Sequence (message id=1).....	14
3.1.8. ChangePassword (message id=9)	14

3.1.9. ChangePasswordAck (message id=10).....	14
3.1.10. ChangePasswordReject (message id=11)	14
3.1.11. SessionReject (message id=5)	15
3.1.12. BusinessMessageReject (message id=12)	Ошибка! Закладка не определена.
3.2. Session interaction scenarios	15
3.2.1. Session binding and termination	15
3.2.2. Session status monitoring	17
3.2.3. Message Numbering	17
3.2.4. Message retransmission request.....	18
3.2.5. Lost session recovery	18
3.2.6. Reset of sequence numbers	19
3.3. Disconnect on SBE-GW buffer overflow	19
3.4. Rejection of application-level messages	19
3.4.1. Rejecting request for adding order.....	20
3.5.2. Rejecting request for cancellation of order	20
3.5.3. Rejecting request for replacement of order.....	21
3.5.4. Rejecting request for mass cancellation of order	21
4. Application level.....	22
4.1. Supported messages	22
4.1.1. NewOrderSingle (message id=13).....	22
4.1.2. OrderCancelRequest (message id=14).....	23
4.1.3. OrderReplaceRequest (message id=16).....	24
4.1.4. OrderMassCancelRequest (message id=15)	25
4.1.5. ExecutionReport (message id=17).....	25
4.1.6. OrderMassCancelReport (message id=18)	28
4.2. Trading interaction scenarios	29

4.2.1. Adding orders.....	29
4.2.2. Order cancellation.....	32
4.2.3. Order mass cancellation.....	34
4.2.4. Replacing order.....	35
4.2.5. ClOrdID uniqueness.....	37
6. XML message schemas.....	38
7. Return codes list.....	38
8. Order cancellation reason codes	38

A. History of changes

Date	Version	Changes
20.03.2022	1.0	Initial document
24.08.2022	1.01	Added new value '8' to TimeInForce field indicating passive only GTC order.
09.09.2022	1.1	Finalized versions of documentation and message schemas before production launch planned for 10.10.2022: <ul style="list-style-type: none"> • Added UTCTimeStamp and UTCTimeOnly data types. These data types are applied to timestamp fields in message schemas • OrderQty and MaxFloor fields are assigne data type uInt64NULL in message schemas to match this document • Added clarifications to OrderQty and MaxFloor fields values

1. Introduction

1.1. Document purpose

This document contains specification of binary protocol for trading at Equities and Currency markets of Moscow Exchange. Specification includes descriptions of presentation, session, and application-level messages of protocol. This specification does not include administrative and technical aspects of network connectivity and security.

1.2. Targeted audience

This document is targeted at business analytics, system architects and software developers involved in projecting and development of software solutions for trading and Equities and Currency markets of MOEX.

1.3. High level description of SBE-GW

SBE-GW gateway is a server application that work at the Exchange side and provides ability to connect and trade at Equities and Currency markets via binary protocol. Gateway supports the following actions:

- Sending orders to trading system
- Canceling and modifying orders
- Getting results of processing orders
- Getting trades resulting from execution of orders

- Requesting and receiving missed messages.
- Marketdata publishing is not included in gateway functionality.

2. Presentation level

Presentation level protocol is based on FIX Simple Binary Encoding (<https://www.fixtrading.org/standards/sbe>); it is assumed that you are familiar with this protocol basics.

2.1. Data Types

Protocol uses the following data types. Please see <https://www.fixtrading.org/standards/sbe-online/> for primitive data types and their Null values.

2.1.1. Integer types

```
<type name="uInt8" primitiveType="uint8"/>
<type name="uInt8NULL" presence="optional" nullValue="255" primitiveType="uint8"/>
<type name="uInt16" primitiveType="uint16"/>
<type name="uInt16NULL" presence="optional" nullValue="65535" primitiveType="uint16"/>
<type name="uInt32" primitiveType="uint32"/>
<type name="uInt32NULL" presence="optional" nullValue="4294967295" primitiveType="uint32"/>
<type name="uInt64" primitiveType="uint64"/>
<type name="uInt64NULL" presence="optional" nullValue="18446744073709551615" primitiveType="uint64"/>
<type name="Int32" primitiveType="int32"/>
<type name="Int32NULL" presence="optional" nullValue="2147483647" primitiveType="int32"/>
<type name="Int64" primitiveType="int64"/>
<type name="Int64NULL" presence="optional" nullValue="9223372036854775807" primitiveType="int64"/>
```

2.1.2. Decimal types

```
<composite name="Decimal2NULL" description="Price type" semanticType="Price">
  <type name="mantissa" description="mantissa" presence="optional" nullValue="9223372036854775807"
primitiveType="int64"/>
  <type name="exponent" description="exponent" presence="constant" primitiveType="int8">-2</type>
</composite>
<composite name="Decimal9NULL" description="Price type" semanticType="Price">
  <type name="mantissa" description="mantissa" presence="optional" nullValue="9223372036854775807"
primitiveType="int64"/>
  <type name="exponent" description="exponent" presence="constant" primitiveType="int8">-9</type>
</composite>
```

2.1.3. Timestamps

```
<type name="UTCTimestamp" presence="optional" nullValue="18446744073709551615" primitiveType="uint64" description="UTC timestamp with nanoseconds precision" />
```

```
<type name="UTCTimeOnly" presence="optional" nullValue="18446744073709551615" primitiveType="uint64" description="Time of day with nanoseconds precision" /> </composite>
```

2.1.4. String types

Fixed length strings.

```
<type name="Char" primitiveType="char"/>
<type name="String4" length="4" primitiveType="char" presence="optional" nullValue=" " />
<type name="String5" length="5" primitiveType="char" presence="optional" nullValue=" " />
<type name="String6" length="6" primitiveType="char" presence="optional" nullValue=" " />
<type name="String10" length="10" primitiveType="char" presence="optional" nullValue=" " />
<type name="String12" length="12" primitiveType="char" presence="optional" nullValue=" " />
<type name="String20" length="20" primitiveType="char" presence="optional" nullValue=" " />
<type name="BoardID" length="4" primitiveType="char" presence="optional" nullValue=" " description="Board ID value in trading system SECB0ARD(4)"/>
<type name="SecurityID" length="12" primitiveType="char" presence="optional" nullValue=" " description="Instrument ID value in trading system SECCODE(12)"/>
```

2.1.5. Enums

```
<enum name="SessionRejectReasonEnum" encodingType="uint8">
  <validValue name="UserNameInvalid" >1</validValue>
  <validValue name="ValueIsIncorrect">5</validValue>
  <validValue name="SystemIsUnavailable" >100</validValue>
  <validValue name="ClOrdIdIsNotUnique" >101</validValue>
</enum>
```

```
<enum name="TerminationCodeEnum" encodingType="uint8">
  <validValue name="Finished" >0</validValue>
  <validValue name="UnspecifiedError">1</validValue>
  <validValue name="ReRequestOutOfBounds" >2</validValue>
  <validValue name="ReRequestInProgress" >3</validValue>
```


TWIME protocol specification for MOEX Equities and Currency markets

```
<validValue name="TooFastClient" >4</validValue>
<validValue name="TooSlowClient" >5</validValue>
<validValue name="MissedHeartbeat" >6</validValue>
<validValue name="InvalidMessage" >7</validValue>
<validValue name="TCPFailure" >8</validValue>
<validValue name="InvalidSequenceNumber">9</validValue>
<validValue name="ServerShutdown" >10</validValue>
</enum>

<enum name="BuySellEnum" encodingType="int8">
  <validValue name="Buy">1</validValue>
  <validValue name="Sell">2</validValue>
</enum>

<enum name="OrdTypeEnum" encodingType="char">
  <validValue name="Market">1</validValue>
  <validValue name="Limit">2</validValue>
  <validValue name="ClosingPeriod" description="Order of the closing period" >B</validValue>
</enum>

<enum name="SplitFlagEnum" encodingType="int8">
  <validValue name="Split" description="Price split allowed">0</validValue>
  <validValue name="One" description="One price only">1</validValue>
</enum>

<enum name="IMMCancelEnum" encodingType="int8">
  <validValue name="Day" description="Put in the queue">0</validValue>
  <validValue name="FoK" description="Fill or kill">4</validValue>
  <validValue name="IoC" description="Withdraw the balance">3</validValue>
  <validValue name="PO" description="Passive Only">8</validValue>
</enum>

<enum name="MMOrderEnum" encodingType="int8">
  <validValue name="MarketMaker">5</validValue>
</enum>
```

```
<enum name="OrderActivationTypeEnum" encodingType="char">
  <validValue name="ClosingAuction" description="Order to the closing auction">C</validValue>
  <validValue name="ActivationTime" description="Order with activation time">T</validValue>
</enum>

<enum name="ExecTypeEnum" encodingType="char">
  <validValue name="New" description="New order">0</validValue>
  <validValue name="Cancel" description="Order cancel">4</validValue>
  <validValue name="Replace" description="Order replace">5</validValue>
  <validValue name="PCancel" description="Pending Cancel">6</validValue>
  <validValue name="Trade" description="Trade">F</validValue>
  <validValue name="PMTrade" description="PreMatched Trade">L</validValue>
  <validValue name="PMTradeC" description="PreMatched Trade Cancel">H</validValue>
</enum>

<enum name="OrderStatusEnum" encodingType="int8">
  <validValue name="New" description="New">0</validValue>
  <validValue name="PFilled" description="Partially filled">1</validValue>
  <validValue name="Filled" description="Filled">2</validValue>
  <validValue name="Canceled" description="Canceled">4</validValue>
  <validValue name="PCanceled" description="Pending Cancel (i.e. result of Order Cancel Request)">6</validValue>
  <validValue name="Suspended" description="Suspended">9</validValue>
</enum>

<enum name="LiquidityTypeEnum" encodingType="char">
  <validValue name="Quote" description="Quoter">E</validValue>
  <validValue name="Internal" description="Internal liquidity">I</validValue>
</enum>

<enum name="LastLiquidityIndEnum" encodingType="int8">
  <validValue name="Add" description="Add liquidity">1</validValue>
  <validValue name="Removed" description="Removed liquidity">2</validValue>
</enum>
```

```
<enum name="TradeTypeEnum" encodingType="int8">  
  <validValue name="Regular">0</validValue>  
  <validValue name="Iceberg">1</validValue>  
</enum>
```

