KASE

Binary protocol TWIME for derivatives market specification

version 5.0.1

Almaty 2022

Table of Contents

1. Introduction ... 5
  1.1. Purpose of the document ... 5
  1.2. Target audience ... 5
  1.3. General information on WireGate ... 5
2. Presentation layer ... 6
  2.1. Data types ... 6
    2.1.1. Integer ... 6
    2.1.2. Decimal ... 6
    2.1.3. String ... 6
    2.1.4. Date and time ... 6
    2.1.5. Enumerations ... 7
    2.1.6. Bitmasks ... 7
  2.2. Header ... 8
  2.3. Message scheme ... 8
3. Session layer ... 9
  3.1. Supported messages ... 9
    3.1.1. Establish (message id=5000) ... 9
    3.1.2. EstablishmentAck (message id=5001) ... 9
    3.1.3. EstablishmentReject (message id=5002) ... 9
    3.1.4. Terminate (message id=5003) ... 10
    3.1.5. RetransmitRequest (message id=5004) ... 10
    3.1.6. Retransmission (message id=5005) ... 11
    3.1.7. Sequence (message id=5006) ... 11
    3.1.8. FloodReject (message id=5007) ... 11
    3.1.9. SessionReject (message id=5008) ... 11
    3.1.10. BusinessMessageReject (message id=5009) ... 11
  3.2. Session interaction scenarios ... 12
    3.2.1. Session binding and termination ... 12
    3.2.2. Session status monitoring ... 13
    3.2.3. Message numbering ... 13
    3.2.4. Message retransmission request ... 13
    3.2.5. Session short-term crash recovery ... 14
    3.2.6. Full session crash recovery ... 14
    3.2.7. Message counter reset ... 15
  3.3. Flood control ... 15
  3.4. Disconnect on WireGate buffer overflow ... 16
  3.5. Request's rejection ... 16
    3.5.1. Rejecting request for adding order ... 16
    3.5.2. Rejecting request for cancellation of order ... 17
    3.5.3. Rejecting request for replacement of order ... 17
    3.5.4. Rejecting request for mass cancellation of order ... 17
    3.5.5. Rejecting request for adding iceberg order ... 18
    3.5.6. Rejecting request for cancellation of iceberg order ... 18
    3.5.7. Rejecting request for replacement of iceberg order ... 18
    3.5.8. Rejection of multi-day orders in a clearing session ... 19
4. Application layer ... 20
  4.1. Supported messages ... 20
    4.1.1. NewOrderSingle (message id=6000) ... 20
    4.1.2. NewOrderIceberg (message id=6008) ... 21
    4.1.3. OrderCancelRequest (message id=6006) ... 21
    4.1.4. OrderIcebergCancelRequest (message id=6009) ... 21
    4.1.5. OrderReplaceRequest (message id=6007) ... 21
    4.1.6. OrderIcebergReplaceRequest (message id=6010) ... 22
    4.1.7. OrderMassCancelRequest (message id=6004) ... 22
    4.1.8. NewOrderSingleResponse (message id=7000) ... 23
    4.1.9. NewOrderIcebergResponse (message id=7013) ... 23
    4.1.10. OrderCancelResponse (message id=7003) ... 24
    4.1.11. OrderReplaceResponse (message id=7005) ... 25
    4.1.12. OrderMassCancelResponse (message id=7007) ... 25
    4.1.13. ExecutionSingleReport (message id=7008) ... 25
    4.1.14. ExecutionMultilegReport (message id=7009) ... 26
    4.1.15. EmptyBook (message id=7010) ... 27
    4.1.16. SystemEvent (message id=7011) ... 27
  4.2. Trading interaction scenarios ... 28
    4.2.1. Adding orders ... 28
      4.2.1.1. Adding a new order on a standard instrument by a client ... 28
      4.2.1.2. Adding a new order on a multi leg instrument by a client ... 30
      4.2.1.3. Adding a new iceberg order by a client ... 30
    4.2.2. Order cancellation (deletion) ... 31

        - 4.2.2.1. Client order cancellation by client ... 31
        - 4.2.2.2. Client order cancellation by Brokerage Firm ... 31
        - 4.2.2.3. Client iceberg order cancellation by client ... 32
        - 4.2.2.4. Client iceberg order cancellation by Brokerage Firm ... 32
    - 4.2.3. Order mass cancellation ... 33
        - 4.2.3.1. Order mass cancellation by client ... 33
        - 4.2.3.2. Client's order mass cancellation by Brokerage Firm ... 33
    - 4.2.4. Order replacement ... 34
        - 4.2.4.1. Order replacement by client ... 34
        - 4.2.4.2. Order replacement by Brokerage Firm ... 35
        - 4.2.4.3. Iceberg order replacement by client ... 36
        - 4.2.4.4. Iceberg order replacement by Brokerage Firm ... 36
    - 4.2.5. Filling orders ... 37
        - 4.2.5.1. Day order ... 37
        - 4.2.5.2. Offsetting IOC order (partial fills) ... 37
        - 4.2.5.3. Trade matching after order had been changed by Brokerage Firm ... 38
        - 4.2.5.4. Filling a multileg order ... 39
        - 4.2.5.5. Trade matching of iceberg order ... 40
    - 4.2.6. Closing a trading session ... 41
    - 4.2.7. Automatic order cancellation option (Cancel On Disconnect) ... 41
    - 4.2.8. Relisting multi-days orders ... 42
    - 4.2.9. Instrument ID ... 43
    - 4.2.10. Client order ID ... 43
- 5. TWIME protocol customization for Brokerage Firms ... 44
- 6. Message scheme ... 45
- 7. List of return codes ... 51

History of changes

Date	Version	Changes
07.06.2021	5.0.0	Changes applied: • Changed texts of error codes: 31, 4142.
27.04.2022	5.0.1	Changes applied: • Added new error codes: 80, 81, 3001, 4280-4282. • Changed texts of error codes: 4017, 4160. • Removed error codes: 4120, 4121, 4168.

1. Introduction

1.1. Purpose of the document

The document overviews specifications of the binary protocol for derivatives market and covers the presentation, session and application layers of the protocol. The document does not cover administrative and technical details of establishing a network connection using the protocol; also, this document does not cover the security support methods.

1.2. Target audience

The document is targeted to business analysts, system architects and developers involved in developing and programming software applications for accessing the derivatives market through the WireGate.

1.3. General information on WireGate

The gateway WireGate is a server side software application that allows users trading applications (trading robots, terminals, technical analysis systems, etc.) to connect to derivatives market tradings using the binary protocol. The gateway provides users with the following possibilities:

- To send orders into the trading system.
- To cancel and replace orders.
- To obtain results of messages processing.
- To obtain messages about matching of orders.
- To request a message resend (in case of a missing message).

Please note that the gateway does not transmit market data.

2. Presentation layer

The presentation layer protocol is based on the FIX Simple Binary Encoding (https://www.fixtrading.org/standards/sbe); it is expected that users have already got some information about this protocol.

2.1. Data types

Within the protocol, the following data types are used:

2.1.1. Integer

```
<type name="Int8" maxValue="126" minValue="-128" nullValue="127"
      presence="optional" primitiveType="int8"
      description="Integer signed, 1 byte"/>

<type name="Int16" maxValue="32766" minValue="-32768" nullValue="32767"
      presence="optional" primitiveType="int16"
      description="Integer signed, 2 bytes"/>

<type name="Int32" maxValue="2147483646" minValue="-2147483648"
      nullValue="2147483647" presence="optional" primitiveType="int32"
      description="Integer signed, 4 bytes"/>

<type name="Int64" maxValue="9223372036854775806" minValue="-9223372036854775808"
      nullValue="9223372036854775807" presence="optional"
      primitiveType="int64" description="Integer signed, 8 bytes"/>

<type name="UInt8" maxValue="254" minValue="0" nullValue="255"
      presence="optional" primitiveType="uint8"
      description="Integer unsigned, 1 byte"/>

<type name="UInt16" maxValue="65534" minValue="0" nullValue="65535"
      presence="optional" primitiveType="uint16"
      description="Integer unsigned, 2 bytes"/>

<type name="UInt32" maxValue="4294967294" minValue="0"
      nullValue="4294967295" presence="optional" primitiveType="uint32"
      description="Integer unsigned, 4 bytes"/>

<type name="UInt64" maxValue="18446744073709551614" minValue="0"
      nullValue="18446744073709551615" presence="optional"
      primitiveType="uint64" description="Integer unsigned, 8 bytes"/>
```

2.1.2. Decimal

```
<composite name="Decimal5" description="Decimal">
      <type name="mantissa" description="mantissa" minValue="-9999999999999999"
            maxValue="9999999999999999" primitiveType="int64" presence="required" />
      <type name="exponent" description="exponent" presence="constant"
            primitiveType="int8">-5</type>
</composite>
```

2.1.3. String

String of symbols with fixed lendth

```
<type name="String7" length="7" primitiveType="char"/>

<type name="String20" length="20" primitiveType="char"/>

<type name="String25" length="25" primitiveType="char"/>
```

2.1.4. Date and time

```
<type name="DeltaMillisecs" description="Milliseconds per one timeout"
      maxValue="60000" minValue="1000" presence="required"
      primitiveType="uint32"/>

<type name="TimeStamp" description="Time in number of nanoseconds since Unix epoch, UTC timezone"
      maxValue="18446744073709551614" minValue="0" nullValue="18446744073709551615"
      presence="optional" primitiveType="uint64"/>
```

2.1.5. Enumerations

```
<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>
    <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="EstablishmentRejectCodeEnum" encodingType="uint8">
    <validValue name="Unnegotiated">0</validValue>
    <validValue name="AlreadyEstablished">1</validValue>
    <validValue name="SessionBlocked">2</validValue>
    <validValue name="KeepaliveInterval">3</validValue>
    <validValue name="Credentials">4</validValue>
    <validValue name="Unspecified">5</validValue>
</enum>

<enum name="SessionRejectReasonEnum" encodingType="uint8">
    <validValue name="ValueIsIncorrect">5</validValue>
    <validValue name="Other">99</validValue>
    <validValue name="SystemIsUnavailable">100</validValue>
    <validValue name="ClOrdIdIsNotUnique">101</validValue>
</enum>

<enum name="TimeInForceEnum" encodingType="uint8">
    <validValue name="Day">0</validValue>
    <validValue name="IOC">3</validValue>
    <validValue name="FOK">4</validValue>
    <validValue name="GTD">6</validValue>
</enum>

<enum name="SideEnum" encodingType="uint8">
    <validValue name="Buy">1</validValue>
    <validValue name="Sell">2</validValue>
    <validValue name="AllOrders">89</validValue>
</enum>

<enum name="ModeEnum" encodingType="uint8">
    <validValue name="DontChangeOrderQty">0</validValue>
    <validValue name="ChangeOrderQty">1</validValue>
    <validValue name="CheckOrderQtyAndCancelOrder">2</validValue>
    <validValue name="FixStyleReplace">3</validValue>
</enum>

<enum name="TradSesEventEnum" encodingType="uint8">
   <validValue name="SessionDataReady">101</validValue>
   <validValue name="IntradayClearingFinished">102</validValue>
   <validValue name="IntradayClearingStarted">104</validValue>
   <validValue name="ClearingStarted">105</validValue>
   <validValue name="ExtensionOfLimitsFinished">106</validValue>
   <validValue name="BrokerRecalcFinished">108</validValue>
</enum>
```

2.1.6. Bitmasks

```
<set name="FlagsSet" encodingType="uint64">
 <choice name="Day" description="Orders: Day order" >0</choice><set name="FlagsSet" encodingType="uint64">
 <choice name="Day" description="Orders: Day order" >0</choice>
 <choice name="IOC" description="Orders: IOC order" >1</choice>
 <choice name="OTC" description="Orders and Trades: OTC order or OTC trade" >2</choice>
 <choice name="PosTransfer" description="Trades: Position transfer trade" >3</choice>
 <choice name="Collateral" description="Orders:
 Client collateral was not checked while adding order" >4</choice>
 <choice name="DontCheckLimits" description="Orders: Do not check limits for options">9</choice>
 <choice name="FOK" description="Orders: FOK order" >19</choice>
 <choice name="Replace" description="Orders:
 The record results from replacing the order" >20</choice>
 <choice name="Cancel" description="Orders:
 The record results from cancelling the order" >21</choice>
 <choice name="MassCancel" description="Orders:
 The record results from mass cancelling" >22</choice>
 <choice name="Clearing" description="Trades: Clearing session trade" >25</choice>
 <choice name="Negotiated" description="Trades: Negotiated trade" >26</choice>
 <choice name="MultiLeg" description="Trades: Multi leg trade" >27</choice>
 <choice name="CrossTrade" description="Orders:
 Flag of cancelling the left balance of the order because of a cross-trade" >29</choice>
 <choice name="COD" description="Orders:
 The record results from cancelling an order via 'Cancel on Disconnect' service">32</choice>
 <choice name="UKS" description="Orders: The record results from cancelling an order via
 'User Kill Switch' service">37</choice>
 <choice name="LiqNettingRF" description="Orders and Trades: The record are formed in the
 process of liquidation netting">40</choice>
 <choice name="ActiveSide" description="Trades: Flag of aggressive side">41</choice>
 <choice name="PassiveSide" description="Trades: Flag of passive side">42</choice>
 <choice name="Synthetic" description="Orders: Flag of the synthetic order">45</choice>
 <choice name="Iceberg" description="Orders and Trades: Iceberg order">47</choice>
 <choice name="DisclosedIceberg" description="Orders: The record results from disclosing
 of Iceberg order">53</choice>
</set>



<set name="SecurityTypeSet" encodingType="uint8">
   <choice name="Future" description="Futures" >0</choice>

</set>
```

2.2. Header

A WireGate message consists of a header and a message body (please note that the header fields always precede the message body fields). The header name is 'messageHeader'. The standard message header contains the following fields:

```
<composite name="messageHeader">
           <type name="blockLength" primitiveType="uint16" description="Message body length"/>
           <type name="templateId" primitiveType="uint16" description="Message ID"/>
           <type name="schemaId" primitiveType="uint16" description="Message scheme ID"/>
           <type name="version" primitiveType="uint16" description="Message scheme version"/>
</composite>
```

2.3. Message scheme

```
<?xml version="1.0" encoding="UTF-8"?>
<sbe:messageSchema byteOrder="littleEndian" id="19781" package="sbe" version="5">
</sbe:messageSchema>
```

Scheme attributes:

Attribute	Details	Value
id	Scheme unique ID	
version	Scheme version	
package	Scheme name or category	"sbe"
byteOrder	Byte order in fields	"littleEndian"

3. Session layer

The session layer provides parties authentication, guarantee message delivery, message sequential procession, connection status control and possibility to recover in case of any failure. The session layer protocol is based on FIXP (https://www.fixtrading.org/standards/fixp); it is expected that users have already got some information about this protocol.

3.1. Supported messages

- **Establish** - Initiates binding the session to the current TCP connection.
- **EstablishmentAck** - Confirms that the session has been successfully bound to the current TCP connection.
- **EstablishmentReject** - Notifies that the session has not been successfully bound to the current TCP connection.
- **Terminate** - Session termination.
- **RetransmitRequest** - Requests message retransmission starting from a specified message number.
- **Retransmission** - Notifies about a message retransmission.
- **Sequence** - Specifies the next message number and is also used as the Heartbeat.
- **FloodReject** - Notifies that number of messages per time unit exceeded the limit.
- **SessionReject** - Notifies about invalid messages sent from the client side.
- **BusinessMessageReject** - Application level message rejection.

Below, there are details on the message fields. Each field contains the following attributes:

- **Tag** – field unique ID;
- **Field** – field name;
- **Mandatory** – defines whether 'nullValue' is a valid value or not:
  - **Y** - the field is mandatory, i.e. 'nullValue' will not be transmitted;
  - **N** - the field non-mandatory, i.e. 'nullValue' may be transmitted;
  - **C** - the field contains a non-'nullValue' value subject to a certain condition.
- **Type** - field type;
- **Details** - field's detailed description.

3.1.1. Establish (message id=5000)

The message Initiates binding the session to the current TCP connection. After the TCP connection has established, the system expects the message 'Establish' to be sent within 10 seconds, otherwise the TCP connection terminates.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Request sending time.
20205	KeepaliveInterval	Y	DeltaMillisecs	Time interval to for sending the Heartbeat messages. The time interval value must be within 1000 - 60000 milliseconds range inclusive.
20206	Credentials	Y	String20	Client ID (login).

3.1.2. EstablishmentAck (message id=5001)

The message confirms that the session has been successfully bound to the current TCP connection.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20207	RequestTimestamp	Y	TimeStamp	Sending time of the message 'Establish'.
20205	KeepaliveInterval	Y	DeltaMillisecs	'Heartbeat' messages time interval.
20208	NextSeqNo	Y	UInt64	Next message sequential number.

3.1.3. EstablishmentReject (message id=5002)

The message notifies that the session has not been successfully bound to the current TCP connection.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20207	RequestTimestamp	Y	TimeStamp	Sending time of the message 'Establish'.
20209	EstablishmentRejectCode	Y	EstablishmentRejectCodeEnum	Establishment reject code: • "0" Unnegotiated - Reserved in FIXP, but cannot pass to WireGate. • "1" AlreadyEstablished - A connection has been already established for this client. • "2" SessionBlocked - User has been blocked. Please contact the technical support for details. • "3" KeepaliveInterval - 'Heartbeat' sending interval exceeds the allowed limit. • "4" Credentials - User not found/Wrong IP address. • "5" Unspecified - Internal server error. Please contact technical support.

3.1.4. Terminate (message id=5003)

Session termination.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20210	TerminationCode	Y	TerminationCodeEnum	Session termination reason: • "0" Finished - Terminated by client's request. • "1" UnspecifiedError - Internal server error. Please contact technical support. • "2" ReRequestOutOfBounds - In reply to 'RetransmitRequest'. The requested messages cannot be retransmit. • "3" ReRequestInProgress - 'RetransmitRequest' is still in progress. • "4" TooFastClient - A client sends too many messages and exceeds their quota. • "5" TooSlowClient - A client does not retrieve messages from TCP socket; too many messages left non-retrieved. • "6" MissedHeartbeat - A client has not sent a single message within the 'KeepaliveInterval'. • "7" InvalidMessage - The message does not comply the protocol standards/Unable to recognize the received message. • "8" TCPFailure - Error in transport layer. • "9" InvalidSequenceNumber - in reply to the message 'Sequence' containing an incorrect 'SequenceNumber' value. • "10" ServerShutdown - Server shuts down normally.

3.1.5. RetransmitRequest (message id=5004)

Requests 'Count' message(s) retransmission starting from number 'FromSeqNo'.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Request sending time.
20211	FromSeqNo	Y	UInt64	Sequence number of the first message to retransmit.

Tag	Field	Mandatory	Type	Details
20212	Count	Y	UInt32	Number of messages to retransmit.

3.1.6. Retransmission (message id=5005)

The message notifies that all the next 'Count' message(s) are retransmitted in reply to the 'RetransmitRequest' (message id=5004).

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20208	NextSeqNo	Y	UInt64	Sequence number of the first message among those sent in reply to message ‘RetransmitRequest’.
20207	RequestTimestamp	Y	TimeStamp	Sending time of the message 'RetransmitRequest'.
20212	Count	Y	UInt32	Number of messages.

3.1.7. Sequence (message id=5006)

It is used only as a Heartbeat-message. When sent from client side to server side, the field 'NextSeqNo' contains 'nullValue'. When sent from server side to client side, the field 'NextSeqNo' contains number of the next application layer message.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20208	NextSeqNo	N	UInt64	Next message sequence number.

3.1.8. FloodReject (message id=5007)

When the message limit is exceeded, the Flood Control system sends a message to user containing service denial notification.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	ClOrdID that was submitted with the message being rejected.
20213	QueueSize	Y	UInt32	Number of messages received from client during the last second.
20214	PenaltyRemain	Y	UInt32	A period of rejection in microseconds; after this specified period, the system will stop rejecting user's messages.

3.1.9. SessionReject (message id=5008)

Notifies about invalid messages sent from the client side.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	ClOrdID that was submitted with the message being rejected.
371	RefTagID	N	UInt32	ID of invalid field
373	SessionRejectReason	Y	SessionRejectReasonEnum	Rejection reason code: • "5" ValueIsIncorrect - Incorrect field value. • "100" SystemIsUnavailable - Trading system is unavailable. • "101" ClOrdIdIsNotUnique - Client order ID is not unique. • "99" Other - Other reason.

3.1.10. BusinessMessageReject (message id=5009)

Notifies of rejection of application layer messages. 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
- cancellation of multi-day orders in clearing session.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	ClOrdID ID from request.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
103	OrdRejReason	Y	Int32	Reason for rejection. For details see 'List or return codes'.

