



TWIME protocol specification for OTC System of Derivatives Market

version 5.6.0

Moscow 2022

Table of Contents

1. Introduction	5
1.1. Document purpose	5
1.2. General description	5
1.3. Terms and definitions	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	8
2.2. Header	9
2.3. Message scheme	9
3. Session Layer	10
3.1. Supported messages	10
3.1.1. Establish (message id=5100)	10
3.1.2. EstablishmentAck (message id=5101)	10
3.1.3. EstablishmentReject (message id=5102)	11
3.1.4. Terminate (message id=5103)	11
3.1.5. RetransmitRequest (message id=5104)	12
3.1.6. Retransmission (message id=5105)	12
3.1.7. Sequence (message id=5106)	12
3.1.8. FloodReject (message id=5107)	12
3.1.9. SessionReject (message id=5108)	12
3.2. Session interaction scenarios	13
3.2.1. Session binding and termination	13
3.2.2. Session status monitoring	13
3.2.3. Message numbering	14
3.2.4. Message retransmission request	14
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
4. Application layer	17
4.1. Supported messages	17
4.1.1. NewStream (message id=8007)	17
4.1.2. CancelStream (message id=8008)	18
4.1.3. RfsQuote (message id=8009)	18
4.1.4. RfsQuoteMassCancel (message id=8011)	19
4.1.5. RfsQuoteHit (message id=8012)	19
4.1.6. RfsConfirmation (message id=8013)	19
4.1.7. NewStreamResponse (message id=9011)	20
4.1.8. NewStreamReject (message id=9012)	21
4.1.9. CancelStreamResponse (message id=9013)	21
4.1.10. CancelStreamReject (message id=9014)	22
4.1.11. RfsQuoteResponse (message id=9015)	22
4.1.12. RfsQuoteReplaceResponse (message id=9016)	23
4.1.13. RfsQuoteReject (message id=9017)	24
4.1.14. RfsQuoteCancelResponse (message id=9018)	24
4.1.15. RfsQuoteMassCancelAck (message id=9020)	24
4.1.16. RfsBestQuoteUpdate (message id=9021)	25
4.1.17. RfsQuoteHitAck (message id=9022)	25
4.1.18. RfsConfirmationAck (message id=9023)	25
4.1.19. RfsExecutionReport (message id=9024)	25
4.1.20. EmptyBook (message id=9009)	26
4.1.21. SystemEvent (message id=9010)	27
4.1.22. Unsolicited messages	27
4.2. Trading Interaction Scenarios	27
4.2.1. Creating of liquidity stream by a liquidity consumer LC	27
4.2.2. Provider LP1 adds a quote in liquidity stream, WireGate sends update of the best quotes to LC	28
4.2.3. Provider LP2 adds a quote into liquidity stream, WireGate sends update of the best quotes to LC	29
4.2.4. Liquidity provider changes the quote	29
4.2.5. Cancellation of quotes by a liquidity provider LP	30
4.2.6. The quote is cancelled due to the end of its lifetime	31
4.2.7. Liquidity stream closing	31
4.2.7.1. Liquidity stream closing by the command from LC	31
4.2.7.2. Automatic liquidity stream closing	32

4.2.8. Liquidity consumer hits the quote	33
4.2.9. Match quotes into a trade and Last Look	33
4.2.9.1. LC hits the quote without Last Look	34
4.2.9.2. LP negotiated order is rejected by the Spectra	34
4.2.9.3. LC negotiated order is rejected by the Spectra	35
4.2.9.4. Provider indicative trade confirmation - Last Look	36
4.2.10. Client quote ID	37
5. Message scheme	38

History of changes

Date	Version	Changes
14.10.2022	5.6.0	<ol style="list-style-type: none"> 1. New value 'SequenceReset' was added into 'TerminationCodeEnum'. 2. Updated message schema.
15.01.2020	3.6.0	<ol style="list-style-type: none"> 1. Specification was renamed to 'TWIME protocol specification for OTC System of Derivatives Market'. 2. New section was added - see Section 4, "Application layer". 3. Message scheme has been changed - see details in the Section 5, "Message scheme". 4. Field AggressorIndicator was deleted from message ExecutionReport for IQS - see ???.
31.10.2019	3.5.0	<ol style="list-style-type: none"> 1. Sections "2.1.6. Bitmasks" and "5. Message scheme" - new bits for 'FlagsSet'. 2. Section "4.1.6. ExecutionReport (message id=9008)" - new values for 'Flags' field: <ul style="list-style-type: none"> • 0x200000000000 – The active side in the trade. The quote that led to the trade when added to the order-book. • 0x400000000000 – The passive side in the trade. The quote from the order-book involved in the trade.

1. Introduction

1.1. Document purpose

The document below describes protocol TWIME for connection to OTC System of Derivatives Market. The specification given covers presentation, session and application layers of the protocol. The specification does not contain neither technical nor administrative details on network connection or security protection methods.