2.2. Message Header

SBE-GW messages contain standard header and message body. Message body fields follow message header fields. Message header name is "messageHeader". Message header fields are:

```
<composite name="messageHeader" description="Template ID and length of message root">  
  <type name="blockLength" primitiveType="uint16" />  
  <type name="templateId" primitiveType="uint16" />  
  <type name="schemaId" primitiveType="uint16" />  
  <type name="version" primitiveType="uint16" />  
</composite>
```

2.3. Message Schema

```
<?xml version="1.0" encoding="UTF-8"?>  
<sbe:messageSchema byteOrder="littleEndian" id="22343" package="sbe" version="0">  
</sbe:messageSchema>
```

Message schema attributes:

Attribute	Description	Value
id	Unique schema ID.	22343
version	Schema version.	0
package	Schema name or category	"sbe"
byteOrder	Byte order in fields.	"littleEndian"

3. Session level

Session level of protocol provides authentication of connection sides, guaranteed message delivery and processing order, controlling of connection state, and restoring session in case of failure. Session level protocol is based on FIXP (<https://www.fixtrading.org/standards/fixp>); it is assumed that you are familiar with this protocol basics.

3.1. Supported message types

Establish – Initiates session over TCP connection.

EstablishmentAck – Confirms session establishment over TCP connection.

EstablishmentReject – Informs about rejection to establish session over TCP connection.

Terminate – Terminates session.

RetransmitRequest – Request to retransmit messages starting from indicated sequence number.

Retransmission – Informs that retransmitted messages will follow this message.

Sequence – Used to set sequence number of next application-level message. Also used as Heartbeat message.

ChangePassword – request to change user password in trading system.

ChangePasswordAck – confirms successful change of password in trading system.

ChangePasswordReject – rejects an attempt to change password.

SessionReject – indicates incorrect message received.

BusinessMessageReject – rejection of application-level message.

Field descriptions for each message are given below. The following columns are used:

Tag – unique field ID;

Field – field name.

Required – indicates if nullValue is valid for a field:

Y – required field, nullValue is invalid;

N – optional field, nullValue is valid;

C – conditionally required, nullValue is invalid in certain cases.

Type – field data type;

Description – detail description of field.

3.1.1. Establish (message id=6)

Initiates session over TCP connection. Establish waiting time since TCP connection is 10 seconds. TCP connection is dropped after this waiting interval.

Tag	Field	Required	Type	Description
<Header>		Y		

Tag	Field	Required	Type	Description
52	SendingTime	Y	UTCTimestamp	Request sending time.
108	KeepaliveInterval	Y	uInt16	Heartbeat messages period in milliseconds. Valid values are between 1000 and 15000, inclusive.
553	Username	Y	String12	Username
554	Password	Y	String8	Password

3.1.2. EstablishmentAck (message id=7)

Confirms session establishment over TCP.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	Sending time of EstablishmentAck.
60	TimeStamp	Y	UTCTimestamp	Establishment message processing timestamp at trading system core.
5979	RequestTime	Y	UTCTimestamp	Request receiving time at SBE-GW server
789	NextSeqNo	Y	uInt64	Sequence number of next application-level message.
108	KeepaliveInterval	Y	uInt16	Heartbeat messages period in milliseconds.

3.1.3. EstablishmentReject (message id=8)

Rejection of session establishment.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time.
60	TimeStamp	Y	UTCTimestamp	Request message processing timestamp at trading system core
5979	RequestTime	Y	UTCTimestamp	Request receiving time at SBE-GW
30002	EstablishmentRejectCode	Y	uInt16	Rejection reason code. Full list of return codes is in file http://ftp.moex.com/pub/ClientsAPI/ASTS/docs/asts_message_list.csv

3.1.4. Terminate (message id=4)

Request to terminate session or confirmation of session termination.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	uInt64	This message sending time.

Tag	Field	Required	Type	Description
20210	TerminationCode	Y	TerminationCodeEnum	Termination reason code: "0" Finished – session terminated per other side request. "1" UnspecifiedError – internal error, contact support team "2" ReRequestOutOfBounds – Reply to RetransmitRequest. Requested messages are not available. "3" ReRequestInProgress - RetransmitRequest is in progress. "4" TooFastClient – Too high message rate from client. "5" TooSlowClient – Client is too slow in reading messages from TCP socket. Session terminated to avoid TCP buffer overflow. "6" MissedHeartbeat – No messages received for longer than KeepaliveInterval. "7" InvalidMessage – Incorrect or unknown message. "8" TCPFailure – Error at transport level "9" InvalidSequenceNumber – Reply to Sequence message with incorrect SequenceNumber. "10" ServerShutdown – Server shutdown in progress.

3.1.5. RetransmitRequest (message id=2)

Request to resend missed messages.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time.
7	BeginSeqNo	Y	uInt64	Sequence number of the first requested message.
30003	Count	Y	uInt32	Number of requested messages.

3.1.6. Retransmission (message id=3)

Retransmission message is used to notify client that the next messages counted by Count value are retransmitted messages in response to a RetransmitRequest (message id=2).

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time.
30001	RequestTimestamp	Y	UTCTimestamp	RetransmitRequest message sending time
789	NextSeqNo	Y	UInt64	Sequence number of the first retransmitted message.

Tag	Field	Required	Type	Description
30003	Count	Y	UInt32	Number of messages.

3.1.7. Sequence (message id=1)

Used only as Heartbeat message. Client must send Sequence messages with nullValue in the NextSeqNo field. SBE-GW outgoing Sequence message contains next sequence number of application-level message in the NextSeqNo field.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
789	NextSeqNo	N	uInt64NULL	Sequence number of next application-level message.

3.1.8. ChangePassword (message id=9)

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	uInt64	This message sending time
554	Password	Y	String10	Current Password
925	NewPassword	Y	String10	New Password

3.1.9. ChangePasswordAck (message id=10)

Tag	Field	Required	Type	Type
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
60	Timestamp	Y	UTCTimestamp	Timestamp of message processing in the trading system core.
5979	RequestTime	Y	UTCTimestamp	Timestamp of receiving request by SBE-GW
554	Password	Y	String10	Password

3.1.10. ChangePasswordReject (message id=11)

Tag	Field	Required	Type	Type
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
60	Timestamp	Y	UTCTimestamp	Timestamp of message processing in the trading system core.
5979	RequestTime	Y	UTCTimestamp	Timestamp of receiving request by SBE-GW
103	RejReason	Y	uInt16	Rejection reason code

3.1.11. SessionReject (message id=5)

SessionReject message is sent by SBE-GW as a reply to invalid incoming message.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
11	ClOrdID	Y	uInt64	ClOrdID field of incoming message
371	RefTagID	N	uInt32	Incorrect field ID.
373	SessionRejectReason	Y	SessionRejectReasonEnum	Message rejection code: "1" UserNameInvalid – Invalid username "5" ValueIsIncorrect – Incorrect field value "100" SystemIsUnavailable – Trading system is not available "101" ClOrdIdIsNotUnique – ClOrdID field in incoming message is not unique. "99" Other - Other.

3.2. Session interaction scenarios

3.2.1. Session binding and termination

To bind a session to the TCP connection, the client side should send the message 'Establish' first. If the message 'Establish' was correct, and the user has been properly authorized, the system replies with the message 'EstablishmentAck', confirming that the session has been successfully bound; otherwise (incorrect message 'Establish' and/or non-authorized user) the system will reply with the message 'EstablishmentReject' containing the rejection reason details.