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 authorised, the system replies with the message 'EstablishmentAck', confirming that the session has been successfully bound; otherwise (incorrect message 'Establish' and/or non-authorised user) the system will reply with the message 'EstablishmentReject' containing the rejection reason details.

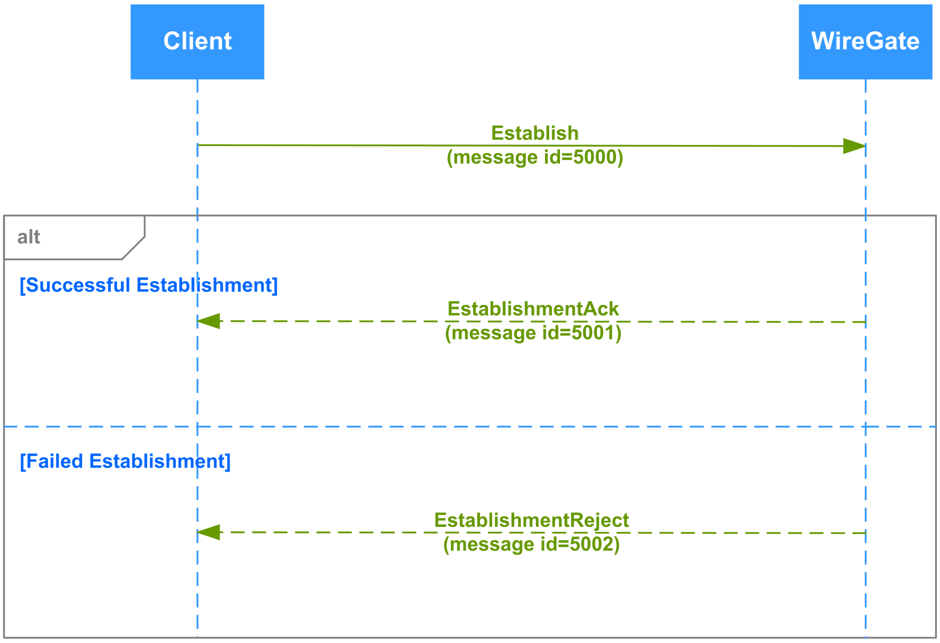

Figure 1. Diagram. Session binding

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

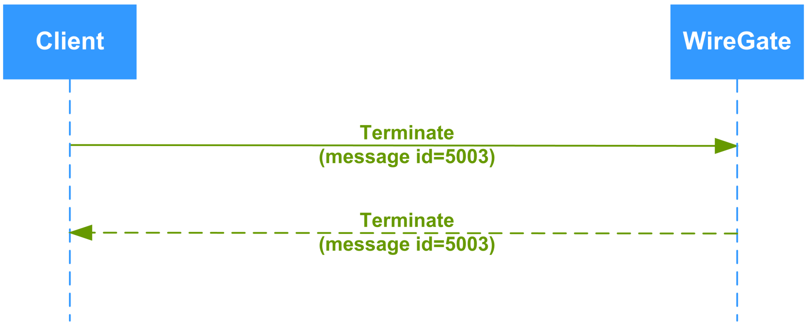

Figure 2. Diagram. Session termination

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 WireGate to each of the sessions.

3.2.2. Session status monitoring

In order to monitor the session status, both sides (client and WireGate) 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'.

WireGate 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 WireGate did not send messages other than ‘Heartbea't 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'.

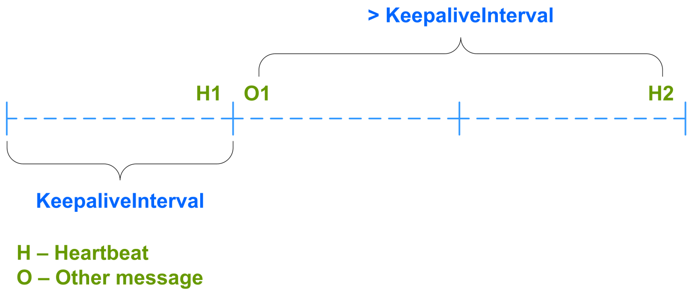

Figure 3. Diagram. Sending 'Heartbeat' messages from WireGate

If the server side does not receive any message from client side within the 'KeepaliveInterval', the client side gets disconnected from WireGate. This disconnection (via COD) takes from 'KeepaliveInterval' up to 2*'KeepaliveInterval' + data transmission time. Please bear this in mind when you try sending a 'Heartbeat' message to verify if connection is still active.

The client side should not send more than 3 'Heartbeat' messages per second to WireGate. The fourth 'Heartbeat' message per second sent will cause a connection termination, when WireGate will send the message 'Terminatet' 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 600 milliseconds.

3.2.3. Message numbering

For consecutive numbering, the client side has to support the message counter in order for messages sent from WireGate, where the very first message EstablishmentAck sent from WireGate 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 WireGate during processing missing messages. All new messages will be dispatched right upon the command ‘RetransmitRequest’ completion.

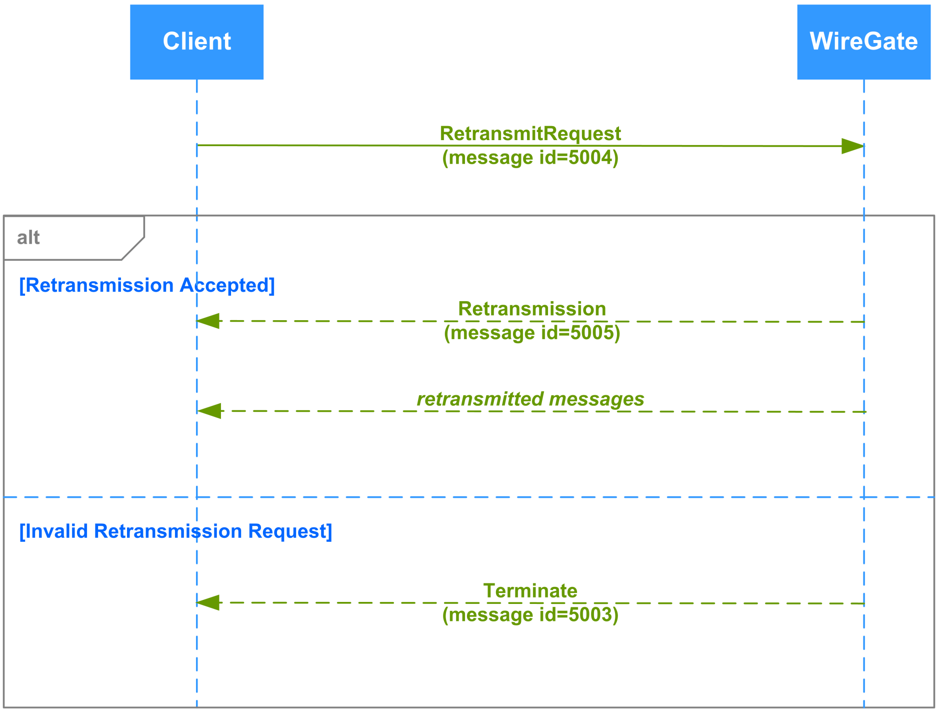

Figure 4. Diagram. Message retransmission request

3.2.5. Session short-term crash recovery

In order 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'. WireGate then will send an application layer message in reply, after one second from receiving command 'RetransmitRequest'. A client can request no more than 10 messages per a single ‘RetransmitRequest’.

3.2.6. Full session crash recovery

In order to receive all session messages, a client side should establish a connection to the recovery service using the appropriate IP address and port number; the message 'Establish' should contain the same login as the one used to connect to WireGate. As the new connection has been established, the client side will obtain a new incoming message sequence number in field 'NextSeqNo' of message 'EstablishmentAck'. Then, the client side should request the missed messages retransmission using command 'RetransmitRequest'. Please note that the connection may be terminated by the server side (error message 'TooSlowClient', see section '3.4. Disconnect on WireGate buffer overflow' of this manual) in case of the client's side inability to process the requested messages on time. A client can request no more than 1000 messages per a single ‘RetransmitRequest’.

After that, the client side is required to establish connection to WireGate, following the steps listed in section '3.2.5. Session short-term crash recovery'.

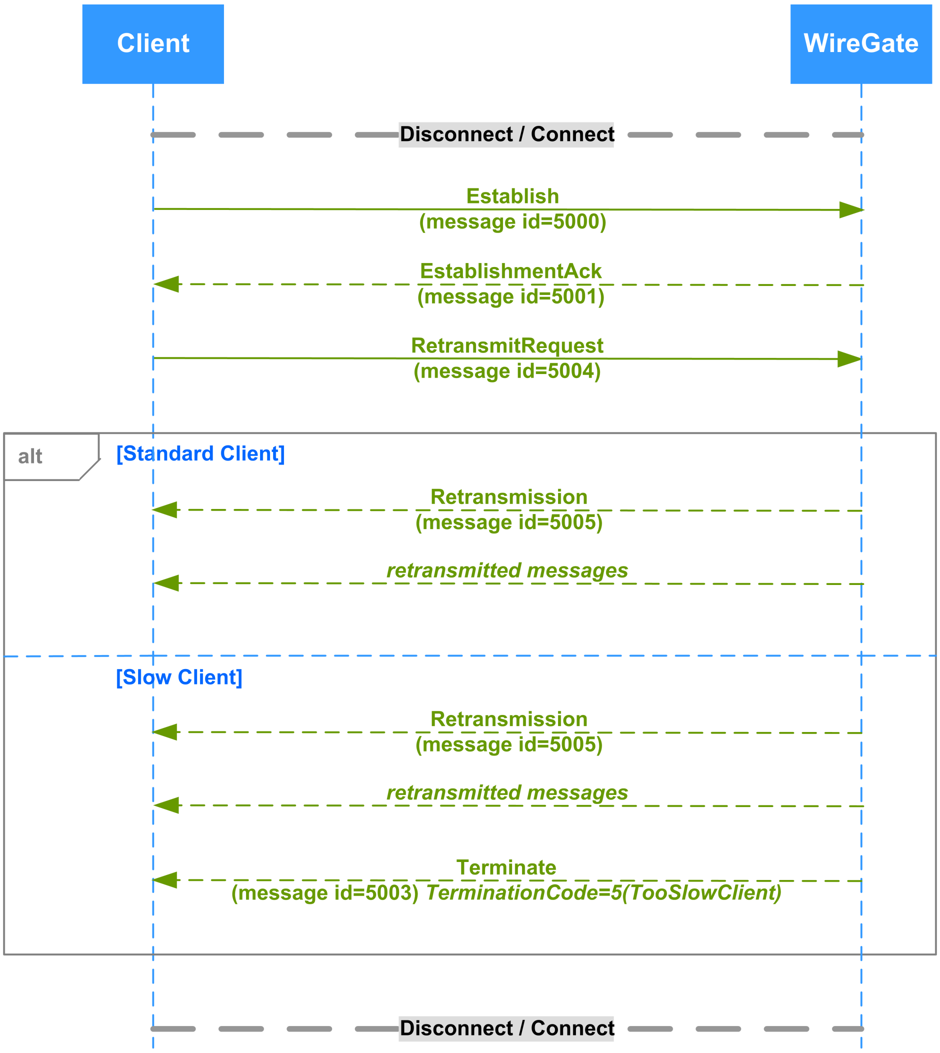

Figure 5. Diagram. Full session crash recovery

3.2.7. Message counter reset

At night, until 10:00 the next day all clients are disconnected from WireGate, all messages of the previous day trading session are being cleaned up, and the message counter resets. Once the message counter has been successfully reset, and connection to WireGate has been reestablished, the client side receives a new message number in field 'NextSeqNo' of message 'EstablishmentAck', which is smaller or equal than the one the client side had before resetting of the message counter. Than the client side has to initialize their message counter using the newly obtained message number. Once the message numbers in WIreGate have been reset, only application layer messages of the evening trading session remain available for recovery (if she is any on this day).

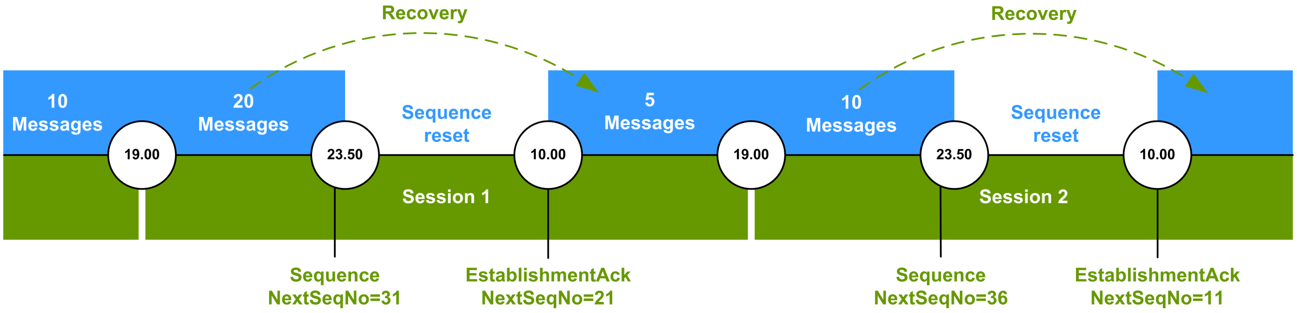

Figure 6. Diagram. Message counter reset

3.3. Flood control

The WireGate maintains a flood control module that restricts clients from sending messages in number greater than specified within a single session. Now clients are offered with a system logins allowing to send 30, 60, 90, etc. (but not more than 3000) trading messages ('NewOrderSingle', 'OrderCancelRequest', 'OrderReplaceRequest' and 'OrderMassCancelRequest') per 1 second.

Once the message number limit is reached by client, the system sends the message 'FloodReject' to client notifying about service denial.

The message number counter counts every received message to evaluate the whole number of messages received within every 1 second time interval. Therefore, if a client exceeds their specified message limit constantly (every next second), their messages will not be processed at all.

If a client exceeds twice the specified message limit, the connection will be terminated by WIreGate via the message 'Terminate' with TerminationCode=4.

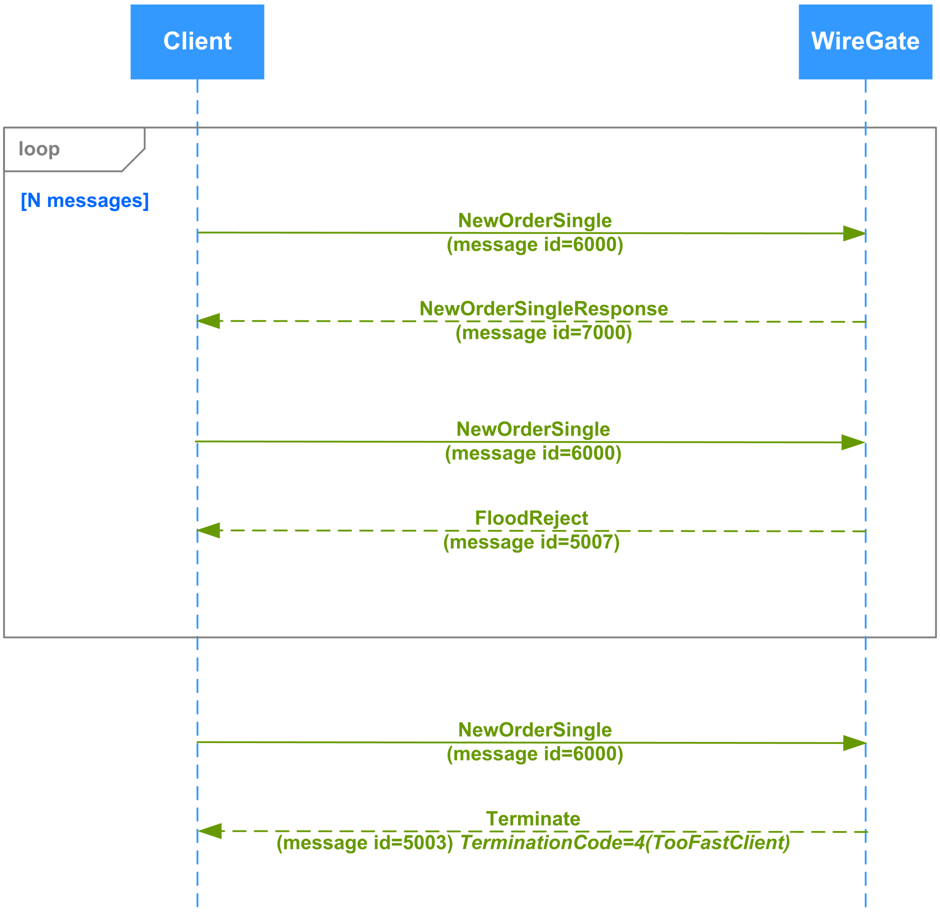

Figure 7. Diagram. Flood control

3.4. Disconnect on WireGate 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 WireGate buffer is overflown.

3.5. Request's rejection

If application-level requests are rejected, WireGate 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
- cancellation of multi-day orders in clearing session.

Session-layer messages 'BusinessMessageReject' are not available in the recovery service described in the sections "3.2.5. Session short-term crash recovery", "3.2.6. Full session crash recovery"

3.5.1. 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'.

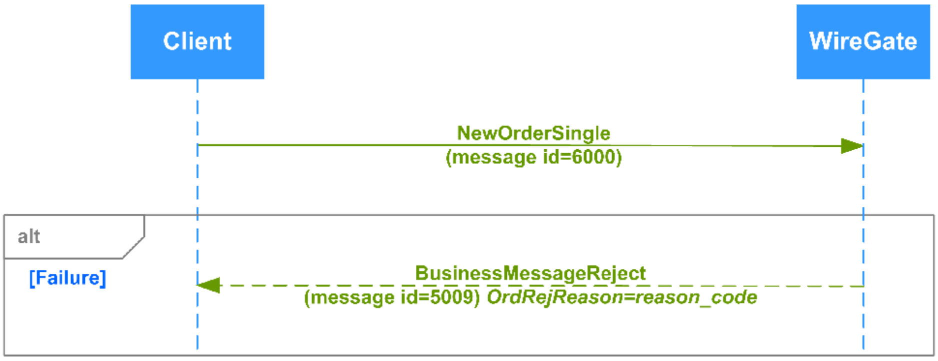

Figure 8. Diagram. Rejecting request for adding order

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'.

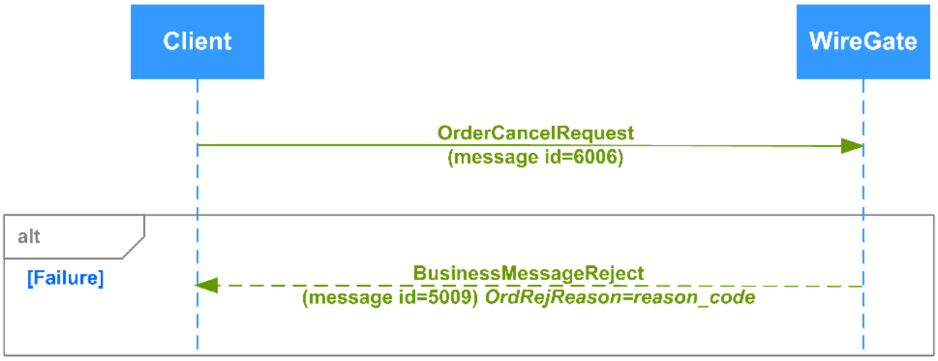

Figure 9. Diagram. Rejecting request for cancellation of order

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'.

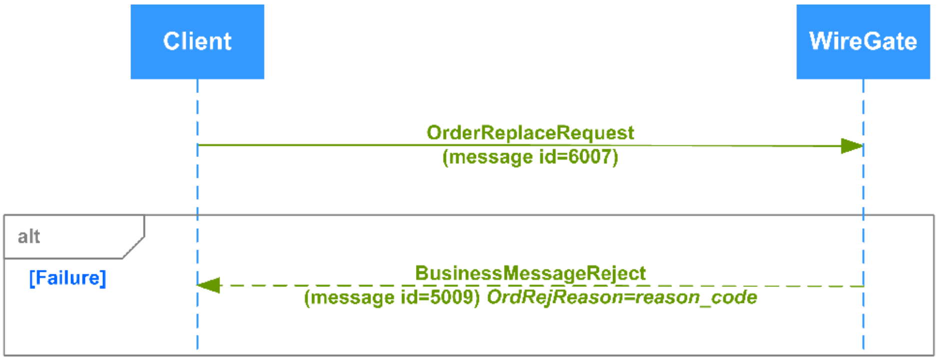

Figure 10. Diagram. Rejecting request for replacement of order

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'.