1.2. General description

OTC System - an additional module of the Derivatives Market trading system, which is a service for attracting large-block liquidity (RFS), which allows members to submit a liquidity request and to withdraw the liquidity provided when a satisfactory offer. For more information of the OTC System of Derivatives Market, please apply to "*Plaza-2 gateway for OTC System of Derivatives Market*" **OTC_P2Gate_en.pdf** [http://ftp.moex.com/pub/OTC/RFS_IQS/prod/OTCGate/Docs/OTC_P2Gate_en.pdf].

1.3. Terms and definitions

This document contains the following terms, definitions and acronyms:

Term	Definition
RFS (Request For Stream)	RFS System of the Derivatives Market - on-demand auction
LC (Liquidity Consumer)	Liquidity Consumer in RFS System
LP (Liquidity Provider)	Liquidity Provider in RFS System
Order	A trading instruction added into the SPECTRA trading system
Indicative Quote (or simply quote)	A trading instruction added into the OTC system without collateral verification
Indicative trade	A trade performed as a result of matching indicative quotes within OTC system.
Trade	A trade performed as a result of matching orders within SPECTRA trading system.

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 length

```
<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"
  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" />

```

2.1.5. Enumerations

```

<enum name="BooleanEnum" encodingType="uint8">
  <validValue name="False" spectra_value="0">0</validValue>
  <validValue name="True" spectra_value="1">1</validValue>
</enum>

<enum name="TerminationCodeEnum" encodingType="uint8">
  <validValue name="Finished" >0</validValue>
  <validValue name="UnspecifiedError" >1</validValue>
  <validValue name="ReRequestOutOfBounds" >2</validValue>
  <validValue name="ReRequestInProgress" >3</validValue>
  <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>
  <validValue name="SequenceReset" >11</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="QuoteMsgIDIsNotUnique" >101</validValue>
</enum>

<enum name="TimeInForceEnum" encodingType="uint8">
  <validValue name="Day" spectra_value="1">0</validValue>
  <validValue name="IOC" spectra_value="2">3</validValue>
</enum>

<enum name="SideEnum" encodingType="uint8">
  <validValue name="Unavailable" spectra_value="0">0</validValue>
  <validValue name="Buy" spectra_value="1">1</validValue>
  <validValue name="Sell" spectra_value="2">2</validValue>
  <validValue name="BothSides" spectra_value="3">89</validValue>
</enum>

<enum name="ModeEnum" encodingType="uint8">
  <validValue name="DontChangeQuoteQty" spectra_value="0">0</validValue>
  <validValue name="ChangeQuoteQty" spectra_value="1">1</validValue>
  <validValue name="CheckQuoteQtyAndCancelQuote" spectra_value="2">2</validValue>
  <validValue name="FixStyleReplace" spectra_value="3">3</validValue>
</enum>

<enum name="MatchTypeEnum" encodingType="uint8">
  <validValue name="AutoMatch" spectra_value="1">4</validValue>
  <validValue name="AutoMatchWithLastLook" spectra_value="0">10</validValue>
</enum>

<enum name="TradSesEventEnum" encodingType="uint16">
  <validValue name="SessionDataReady" spectra_value="1">101</validValue>
  <validValue name="IntradayClearingFinished" spectra_value="2">102</validValue>
  <validValue name="IntradayClearingStarted" spectra_value="4">104</validValue>