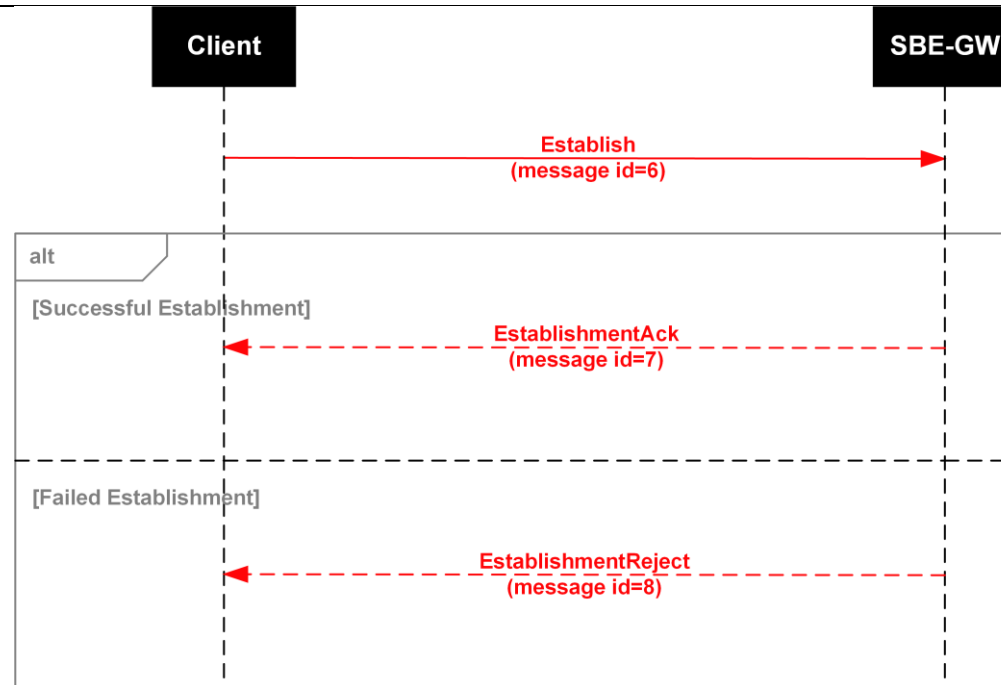


Figure 1. Diagram. Session binding

To terminate the session, the client side should send the message 'Terminate' and then wait the response message 'Terminate' from SBE-GW to be received, with 'TerminationCode=0'.

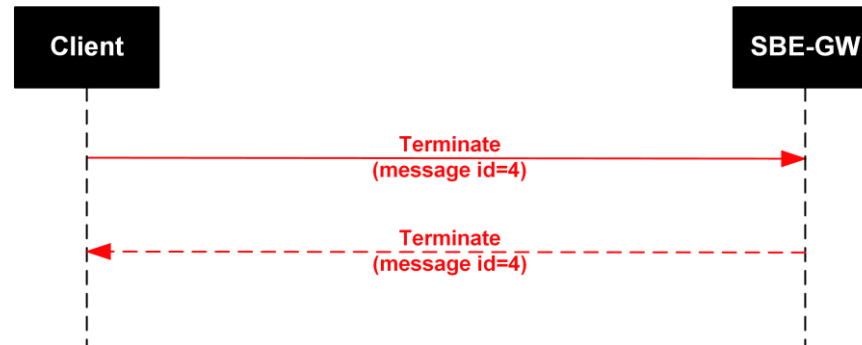


Figure 2. Diagram. Session termination

Please note that a TCP connection should not be established by the client side with the same IP address within a time interval less than 1 second after the last connection has been terminated. Otherwise, the TCP connection will be rejected.

Trying to establish two or more simultaneous TCP connections, with sending equal TWIME login IDs ('Credentials' in message 'Establish'), may cause each of the connections to terminate with an error. After that, message 'EstablishmentReject' containing 'EstablishmentRejectCode=1 (AlreadyEstablished)' will be sent by SBE-GW to each of the sessions.

3.2.2. Session status monitoring

To monitor the session status, both sides (client and SBE-GW) should send the messages 'Heartbeat' (here: the message 'Sequence') to each other with specified frequency. The frequency value is specified within fields 'KeepaliveInterval' in both client side message 'Establish' and server side message 'EstablishmentAck'. SBE-GW guarantees to send messages not less frequently than once per interval. However, the messages sent must not necessary be of the 'Heartbeat' type. Intervals are calculated on a fixed grid, the size of which is set via 'KeepaliveInterval'. If SBE-GW did not send messages other than 'Heartbeat' during this interval, it will send a 'Heartbeat' message at the very end of the interval. If there are other messages in the interval, the 'Heartbeat' message is not sent. Thus, there is at least one message in each interval, but the interval between messages can be more than 'KeepaliveInterval'.

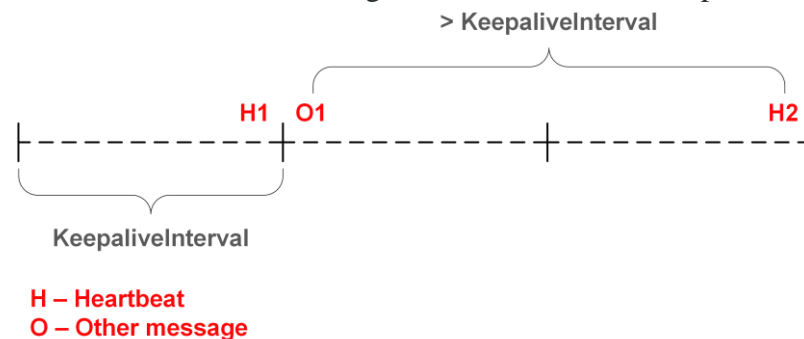


Figure 3. Diagram. Sending 'Heartbeat' messages from SBE-GW

If there is no message sent by client on the specified time interval (KeepaliveInterval) then SBE-GW server disconnect client with Cancel on Disconnect function. Disconnection is processed in one or two intervals plus transmission time. User should take into account transmission time when sending Heartbeat messages.

The client side should not send more than 3 'Heartbeat' messages per second to SBE-GW. The fourth 'Heartbeat' message per second sent will cause a connection termination, when SBE-GW will send the message 'Terminate' containing the termination reason 'TooFastClient' to the client side. To avoid breaking the connection with the reason 'TooFastClient', it is recommended to set the interval between consecutive 'Heartbeat' messages for at least 1000 milliseconds.

3.2.3. Message Numbering

For consecutive numbering, the client side has to support the message counter in order for messages sent from SBE-GW, where the very first message EstablishmentAck sent from SBE-GW after establishing a connection contains the initial number of incoming messages. The next application layer messages will increase the counter value by 1, while the session layer messages so not affect the counter.

3.2.4. Message retransmission request

In case of a missing message, the client side can request for the message retransmission procedure by sending the message 'RetransmitRequest' containing sequence number of the first message along with the number of messages to retransmit. After the request has been confirmed, the missing messages will be retransmitted.

No new messages are sent to client from SBE-GW during processing missing messages. All new messages will be dispatched right upon the command 'RetransmitRequest' completion.

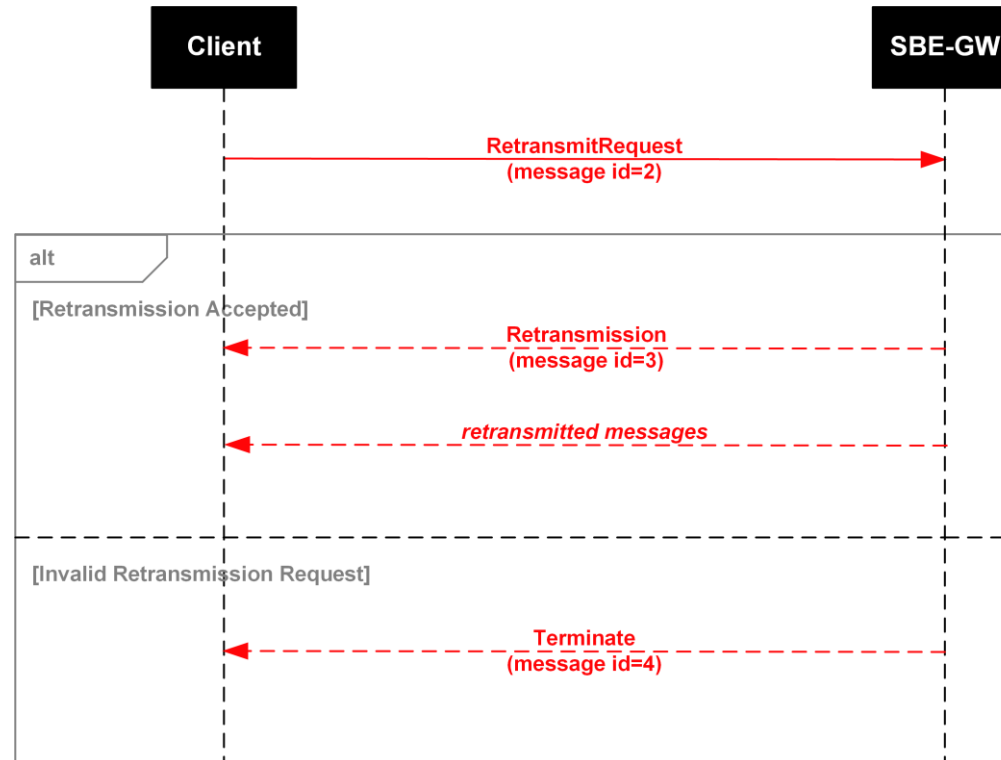


Figure 4. Diagram. Message retransmission request

3.2.5. Lost session recovery

To recover session after a crash, the client should compare the value in the field 'NextSeqNo' of the message 'EstablishmentAck' with that of the incoming message counter. If the value in the field 'NextSeqNo' is greater than that of the incoming message counter, the client side should request retransmission of the missing messages using the command 'RetransmitRequest'. SBE-GW then will send an application layer message in reply, after one second from receiving command 'RetransmitRequest'. A client can request no more than 1000 messages per a single 'RetransmitRequest'