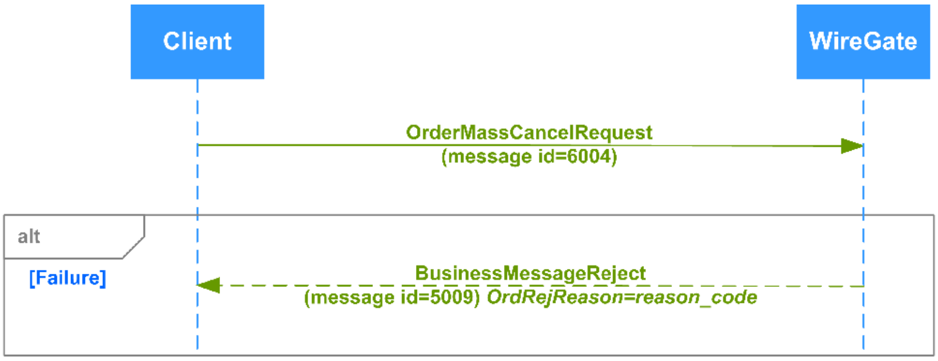

Figure 11. Diagram. Rejecting request for mass cancellation of order

3.5.5. Rejecting request for adding iceberg order

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

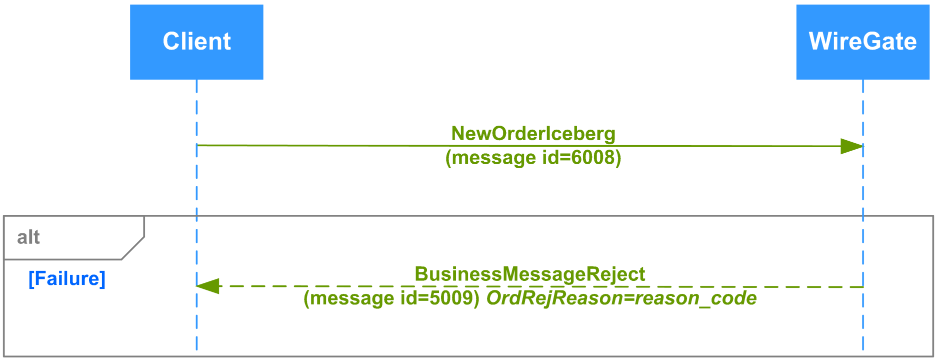

Figure 12. Diagram. Rejecting request for adding iceberg order

3.5.6. Rejecting request for cancellation of iceberg order

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

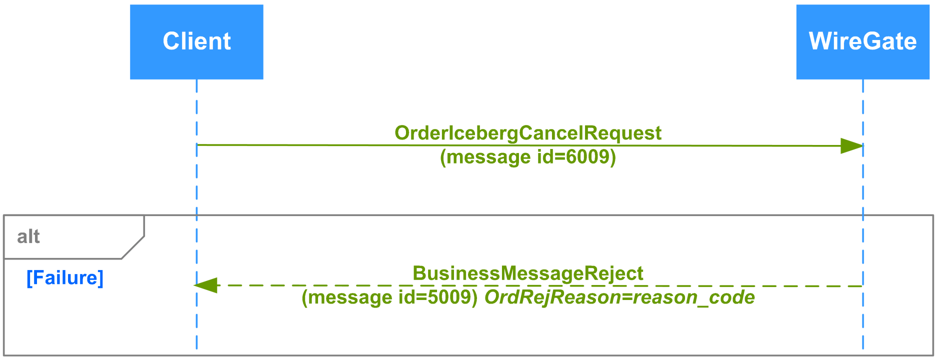
Client
WireGate
OrderIcebergCancelRequest
(message id=6009)
alt
[Failure]
BusinessMessageReject
(message id=5009) *OrdRejReason=reason_code*

Figure 13. Diagram. Rejecting request for cancellation of iceberg order

3.5.7. Rejecting request for replacement of iceberg order

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

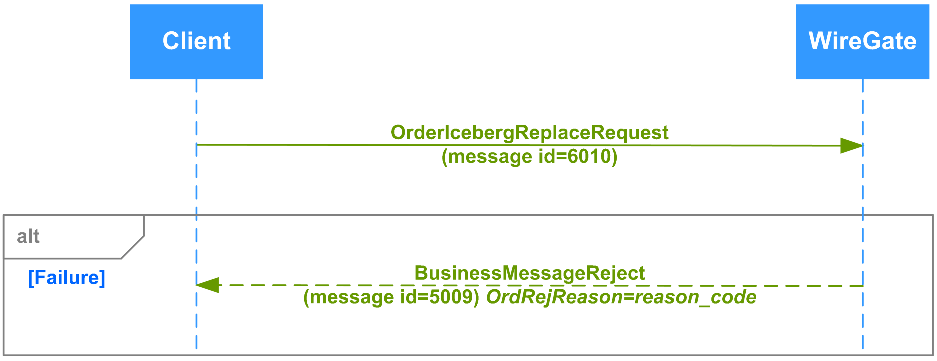
Client
WireGate
OrderIcebergReplaceRequest
(message id=6010)
alt
[Failure]
BusinessMessageReject
(message id=5009) OrdRejReason=reason_code

Figure 14. Diagram. Rejecting request for replacement of iceberg order

3.5.8. Rejection of multi-day orders in a clearing session

Multi-day orders are the orders with a fixed expiration date. Such orders will be relisted for the next trading session automatically. Upon relisting an order, the trading system verifies the instrument and the client availability, and the collateralization. In case of a negative result, the order will be rejected. The system sends a 'BusinessMessageReject' message for each rejected order.

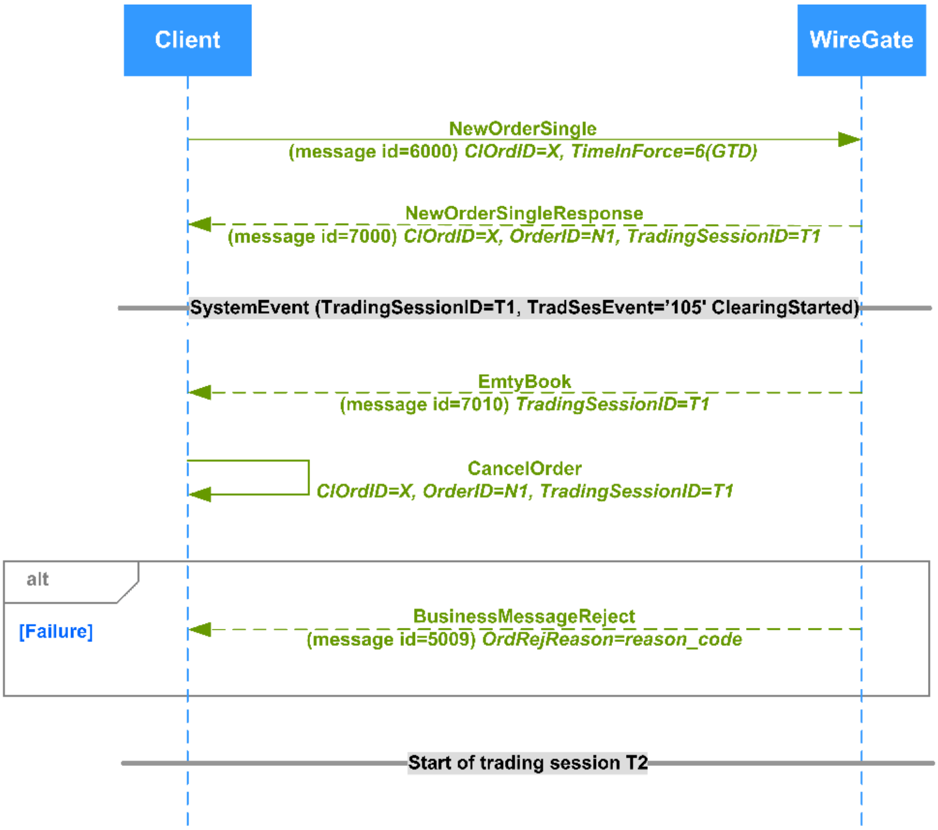
Client
WireGate
NewOrderSingle
(message id=6000) ClOrdID=X, TimeInForce=6(GTD)
NewOrderSingleResponse
(message id=7000) ClOrdID=X, OrderID=N1, TradingSessionID=T1
SystemEvent (TradingSessionID=T1, TradSesEvent='105' ClearingStarted)
EmtyBook
(message id=7010) TradingSessionID=T1
CancelOrder
ClOrdID=X, OrderID=N1, TradingSessionID=T1
alt
[Failure]
BusinessMessageReject
(message id=5009) OrdRejReason=reason_code
Start of trading session T2

Figure 15. Diagram. Rejection of multi-day orders in a clearing session

4. Application layer

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.
- **NewOrderIceberg** – Adding a new iceberg order. The message is sent from client side to server side.
- **OrderCancelRequest** – Order cancellation. The message is sent from client side to server side.
- **OrderIcebergCancelRequest** – Iceberg 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.
- **OrderIcebergReplaceRequest** – Iceberg 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.
- **NewOrderSingleResponse** – Order adding confirmation. The message is sent from server side to client side.
- **NewOrderIcebergResponse** – Iceberg order adding confirmation. The message is sent from server side to client side.
- **OrderCancelResponse** – Order cancellation confirmation. The message is sent from server side to client side.
- **OrderReplaceResponse** – Order replacement confirmation. The message is sent from server side to client side.
- **OrderMassCancelResponse** – Order mass cancellation confirmation. The message is sent from server side to client side.
- **ExecutionSingleReport** – Order matching into trade confirmation. The message is sent from server side to client side.
- **ExecutionMultilegReport** – Multileg order matching into trade confirmation. The message is sent from server side to client side.
- **EmptyBook** – Closing of trading session. The message is sent from server side to client side.
- **SystemEvent** - Trading system events. The message is sent from server side to client side.

4.1.1. NewOrderSingle (message id=6000)

Adding orders for any instruments.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the sent order.
432	ExpireDate	C	TimeStamp	Order expiration date. Mandatory for orders with TimeIn-Force=6 (GTD).
44	Price	Y	Decimal5	Price.
48	SecurityID	Y	Int32	Instrument numeric ID.
583	ClOrdLinkID	Y	Int32	External ID.
38	OrderQty	Y	UInt32	Number of instrument units.
59	TimeInForce	Y	TimeInForceEnum	Order type: • '0' Day –Day order (the order remains in order book after being partly matched). • '3' IOC – Offsetting order (the order is cancelled after auction ends). • '4' FOK – Fill-or-Kill order. • '6' GTD
54	Side	Y	SideEnum	Order side: • '1' - Buy

Tag	Field	Manda-tory	Type	Details
				• '2' - Sell
1	Account	Y	String7	Client 7-symbol ID.

4.1.2. NewOrderIceberg (message id=6008)

Adding iceberg orders for any instruments.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the sent order.
432	ExpireDate	N	TimeStamp	Order expiration date.
44	Price	Y	Decimal5	Price.
48	SecurityID	Y	Int32	Instrument numeric ID.
583	ClOrdLinkID	Y	Int32	External ID.
1138	DisplayQty	Y	UInt32	The number of instrument units in the constant component of the volume of the pop-up (visible part) iceberg order.
20036	DisplayVarianceQty	N	UInt32	The value of the random deviation of the volume of the pop-up part of the iceberg order.
38	OrderQty	Y	UInt32	Number of instrument units for the entire order.
54	Side	Y	SideEnum	Order side: • '1' - Buy • '2' - Sell
1	Account	Y	String7	Client 7-symbol ID.

4.1.3. OrderCancelRequest (message id=6006)

Request to cancel the order.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the cancel request.
37	OrderID	Y	Int64	SPECTRA ID of the order to cancel.
48	SecurityID	Y	Int32	Instrument numeric ID.
1	Account	Y	String7	Client 7-symbol ID.

4.1.4. OrderIcebergCancelRequest (message id=6009)

Request to cancel the iceberg order. Both the visible part number of the iceberg ('DisplayOrderID' from 'NewOrderIcebergResponse' message) and the identifier of the entire iceberg order ('OrderID') can be indicated as the identifier of the canceled order ('OrderID). Please note, that a command will work on 'DisplayOrderID' only if the visible part with such a number is still in the system (has not been matched), otherwise an error will be returned about the absence of an order with such a number. Therefore, we recommend working with iceberg orders on 'OrderID'.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the cancel request.
37	OrderID	Y	Int64	SPECTRA ID of the order to cancel.
48	SecurityID	Y	Int32	Instrument numeric ID.
1	Account	Y	String7	Client 7-symbol ID.

4.1.5. OrderReplaceRequest (message id=6007)

Request to change price/volume of an already added order.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the replace request.
37	OrderID	Y	Int64	SPECTRA ID of the order to cancel.
44	Price	Y	Decimal5	New order price.
38	OrderQty	Y	UInt32	New instrument unit quantity.
583	ClOrdLinkID	Y	Int32	External ID.
48	SecurityID	Y	Int32	Instrument numeric ID.
20019	Mode	Y	ModeEnum	Modes: • '0' DontChangeOrderQty - Do not change order's volume. The current volume remains in the system, the received value will be ignored. • '1' ChangeOrderQty - Change order's volume. If found in the system, the order will be replaced with a new one with the new price and volume. • '2' CheckOrderQtyAndCancelOrder - Cancel order. If the order's volume is not equal to that of the received one, the old order will be cancelled and removed from the system. Otherwise, the old order will be replaced with the new one. • '3' FixStyleReplace - Set new order's volume according to that of the received one excluding the filled part (not less than 0). If the received volume is less or equal to that of the filled part, the order will be cancelled.
1	Account	Y	String7	Client 7-symbol ID.

4.1.6. OrderIcebergReplaceRequest (message id=6010)

Request to change price of an already added iceberg order. Volume cannot be changed. Both the visible part number of the iceberg ('DisplayOrderID' from 'NewOrderIcebergResponse' message) and the identifier of the entire iceberg order ('OrderID') can be indicated as the identifier of the changed order ('OrderID'). Please note, that a command will work on 'DisplayOrderID' only if the visible part with such a number is still in the system (has not been matched), otherwise an error will be returned about the absence of an order with such a number. Therefore, we recommend working with iceberg orders on 'OrderID'.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the replace request.
37	OrderID	Y	Int64	SPECTRA ID of the order to cancel.
44	Price	Y	Decimal5	New order price.
583	ClOrdLinkID	Y	Int32	External ID.
48	SecurityID	Y	Int32	Instrument numeric ID.
1	Account	Y	String7	Client 7-symbol ID.

4.1.7. OrderMassCancelRequest (message id=6004)

Mass cancellation for orders under a criteria.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the mass cancellation request.
583	ClOrdLinkID	Y	Int32	External ID. If contains a value other than 0, all orders with the appropriate ClOrdLinkID will be cancelled. The values of the 'Side', 'SecurityGroup', and 'SecurityID' fields are ignored, but their values must be within the allowed range.
48	SecurityID	N	Int32	Instrument numeric ID. If the field ‘SecurityID’ contains value ‘nullValue’ or 0, then this criteria will not be applied for selecting orders.

Tag	Field	Mandatory	Type	Details
167	SecurityType	Y	SecurityTypeSet	Instrument type: • 0x1 - Futures
54	Side	Y	SideEnum	Order type depending on side: • '1' – Buy orders. • '2' – Sell orders. • '89' – All orders.
1	Account	Y	String7	Client 7-symbol ID. If the last three symbols in the field ‘Account’ are equal to the string ‘%%%’, then all orders for all client accounts will be cancelled.
1151	SecurityGroup	N	String25	Underlying asses code. If the field ‘SecurityGroup’ contains empty string or ‘%’, then all orders for all contracts will be cancelled.

4.1.8. NewOrderSingleResponse (message id=7000)

New order adding confirmation.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	ClOrdID that was submitted with the 'NewOrderSingle' message being accepted.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
432	ExpireDate	C	TimeStamp	Order expiration date. Mandatory for orders with TimeInForce=6 (GTD).
37	OrderID	Y	Int64	SPECTRA ID of the order.
20215	Flags	Y	FlagsSet	The field is a bit mask: • 0x1 - Day order • 0x2 - IOC order • 0x4 - OTC order • 0x10 - Client collateral was not checked while adding order • 0x200 - Do not check limits for options • 0x80000 - FOK order
44	Price	Y	Decimal5	Price.
48	SecurityID	Y	Int32	Instrument numeric ID.
38	OrderQty	Y	UInt32	Number of instrument units.
336	TradingSessionID	Y	Int32	Trading session ID
583	ClOrdLinkID	N	Int32	External ID.
54	Side	Y	SideEnum	Order side: • '1' - Buy • '2' - Sell

4.1.9. NewOrderIcebergResponse (message id=7013)

- New iceberg order adding confirmation.
- Disclosing of iceberg order.

Tag	Field	Mandatory	Type	Details
<Header>		Y		

Tag	Field	Mandatory	Type	Details
11	ClOrdID	Y	UInt64	ClOrdID that was submitted with the 'NewOrderIceberg' message being accepted.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
432	ExpireDate	N	TimeStamp	Order expiration date.
37	OrderID	Y	Int64	SPECTRA ID of entire order.
20037	DisplayOrderID	Y	Int64	ID of visible part.
20215	Flags	Y	FlagsSet	The field is a bit mask: • 0x1 - Day order • 0x200 - Do not check limits for options • 0x800000000000 - Iceberg order • 0x20000000000000 - Disclosing of iceberg order.
44	Price	Y	Decimal5	Price.
48	SecurityID	Y	Int32	Instrument numeric ID.
38	OrderQty	Y	UInt32	Number of instrument units in entire order.
1138	DisplayQty	Y	UInt32	Number of instrument units in visible part.
20036	DisplayVarianceQty	Y	UInt32	Value of random deviation of visible iceberg part.
336	TradingSessionID	Y	Int32	Trading session ID
583	ClOrdLinkID	N	Int32	External ID.
54	Side	Y	SideEnum	Order side: • '1' - Buy • '2' - Sell

4.1.10. OrderCancelResponse (message id=7003)

Order cancellation confirmation.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
11	ClOrdID	N	UInt64	ClOrdID from the 'OrderCancelRequest' message in the case of single order cancellation. ClOrdID assigned to the order when it was added or replaced in the case of mass order cancellation. ClOrdID='nullValue' is the sign of unsolicited message.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
37	OrderID	Y	Int64	SPECTRA ID of the order (for iceberg order - SPECTRA ID of entire order).
20215	Flags	Y	FlagsSet	The field is a bit mask: • 0x100000 - The record results from replacing the order • 0x200000 - The record results from cancelling the order • 0x400000 - The record results from mass cancelling • 0x20000000 - Flag of cancelling the left balance of the order because of a cross-trade • 0x100000000 - The record results from cancelling an order via ‘Cancel on Disconnect’ service • 0x2000000000 - The record results from cancelling an order via 'User Kill Switch' service • 0x800000000000 - Iceberg order
38	OrderQty	Y	UInt32	Instrument units quantity (for iceberg order - instrument units quantity in entire order).

Tag	Field	Manda-tory	Type	Details
336	TradingSessionID	Y	Int32	Trading session ID
583	ClOrdLinkID	N	Int32	External ID.

4.1.11. OrderReplaceResponse (message id=7005)

Order replacement confirmation.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	N	UInt64	ClOrdID that was submitted with the order replace request being accepted.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
37	OrderID	Y	Int64	SPECTRA ID of the order (for iceberg order - SPECTRA ID of entire order).
20216	PrevOrderID	Y	Int64	Previous order ID.
20215	Flags	Y	FlagsSet	The field is a bit mask: • 0x1 - Day order • 0x4 - OTC order • 0x10 - Client collateral was not checked while adding order • 0x200 - Do not check limits for options • 0x100000 - The record results from replacing the order
44	Price	Y	Decimal5	Price.
38	OrderQty	Y	UInt32	Instrument units quantity (for iceberg order - instrument units quantity in entire order).
336	TradingSessionID	Y	Int32	Trading session ID
583	ClOrdLinkID	N	Int32	External ID.

4.1.12. OrderMassCancelResponse (message id=7007)

Order mass cancellation confirmation.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	ClOrdID that was submitted with the mass cancellation request.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
533	TotalAffectedOrders	N	Int32	Orders cancelled.