```

```

    <validValue name="ClearingStarted"          spectra_value="5">105</validValue>
    <validValue name="ExtensionOfLimitsFinished" spectra_value="6">106</validValue>
    <validValue name="BrokerRecalcFinished"     spectra_value="8">108</validValue>
    <validValue name="OtcSessionInited"        spectra_value="10000">10100</validValue>
    <validValue name="OtcSessionStarted"       spectra_value="10001">10101</validValue>
    <validValue name="OtcSessionSuspended"     spectra_value="10002">10102</validValue>
    <validValue name="OtcSessionStoped"       spectra_value="10003">10103</validValue>
    <validValue name="OtcSessionFinished"     spectra_value="10004">10104</validValue>
</enum>

<enum name="SecurityTypeEnum" encodingType="uint8">
    <validValue name="Future"                  spectra_value="0">0</validValue>
    <validValue name="Option"                  spectra_value="1">1</validValue>
    <validValue name="Multileg"                spectra_value="2">2</validValue>
</enum>

<enum name="StatusEnum" encodingType="uint8">
    <validValue name="Matched"                 spectra_value="0">0</validValue>
    <validValue name="WaitConfirm"            spectra_value="2">1</validValue>
    <validValue name="Confirmed"              spectra_value="4">2</validValue>
    <validValue name="Failed"                 spectra_value="3,10,11,12,13,14">3</validValue>
    <validValue name="Success"                spectra_value="9">4</validValue>
</enum>

<enum name="RejectReasonEnum" encodingType="uint8">
    <validValue name="NotApplicable"          spectra_value="0,2,4,9">0</validValue>
    <validValue name="NotConfirmed"           spectra_value="3">1</validValue>
    <validValue name="ActiveSideError"        spectra_value="10,12,13,14">2</validValue>
    <validValue name="PassiveSideError"       spectra_value="11,12,13,14">3</validValue>
</enum>

<enum name="StreamExposureDurationEnum" encodingType="uint8">
    <validValue name="NotApplicable"          spectra_value="0">0</validValue>
    <validValue name="Duration30sec"          spectra_value="1">1</validValue>
    <validValue name="Duration60sec"          spectra_value="2">2</validValue>
    <validValue name="Duration90sec"          spectra_value="3">3</validValue>
    <validValue name="Duration120sec"         spectra_value="4">4</validValue>
</enum>

<enum name="SpeedBumpTypeEnum" encodingType="uint8">
    <validValue name="NotApplicable"          spectra_value="0">0</validValue>
    <validValue name="Duration200ms"          spectra_value="1">1</validValue>
    <validValue name="Duration500ms"          spectra_value="2">2</validValue>
    <validValue name="Duration1000ms"         spectra_value="3">3</validValue>
    <validValue name="Duration3000ms"         spectra_value="4">4</validValue>
</enum>

<enum name="CancelReasonEnum" encodingType="uint8">
    <validValue name="Deal"                   spectra_value="1">1</validValue>
    <validValue name="LCDoesntHaveEnoughMoney" spectra_value="2">2</validValue>
    <validValue name="CancelByLC"             spectra_value="3">3</validValue>
    <validValue name="TimeOut"                spectra_value="4">4</validValue>
    <validValue name="CancelByAdministrator" spectra_value="5">5</validValue>
    <validValue name="EndOfTradingSession"    spectra_value="6">6</validValue>
</enum>

```

2.1.6. Bitmasks

```

<set name="FlagsSet" encodingType="uint64">
    <choice name="Day"                description="Quotes: Day"
    <choice name="IOC"                description="Quotes: IOC"
    <choice name="Replace"            description="Quotes: The record results from replacing the quote"
    <choice name="Cancel"             description="Quotes: The record results from cancelling the quote"
    <choice name="MassCancel"         description="Quotes: The record results from mass cancelling"
    <choice name="MultiLeg"           description="Quotes: Multi leg"
    <choice name="FineOperation"      description="Quotes: Flag of cancelling the quote because of fine"
    <choice name="ActiveSide"         description="Trades: Quote initiator is aggressor"
    <choice name="PassiveSide"        description="Trades: Quote initiator is passive"
    <choice name="TimeOut"            description="Quotes: Delete quote due timeout"
    <choice name="AutoMatch"          description="Quotes and trades: Auto match without last look"
</set>