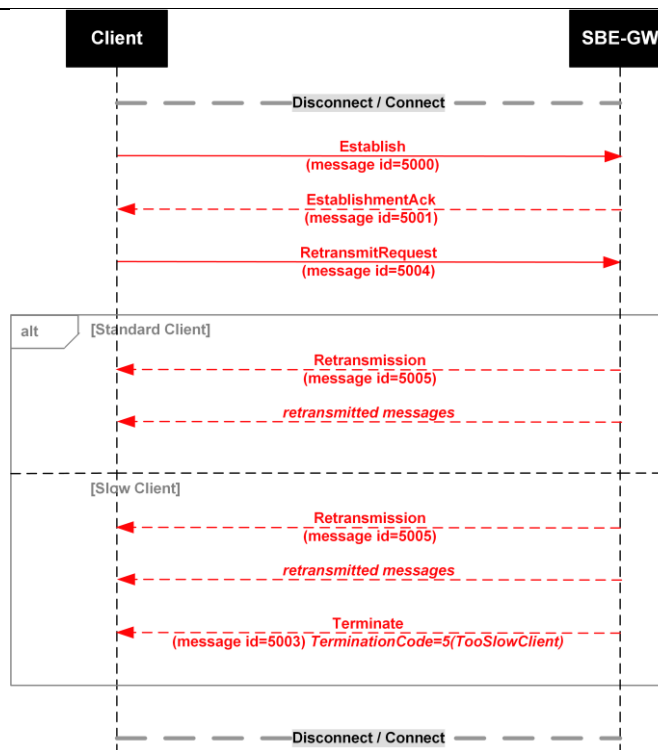


Figure 5. Diagram. Lost session recovery

3.2.6. Reset of sequence numbers

SBE-GW stops with the trading system stop at the end of trading day. SBE-GW resets sequence numbers every new trading day at application start. Server messages from previous day are not available.

3.3. Disconnect on SBE-GW buffer overflow

If a client cannot retrieve messages from TCP socket for some reason, this client will be disconnected via the message 'Terminate' with TerminationCode=5 as soon as the SBE-GW buffer is overflowed.

3.4. Rejection of application-level messages

If application-level requests are rejected, SBE-GW sends a 'BusinessMessageReject' session-level message to the client. A message may be sent in the following cases:

- rejection of requests to add, delete or move orders

- rejection of requests for mass cancellation of orders

Session-layer messages 'BusinessMessageReject' are not available in the recovery service described in the section [3.2.5. Lost session recovery](#).

3.4.1. Rejecting request for adding order

The client side sends a new order into the trading system (message 'NewOrderSingle'). The trading system rejects it with sending out message 'BusinessMessageReject'.

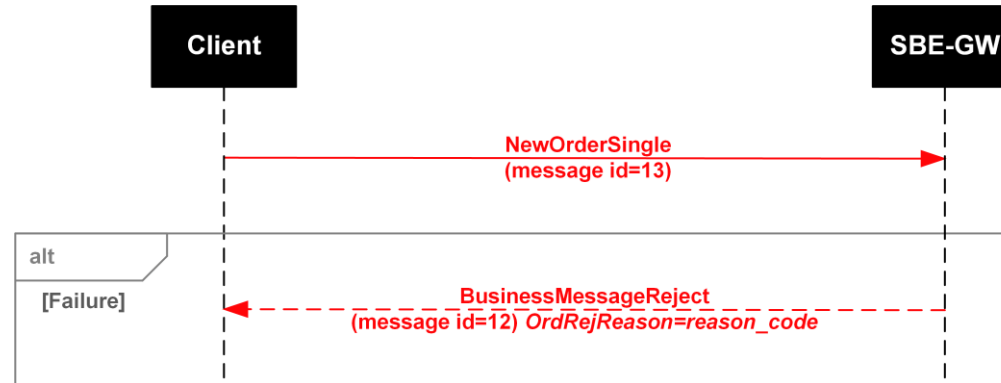


Figure 6. Diagram. Rejecting request for adding order

3.5.2. Rejecting request for cancellation of order

The client side sends a request to cancel the order (message 'OrderCancelRequest'). The trading system rejects the request with sending out message 'BusinessMessageReject'.

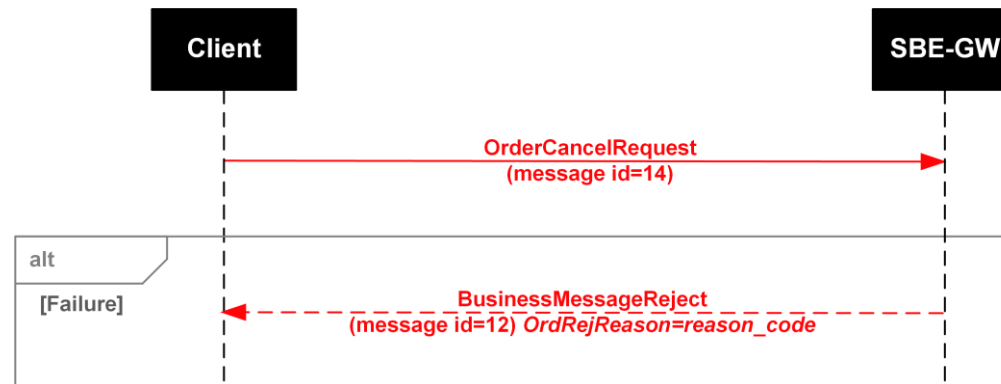


Figure 7. Diagram. Rejecting request for cancellation of order

3.5.3. Rejecting request for replacement of order

The client side sends a request to replace the order (message 'OrderReplaceRequest'). The trading system rejects the request with sending out message 'BusinessMessageReject'.

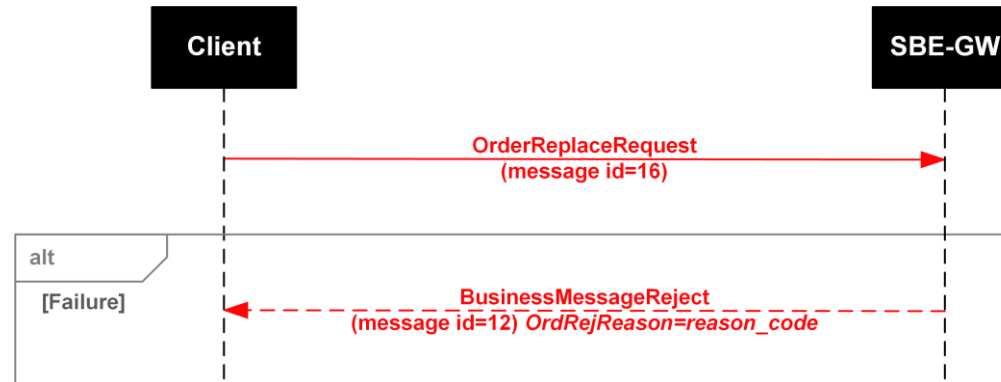


Figure 8. Diagram. Rejecting request for replacement of order

3.5.4. Rejecting request for mass cancellation of order

The client side sends a request for mass cancellation of orders (message 'OrderMassCancelRequest'). The trading system rejects the request with sending out message 'BusinessMessageReject'.

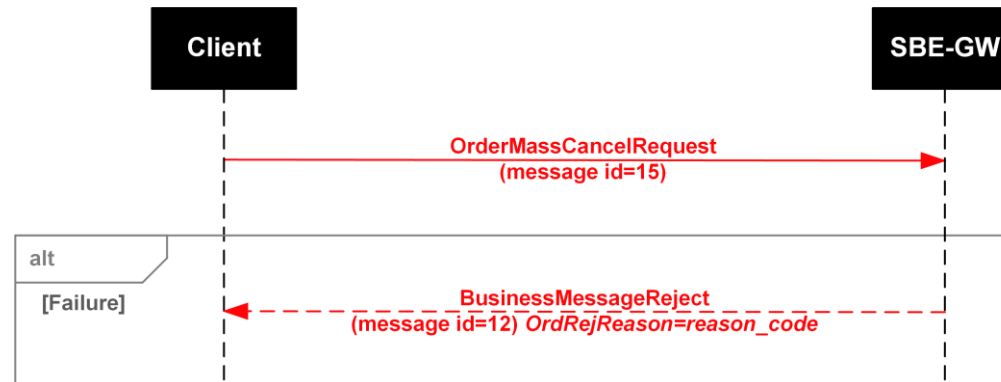


Figure 9. Diagram. Rejecting request for mass cancellation of order

4. Application level

The application layer protocol is based on the standard FIX protocol ver. 5.0 SP2 (<https://www.fixtrading.org/standards/fix-5-0-sp-2>); it is expected that users have already got some information about this protocol.