4.1.13. ExecutionSingleReport (message id=7008)

Order matching confirmation.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client specified identifier of the order.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
37	OrderID	Y	Int64	SPECTRA ID of the order (for iceberg order - SPECTRA ID of entire order).
880	TrdMatchID	Y	Int64	Matched trade ID.
20215	Flags	Y	FlagsSet	The field is a bit mask: • 0x1 – Day-trade

Tag	Field	Manda-tory	Type	Details
				• 0x2 – IOC-trade • 0x4 – OTC-trade • 0x8 – Position transfer trade • 0x2000000 – Clearing session trade • 0x4000000 – Negotiated trade • 0x8000000 – Multi-leg trade • 0x10000000000 - The record are formed in the process of liquidation netting • 0x20000000000 – The active side in the trade. The order that led to the trade when added to the order-book • 0x40000000000 – The passive side in the trade. The order from the order-book involved in the trade • 0x800000000000 - Iceberg
31	LastPx	Y	Decimal5	Matched trade price.
32	LastQty	Y	UInt32	Instrument units in trade (for iceberg order - instrument units in operation for entire order).
38	OrderQty	Y	UInt32	Instrument units left in order (for iceberg order – remaining instrument units in entire order).
336	TradingSessionID	Y	Int32	Trading session ID
583	ClOrdLinkID	N	Int32	External ID.
48	SecurityID	Y	Int32	Instrument numeric ID.
54	Side	Y	SideEnum	Order side: • '1' - Buy • '2' - Sell

4.1.14. ExecutionMultilegReport (message id=7009)

Multi leg order matching confirmation.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
11	ClOrdID	Y	UInt64	Client ID of the order.
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
37	OrderID	Y	Int64	SPECTRA ID of the order (for iceberg order - SPECTRA ID of entire order).
880	TrdMatchID	Y	Int64	Matched trade ID.
20215	Flags	Y	FlagsSet	The field is a bit mask: • 0x1 – Day-trade • 0x2 – IOC-trade • 0x4 – OTC-trade • 0x8 – Position transfer trade • 0x2000000 – Clearing session trade • 0x4000000 – Negotiated trade • 0x8000000 – Multi-leg trade • 0x10000000000 - The record are formed in the process of liquidation netting

Tag	Field	Manda-tory	Type	Details
				• 0x20000000000 – The active side in the trade. The order that led to the trade when added to the order-book • 0x40000000000 – The passive side in the trade. The order from the order-book involved in the trade • 0x800000000000 - Iceberg
31	LastPx	Y	Decimal5	Matched trade price.
566	LegPrice	Y	Decimal5	First leg price
32	LastQty	Y	UInt32	Instrument units in trade (for iceberg order - instrument units in operation for entire order).
38	OrderQty	Y	UInt32	Instrument units left in order (for iceberg order – remaining instrument units in entire order).
336	TradingSessionID	Y	Int32	Trading session ID.
583	ClOrdLinkID	N	Int32	External ID.
48	SecurityID	Y	Int32	Instrument numeric ID.
54	Side	Y	SideEnum	Order side: • '1' - Buy • '2' - Sell

4.1.15. EmptyBook (message id=7010)

Closing of trading session.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
336	TradingSessionID	Y	Int32	Trading session ID.

4.1.16. SystemEvent (message id=7011)

Trading system events.

Tag	Field	Manda-tory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time of operation.
336	TradingSessionID	Y	Int32	Trading session ID.
1368	TradSesEvent	Y	TradSesEventEnum	Event type: • '101' SessionDataReady - Data has been successfully uploaded from clearing system to trading system and now ready for new trading session. • '102' IntradayClearingFinished - Intraday clearing session procedures have finished. • '104' IntradayClearingStarted - Intraday clearing session has started. • '105' ClearingStarted - The main clearing session has started. • '106' ExtensionOfLimitsFinished - Extension of limits has finished. • '108' BrokerRecalcFinished - Funds have been recalculated after closing the intraday clearing session.

4.2. Trading interaction scenarios

4.2.1. Adding orders

4.2.1.1. Adding a new order on a standard instrument by a client

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 'NewOrderSingleResponse' in reply, or rejects it with sending out message 'BusinessMessageReject'.

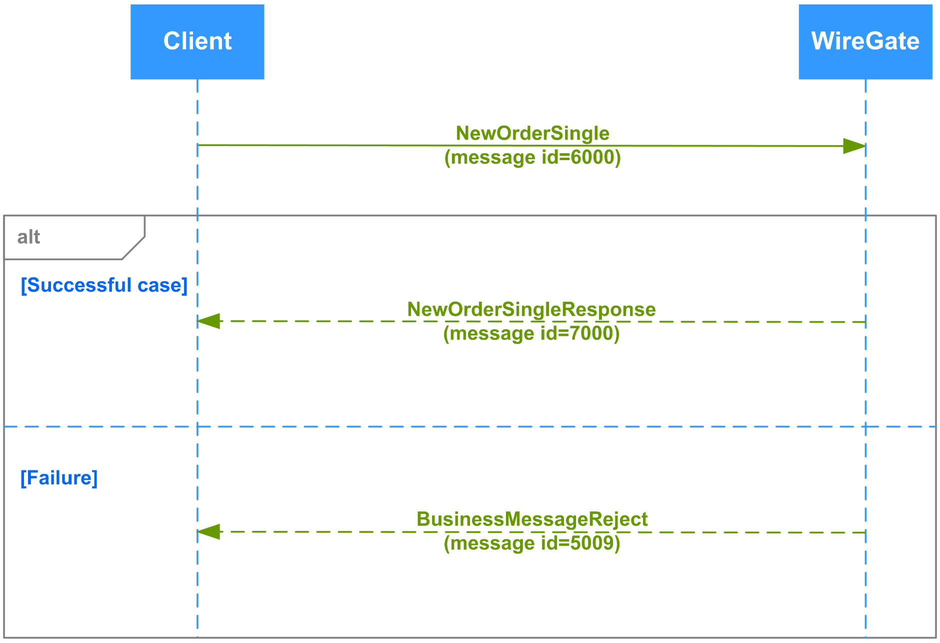

Figure 16. Diagram. Adding an order with TimeInForce=0(Day)

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.

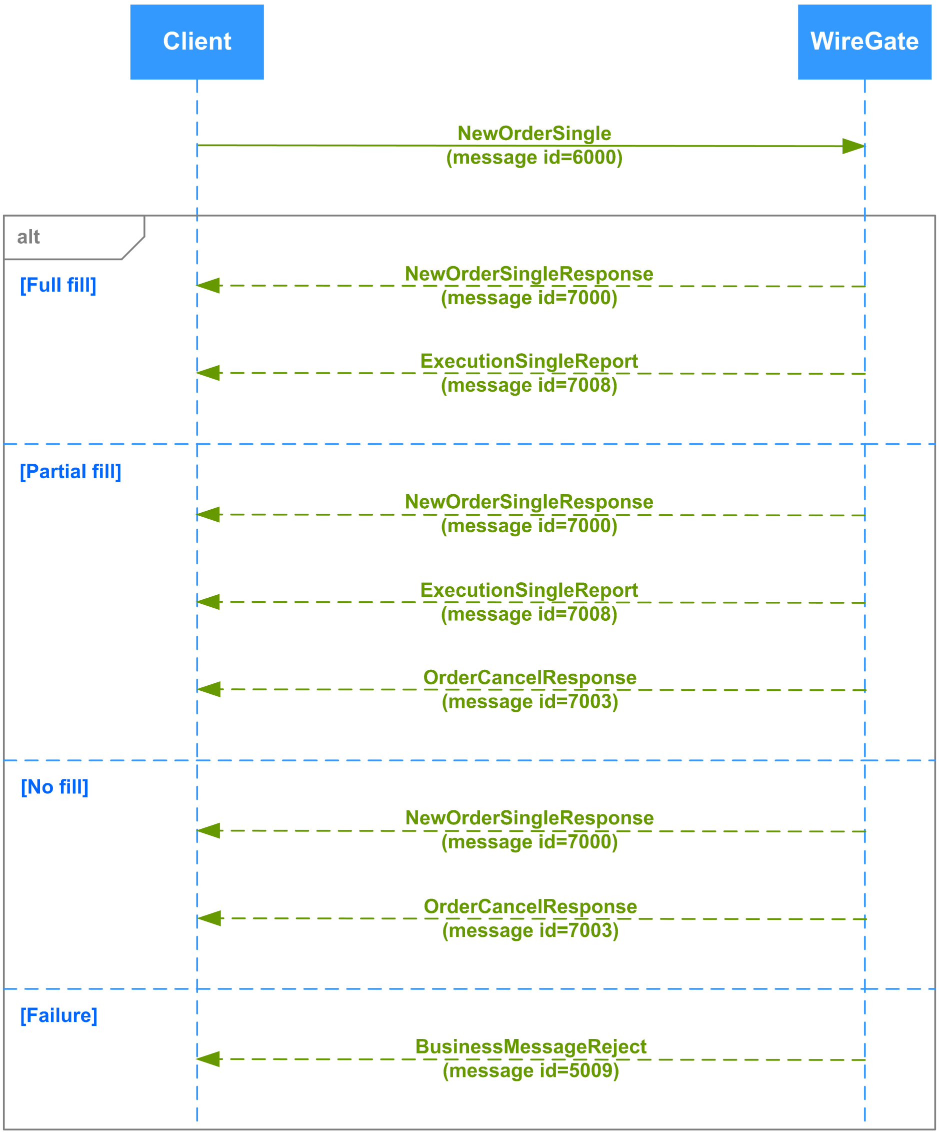

Figure 17. Diagram Adding an order with TimeInForce=3(IOC)

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.

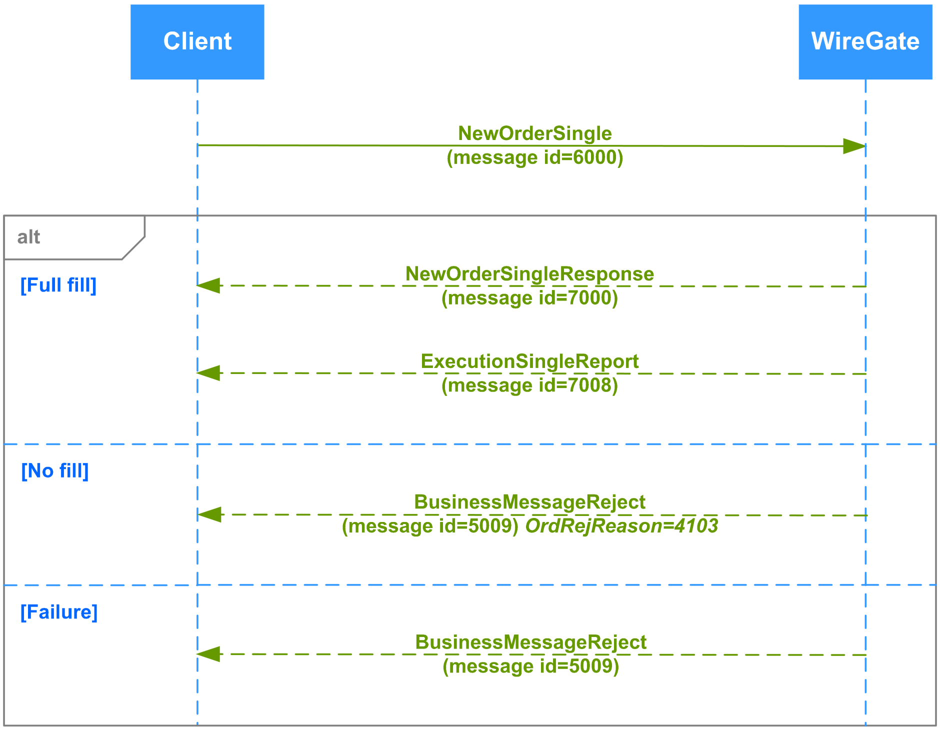

Figure 18. Diagram. Adding an order with TimeInForce=4(FOK)

4.2.1.2. Adding a new order on a multi leg instrument by a client

The client side sends a new order into the trading system (message 'NewOrderSingle'). The trading system either confirms the receiving with sending message 'NewOrderSingleResponse' in reply, or rejects it with sending out message 'BusinessMessageReject'.

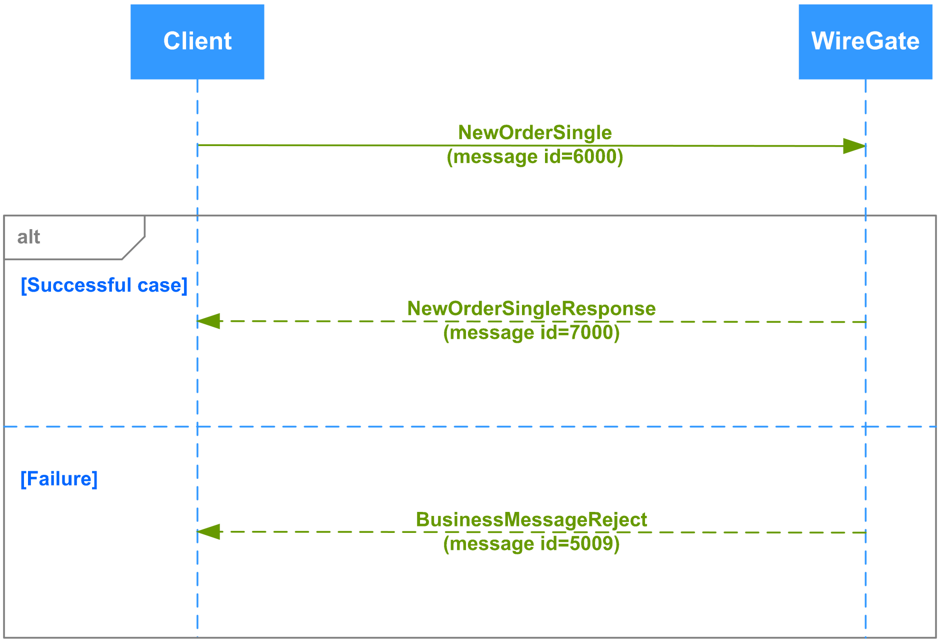

Figure 19. Diagram. Adding a new order on a multi leg instrument

4.2.1.3. Adding a new iceberg order by a client

The client side sends a new iceberg order into the trading system (message 'NewOrderIceberg'). The trading system either confirms the receiving with sending message 'NewOrderIcebrgResponse' in reply, or rejects it with sending out message 'BusinessMessageReject'.

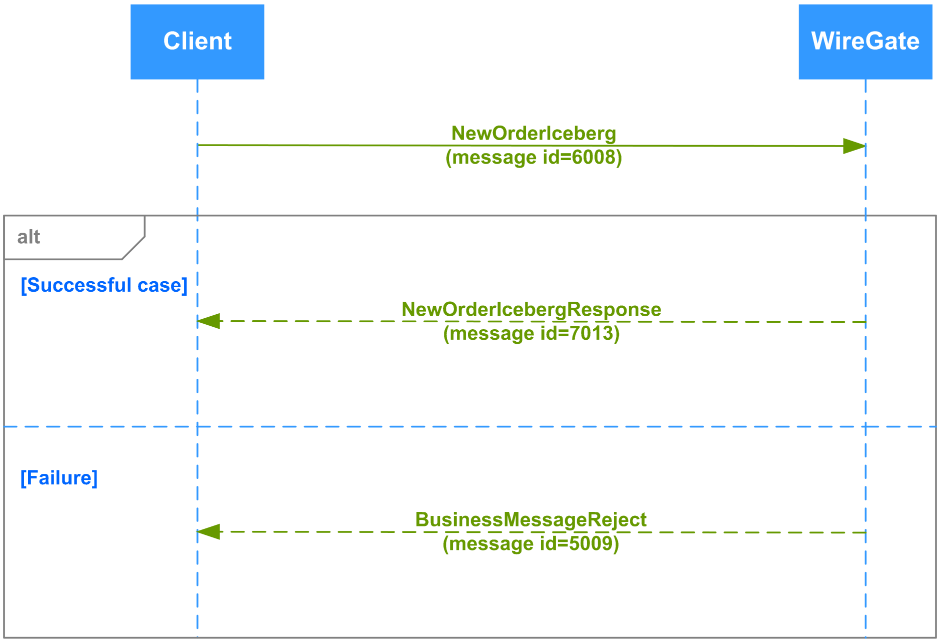

Figure 20. Diagram. Adding a new iceberg order

4.2.2. Order cancellation (deletion)

4.2.2.1. Client order cancellation by client

Any order that has been successfully added into the trading system can be cancelled by the client side using its exchange ID (OrderID). If the order had been already cancelled or already matched into a trade, the cancellation request will be rejected by the trading system.

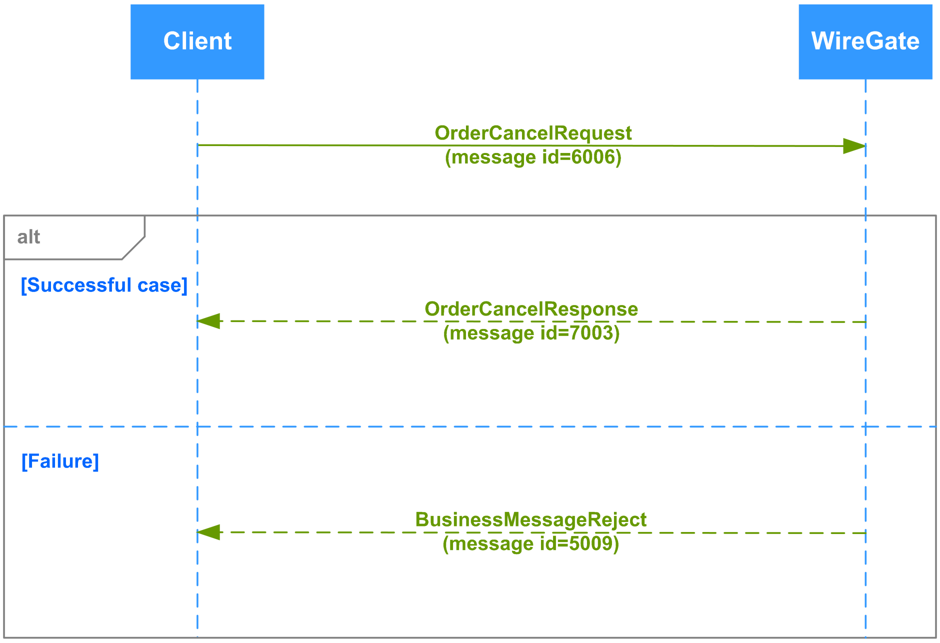

Figure 21. Diagram. Client order cancellation (by client)

4.2.2.2. Client order cancellation by Brokerage Firm

Scenario 1. A Brokerage Firm cancels a client order using its own login (message OrderCancelRequest'). Once an order has been canceled, WireGate confirms the cancellation by sending the message 'OrderCancelResponse' to BF; all other clients will receive the unsolicited messages 'OrderCancelResponse' where the field 'ClOrdID' contains 'nullValue'.

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

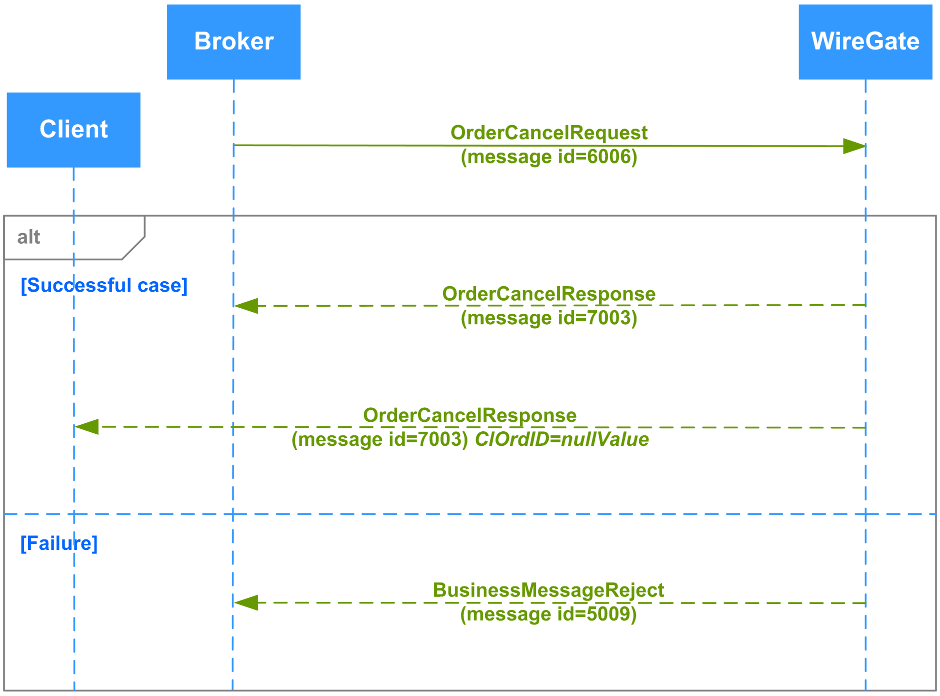

Figure 22. Diagram. Client order cancellation (by Brokerage Firm)

Scenario 2. A Brokerage Firm cancels a client's order using its own login either via the Plaza-2 gateway or via the trading terminal application. Once the order has been successfully cancelled, the client will receive the unsolicited message 'OrderCancelResponse' where the field 'ClOrdID' contains 'nullValue'.

4.2.2.3. Client iceberg order cancellation by client

Any iceberg order that has been successfully added into the trading system can be cancelled by the client side using its exchange ID ('OrderID'). If the order had been already cancelled or already matched into a trade, the cancellation request will be rejected by the trading system.