```



```
<set name="StreamFlagsSet" encodingType="uint64">
  <choice name="AutoMatch" description="Auto match without last look">0</choice>
  <choice name="ClosedStream" description="Stream is closed">1</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 package="sbe" byteOrder="littleEndian" id="20809" version="0">
</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.

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=5100)

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=5101)

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.

Tag	Field	Mandatory	Type	Details
20208	NextSeqNo	Y	UInt64	Next message sequential number.

3.1.3. EstablishmentReject (message id=5102)

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	EstablishmentReject-Code	Y	EstablishmentRejectCodeEnum	Establishment reject code: <ul style="list-style-type: none"> "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=5103)

Session termination.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20210	TerminationCode	Y	TerminationCodeEnum	Session termination reason: <ul style="list-style-type: none"> "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. "11" SequenceReset - Message counter reset.

3.1.5. RetransmitRequest (message id=5104)

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.
20212	Count	Y	UInt32	Number of messages to retransmit.

3.1.6. Retransmission (message id=5105)

The message notifies that all the next 'Count' message(s) are retransmitted in reply to the 'RetransmitRequest' (message id=5104).

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=5106)

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=5107)

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		
1166	QuoteMsgID	Y	UInt64	User ID of the rejected quote.
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=5108)

Notifies about invalid messages sent from the client side.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	User ID of the rejected quote
371	RefTagID	N	UInt32	ID of invalid field
373	SessionRejectReason	Y	SessionRejectReasonEnum	Rejection reason code: <ul style="list-style-type: none"> "5" ValueIsIncorrect - Incorrect field value. "99" Other - Other reason. "100" SystemIsUnavailable - Trading system is unavailable.

Tag	Field	Mandatory	Type	Details
				<ul style="list-style-type: none"> "101" QuoteMsgIDsNotUnique - Non-unique user ID of the quote.

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-authorized user) the system will reply with the message 'EstablishmentReject' containing the rejection reason details.

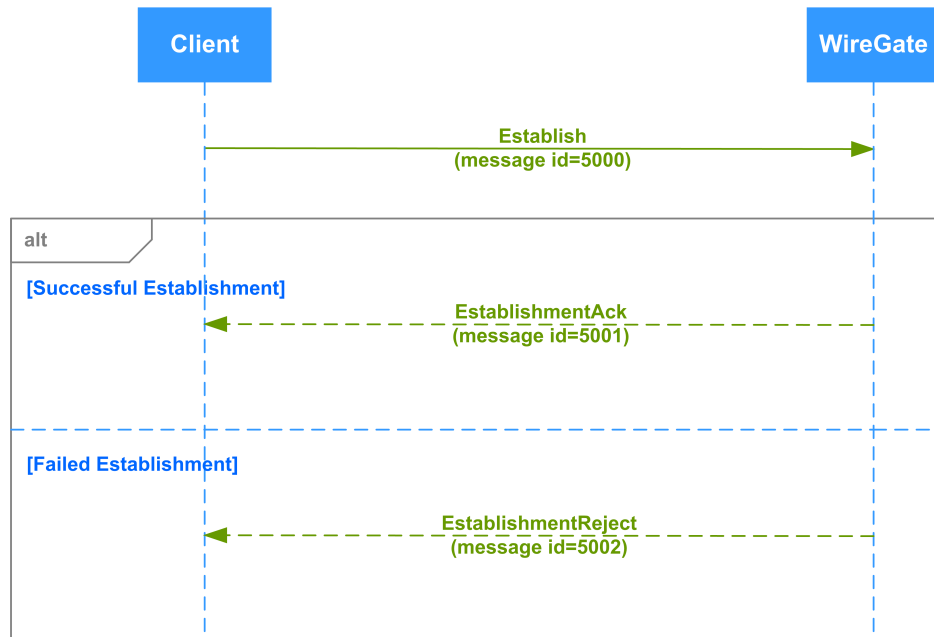


Figure 1. Diagram. Session binding

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

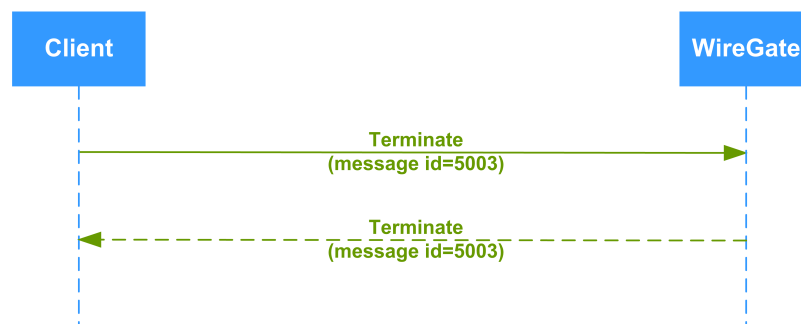


Figure 2. Diagram. Session termination

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.

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 $2 * \text{KeepaliveInterval}$ + 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 Terminate containing the termination reason 'TooFastClient' to the client side.

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.