4.1. Supported messages

- **NewOrderSingle** – Adding a new order. The message is sent from client side to server side.
- **OrderCancelRequest** – Order cancellation. The message is sent from client side to server side.
- **OrderReplaceRequest** – Order replacement. The message is sent from client side to server side.
- **OrderMassCancelRequest** – Mass cancellation of orders. The message is sent from client side to server side.
- **OrderMassCancelReport** – Reply to successfully processed OrderMassCancelRequest message
- **ExecutionReport** – Report adding, cancelling, changing orders, and trades. The message is sent from server side to client side.

4.1.1. NewOrderSingle (message id=13)

Placing new order.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
11	ClOrdID	Y	uInt64	Unique identifier for Order as assigned by the client. Uniqueness must be guaranteed within a single trading day.
168	EffectiveTime	C	UTCTimeOnly	Order activation time. Null for immediate activation
44	Price	C	Decimal9NULL	Price. Null for market orders.
38	OrderQty	C	uInt64NULL	Number of lots in the order. Must be Null, if order volume is defined via CashOrderQty(152) field.
111	MaxFloor	C	uInt64NULL	Visible quantity of iceberg order. Expressed in lots. Null, if hidden part is zero. Allowed values of MaxFloor and OrderQty fields are defined in Trading Rules.
152	CashOrderQty	C	Decimal2NULL	Order volume in the units of settlement currency. Must be Null, if order quantity is specified in OrderQty(38) field.
54	Side	Y	BuySellEnum	Order direction: "1" – Buy "2" – Sell

TWIME protocol specification for MOEX Equities and Currency markets

Tag	Field	Required	Type	Description
40	OrdType	Y	OrdTypeEnum	Order Type: "1" – market "2" – limit "B" – order for closing period
1090	MaxPriceLevels	Y	SplitFlagEnum	"0" – execution by multiple price levels "1" – execution at single price level only
59	TimeInForce	Y	IMMCancelEnum	Time in force: "0" Day – active until canceled at the trading session end "3" IOC – Immediate or cancel "4" FOK – Fill-or-Kill "9" PO – limit order is accepted only if it doesnot result to trade upon registration
529	OrderRestriction	C	MMOrderEnum	"5" – order placed by market maker (equities market only). Null – regular order
5202	TradeThruTime	C	OrderActivationTypeEnum	"C" – an order to be activated in the closing auction "T" – an order with activation time Null – regular order
10526	LiquidityType	C	LiquidityTypeEnum	"E" – quote of external liquidity provider (equities market only) "I" – execution allowed only by internal liquidity orders (equities market only) Null – execution allowed by all matching orders
1	Account	Y	String12	Trading account
526	SecondaryClOrdID	N	String12	To be transferred to the EXTREF field in the Orders table of trading system
448	ClientCode	N	String12	Client code
20336	Board	Y	BoardID	Trading board ID, SECBOARD
55	Symbol	Y	SecurityID	Instrument code
10600	Brokerref	N	String20	To be transferred to the BROKERREF field in the Orders table of trading system

4.1.2. OrderCancelRequest (message id=14)

Request to cancel existing order.

Tag	Field	Required	Type	Description
-----	-------	----------	------	-------------

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
11	ClOrdID	Y	uInt64	Unique identifier for OrderCancelRequest as assigned by the client. Uniqueness must be guaranteed within a single trading day.
41	OrigClOrdID	C	uInt64NULL	ClOrdID value of an order to be canceled. Conditionally required if the OrderID is not specified.
37	OrderID	C	uInt64NULL	Exchange number of an order to be canceled. Conditionally required if the OrigClOrdID field is not specified. If this field is not Null, then OrigClOrdID value is ignored.

4.1.3. OrderReplaceRequest (message id=16)

Request to change order's quantity or price.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
11	ClOrdID	Y	uInt64	Unique identifier for OrderReplaceRequest as assigned by the client. Uniqueness must be guaranteed within a single trading day.
37	OrderID	Y	uInt64NULL	ClOrdID value of an order to be replaced. Conditionally required if the OrderID is not specified.
41	OrigClOrdID	C	uInt64NULL	Exchange number of an order to be replaced. Conditionally required if the OrigClOrdID field is not specified. If this field is not Null, then OrigClOrdID value is ignored.
44	Price	N	Decimal9NULL	New price. Null value has a meaning of keeping price unchanged
38	OrderQty	Y	uInt64NULL	New quantity. Null value has a meaning of keeping quantity unchanged
54	Side	Y	BuySellEnum	Order side. Must match order side of original order.
1	Account	Y	String12	Trading account. Must match the value of original order.
526	SecondaryClOrdID	N	String12	To be transferred to the EXTREF field in the Orders table of trading system
448	ClientCode	N	String12	Client code. Must match the value of original order.
20336	Board	Y	BoardID	Trading board ID, SECBOARD. Must match the value of original order.
55	Symbol	Y	SecurityID	Instrument code. Must match the value of original order.
10600	Brokerref	N	String20	To be transferred to the BROKERREF field in the Orders table of trading system

4.1.4. OrderMassCancelRequest (message id=15)

Request to cancel multiple orders that have field values matching not-Null values in this request.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
11	ClOrdID	Y	uInt64	Unique identifier for OrderMassCancelRequest as assigned by the client. Uniqueness must be guaranteed within a single trading day.
54	Side	N	BuySellEnum	Direction of orders to be canceled: "1" – buy. "2" – sell. NULL – any direction.
1	Account	N	String12	Trading account
526	SecondaryClOrdID	N	String12	EXTREF field value of orders to be canceled
448	ClientCode	N	String12	Client code value of orders to be canceled
20336	Board	N	BoardID	BoardID values
55	Symbol	N	SecurityID	Symbol value of orders to be canceled

4.1.5. ExecutionReport (message id=17)

The ExecutionReport message is used to:

- Confirm the receipt of an order;
- Confirm changes to an existing order (i.e. accept cancel request);
- Relay order status information;
- Relay fill information on working orders;
- Reject orders;
- Relay information on pre-matched trades.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
60	Timestamp	Y	UTCTimestamp	Timestamp of message processing in the trading system core.
5979	RequestTime	C	UTCTimestamp	Timestamp of receiving request by SBE-GW. Required for replies to NewOrderSingle, OrderReplaceRequest, OrderCancelRequest messages. Null for messages with trade or notifications of cancelling order by other session or trading system.

TWIME protocol specification for MOEX Equities and Currency markets

Tag	Field	Required	Type	Description
11	ClOrdID	Y	uInt64	Client ID of an order: <ul style="list-style-type: none"> • For new order – ClOrdID fields value of NewOrderSingle or OrderReplaceRequest message • For reply to the sender of OrderCancelRequest message – ClOrdID value of OrderCancelRequest • For notification of cancelling order by other session or trading system – ClOrdID field value in the message that resulted in order registration
168	EffectiveTime	C	UTCTimeOnly	Order activation time. Null, if an order is activated immediately
37	OrderID	Y	uInt64NULL	Exchange order ID.
9945	OrigOrderID	C	uInt64NULL	Exchange order ID of an order that was replaced via OrderReplaceRequest
278	MDEntryID	Y	uInt64NULL	Publish order ID in FAST or SIMBA marketdata feed.
41	OrigClOrdID	C	uInt64NULL	OrigClOrdID field of OrderCancelRequest or OrderReplaceRequest. Required, if specified in requests. NULL in other cases.
880	TrdMatchID	C	uInt64NULL	Exchange Trade ID
44	Price	C	Decimal9NULL	Order price.
38	OrderQty	C	Int64Null	Number of lots in the order. Null, if order volume is defined via CashOrderQty(152) field.
111	MaxFloor	C	Int64Null	Visible quantity of iceberg order? Expressed in lots. Null, if hidden part is zero.
152	CashOrderQty	C	Decimal2NULL	Order volume in the units of settlement currency. Null, if order quantity is specified in OrderQty(38) field.
31	LastPx	C	Decimal9NULL	Price of reported trade
32	LastQty	C	uInt64NULL	Number of lots in reported trade
151	LeavesQty	Y	uInt64NULL	Number of lots available for further execution
84	CxlQty	C	uInt64NULL	Canceled number of lots, for ExecType=4
18181	PreMatchedCumQty	C	uInt64NULL	Number of lots in a state of waiting for PreMatch trades confirmation (only for equities market).
34	MsgSeqNum	Y	uInt32	Message sequence number
9947	OrdCancelReason	C	uInt8NULL	Order cancellation reason code. See appendix.
150	ExecType	Y	ExecTypeEnum	Execution type reported: "0" – New order