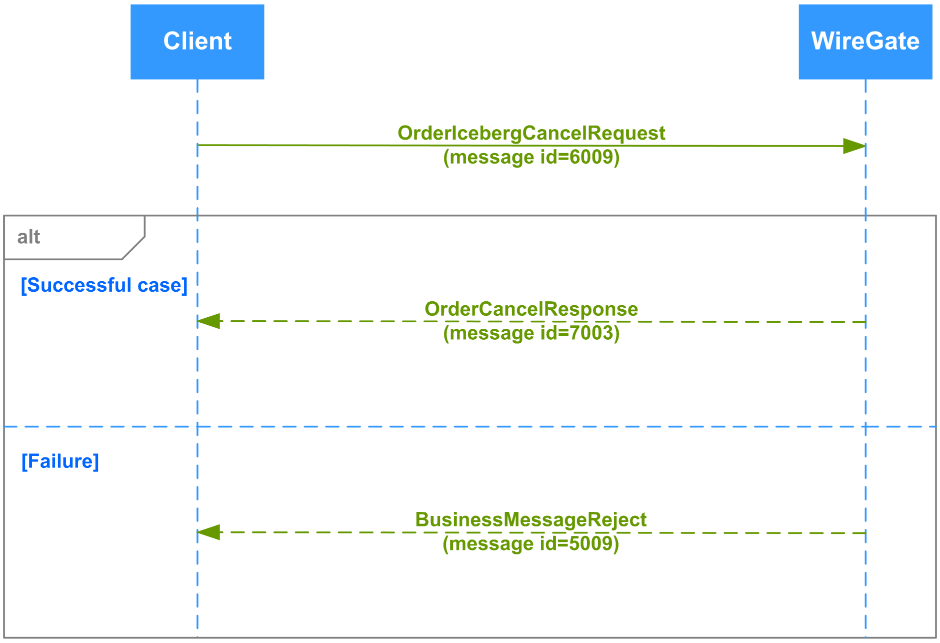

Figure 23. Diagram. Iceberg order cancellation by client

4.2.2.4. Client iceberg order cancellation by Brokerage Firm

Scenario 1. A Brokerage Firm cancels a client iceberg order using its own login (message 'OrderIcebergCancelRequest') via WireGate. Once an order has been canceled, WireGate confirms the cancellation by sending the message 'OrderCancelResponse' to BF; a client, who set the order, will receive the unsolicited messages 'OrderCancelResponse' where the field 'ClOrdID' contains 'nullValue'.

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

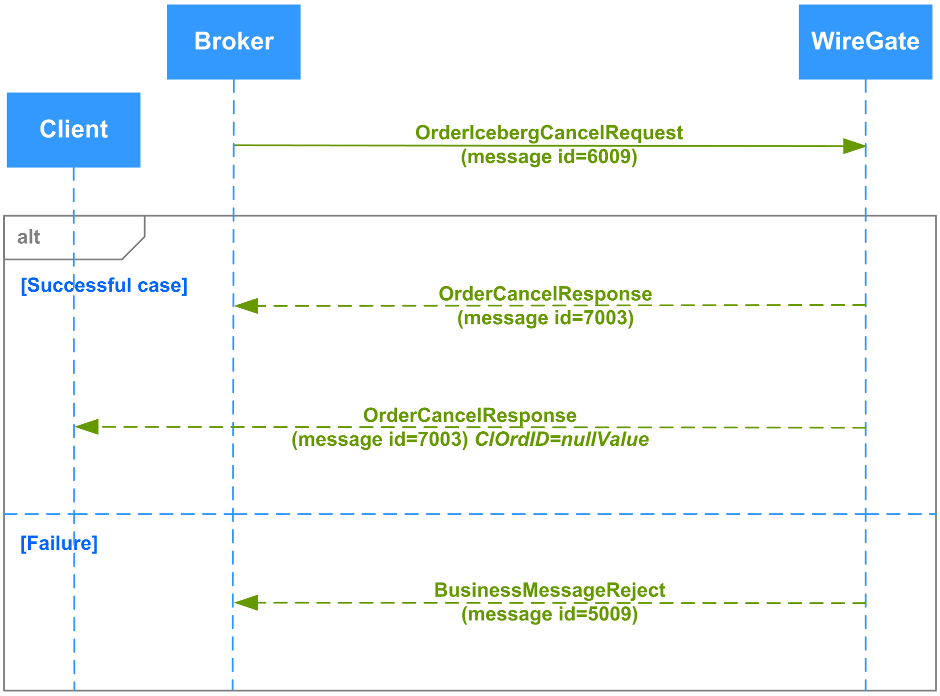

Figure 24. Diagram. Client iceberg order cancellation by Brokerage Firm

Scenario 2. A Brokerage Firm cancels a client's iceberg order using its own login either via the Plaza-2 gateway or via the trading terminal application. Once the order has been successfully cancelled, the client will receive the unsolicited 'OrderCancelResponse' message where the field 'ClOrdID' contains 'nullValue'.

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'. For every single order cancelled due to the request, there is a separate reply message 'OrderCancelResponse' sent by the server side. Also, the server side sends the message 'OrderMassCancelResponse' after processing the request.

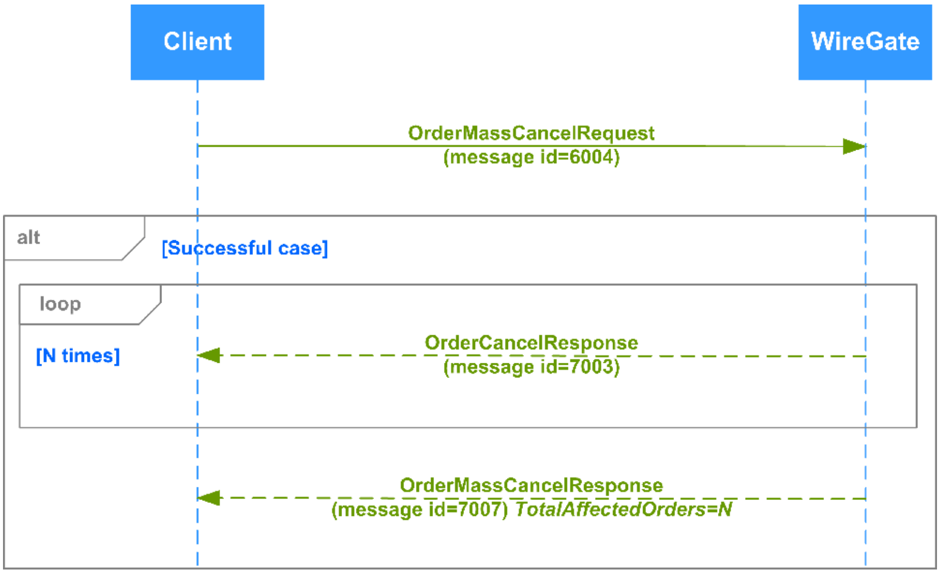

Figure 25. Diagram. Order mass cancellation by client

4.2.3.2. Client's order mass cancellation by Brokerage Firm

Scenario 1. A Brokerage Firm originates mass cancellation of its client's orders using a Brokerage Firm login (message 'OrderMassCancelRequest'). Once a single client's message has been cancelled, the BF receives the message 'OrderCancelResponse' for each cancelled order while the client receives unsolicited messages 'OrderCancelResponse' where the field 'ClOrdID' contains 'nullValue' for each cancelled message. This results in the following:

- the Brokerage Firm receives message 'OrderCancelResponse' for each cancelled order. The field 'ClOrdID' contains 'ClOrdID' assigned to the order on its addition or replacement. Also, the Brokerage Firm receives the message 'OrderMassCancelResponse' after processing the request ;
- each WireGate client with their login linked up with the client account will receive the unsolicited message 'OrderCancelResponse' where the field 'ClOrdID' value is 'nullValue'.

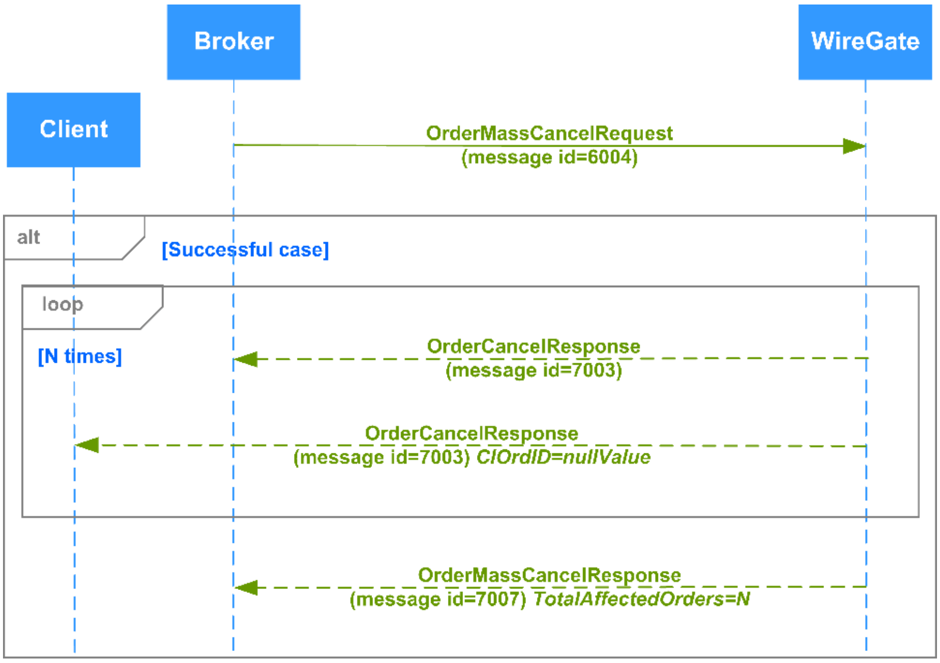

Figure 26. Diagram. Order mass cancellation using client trading code by Brokerage Firm

Scenario 2. A Brokerage Firm originates mass cancellation of its client's orders using its own login. Once the the cancellation has been successful, the client receives the unsolicited message 'OrderCancelResponse' via WireGate, where the field 'ClOrdID' contains 'nullValue'.

4.2.4. Order replacement

4.2.4.1. Order replacement by client

A request to change price/volume of an already added order.

Scenario 1. Changing an order using one of the modes: 0(DontChangeOrderQty), 1(ChangeOrderQty).

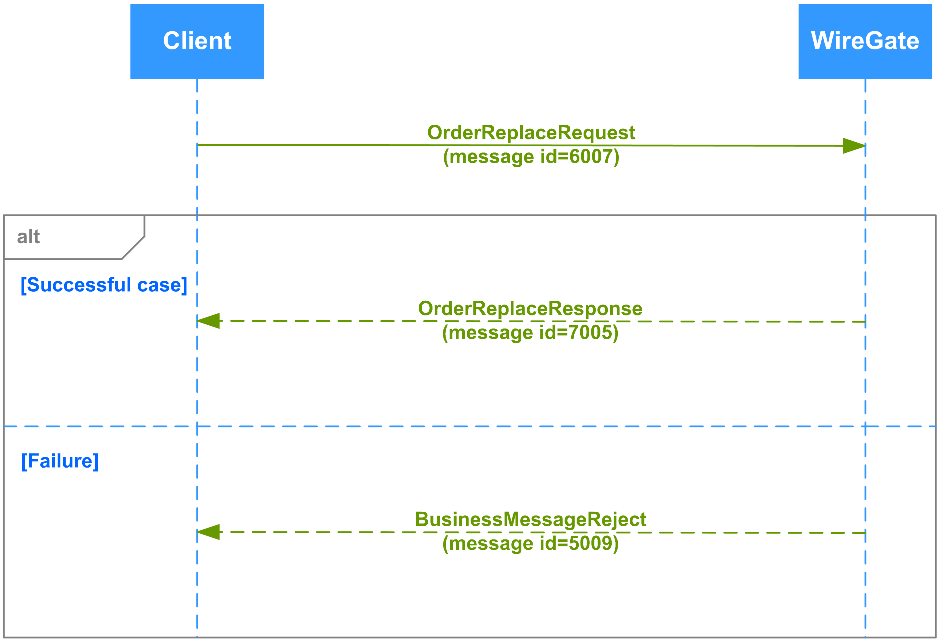

Figure 27. Diagram. Changing an order using one of the modes: 0(DontChangeOrderQty), 1(ChangeOrderQty)

Scenario 2. Changing an order using one of the modes: 2(CheckOrderQtyAndCancelOrder), 3(FixStyleReplace).

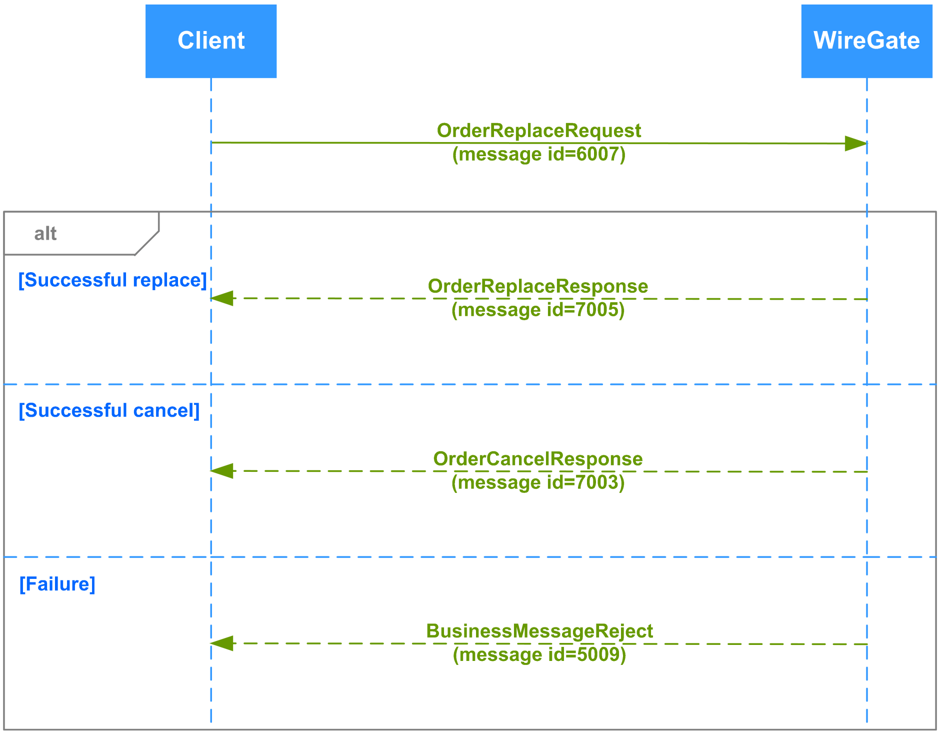

Figure 28. Diagram. Changing an order using one of the modes: 2(CheckOrderQtyAndCancelOrder), 3(FixStyleReplace)

4.2.4.2. Order replacement by Brokerage Firm

Scenario 1. A Brokerage Firm replaces a client's order using its own login via WIreGate (message 'OrderReplaceRequest'). Once an order has been replaced, the BF receives the confirmation message 'OrderReplaceResponse', while the client receives the unsolicited message 'OrderReplaceResponse', where the field 'ClOrdID' contains 'nullValue'.

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

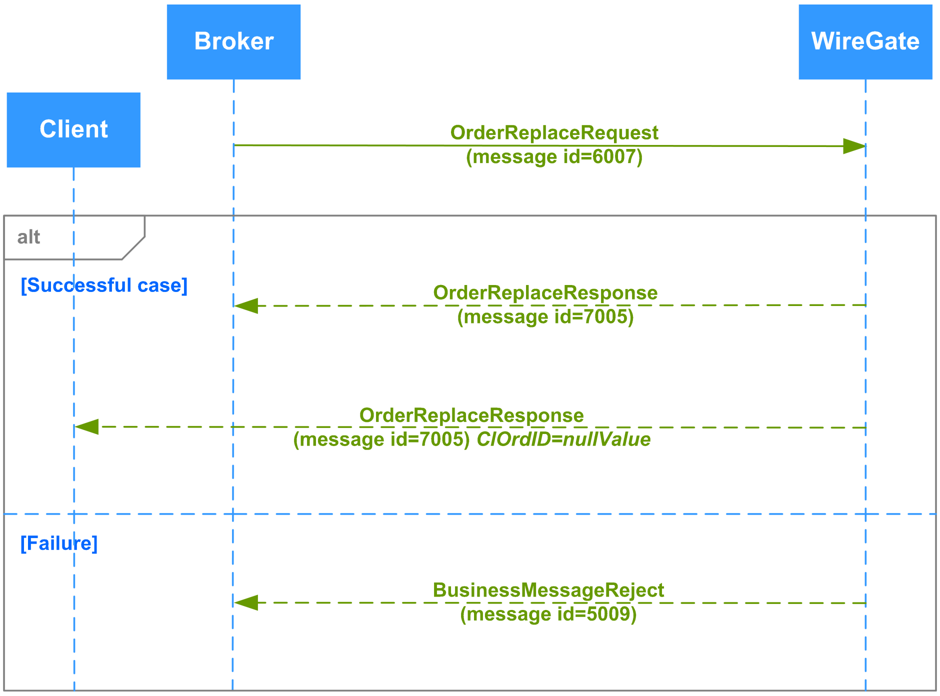

Figure 29. Diagram. Order replacement by Brokerage Firm

Attention! Once a WIreGate 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 Plaza-2 gate.

Scenario 2. A Brokerage Firm replaces a client's order using its own login either via the Plaza-2 gateway or via the trading terminal application. Once an order has been successfully replaced, the client receives the unsolicited message 'OrderReplaceResponse', where the field 'ClOrdID' contains 'nullValue', via WireGate.

4.2.4.3. Iceberg order replacement by client

Any iceberg order that has been successfully added into the trading system can be replaced by the client. side using its exchange ID ('OrderID'). For an iceberg order only the price can be changed, the volume is not available for change. If the order had been already cancelled or already matched into a trade, the replacement request will be rejected by the trading system.

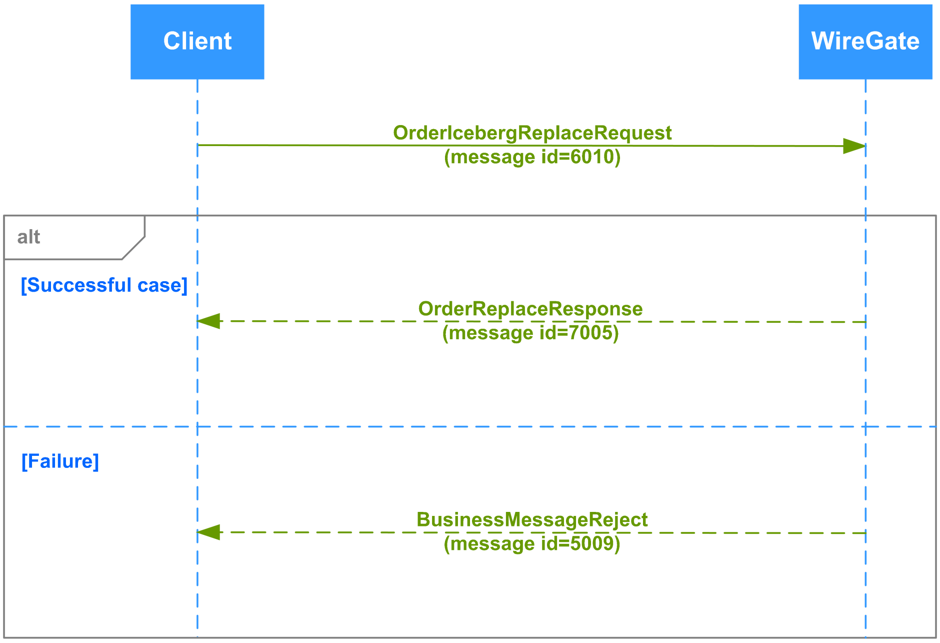

Figure 30. Diagram. Iceberg order replacement by client

4.2.4.4. Iceberg order replacement by Brokerage Firm

Scenario 1. A Brokerage Firm replaces a client's iceberg order using its own login via WIreGate ('OrderIcebergReplaceRequest' message). Once an order has been replaced, the BF receives the confirmation 'OrderReplaceResponse' message, and the client, who set an iceberg order, receives 'OrderReplaceResponse' unsolicited message, where the field 'ClOrdID' contains 'nullValue'.