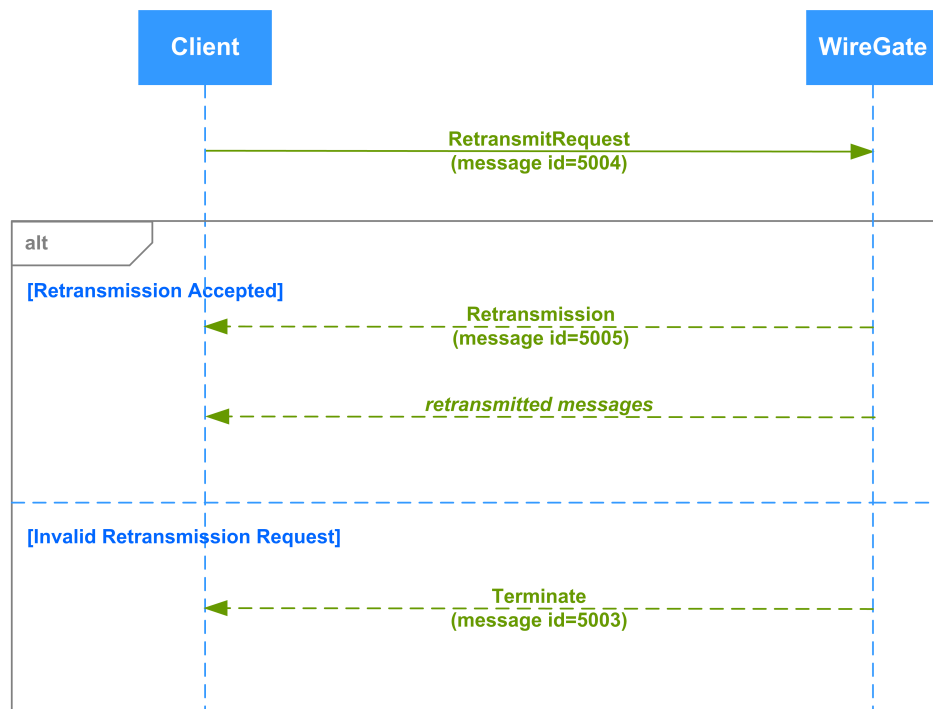


Figure 3. Diagram. Message retransmission request

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. A client can request no more than 1000 messages per a single 'RetransmitRequest'.

3.2.6. Full session crash recovery

In order to receive all session messages the client side should 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 '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 'Section 3.2.5, "Session short-term crash recovery"'.

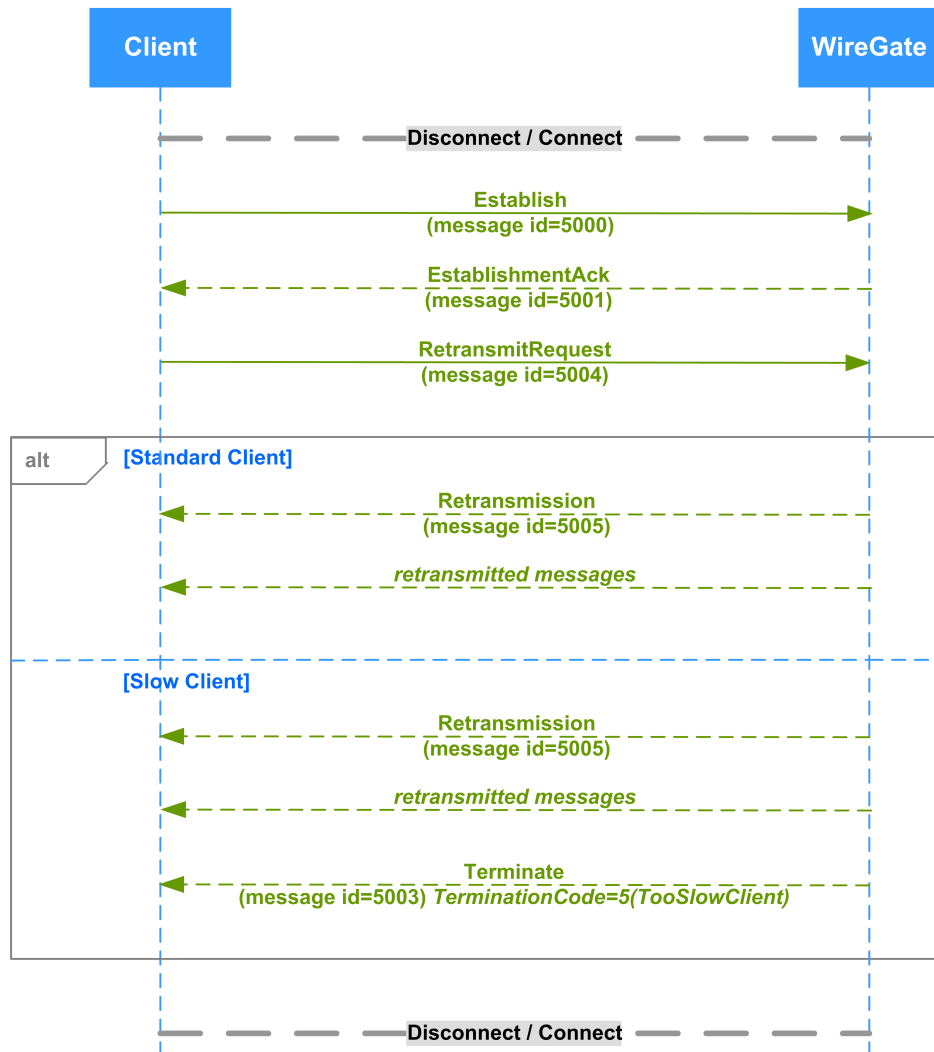


Figure 4. Diagram. Full session crash recovery

3.2.7. Message counter reset

During the time period 11:50 PM - 10:00 AM 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. Then 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.

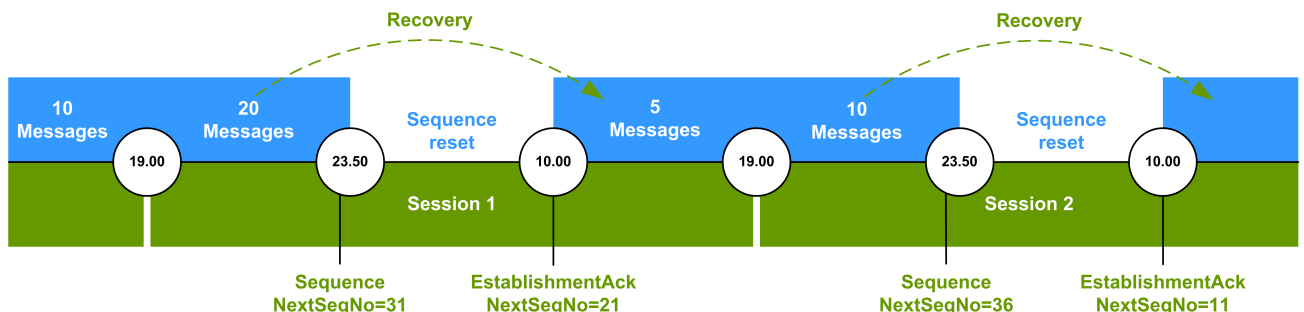


Figure 5. Diagram. Message counter reset

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 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.

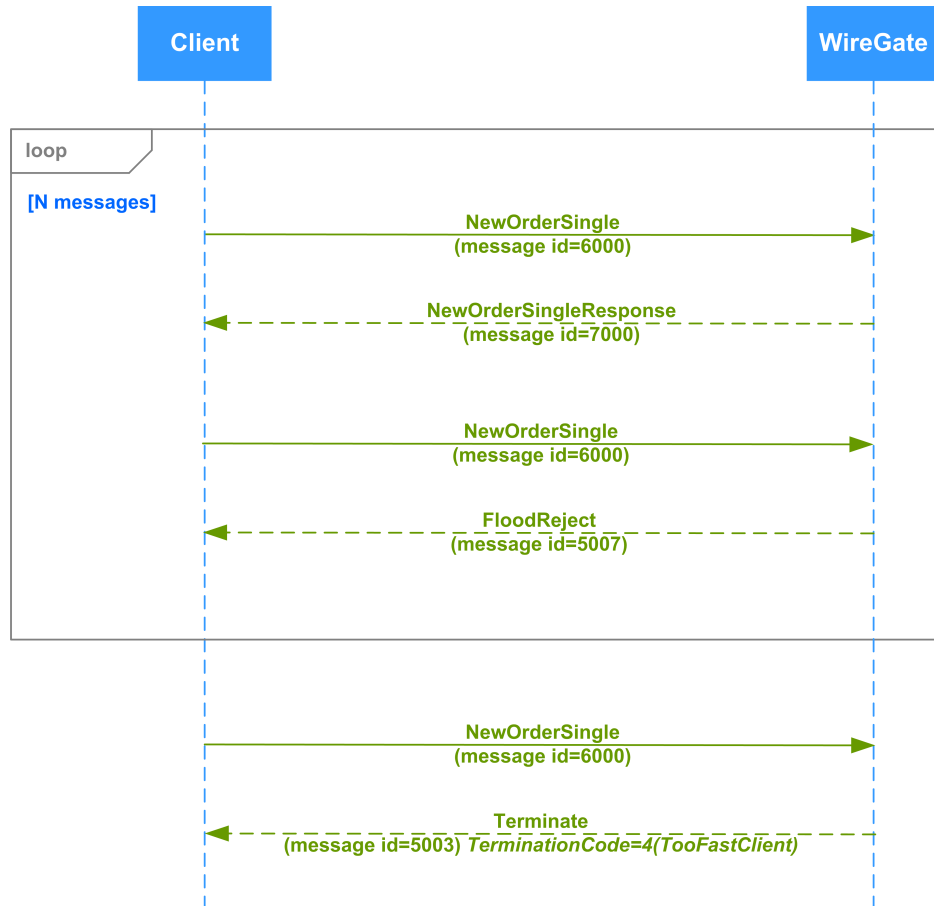


Figure 6. Diagram. Flood control

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 overflowed.

4. Application layer

4.1. Supported messages

From client side to WireGate:

- **NewStream** - Adding a new liquidity stream (performed by the liquidity consumer - LC)
- **CancelStream** - Closing the liquidity stream by the liquidity consumer
- **RfsQuote** - Adding / changing quotes by the liquidity providers (LP)
- **RfsQuoteMassCancel** - Mass quotes deletion by the liquidity provider
- **RfsQuoteHit** - Quote acceptance by the liquidity consumer
- **RfsConfirmation** - Indicative trade confirmation by the liquidity provider

From WireGate to client side:

- **NewStreamResponse** - Successfully stream creation for the liquidity consumer. Stream creation notification for the liquidity providers.
- **NewStreamReject** - Reject a request to add a new liquidity stream for LC
- **CancelStreamResponse** - Successfully stream deletion for LC. Stream deletion notification for the liquidity providers.
- **CancelStreamReject** - Reject stream deletion request for LC
- **RfsQuoteResponse** - Successfully quote adding (response for LP)
- **RfsBestQuoteUpdate** - Best provider quote for LC
- **RfsQuoteCancelResponse** - Successfully quote deletion (response for LP)
- **RfsQuoteReplaceResponse** - Successfully quote moving (response for LP)
- **RfsQuoteReject** - Reject add / move quote request (response for LP)
- **RfsQuoteMassCancelAck** - Message about the results of quote mass deletion (response for LP)
- **RfsQuoteHitAck** - Reply to quote confirmation for LC
- **RfsConfirmationAck** - Notification of successful/ unsuccessful indicative trade confirmation.
- **RfsExecutionReport** - Quotes have matched into indicative trade. Notifications of the trade status changes to the LC and to the LP.
- **EmptyBook** - Empty order-book at client side.
- **SystemEvent** - OTC System events

4.1.1. NewStream (message id=8007)

The liquidity consumer has the opportunity to submit a request to the OTC system to open a liquidity stream, in which providers can add their quotes. Operation is available only LC.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Custom ID of the request to create the stream