TWIME protocol specification for MOEX Equities and Currency markets

Tag	Field	Required	Type	Description
				"4" – Order Canceled "5" – Order Replaced "6" – Pending cancellation of order "F" – Trade "L" – Pre-matched trade (equities market only) "H" – Pre-matched trade canceled (equities market only).
39	OrdStatus	Y	OrderStatusEnum	"0" – New "4" – Canceled "2" – Filled "6" – Pending cancellation "1" – Partially filled "9" – Waiting for activation event
6636	StipulationValue	C	TradeTypeEnum	Trade type: "0" – regular trade "4" – a trade with this iceberg order
54	Side	Y	BuySellEnum	Order side: "1" – buy "2" – sell
40	OrdType	Y	OrdTypeEnum	Order type: "1" – market "2" – limit "B" – order in closing period
1090	MaxPriceLevels	Y	SplitFlagEnum	"0" – execution allowed by multiple price levels "1" – execution allowed by single price levels
59	TimeInForce	Y	IMMCancelEnum	Time in force: "0" Day – active until canceled at the trading session end "3" IOC – Immediate or cancel "4" FOK – Fill-or-Kill "8" PO – passive only
529	OrderRestriction	C	MMOrderEnum	"5" – order placed by market maker (equities market only). Null – regular order
5202	TradeThruTime	C	OrderActivationTypeEnum	"C" – an order to be activated in the closing auction "T" – an order with activation time

Tag	Field	Required	Type	Description
				Null – regular order
10526	LiquidityType	C	LiquidityTypeEnum	"E" – quote of external liquidity provider (equities market only) "I" – execution allowed only by internal liquidity orders (equities market only) Null – execution allowed by all matching orders
851	LastLiquidityInd	C	LastLiquidityIndEnum	"1" – added liquidity "2" – removed liquidity
1	Account	Y	String12	Trading account
526	SecondaryClOrdID	N	String12	To be transferred to the EXTREF field in the Orders table of trading system
448	ClientCode	N	String12	Client code
20336	Board	Y	BoardID	Trading board ID, SECBOARD
55	Symbol	Y	SecurityID	Instrument code
10600	Brokerref	N	String20	To be transferred to the BROKERREF field in the Orders table of trading system

4.1.6. OrderMassCancelReport (message id=18)

The OrderMassCancelReport message is used to acknowledge an OrderMassCancelRequest message.

Tag	Field	Required	Type	Description
	<Header>	Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
60	Timestamp	Y	UTCTimestamp	Timestamp of message processing in the trading system core.
5979	RequestTime	C	UTCTimestamp	Timestamp of receiving request by SBE-GW.
11	ClOrdID	Y	uInt64	Unique identifier for MassCancelRequest as assigned by the client. Uniqueness must be guaranteed within a single trading day
533	TotalAffectedOrders	N	uInt64	Number of canceled orders.
34	MsgSeqNum	Y	uInt32	Message sequence number

4.1.7. BusinessMessageReject (message id=12)

BusinessMessageReject is sent for rejection of application-level message.

Tag	Field	Required	Type	Description
<Header>		Y		
52	SendingTime	Y	UTCTimestamp	This message sending time
60	Timestamp	Y	UTCTimestamp	Timestamp of message processing in the trading system core.
5979	RequestTime	Y	UTCTimestamp	Timestamp of receiving request by SBE-GW
11	ClOrdID	Y	uInt64	ClOrdID field of incoming message.
34	MsgSeqNum	Y	uInt32	Sequence number
103	OrdRejReason	Y	uInt16	Message rejection reason code. See reason codes in http://ftp.moex.com/pub/ClientsAPI/ASTS/docs/asts_message_list.csv file.

4.2. Trading interaction scenarios

4.2.1. Adding orders

4.2.1.1. Adding a new order

Scenario 1. Adding an order with 'TimeInForce'=0(Day). The client side sends a new order into the trading system (message 'NewOrderSingle'). The trading system either confirms the receiving with sending message 'ExecutionReport' in reply, or rejects it with sending out message 'BusinessMessageReject'.

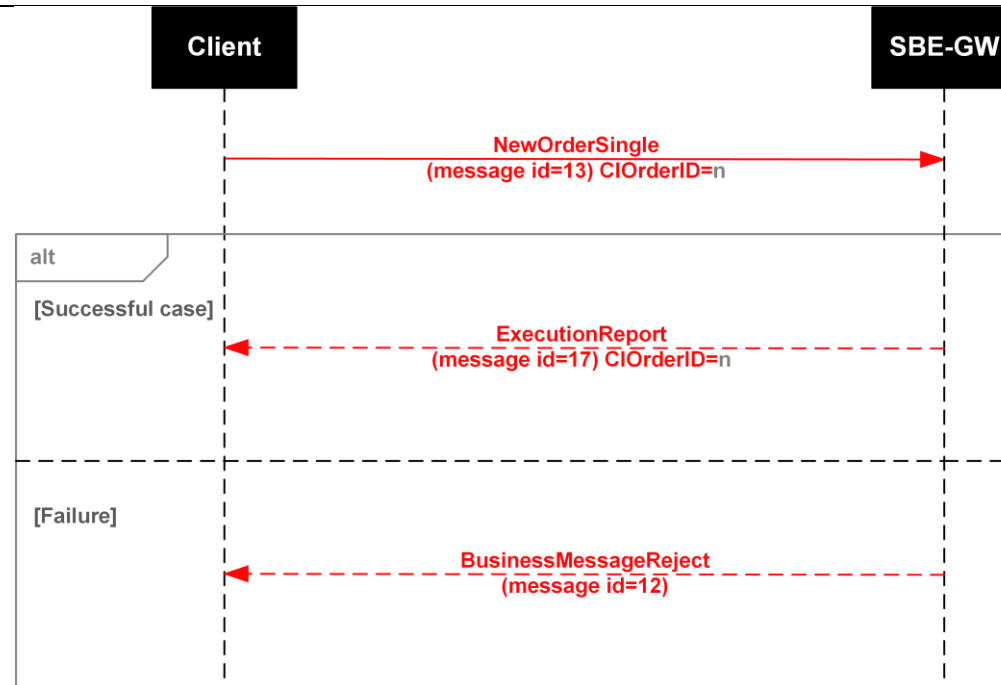


Figure 10. Diagram. Adding an order with *TimeInForce=0(Day)*

Scenario 2. Adding an order with 'TimeInForce'=3(IOC). An order with 'TimeInForce'=3(IOC) can be completely filled, or partially filled with its remaining part cancelled, or fully cancelled and removed from the system.

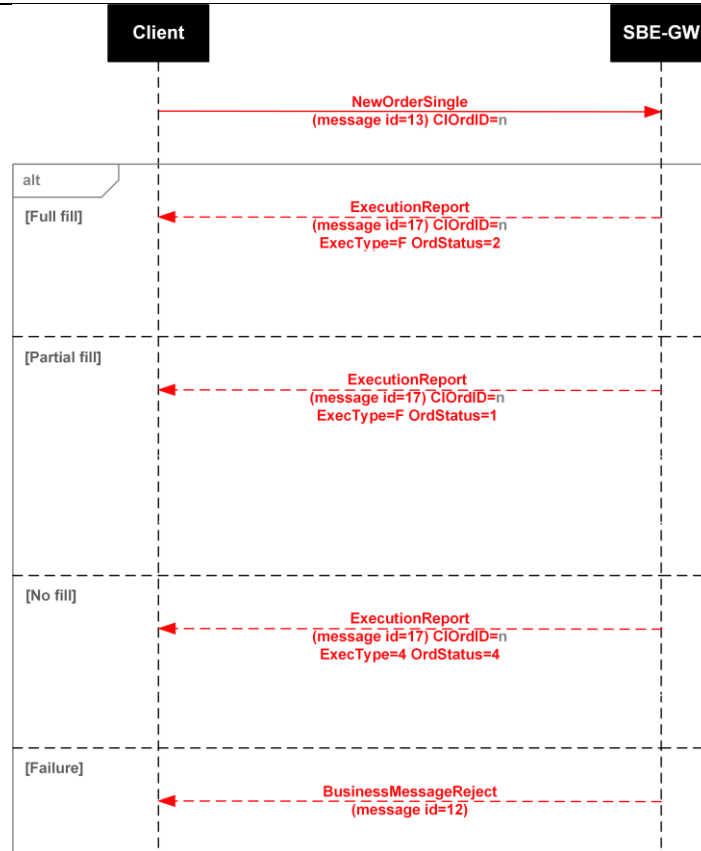


Figure 11. Diagram Adding an order with TimeInForce=3(IOC)

Scenario 3. Adding an order with 'TimeInForce'=4(FOK). An order with 'TimeInForce'=4(FOK) can be either filled completely or rejected by the system.