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

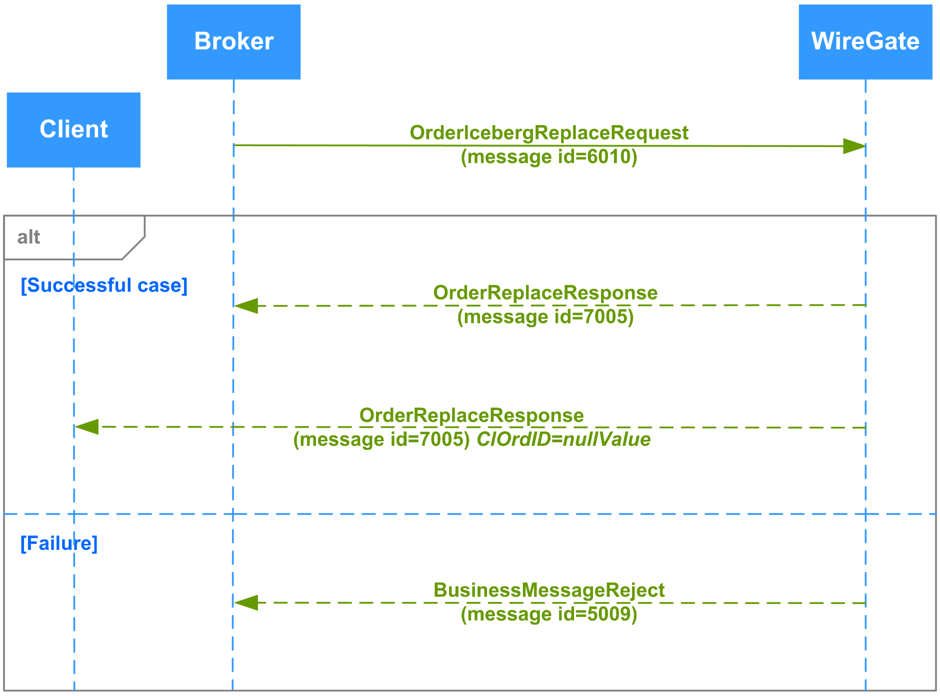

Figure 31. Diagram. Iceberg order replacement by Brokerage Firm

Attention! Once a WIreGate 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 Plaza-2 gate.

Scenario 2. A Brokerage Firm replaces a client's iceberg order using its own login either via the Plaza-2 gateway or via the trading terminal application. Once an order has been successfully replaced, the client receives 'OrderReplaceResponse' unsolicited message, where the field 'ClOrdID' contains 'nullValue', via WireGate.

4.2.5. Filling orders

4.2.5.1. Day order

A client side sends a new order into the trading system (message 'NewOrderSingle'). The trading system confirms receiving by sending message 'NewOrderSingleResponse' in reply. If the Exchange is ready to fill the order immediately, than the order is matched into trade (the trading system replies with message 'ExecutionSingleReport'). Partial fills will cause the message 'ExecutionSingleReport' to be sent several times.

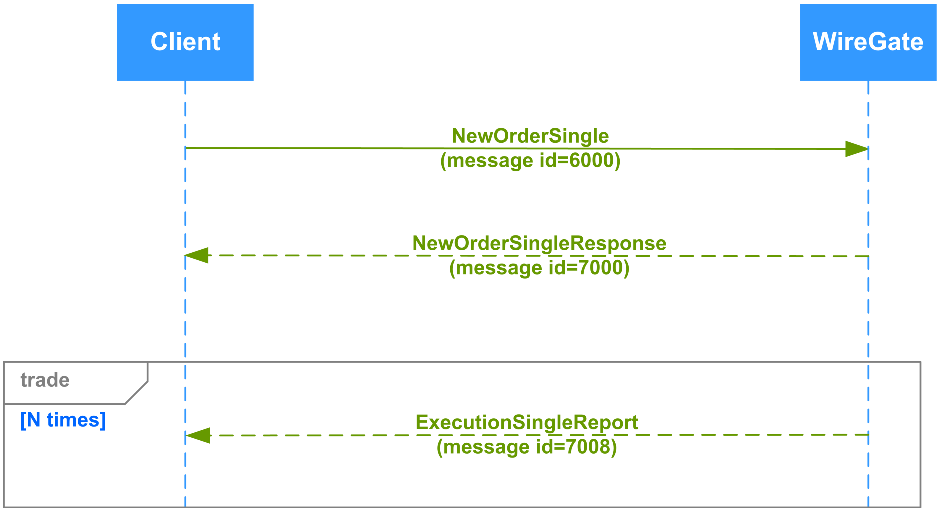

Figure 32. Diagram. Matching day order

4.2.5.2. Offsetting IOC order (partial fills)

In case of partial fills of the offsetting IOC order its unfilled part will be cancelled at auction end. A client side sends a new order into the trading system (message 'NewOrderSingle'). The trading system confirms the receiving by sending message 'NewOrderSingleResponse' in reply. After filling a part of the order, the trading system replies with the message 'ExecutionSingleReport'; the unfilled part of the order is cancelled (message 'OrderCancelResponse').

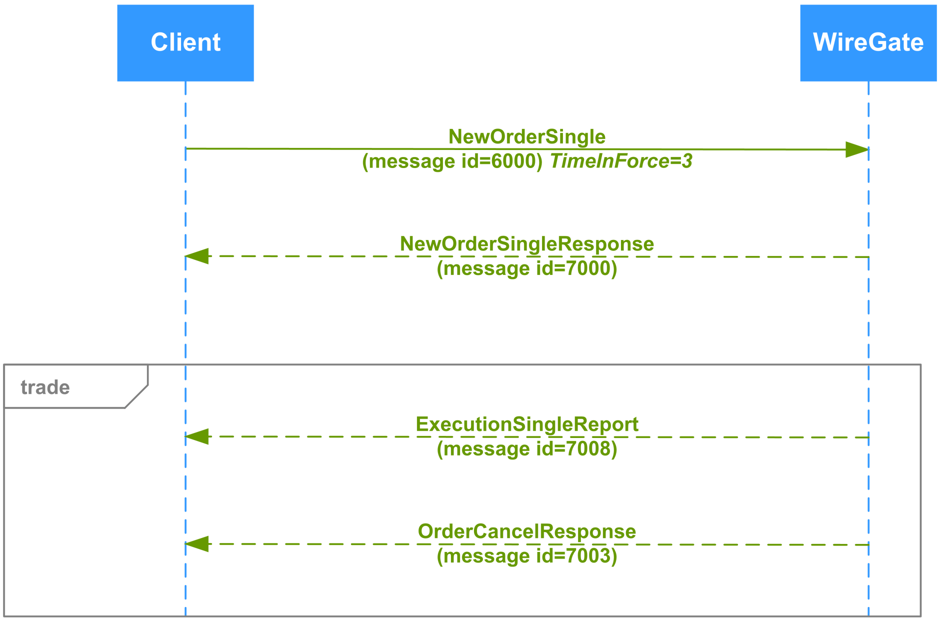

Figure 33. Diagram. Partial filling of the offsetting IOC order.

4.2.5.3. Trade matching after order had been changed by Brokerage Firm

One order added by Client1. Another order added by Client2. The Client2's order has been replaced by BF, which results in message 'OrderReplaceResponse' (operation successfully completed) sent by WIreGate to the BF. The Client2 receives the unsolicited message 'OrderReplaceResponse'. After the order has been replaced, both orders are matched into a trade, and BF and Client1 receive messages 'ExecutionSingleReport' from WireGate.

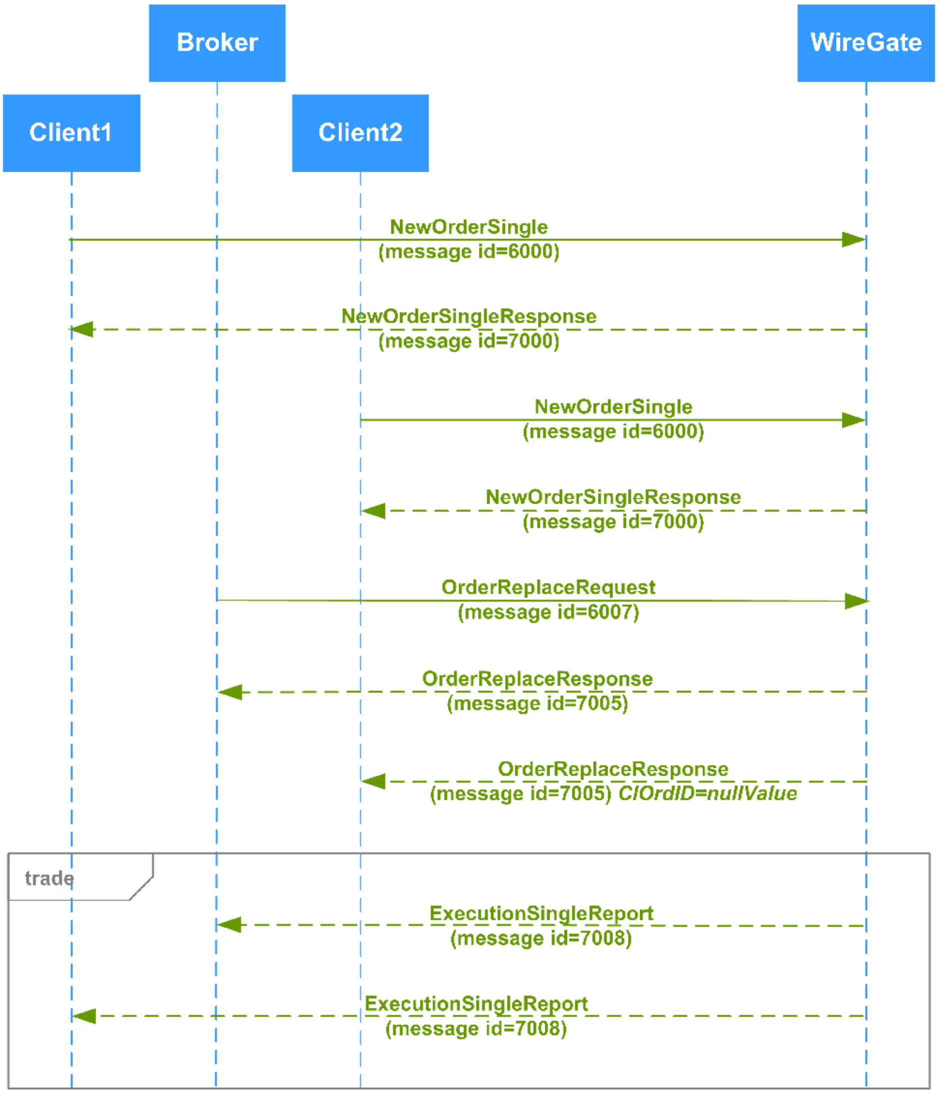

Figure 34. Diagram. Trade matching after order had been changed by Brokerage Firm

4.2.5.4. Filling a multileg order

Currently, multileg instruments are represented as calendar spreads on futures.

A client side sends a new order into the trading system (message 'NewOrderSingle'). The trading system confirms receiving by sending message 'NewOrderSingleResponse' in reply. If the Exchange is ready to fill the order immediately, then the order is matched into trade, and the System replies with a single message 'ExecutionMultilegReport' and two messages 'ExecutionSingleReport' as technical trades according to the instrument's legs.

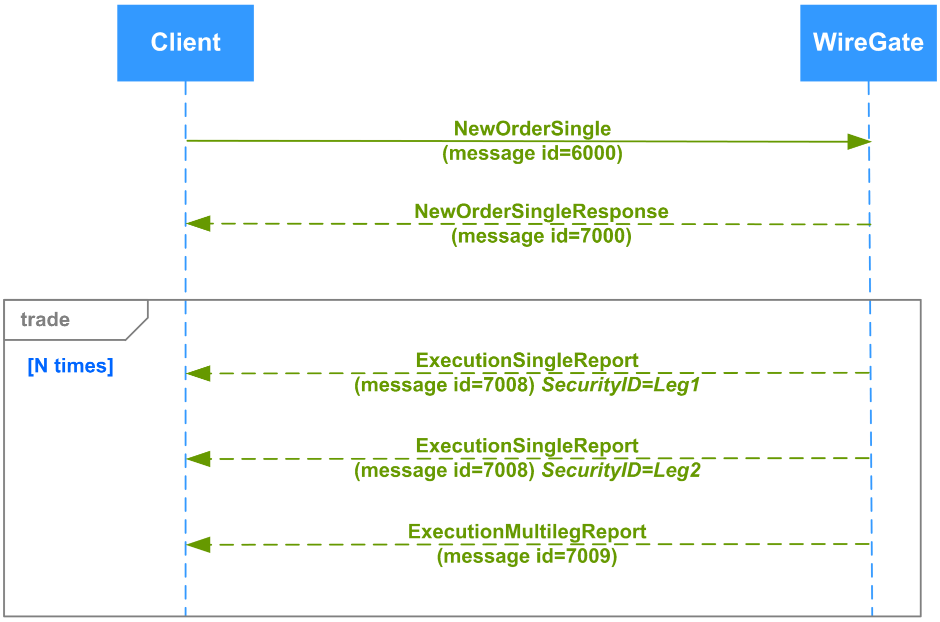

Figure 35. Diagram. Filling a multileg order

4.2.5.5. Trade matching of iceberg order

A client side sends a new iceberg order into the trading system ('NewOrderIceberg' message). The trading system confirms receiving by sending 'NewOrderIcebergResponse' message in reply. If the Exchange is ready to fill the visible part of iceberg order immediately, than the order is matched into trade. In this case, the trading system replies with 'ExecutionSingleReport' message where the 'Flags' field contains 'Iceberg' flag. Partial matching of visible part will cause 'ExecutionSingleReport' message to be sent several times.

New portion of the iceberg order pops up after matching all visible part to the trades. In this case, the trading system replies with 'NewOrderIcebergResponse' message where the 'Flags' field contains 'DisclosedIceberg' flag, and the 'DisplayOrderID' field contains the identifier of the new visible part, and the 'DisplayQty' field contains the number of instrument units in the visible part. Partial matching of visible part will cause 'ExecutionSingleReport' message to be sent several times.

The new visible part of the iceberg order is also matched to the trades and the next portion pops up, this continues until the entire iceberg order is exhausted. Accordingly, 'ExecutionSingleReport' messages will alternate with 'NewOrderIcebergResponse' messages where the 'Flags' field contains 'DisclosedIceberg' flag, and to be sent several times.

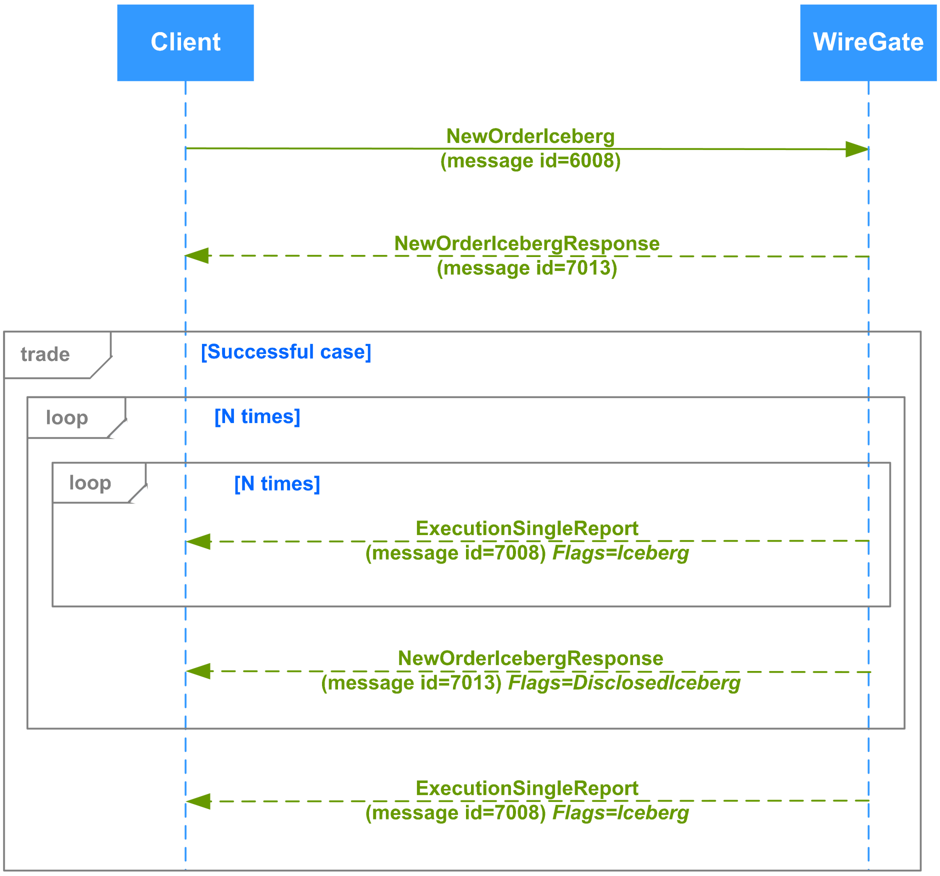

Figure 36. Diagram. Trade matching of iceberg order

4.2.6. Closing a trading session

The message 'EmptyBook' indicates closing of the current trading session. This results in automatic cancellation of all day orders in the trading system, so that the client side are to cancel all orders of that type, too.

4.2.7. Automatic order cancellation option (Cancel On Disconnect)

The trading system provides a client connection control feature ('Cancel On Disconnect' or 'COD'). This option allows to automatically cancel all client's active orders (anonymous orders without specified expiration time) on disconnect. The service enables by the client's request.

When an ID connects to the trading system having the 'Cancel On Disconnect' option enabled, the trading system starts to monitor its connection activity in the 'COD' mode.

The connection activity monitoring algorithm is as following:

- The system monitors the client's activity on transaction layer by checking whether or not the client periodically sends 'Heartbeat' messages (for details see '3.2.2. Session status monitoring').
- If the client does not send a single 'Heartbeat' message within the specified time period, or lost TCP-connection to WireGate, all they active orders are automatically cancelled.

Order cancellation conditions are as following:

- The session was closed by client via command ''Terminate'. Orders will be cancelled on disconnect.
- Client has lost connection to WireGate or become unable to operate properly due to an error. Once a connection loss has been detected, all orders will be cancelled.
- Client has been disconnected due to heartbeats timeout (specified in the message 'Establish'). Orders will be cancelled on disconnect.
- There may occur an issue when WireGate server becomes unable to operate properly; it loses connection with its clients while the client side does not inform the Trading System about the connection loss. The Trading System cannot handle such issues; if occurs, the issue should be resolved on the client side.

All orders added by clients with COD-mode enabled are cancelled when the evening trading session ends and when the Trading System has been restored after a failure.

When a next time the client connects to WireGate, they will receive the messages informing on all cancelled orders.

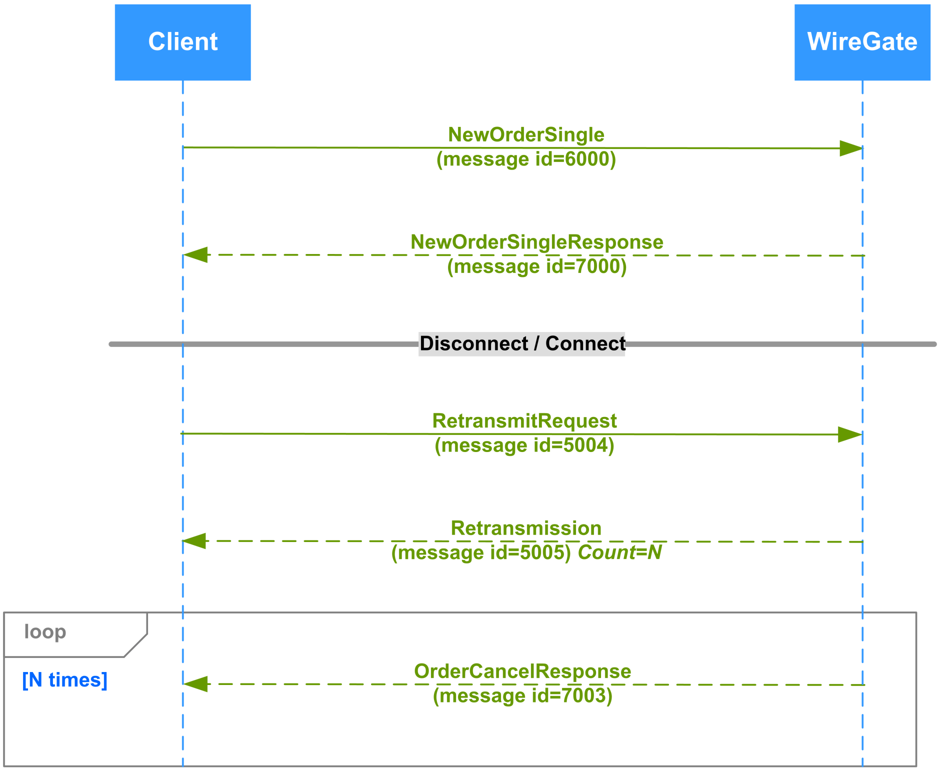

Figure 37. Diagram. Automatic order cancellation

4.2.8. Relisting multi-days orders

Multi-day orders are the orders with a fixed expiration date. Such orders will be automatically relisted for the next trading session, with a new number, and with a reference to the client order ID (field 'ClOrdID') taken from the last trading session. Upon relisting an order, the trading system verifies the instrument and the client availability, and the funds sufficiency. In case of a negative result, the order will be rejected.