110	MinQty	Y	UInt64	Request volume (number of instrument units). Volume has a lower limit depending on the underlying asset. The user cannot set a value less than the minimum allowed volume.
20027	ExternalID	N	UInt64	Stream external ID.
48	SecurityID	Y	UInt32	Instrument ID.
54	Side	Y	SideEnum	Direction: <ul style="list-style-type: none"> • '1' - Buy. • '2' - Sell. • '89' - Both direction.
21011	StreamExposureDuration	Y	StreamExposureDurationEnum	Stream duration: <ul style="list-style-type: none"> • "1" - 30 seconds.

Tag	Field	Mandatory	Type	Details
				<ul style="list-style-type: none"> "2" - 60 seconds. "3" - 90 seconds. "4" - 120 seconds.
574	MatchType	Y	MatchTypeEnum	The type of allowable quotes: <ul style="list-style-type: none"> "4" - Only firm quote is allowable, auto match (without Last Look). "10" - Any quote is allowable; indicative quote (with Last Look) and firm quote (without Last Look).
21000	SpeedBumpType	Y	SpeedBumpTypeEnum	Limitations on the frequency of quotes updates by liquidity providers. The time during which it is impossible to move or cancel the quote after it is set. Possible values: <ul style="list-style-type: none"> "0" - Not limited "1" - 200 milliseconds. "2" - 500 milliseconds. "3" - 1000 milliseconds. "4" - 3000 milliseconds.
1	Account	Y	String7	Client 7-symbol ID
21001	TextToLP	N	String20	Comment for LP
58	Text	N	String20	Comment

4.1.2. CancelStream (message id=8008)

Close the liquidity stream. Operation is available to LC.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Custom ID of the request to close the stream
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
1	Account	Y	String7	Client 7-symbol ID.

4.1.3. RfsQuote (message id=8009)

Add / replace quote by provider. Operation is available to LP.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Custom ID of the request to add / replace quote
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
133	OfferPx	Y	Decimal5	Sell quote price
20025	OfferExternalID	N	UInt64	External ID of sell quote
132	BidPx	Y	Decimal5	Buy quote price
20024	BidExternalID	N	UInt64	External ID of buy quote
1629	ExposureDuration	Y	UInt64	Quote lifetime in microseconds.
574	MatchType	Y	MatchTypeEnum	Quote type: <ul style="list-style-type: none"> "4" - Firm quote (without Last Look). "10" - Indicative quote (with Last Look).
54	Side	Y	SideEnum	Direction:

Tag	Field	Mandatory	Type	Details
				<ul style="list-style-type: none"> '1' - Buy. '2' - Sell. '89' - Both direction.
1	Account	Y	String7	Client 7-symbol ID.
21003	OfferText	N	String20	Sell quote comment.
21004	BidText	N	String20	Buy quote comment.

4.1.4. RfsQuoteMassCancel (message id=8011)

Cancellation of quotes by provider.

The provider can delete the quote group by sending the message 'RfsQuoteMassCancel', in which one of the fields is filled:

- SecurityID - delete quotes with a given instrument identifier;
- Account - delete quotes with a given client code;
- ExternalID - delete quotes with a given external identifier.