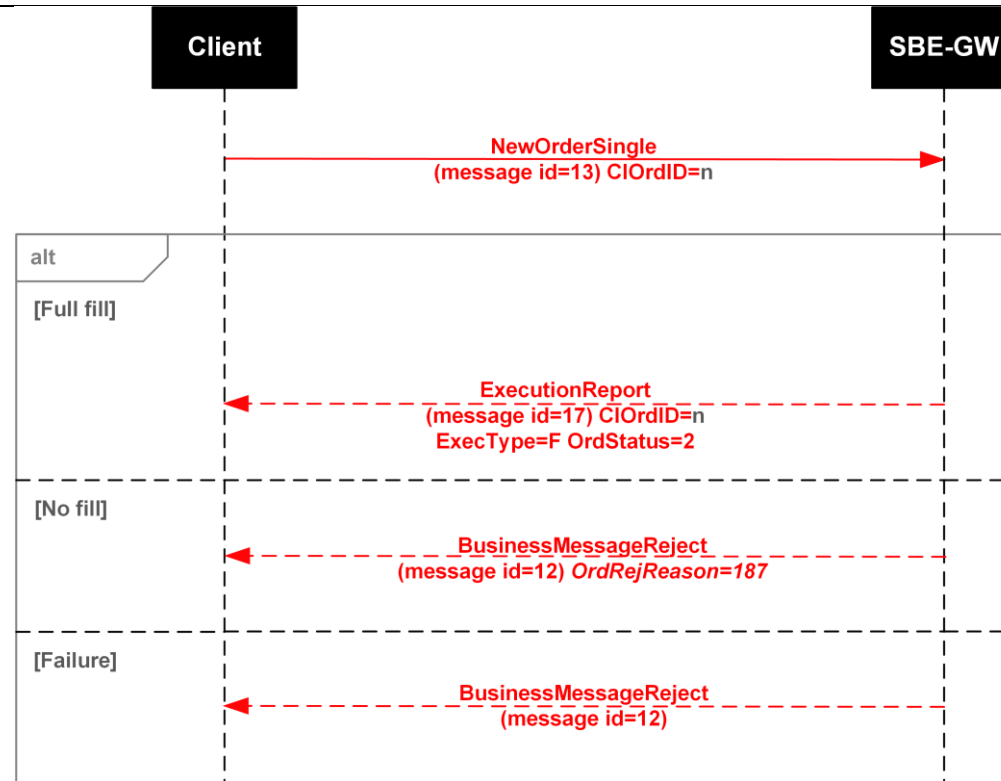


Figure 12. Diagram. Adding an order with TimeInForce=4(FOK)

4.2.2. Order cancellation

4.2.2.1. Canceling own order

Client can cancel previously created order via OrderCancelRequest specifying either ClOrdID of an order or Exchange OrderID. If specified, Exchange OrderID takes precedence. OrderCancelRequest is rejected if an order is filled or canceled.

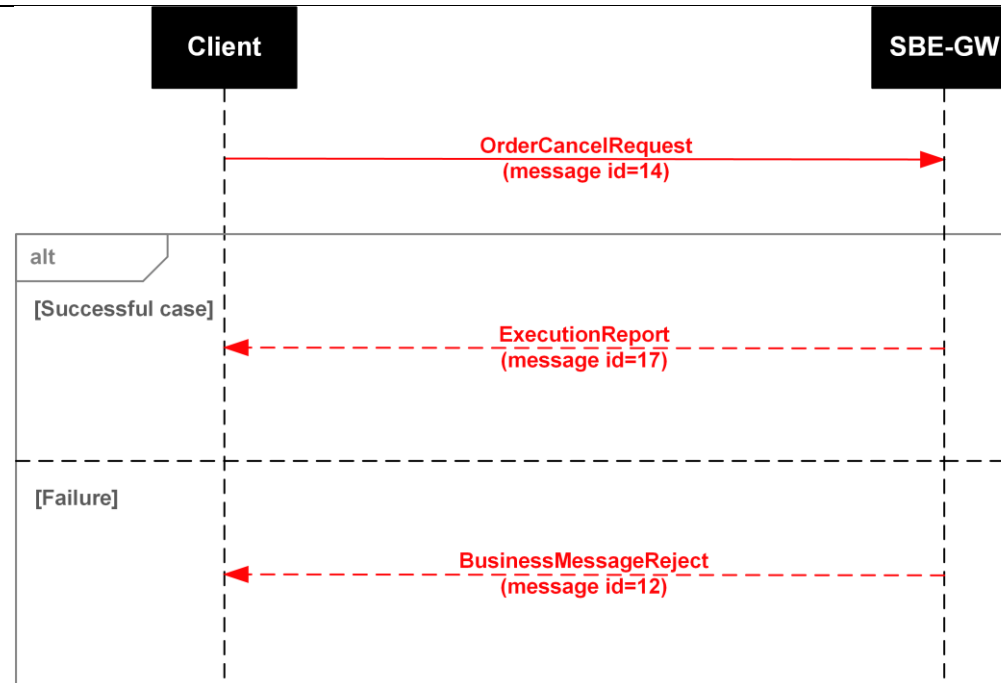


Figure 13. Diagram. Client order cancellation (by client)

4.2.2.2. Canceling order by other user (broker)

Scenario. A Brokerage Firm cancels a client order using its own login (message OrderCancelRequest). Once an order has been canceled, SBE-GW confirms the cancellation by sending the message ExecutionReport to BF; order’s owner will receive the unsolicited message ExecutionReport where the field ClOrdID contains value of ClOrdID field in a message that resulted in order registration.

The cancellation request may also be rejected by the system (message BusinessMessageReject).

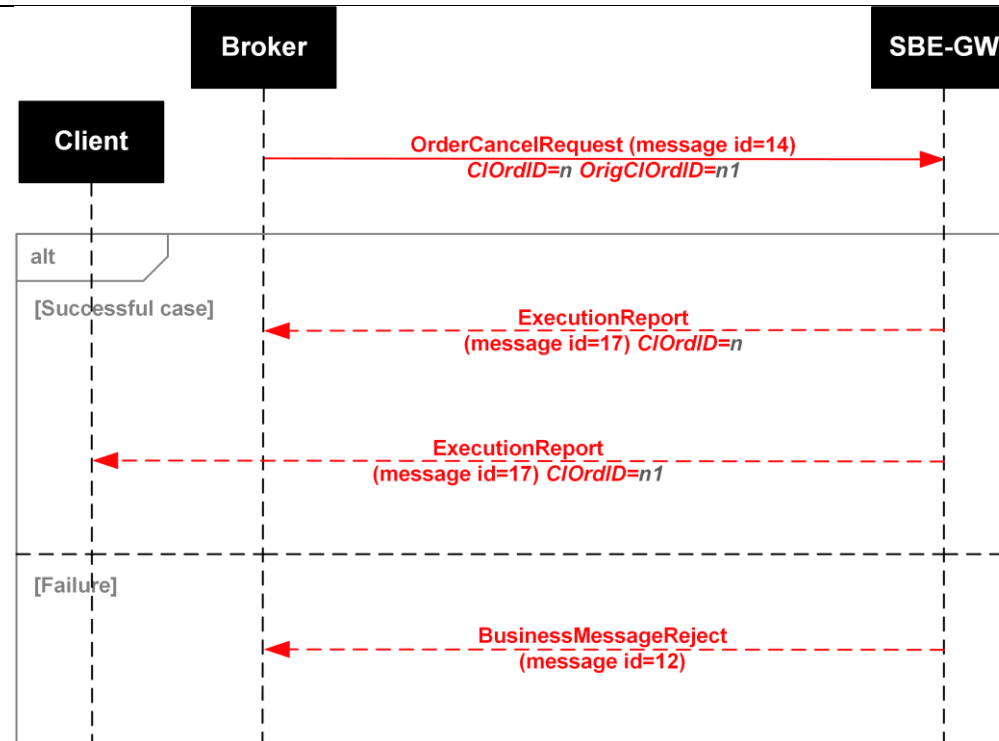


Figure 14. Diagram. Client order cancellation (by Brokerage Firm)

4.2.3. Order mass cancellation

4.2.3.1. Order mass cancellation by client

The client side can request mass cancellation of orders via the message OrderMassCancelRequest. The server side sends the message OrderMassCancelReport after processing the request.

Order owners receive unsolicited messages ExecutionReport, where the field COrdID contains value of COrdID field in a message that resulted in order registration.

The OrderMassCancelRequest may also be rejected by the system (message BusinessMessageReject).