The multi-day orders are relisted during the Evening clearing session. After an order has been successfully relisted, the clients will receive either message 'NewOrderSingleResponse' (adding a new order).

On the expiration date, a multi-day order will be automatically cancelled after closing the Evening trading session (in case there is any on that day).

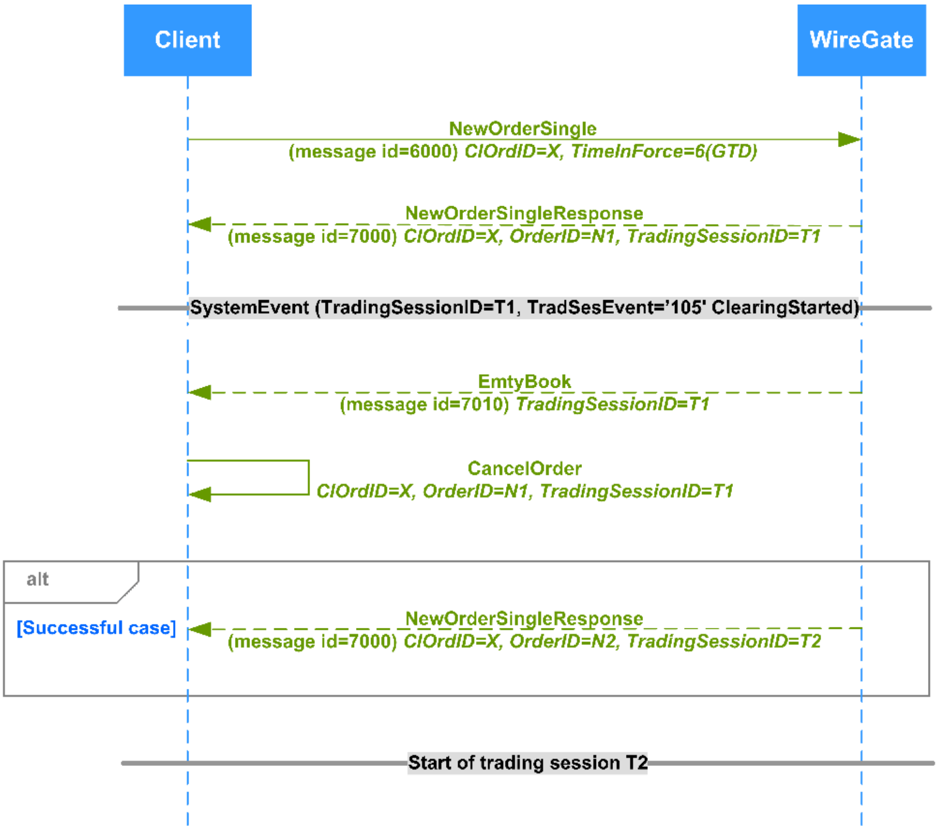
Client
WireGate
NewOrderSingle
(message id=6000) ClOrdID=X, TimeInForce=6(GTD)
NewOrderSingleResponse
(message id=7000) ClOrdID=X, OrderID=N1, TradingSessionID=T1
SystemEvent (TradingSessionID=T1, TradSesEvent='105' ClearingStarted)
EmtyBook
(message id=7010) TradingSessionID=T1
CancelOrder
ClOrdID=X, OrderID=N1, TradingSessionID=T1
alt
[Successful case]
NewOrderSingleResponse
(message id=7000) ClOrdID=X, OrderID=N2, TradingSessionID=T2
Start of trading session T2

Figure 38. Diagram. Relisting multi-days orders

4.2.9. Instrument ID

The numeric instrument ID should be obtained either from the Plaza-2 gateway or from the FAST gateway. In case of using the Plaza-2 gateway, a futures contract numeric ID is contained within the field 'isin_id' of the table 'fut_sess_contents' in the stream 'FORTS_REFDATA_REPL', and an option contract ID is contained within the field 'isin_id' of the table 'opt_sess_contents' in the stream 'FORTS_REFDATA_REPL'. In case of using the FAST gateway, a numeric ID is contained within the field 'SecurityId' of the message 'Security Definition'.

4.2.10. Client order ID

The WireGate checks the identifiers uniqueness. A client is required to provide unique 'ClOrdID' for orders with the session-long lifetime during a single trading session. A client is also required to provide the unique 'ClOrdID' for multi-day orders within their lifetime. Any non-unique 'ClOrdID' sent by client side will be rejected by the WireGate; also, the reply message 'SessionReject' containing error code 101 (ClOrdIdIsNotUnique) will be sent to the client side.

5. TWIME protocol customization for Brokerage Firms

The section contains decription of the TWIME protocol message customization scheme. This customization option may be useful, for example, to Brokerage Firms, which use pre-trading systems.

A Brokerage Firm is able to create and use their own messages, if needed. For such messages, there is a dedicated range of IDs (message id) reserved, from 30000 to 39999, inclusively. For fields, there is also a dedicated range of IDs (field id) reserved, from 30000 to 39999, inclusively.

Please note, that for the fields with IDs from 30000 to 39999, inclusively, there is a possibility of field names overlapping the ones reserved by the Exchange trading system (see 6. Message scheme). To avoid this, it is recomended to Brokerage Firm to add a prefix to the field names in the messages they create.

```
Message scheme example:
<message name="BrokerReject" id="30000">
<field name="ClOrdID" id="11" type="UInt64"/>
<field name="RefTagID" id="371" type="UInt32"/>
<field name="BrokerRejectReason" id="30000" type="UInt64"/>
<field name="BrokerRejectReasonText" id="30001" type="String25"/>
</message>
```

The example above shows a message containing a specific range of error codes, designated to transmit an error description in text format. An end client, on their side, will have to merge the Exchange's scheme with the Brokerage Firm's scheme upon building their TWIME application. Please note, that it is also necessary to add a new message processing algorythm into the program code.

It is important for a Brokerage Firm to comply with the principles of dividing messages between the session and application layers (for more details see 3. Session layer, 4. Application layer). In order to add an application layer message into the TWIME protocol, a Brokerage Firm is also obliged to provide the appropriate algorythms for session recovery with the missing message request, as it is stated in sections 3.2.3. Message numbering, 3.2.4. Message retransmission request, 3.2.5. Session short-term crash recovery, 3.2.6. Full session crash recovery.

6. Message scheme

The current message scheme is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="twime.xsl" type="text/xsl"?>
<sbe:messageSchema package="sbe" byteOrder="littleEndian" id="19781" version="5"
xmlns:sbe="http://fixprotocol.io/2016/sbe"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://fixprotocol.io/2016/sbe sbe.xsd">
  <types>
    <type name="Int8"  primitiveType="int8"  minValue="-128"
          maxValue="126" nullValue="127" presence="optional" />
    <type name="Int16" primitiveType="int16" minValue="-32768"
          maxValue="32766" nullValue="32767" presence="optional" />
    <type name="Int32" primitiveType="int32" minValue="-2147483648"
          maxValue="2147483646" nullValue="2147483647" presence="optional" />
    <type name="Int64" primitiveType="int64" minValue="-9223372036854775808"
          maxValue="9223372036854775806" nullValue="9223372036854775807" presence="optional" />

    <type name="UInt8"  primitiveType="uint8" minValue="0" maxValue="254"
          nullValue="255" presence="optional" />
    <type name="UInt16" primitiveType="uint16" minValue="0" maxValue="65534"
          nullValue="65535" presence="optional" />
    <type name="UInt32" primitiveType="uint32" minValue="0" maxValue="4294967294"
          nullValue="4294967295" presence="optional" />
    <type name="UInt64" primitiveType="uint64" minValue="0" maxValue="18446744073709551614"
          nullValue="18446744073709551615" presence="optional" />

    <type name="String7"  primitiveType="char" length="7"/>
    <type name="String20" primitiveType="char" length="20"/>
    <type name="String25" primitiveType="char" length="25"/>

    <type name="DeltaMillisecs" primitiveType="uint32" minValue="1000" maxValue="60000"
          presence="required" />
    <type name="TimeStamp" primitiveType="uint64" minValue="0" maxValue="18446744073709551614"
          nullValue="18446744073709551615" presence="optional"
          description="Time in number of nanoseconds since Unix epoch, UTC timezone" />

    <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>
      <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="EstablishmentRejectCodeEnum" encodingType="uint8">
      <validValue name="Unnegotiated"      >0</validValue>
      <validValue name="AlreadyEstablished">1</validValue>
      <validValue name="SessionBlocked"    >2</validValue>
      <validValue name="KeepaliveInterval" >3</validValue>
      <validValue name="Credentials"       >4</validValue>
      <validValue name="Unspecified"       >5</validValue>
    </enum>

    <enum name="SessionRejectReasonEnum" encodingType="uint8">
      <validValue name="ValueIsIncorrect"   >5</validValue>
      <validValue name="Other"              >99</validValue>
      <validValue name="SystemIsUnavailable">100</validValue>
      <validValue name="ClOrdIdIsNotUnique" >101</validValue>
    </enum>

    <enum name="TimeInForceEnum" encodingType="uint8">
      <validValue name="Day">0</validValue>
      <validValue name="IOC">3</validValue>
      <validValue name="FOK">4</validValue>
      <validValue name="GTD">6</validValue>
    </enum>

    <enum name="SideEnum" encodingType="uint8">
      <validValue name="Buy"      >1</validValue>
      <validValue name="Sell"     >2</validValue>
      <validValue name="AllOrders">89</validValue>
    </enum>

    <enum name="ModeEnum" encodingType="uint8">
      <validValue name="DontChangeOrderQty"         >0</validValue>
      <validValue name="ChangeOrderQty"             >1</validValue>
      <validValue name="CheckOrderQtyAndCancelOrder">2</validValue>
      <validValue name="FixStyleReplace"            >3</validValue>
    </enum>

    <set name="SecurityTypeSet" encodingType="uint8">
      <choice name="Future" description="Futures"               >0</choice>

    </set>

    <enum name="TradSesEventEnum" encodingType="uint8">
      <validValue name="SessionDataReady"         >101</validValue>
      <validValue name="IntradayClearingFinished" >102</validValue>
      <validValue name="IntradayClearingStarted"  >104</validValue>
      <validValue name="ClearingStarted"          >105</validValue>
      <validValue name="ExtensionOfLimitsFinished">106</validValue>
      <validValue name="BrokerRecalcFinished"     >108</validValue>
    </enum>

    <set name="FlagsSet" encodingType="uint64">
      <choice name="Day" description="Orders: Day order">0</choice>
      <choice name="IOC" description="Orders: IOC order">1</choice>
      <choice name="OTC" description="Orders and Trades: OTC order or OTC trade">2</choice>
      <choice name="PosTransfer" description="Trades: Position transfer trade">3</choice>
      <choice name="Collateral" description="Orders: Client collateral was not checked
      while adding order">4</choice>
      <choice name="DontCheckLimits" description="Orders: Do not check limits for options">9</choice>
      <choice name="FOK" description="Orders: FOK order">19</choice>
      <choice name="Replace" description="Orders:
      The record results from replacing the order">20</choice>
      <choice name="Cancel" description="Orders:
      The record results from cancelling the order">21</choice>
      <choice name="MassCancel" description="Orders:
      The record results from mass cancelling">22</choice>
      <choice name="Clearing" description="Trades: Clearing session trade">25</choice>
      <choice name="Negotiated" description="Trades: Negotiated trade">26</choice>
      <choice name="MultiLeg" description="Trades: Multi leg trade">27</choice>
      <choice name="CrossTrade" description="Orders:
      Flag of cancelling the left balance of the order because of a cross-trade">29</choice>
      <choice name="COD" description="Orders: The record results from cancelling an order via
      'Cancel on Disconnect' service">32</choice>
      <choice name="UKS" description="Orders: The record results from cancelling an order via
      'User Kill Switch' service">37</choice>
      <choice name="LiqNettingRF" description="Orders and Trades: The record are formed in the
      process of liquidation netting">40</choice>
      <choice name="ActiveSide" description="Trades: Flag of aggressive side">41</choice>
      <choice name="PassiveSide" description="Trades: Flag of passive side">42</choice>
      <choice name="Synthetic" description="Orders: Flag of the synthetic order">45</choice>
      <choice name="Iceberg" description="Orders and Trades: Iceberg order">47</choice>
      <choice name="DisclosedIceberg" description="Orders: The record results from disclosing
      of Iceberg order">53</choice>
    </set>



    <composite name="Decimal5" description="Decimal">
      <type name="mantissa" description="mantissa" minValue="-9999999999999999"
            maxValue="9999999999999999" primitiveType="int64" presence="required" />
      <type name="exponent" description="exponent" presence="constant" primitiveType="int8">-5</type>
    </composite>
```

```
  <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>
</types>

<sbe:message name="Establish" id="5000">
    <field name="Timestamp"         id="20204" type="TimeStamp" />
    <field name="KeepaliveInterval" id="20205" type="DeltaMillisecs" />
    <field name="Credentials"       id="20206" type="String20" />
</sbe:message>

<sbe:message name="EstablishmentAck" id="5001">
    <field name="RequestTimestamp"  id="20207" type="TimeStamp" />
    <field name="KeepaliveInterval" id="20205" type="DeltaMillisecs" />
    <field name="NextSeqNo"         id="20208" type="UInt64" />
</sbe:message>

<sbe:message name="EstablishmentReject" id="5002">
    <field name="RequestTimestamp"        id="20207" type="TimeStamp" />
    <field name="EstablishmentRejectCode" id="20209" type="EstablishmentRejectCodeEnum" />
</sbe:message>

<sbe:message name="Terminate" id="5003">
    <field name="TerminationCode" id="20210" type="TerminationCodeEnum" />
</sbe:message>

<sbe:message name="RetransmitRequest" id="5004">
    <field name="Timestamp" id="20204" type="TimeStamp" />
    <field name="FromSeqNo" id="20211" type="UInt64" />
    <field name="Count"     id="20212" type="UInt32" />
</sbe:message>

<sbe:message name="Retransmission" id="5005">
    <field name="NextSeqNo"        id="20208" type="UInt64" />
    <field name="RequestTimestamp" id="20207" type="TimeStamp" />
    <field name="Count"            id="20212" type="UInt32" />
</sbe:message>

<sbe:message name="Sequence" id="5006">
    <field name="NextSeqNo" id="20208" type="UInt64" />
</sbe:message>

<sbe:message name="FloodReject" id="5007">
    <field name="ClOrdID"       id="11"    type="UInt64" />
    <field name="QueueSize"     id="20213" type="UInt32" />
    <field name="PenaltyRemain" id="20214" type="UInt32" />
</sbe:message>

<sbe:message name="SessionReject" id="5008">
    <field name="ClOrdID"             id="11"  type="UInt64" />
    <field name="RefTagID"            id="371" type="UInt32" />
    <field name="SessionRejectReason" id="373" type="SessionRejectReasonEnum" />
</sbe:message>

<sbe:message name="BusinessMessageReject" id="5009">
    <field name="ClOrdID"      id="11"    type="UInt64" />
    <field name="Timestamp"    id="20204" type="TimeStamp" />
    <field name="OrdRejReason" id="103"   type="Int32" />
</sbe:message>

<sbe:message name="NewOrderSingle" id="6000">
    <field name="ClOrdID"     id="11"    type="UInt64" />
    <field name="ExpireDate"  id="432"   type="TimeStamp" />
    <field name="Price"       id="44"    type="Decimal5" />
    <field name="SecurityID"  id="48"    type="Int32" />
    <field name="ClOrdLinkID" id="583"   type="Int32" />
    <field name="OrderQty"    id="38"    type="UInt32" />
    <field name="TimeInForce" id="59"    type="TimeInForceEnum" />
    <field name="Side"        id="54"    type="SideEnum" />

    <field name="Account"     id="1"     type="String7" />
</sbe:message>

<sbe:message name="NewOrderIceberg" id="6008" sinceVersion="4">
    <field name="ClOrdID"            id="11"    type="UInt64" />
    <field name="ExpireDate"         id="432"   type="TimeStamp" />
    <field name="Price"              id="44"    type="Decimal5" />
    <field name="SecurityID"         id="48"    type="Int32" />
    <field name="ClOrdLinkID"        id="583"   type="Int32" />
    <field name="DisplayQty"         id="1138"  type="UInt32" />
    <field name="DisplayVarianceQty" id="20036" type="UInt32" />
    <field name="OrderQty"           id="38"    type="UInt32" />
    <field name="Side"               id="54"    type="SideEnum" />

    <field name="Account"            id="1"     type="String7" />
</sbe:message>

<sbe:message name="OrderCancelRequest" id="6006" sinceVersion="3">
    <field name="ClOrdID"     id="11"    type="UInt64" />
    <field name="OrderID"     id="37"    type="Int64" />
    <field name="SecurityID"  id="48"    type="Int32" />

    <field name="Account"     id="1"     type="String7" />
</sbe:message>

<sbe:message name="OrderIcebergCancelRequest" id="6009" sinceVersion="4">
    <field name="ClOrdID"     id="11"    type="UInt64" />
    <field name="OrderID"     id="37"    type="Int64" />
    <field name="SecurityID"  id="48"    type="Int32" />

    <field name="Account"     id="1"     type="String7" />
</sbe:message>

<sbe:message name="OrderReplaceRequest" id="6007" sinceVersion="3">
    <field name="ClOrdID"     id="11"    type="UInt64" />
    <field name="OrderID"     id="37"    type="Int64" />
    <field name="Price"       id="44"    type="Decimal5" />
    <field name="OrderQty"    id="38"    type="UInt32" />
    <field name="ClOrdLinkID" id="583"   type="Int32" />
    <field name="SecurityID"  id="48"    type="Int32" />
    <field name="Mode"        id="20019" type="ModeEnum" />

    <field name="Account"     id="1"     type="String7" />
</sbe:message>

<sbe:message name="OrderIcebergReplaceRequest" id="6010" sinceVersion="4">
    <field name="ClOrdID"     id="11"    type="UInt64" />
    <field name="OrderID"     id="37"    type="Int64" />
    <field name="Price"       id="44"    type="Decimal5" />
    <field name="ClOrdLinkID" id="583"   type="Int32" />
    <field name="SecurityID"  id="48"    type="Int32" />

    <field name="Account"     id="1"     type="String7" />
</sbe:message>

<sbe:message name="OrderMassCancelRequest" id="6004">
    <field name="ClOrdID"       id="11"   type="UInt64" />
    <field name="ClOrdLinkID"   id="583"  type="Int32" />
    <field name="SecurityID"    id="48"   type="Int32" />
    <field name="SecurityType"  id="167"  type="SecurityTypeSet" />
    <field name="Side"          id="54"   type="SideEnum" />
    <field name="Account"       id="1"    type="String7" />
    <field name="SecurityGroup" id="1151" type="String25" />
</sbe:message>

<sbe:message name="OrderMassCancelByBFLimitRequest" id="6005" sinceVersion="2">
    <field name="ClOrdID" id="11" type="UInt64"  />
    <field name="Account" id="1"  type="String7" />
</sbe:message>
```

```
<sbe:message name="NewOrderSingleResponse" id="7000">
    <field name="ClOrdID"          id="11"    type="UInt64" />
    <field name="Timestamp"        id="20204" type="TimeStamp" />
    <field name="ExpireDate"       id="432"   type="TimeStamp" />
    <field name="OrderID"          id="37"    type="Int64" />
    <field name="Flags"            id="20215" type="FlagsSet" />
    <field name="Price"            id="44"    type="Decimal5" />
    <field name="SecurityID"       id="48"    type="Int32" />
    <field name="OrderQty"         id="38"    type="UInt32" />
    <field name="TradingSessionID" id="336"   type="Int32" />
    <field name="ClOrdLinkID"      id="583"   type="Int32" />
    <field name="Side"             id="54"    type="SideEnum" />
</sbe:message>

<sbe:message name="NewOrderIcebergResponse" id="7013" sinceVersion="4">
    <field name="ClOrdID"            id="11"    type="UInt64" />
    <field name="Timestamp"          id="20204" type="TimeStamp" />
    <field name="ExpireDate"         id="432"   type="TimeStamp" />
    <field name="OrderID"            id="37"    type="Int64" />
    <field name="DisplayOrderID"     id="20037" type="Int64" />
    <field name="Flags"              id="20215" type="FlagsSet" />
    <field name="Price"              id="44"    type="Decimal5" />
    <field name="SecurityID"         id="48"    type="Int32" />
    <field name="OrderQty"           id="38"    type="UInt32" />
    <field name="DisplayQty"         id="1138"  type="UInt32" />
    <field name="DisplayVarianceQty" id="20036" type="UInt32" />
    <field name="TradingSessionID"   id="336"   type="Int32"  />
    <field name="ClOrdLinkID"        id="583"   type="Int32" />
    <field name="Side"               id="54"    type="SideEnum" />
</sbe:message>