The provider deletes one quote by sending the 'RfsQuoteMassCancel' message with the filled fields 'AuctionID' and 'Side'

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Client ID for mass cancellation request.
21002	AuctionID	N	UInt64	The liquidity stream ID assigned by RFS servis. ID is unique during the trading day.
20027	ExternalID	N	UInt64	Quote external ID.
48	SecurityID	N	Int32	Instrument ID.
54	Side	N	SideEnum	Quote direction: <ul style="list-style-type: none"> '1' - Buy quote. '2' - Sell quote. '89' - Both sides.
1	Account	N	String7	Client 7-symbol ID.

4.1.5. RfsQuoteHit (message id=8012)

Consumer hits the quote. Operation is available to LC.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Custom ID of the request to hit quote.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
44	Price	Y	Decimal5	Quote price.
54	Side	Y	SideEnum	Direction: <ul style="list-style-type: none"> '1' - Buy. '2' - Sell. '89' - Both sides.
58	Text	N	String20	Comment LC.

4.1.6. RfsConfirmation (message id=8013)

Indicative trade confirmation by liquidity provider.

Before performing an indicative trade, provider should confirm this indicative trade with one of two methods, as follows:

- automatically, when a quote adds into the system;
- manually, by sending a confirmation message.

To confirm indicative trades automatically, one should specify value 'MatchType=4' in field 'MatchType (Tag=574)' of message **RfsQuote**.

To confirm indicative trades manually, one should send a message **RfsConfirmation** containing the indicative trade ID, in reply to received message **ExecutionReport** with the indicative trade status '1' (wait for confirmation). Note that the time for sending the reply message is strictly limited. Once the message could not be sent in the time frame specified, the indicative trade will be considered as unconfirmed.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Custom ID of the request to confirm the indicative trade.
17	ExecID	Y	UInt64	Indicative trade ID

4.1.7. NewStreamResponse (message id=9011)

Liquidity stream is added successfully.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
1166	QuoteMsgID	Y	UInt64	QuoteReqID that was submitted with the 'NewStream' message.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
110	MinQty	Y	UInt64	Liquidity volume (number of instrument units).
20027	ExternalID	N	UInt64	Stream external ID.
48	SecurityID	Y	UInt32	Instrument ID.
336	TradingSessionID	Y	Int32	Trading session ID.
21005	StreamFlags	Y	StreamFlagsSet	The type of allowable quotes: <ul style="list-style-type: none"> • "4" - Only firm quote is allowable, auto match (without Last Look). • "10" - Any quote is allowable; indicative quote (with Last Look) and firm quote (without Last Look).
167	SecurityType	Y	SecurityTypeEnum	Instrument type: <ul style="list-style-type: none"> • '0' - Futures. • '1' - Option. • '2' - Multi-leg.
54	Side	Y	SideEnum	Direction: <ul style="list-style-type: none"> • '1' - Buy. • '2' - Sell. • '89' - Both direction.
21011	StreamExposureDuration	Y	StreamExposureDurationEnum	Stream duration: <ul style="list-style-type: none"> • "1" - 30 seconds. • "2" - 60 seconds. • "3" - 90 seconds. • "4" - 120 seconds.