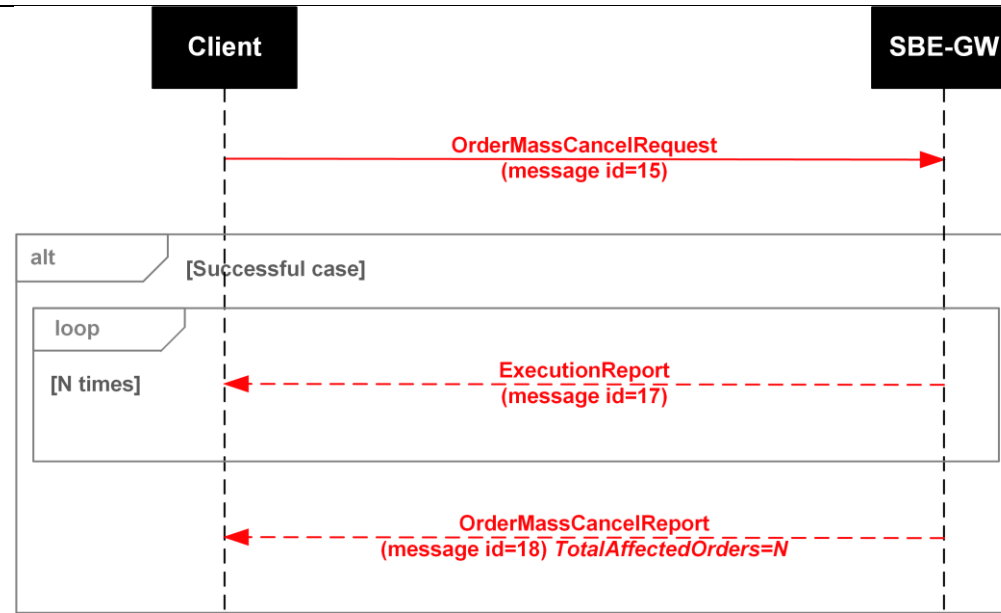


Figure 15. Diagram. Order mass cancellation

4.2.4. Replacing order

4.2.4.1. Replacing own order

Client can change price and quantity of own order.

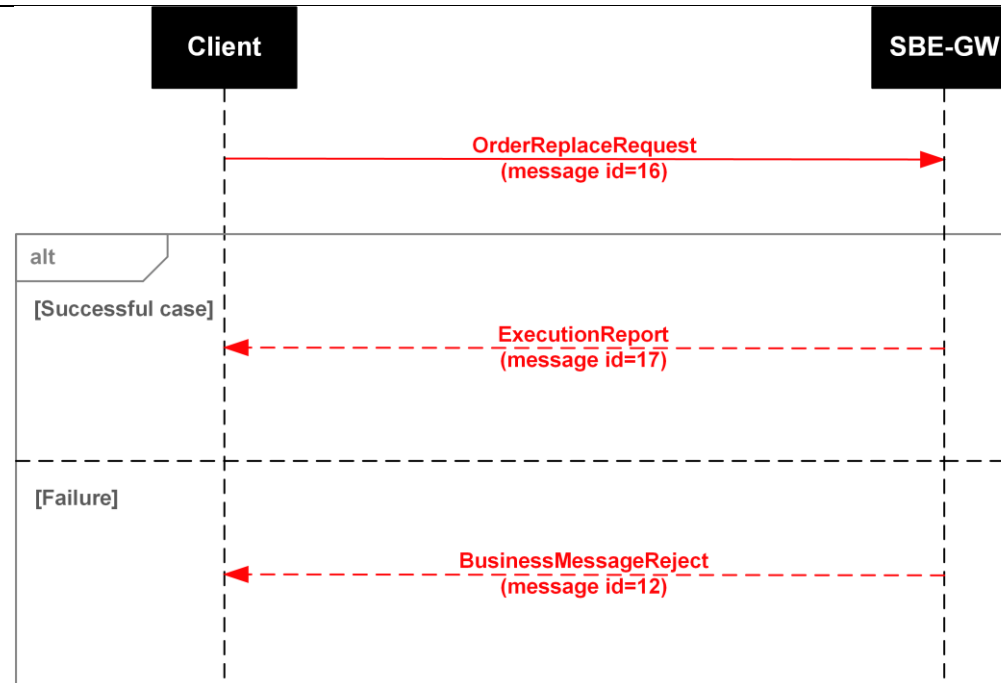


Figure 16. Diagram. Order replacement

4.2.4.2. Replacing order by other user (broker)

Scenario. Other user sends OrderReplaceRequest to SBE-GW. SBE-GW replies with Execution Report, ExecType=5 (Replaced). Owner of original order receives unsolicited Execution report message, ExecType=4 (Canceled), where the field ClOrdID contains value of ClOrdID field in a message that resulted in order registration.

The OrderReplaceRequest may also be rejected by the system (message BusinessMessageReject).

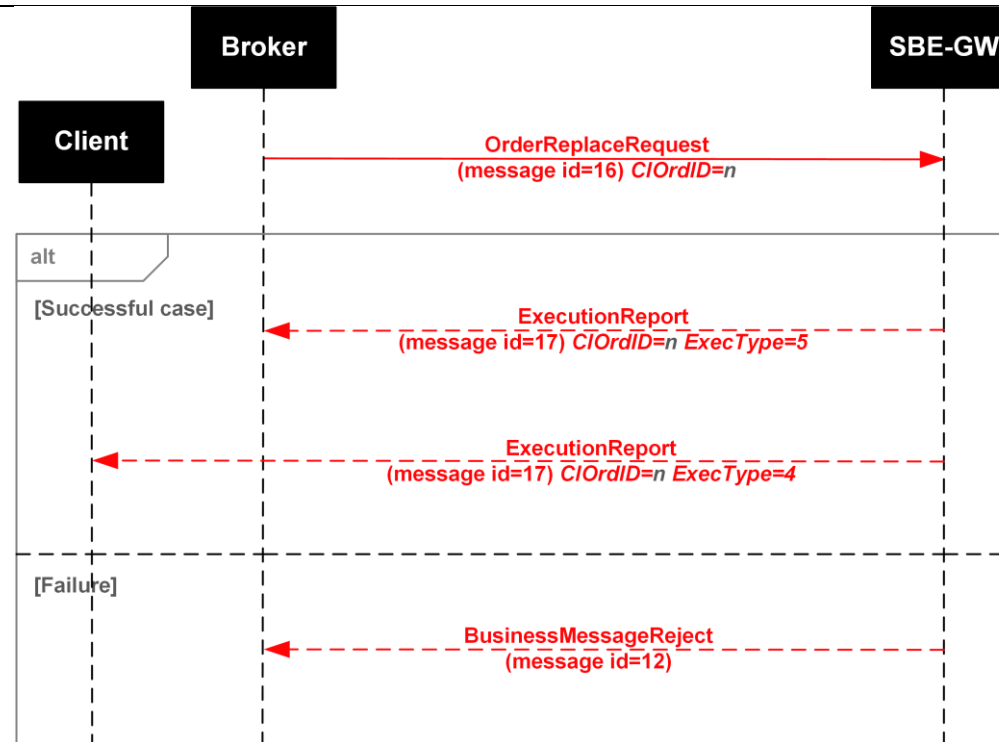


Figure 17. Diagram. Order replacement by other user (broker)

Note: Once a SBE-GW client (login) performs a replacement of another client's order, they become the owner of the replaced order and start receiving all necessary notifications regarding the order. Since that, the previous owner no more receives any notification regarding this order; to obtain information about their orders and trades, the clients are advised to use a Drop Copy service such as FIX Drop Copy or ASTS Bridge.

4.2.5. CIOrdID uniqueness

The SBE-GW checks the identifiers uniqueness. A client is required to provide unique 'CIOrdID' for orders with the session-long lifetime during a single trading session. Any nonunique 'CIOrdID' sent by client side will be rejected by the SBE-GW; also, the reply message 'SessionReject' containing error code 101 (CIOrdIdIsNotUnique) will be sent to the client side.

6. XML message schemas

Current message schemes are available in *.xml files placed at publicly available resource <http://ftp.moex.com/pub/TWIME/ASTS/>

7. Return codes list

Regularly updated list of return codes is publicly available at address http://ftp.moex.com/pub/ClientsAPI/ASTS/docs/asts_message_list.csv

8. Order cancellation reason codes

Cancellation reason codes are identical for all connectivity interfaces.

Constant	Description
00	By trading rules
01	Price move limit breached
02	Iceberg order
03	Market order
04	Market order with price
05	Canceled by operator
06	Closing auction price conditions
07	Canceled due to depo account suspension
08	Cancellation of unconfirmed negotiated orders
09	Cancellation of unconfirmed transfers
10	Cancellation of orders from external systems
11	Cancellation of orders awaiting activation event
12	Cancellation of orders awaiting activation time
13	Canceled during the market suspension sequence
14	Canceled during the board suspension sequence
15	Canceled during the board trading mode change
16	Canceled during the security suspension sequence
17	Canceled during the security trading mode change

Constant	Description
18	Canceled during the default management
19	Canceled to start early settlement
20	Trading closed for bankacc
21	Canceled on single limit check
22	Cancel orders time event
23	Cross trade check
24	Market maker trade check
25	Not enough collateral (on activation)
26	Canceled during the firm suspension sequence
27	Canceled during the user suspension sequence
28	Canceled during the trading account suspension sequence
29	Canceled during the instrument group suspension sequence
30	Value of the market order is less then one lot
31	Canceled during the user's trading suspension sequence
32	Canceled due to the prohibited period order
33	Canceled during the client code suspension sequence
34	Cancellation of a passive order during cross trade check
35	Market order depth limit breached
36	Cancellation of active quotes due to the trading session end
37	Internal liquidity order is rejected before trade conclusion with external quote
38	Quote is rejected before matching with active order
39	Canceled due to liquidity provider confirmation timeout
40	Declined by liquidity provider
41	Canceled due to collateral deficiency for external quote of liquidity provider