<sbe:message name="OrderCancelResponse" id="7003">
    <field name="ClOrdID"          id="11"    type="UInt64" />
    <field name="Timestamp"        id="20204" type="TimeStamp" />
    <field name="OrderID"          id="37"    type="Int64" />
    <field name="Flags"            id="20215" type="FlagsSet" />
    <field name="OrderQty"         id="38"    type="UInt32" />
    <field name="TradingSessionID" id="336"   type="Int32" />
    <field name="ClOrdLinkID"      id="583"   type="Int32" />
</sbe:message>

<sbe:message name="OrderReplaceResponse" id="7005">
    <field name="ClOrdID"          id="11"    type="UInt64" />
    <field name="Timestamp"        id="20204" type="TimeStamp" />
    <field name="OrderID"          id="37"    type="Int64" />
    <field name="PrevOrderID"      id="20216" type="Int64" />
    <field name="Flags"            id="20215" type="FlagsSet" />
    <field name="Price"            id="44"    type="Decimal5" />
    <field name="OrderQty"         id="38"    type="UInt32" />
    <field name="TradingSessionID" id="336"   type="Int32" />
    <field name="ClOrdLinkID"      id="583"   type="Int32" />
</sbe:message>

<sbe:message name="OrderMassCancelResponse" id="7007">
    <field name="ClOrdID"             id="11"    type="UInt64" />
    <field name="Timestamp"           id="20204" type="TimeStamp" />
    <field name="TotalAffectedOrders" id="533"   type="Int32" />
</sbe:message>

<sbe:message name="ExecutionSingleReport" id="7008">
    <field name="ClOrdID"          id="11"    type="UInt64" />
    <field name="Timestamp"        id="20204" type="TimeStamp" />
    <field name="OrderID"          id="37"    type="Int64" />
    <field name="TrdMatchID"       id="880"   type="Int64" />
    <field name="Flags"            id="20215" type="FlagsSet" />
    <field name="LastPx"           id="31"    type="Decimal5" />
    <field name="LastQty"          id="32"    type="UInt32" />
    <field name="OrderQty"         id="38"    type="UInt32" />
    <field name="TradingSessionID" id="336"   type="Int32" />
    <field name="ClOrdLinkID"      id="583"   type="Int32" />
    <field name="SecurityID"       id="48"    type="Int32" />
    <field name="Side"             id="54"    type="SideEnum" />
```

```
  </sbe:message>

  <sbe:message name="ExecutionMultilegReport" id="7009">
      <field name="ClOrdID"          id="11"    type="UInt64" />
      <field name="Timestamp"        id="20204" type="TimeStamp" />
      <field name="OrderID"          id="37"    type="Int64" />
      <field name="TrdMatchID"       id="880"   type="Int64" />
      <field name="Flags"            id="20215" type="FlagsSet" />
      <field name="LastPx"           id="31"    type="Decimal5" />
      <field name="LegPrice"         id="566"   type="Decimal5" />
      <field name="LastQty"          id="32"    type="UInt32" />
      <field name="OrderQty"         id="38"    type="UInt32" />
      <field name="TradingSessionID" id="336"   type="Int32" />
      <field name="ClOrdLinkID"      id="583"   type="Int32" />
      <field name="SecurityID"       id="48"    type="Int32" />
      <field name="Side"             id="54"    type="SideEnum" />
  </sbe:message>

  <sbe:message name="EmptyBook" id="7010">
      <field name="Timestamp"        id="20204" type="TimeStamp" />
      <field name="TradingSessionID" id="336"   type="Int32" />
  </sbe:message>

  <sbe:message name="SystemEvent" id="7011">
      <field name="Timestamp"        id="20204" type="TimeStamp" />
      <field name="TradingSessionID" id="336"   type="Int32" />
      <field name="TradSesEvent"     id="1368"  type="TradSesEventEnum" />
  </sbe:message>

</sbe:messageSchema>
```

7. List of return codes

Return code	Description
-1	Error performing operation.
0	Operation successful.
1	User not found.
2	Brokerage Firm code not found.
3	Session inactive.
4	Session halted.
5	Error performing operation.
6	Insufficient rights to perform operation.
7	Cannot perform operation: incorrect Clearing Firm account.
8	Insufficient rights to perform order deletion.
9	Operations with orders are blocked for the firm by the Clearing Centre.
10	Insufficient funds to reserve.
12	Options premium exceeds the limit allowed.
13	Total amount of positions exceeds the market limit.
14	Order not found.
25	Unable to add order: prohibited by the Trading Administrator.
26	Unable to open position: prohibited by Trading Administrator.
27	Unable to open short position: prohibited by Trading Administrator.
28	Unable to perform operation: insufficient rights.
31	Matching order for the same account/UIN is not allowed.
32	Trade price exceeds the limit allowed.
33	Operations with orders are blocked for this firm by the Clearing Administrator.
34	Cannot perform operation: wrong client code.
35	Invalid input parameters.
36	Cannot perform operation: wrong underlying.
37	Multi-leg orders cannot be moved.
38	Negotiated orders cannot be moved.
39	Price is not a multiple of the tick size.
40	Unable to add Negotiated order: counterparty not found.
41	User's trading rights have expired or are not valid yet.
42	Operations are prohibited by Chief Trader of Clearing Firm.
44	Clearing Firm's Chief Trader flag not found for this firm.
45	Unable to add Negotiated orders: no RTS code found for this firm.
46	Only Negotiated orders are allowed for this security.
47	There was no trading in this security during the session specified.
48	This security is being delivered. Only Negotiated orders from all Brokerage Firms within the same Clearing Firm are allowed.
49	Unable to add Negotiated order: a firm code must be specified.
50	Order not found.
53	Error setting input parameter - amount.
54	Unable to perform operation: exceeded operations quota for this client.
56	Unable to perform operations using this login/code pair: insufficient rights. Please contact the Trading Administrator.
57	Unable to connect to the Exchange server: insufficient rights. Please contact the Trading Administrator.
58	Unable to add orders without verifying client funds sufficiency: insufficient rights.
60	Auction halted for all risk-netting instruments.
61	Trading halted in all risk-netting instruments.
62	Trading halted on the MOEX Derivatives Market.

Return code	Description
63	Auction halted in all risk-netting instruments with this underlying.
64	Trading halted in all risk-netting instruments with this underlying.
65	Trading halted on all boards in all securities with this underlying.
66	Trading halted in this risk-netting instrument.
67	Unable to open positions in this risk-netting instrument: prohibited by the Trading Administrator.
68	Unable to add orders for all risk-netting instruments: prohibited by the Brokerage Firm.
69	Unable to add orders for all risk-netting instruments: prohibited by the Chief Trader.
70	Trading operation is not supported.
71	Position size exceeds the allowable limit.
72	Order is being moved.
73	Aggregated buy order quantity exceeds the allowable limit.
74	Aggregated sell order quantity exceeds the allowable limit.
75	Non-trading operation was unsuccessful due to timeout.
76	No record to delete.
77	No identification data for the specified trading account.
78	Clearing Firm code not found.
79	Operations are prohibited by the Clearing Administrator.
80	Non trading operation is not supported.
81	Cannot perform operation: input validation error of data relevance.
200	Collateral calculation parameters are being changed by the Trading Administrator.
201	Collateral calculation parameters are being changed by the Trading Administrator.
202	Collateral calculation parameters are being changed by the Trading Administrator.
203	Collateral calculation parameters are being changed by the Trading Administrator.
204	Collateral calculation parameters are being changed by the Trading Administrator.
205	Collateral calculation parameters are being changed by the Trading Administrator.
206	Collateral calculation parameters are being changed by the Trading Administrator.
207	Collateral calculation parameters are being changed by the Trading Administrator.
208	Collateral calculation parameters are being changed by the Trading Administrator.
300	Prohibition of all trading operations due to revocation/ suspension of the dealer license of this CF.
301	Prohibition of opening positions due to revocation/ suspension of the dealer license of this CF.
302	Prohibition of all trading operations due to revocation/ suspension of the brokerage license of this CF.
303	Prohibition of opening positions due to revocation/ suspension of the brokerage license of this CF.
304	Prohibition of all trading operations due to revocation/ suspension of the license of exchange intermediary for this CF.
305	Prohibition of opening positions due to revocation/ suspension of the license of exchange intermediary for this CF
306	Prohibition of all trading operations due to revocation/ suspension of the asset management license of this CF.
307	Prohibition of opening positions due to revocation/ suspension of the asset management license of this CF.
310	Unable to add order: prohibited by Clearing Administrator.
311	Unable to open position: prohibited by Clearing Administrator.
312	Unable to open short position: prohibited by Clearing Administrator.
314	Unable to add orders in the client account: prohibited by the Trader.
315	Unable to open position in the client account: prohibited by the Trader.
316	Unable to open short position in the client account: prohibited by the Trader.
317	Amount of buy/sell orders exceeds the limit allowed.
318	Unable to add order for the client account: client does not have a deposit account for settlement of Money Market securities. Prohibited by Clearing Administrator.
320	Amount of active orders exceeds the limit allowed for the client account for this security.
331	Insufficient funds in the Settlement Account.
332	Insufficient client funds.

Return code	Description
333	Insufficient Brokerage Firm funds.
335	Unable to buy: amount of securities exceeds the limit set for the client.
336	Unable to buy: amount of securities exceeds the limit set for the Brokerage Firm.
337	Unable to sell: amount of securities exceeds the limit set for the client.
338	Unable to sell: amount of securities exceeds the limit set for the Brokerage Firm.
339	Collateral recalculation in progress.
380	Trading restricted while intraday clearing is in progress.
381	Trading restricted while intraday clearing is in progress: cannot delete orders.
382	Trading restricted while intraday clearing is in progress: cannot move orders.
383	Non-trading operations restricted while intraday clearing is in progress.
680	Insufficient client funds.
681	Insufficient Clearing Firm funds.
682	Insufficient funds to increase position.
3000	Modification and cancellation of the quote is prohibited due to Speed bump.
3001	Operation is prohibited.
4000	Invalid input parameters.
4001	Unable to perform operation: insufficient rights.
4002	Unable to change trading limit for the client: no active trading sessions.
4004	Unable to change trading limit for the client: client code not found.
4005	Unable to change the trading limit for the client: insufficient funds.
4006	Invalid input parameters: this "Operating mode" is not supported.
4007	Invalid input parameters: the "Funds limit" parameter is not a number.
4008	Invalid input parameters: the "Clients collateral ratio" parameter is not a number.
4009	Invalid input parameters: invalid value for "Clients collateral ratio" parameter.
4010	Invalid input parameters: invalid value for "Minus check flag" parameter.
4011	Invalid input parameters: invalid value for "Flag of automatic adjustment of the limit by the amount of income after clearing" parameter.
4012	Unable to set trading limit for the client: error performing operation.
4013	Unable to set trading limit for the client: error performing operation.
4014	Unable to change parameters: no active trading sessions.
4015	Unable to change parameters: client code not found.
4016	Unable to change parameters: underlying's code not found.
4017	Invalid input parameters: invalid value for "Funds limit" parameter.
4018	Collateral calculation parameters are being changed by the Trading Administrator.
4021	Unable to set requested amount of pledged funds for Clearing Firm: insufficient amount of free funds.
4022	Unable to set requested amount of funds for Clearing Firm: insufficient amount of free funds.
4023	Unable to change trading limit for the Brokerage Firm: no active trading sessions.
4024	Unable to change trading limit for the Brokerage Firm: the Brokerage Firm is not registered for trading.
4025	Unable to set requested amount of pledged funds for the Brokerage Firm: insufficient amount of free funds in the Clearing Firm.
4026	Unable to set requested amount of funds for the Brokerage Firm: insufficient amount of free funds in the balance of the Separate Account.
4027	Unable to set requested amount of pledged funds for the Clearing Firm: insufficient amount of pledged funds in the balance of the Separate Account.
4028	Unable to set requested amount of funds for the Brokerage Firm: insufficient amount of free funds in the Clearing Firm.
4030	Unable to change parameters for the Brokerage Firm: no active sessions.
4031	Unable to change parameters for the Brokerage Firm: Brokerage Firm code not found.
4032	Unable to change parameters for the Brokerage Firm: underlying's code not found.
4033	Unable to change parameters for the Brokerage Firm: insufficient rights to trade this underlying.

Return code	Description
4034	Transfer of pledged funds from the Separate account is prohibited.
4035	Transfer of collateral is prohibited.
4040	Unable to change Brokerage Firm limit on risk-netting: no active sessions.
4041	Unable to change Brokerage Firm limit on risk-netting: Brokerage Firm is not registered for trading.
4042	Unable to change Brokerage Firm limit on risk-netting: Brokerage Firm code not found.
4043	Unable to change Brokerage Firm limit on risk-netting: error performing operation.
4044	Unable to change Brokerage Firm limit on risk-netting: error performing operation.
4045	Unable to delete Brokerage Firm limit on risk-netting: error performing operation.
4046	Unable to remove Chief Trader's restriction on trading in risk-netting instruments: insufficient rights.
4050	Unable to process the exercise request: restricted by the Chief Trader.
4051	Unable to process the exercise request: restricted by the Brokerage Firm.
4052	Unable to process the exercise request: wrong client code and/or security.
4053	Unable to process the exercise request: cannot delete orders during the intraday clearing session.
4054	Unable to process the exercise request: cannot change orders during the intraday clearing session.
4055	Unable to process the exercise request: order number not found.
4060	Unable to process the exercise request: insufficient rights.
4061	Unable to process the exercise request: deadline for submitting requests has passed.
4062	Unable to process the exercise request: client code not found.
4063	Unable to process the exercise request: request not found.
4064	Unable to process the exercise request: insufficient rights.
4065	Unable to process the exercise request: option contract not found.
4066	Unable to process the exercise request: request to disable automatic exercise may only be submitted on the option's expiration date.
4067	Unable to process the exercise request: error performing operation.
4068	Unable to process the exercise request: error performing operation.
4069	Unable to process the exercise request: error performing operation.
4070	Unable to process the exercise request: insufficient amount of positions in the client account.
4090	No active sessions.
4091	Client code not found.
4092	Underlying's code not found.
4093	Futures contract not found.
4094	Futures contract does not match the selected underlying.
4095	Partial selection of futures contracts not accepted: underlying flag set 'For all'.
4096	Unable to remove limit: no limit set.
4097	Unable to remove: the Chief Trader's restriction cannot be removed by Brokerage Firm trader.
4098	Security not found in the current trading session.
4099	Both securities must have the same underlying.
4100	Exercise date of the near leg of a multi-leg order must not be later than that of the far leg.
4101	Unable to make a multi-leg order: lots are different.
4102	No position to move.
4103	The FOK order has not been fully matched.
4104	Anonymous repo order must contain a repo type.
4105	Order containing a repo type is restricted in this multi-leg order.
4106	Multi-leg orders can be added only on the Money Market.
4107	This procedure is not eligible for adding orders for multi-leg securities.
4108	Unable to trade risk-netting instruments in T0: insufficient rights.
4109	Rate/swap price is not a multiple of the tick size.
4110	The near leg price differs from the settlement price.

Return code	Description
4111	The rate/swap price exceeds the limit allowed.
4112	Unable to set restrictions for multi-leg futures.
4115	Unable to transfer funds between Brokerage Firm accounts: no active sessions.
4116	Unable to transfer funds between Brokerage Firm accounts: the donor Brokerage Firm is not registered for trading.
4117	Unable to transfer funds between Brokerage Firms: the receiving Brokerage Firm is not registered for trading.
4118	Broker Firm does not have sufficient amount of free funds.
4119	Brokerage Firm does not have sufficient amount of collateral.
4122	Clearing Firm does not have sufficient amount of free funds.
4123	Brokerage Firm does not have sufficient amount of collateral.
4124	Brokerage Firm code not found.
4125	Unable to transfer funds between accounts of different Clearing Firms.
4126	Unable to transfer: error while transferring.
4127	Insufficient free funds in the Settlement Account.
4128	Brokerage firm does not have sufficient amount of free funds.
4129	Insufficient amount of free funds in the balance of the Separate Account.
4130	Clearing Firm does not have sufficient amount of free funds.
4131	Brokerage Firm code not found.
4132	Unable to withdraw: error in withdrawal logic.
4133	No requests to cancel.
4134	Brokerage Firm does not have sufficient amount of funds.
4135	Clearing firm does not have sufficient amount of funds.
4136	Prohibited to transfer pledged funds.
4137	Brokerage Firm does not have sufficient amount of pledged funds.
4138	Insufficient funds to withdraw from the Settlement Account.
4139	Insufficient free collateral in the Settlement Account.
4140	Unable to transfer: position not found.
4141	Unable to transfer: insufficient number of open positions.
4142	Cannot transfer positions from the client account to an account with a different UIN.
4143	Unable to transfer position: the Brokerage Firms specified belong to different Clearing Firms.
4144	Cannot transfer positions to 'XXYY000' Brokerage Firm account.
4145	Unable to transfer positions for the selected Brokerage Firm: restricted by the Trading Administrator.
4146	Transferring positions in the selected securities is prohibited.
4147	Option contract not found.
4148	Settlement Account does not have sufficient amount of pledged funds.
4149	Settlement Account does not have sufficient amount of funds.
4150	Unable to balance risk using specified futures instrument.
4151	Specified FX Market Firm code not found.
4152	Specified FX Market Settlement Account not found.
4153	Specified FX Market financial instrument not found.
4154	Unable to add request for FX Market: the required parameters are not registered in the system.
4155	Required Administrator login for adding a risk balancing request is not registered in the system.
4160	Unable to perform operation. To transfer funds between settlement accounts, you are required to apply to CC.
4161	Withdrawal is prohibited. Settlement account is included in the Unified Collateral Pool.
4162	Unable to perform operation. The Brokerage Firms must be of the same Settlement account.
4164	Unable to perform operation. It is prohibited to change settings for client accounts.
4165	Unable to perform operation. Only Clearing Firm logins are able to perform the operation.
4166	Incorrect combination of flag values.
4167	Settlement Account not found.

Return code	Description
4169	Cannot perform operation: the operation is available for Clearing Firm/Brokerage Firm login only.
4170	Cannot perform operation: incorrect Brokerage Firm account.
4171	Cannot perform operation: incorrect client account.
4172	Cannot perform operation: insufficient rights for the Clearing Member.
4173	Cannot perform operation: insufficient rights for the Trading Member.
4174	GTD multileg order is canceled by trading system.
4175	The Clearing Member has the option to take into funds only on the Settlement Account.
4200	Cannot confirm request. Trading participant's MASTER login is not connected.
4201	Cannot confirm request. Price value in request exceeded the current price value.
4202	Cannot confirm request. Maximum number of contracts exceeded in request.
4203	Cannot confirm request. Negotiated mode is not allowed.
4204	Cannot confirm request. Maximum volume in Russian Ruble exceeded in request.
4205	Cannot confirm request. Amount in Russian Ruble exceeded total available amount in requests per trading day.
4206	Cannot confirm request. Number of buy orders exceeded maximum available number in position.
4207	Cannot confirm request. Number of sell orders exceeded maximum available number in position.
4208	Cannot confirm request. Total quantity of simultaneous restrictions on position size for different clearing register exceeded for given SMA login.
4220	Trading operations for user are prohibited.
4221	Unable to perform operation: Clearing Member and Trading Member represent the same entity.
4222	Unable to perform operation with orders: insufficient rights for Clearing Member.
4224	Unable to perform operation: insufficient rights for active MASTER logins.
4225	Clearing member is under liquidation netting process, all operations are prohibited.
4226	All trading operations are prohibited for this BF during the morning session, except for orders cancellation operations.
4230	Orders will not be cancelled: collateral requirements are met for the Brokerage Firm.
4258	Negotiated iceberg orders are prohibited.
4259	Change of only a single iceberg order is possible.
4260	The iceberg visible part size is less than the minimum acceptable value.
4261	The iceberg visible part size is more than the iceberg order volume.
4262	The random addition size is more than the maximum acceptable value.
4264	The random addition size is less than zero.
4266	Trading system administrator lock mode is set for Settlement Account.
4268	Iceberg order can be changed only at the price.
4269	Expiration order date cannot be indicated in the negotiated order.
4280	Invalid input parameters: "Client Code" parameter was not specified.
4281	Invalid input parameters: invalid value for "Prohibit Type" parameter.
4282	Invalid input parameters: for the parameter "Operating mode" = 12, the "Prohibit mask" = 0 cannot be set.
9999	Too many transactions sent from this login.
10000	System level error while processing message.
10001	Undefined message type.
10004	Invalid message type.
10005	MQ address is too large
10006	Error parsing message.