21000	SpeedBumpType	Y	SpeedBumpTypeEnum	Limitation on the frequency of quotes updates by liquidity providers. The time during which it is impossible to move or cancel the quote after it is set. Possible values: <ul style="list-style-type: none"> • "0" - Not limited

Tag	Field	Mandatory	Type	Details
				<ul style="list-style-type: none"> "1" - 200 milliseconds. "2" - 500 milliseconds. "3" - 1000 milliseconds. "4" - 3000 milliseconds.
21001	TextToLP	N	String20	Comment for LP
58	Text	N	String20	Comment.
21007	TagOfLC	N	String64	Consumer tag.

4.1.8. NewStreamReject (message id=9012)

Liquidity request is rejected.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
1166	QuoteMsgID	Y	UInt64	QuoteReqID that was submitted with the 'NewStream' message.
300	QuoteRejectReason	Y	Int32	Error code.

4.1.9. CancelStreamResponse (message id=9013)

Liquidity stream is closed successfully.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
1166	QuoteMsgID	Y	UInt64	QuoteReqID that was submitted with the 'CancelStream' message.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
110	MinQty	Y	UInt64	Liquidity volume (number of instrument units).
20027	ExternalID	N	UInt64	Stream external ID.
17	ExecID	Y	UInt64	Indicative trade ID.
48	SecurityID	Y	UInt32	Instrument ID.
336	TradingSessionID	Y	Int32	Trading session ID.
54	Side	Y	SideEnum	Direction: <ul style="list-style-type: none"> '1' - Buy. '2' - Sell. '89' - Both direction.
21011	StreamExposureDuration	Y	StreamExposureDurationEnum	Stream duration: <ul style="list-style-type: none"> "1" - 30 seconds. "2" - 60 seconds. "3" - 90 seconds. "4" - 120 seconds.
21005	StreamFlags	Y	StreamFlagsSet	The type of allowable quotes: <ul style="list-style-type: none"> "4" - Only firm quote is allowable, auto match (without Last Look). "10" - Any quote is allowable; indicative quote (with Last Look) and firm quote (without Last Look).
167	SecurityType	Y	SecurityTypeEnum	Instrument type:

Tag	Field	Mandatory	Type	Details
				<ul style="list-style-type: none"> '0' - Futures. '1' - Option. '2' - Multi-leg.
21000	SpeedBumpType	Y	SpeedBumpTypeEnum	Limitation on the frequency of quotes updates by liquidity providers. The time during which it is impossible to move or cancel the quote after it is set. Possible values: <ul style="list-style-type: none"> "0" - Not limited "1" - 200 milliseconds. "2" - 500 milliseconds. "3" - 1000 milliseconds. "4" - 3000 milliseconds.
21008	CancelReason	Y	CancelReasonEnum	Reason for closing: <ul style="list-style-type: none"> "1" - Trade. "2" - LC doesn't have enough money. "3" - Cancel by LC. "4" - TimeOut. "5" - Cancel by Administrator. "6" - End of trading session.
21001	TextToLP	N	String20	Comment for LP
58	Text	N	String20	Comment LC.
21007	TagOfLC	N	String64	Consumer tag.

4.1.10. CancelStreamReject (message id=9014)

Close stream request is rejected.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
1166	QuoteMsgID	Y	UInt64	QuoteReqID that was submitted with the 'CancelStream' message being accepted.
300	QuoteRejectReason	Y	Int32	Error code.

4.1.11. RfsQuoteResponse (message id=9015)

Quote is added successfully by LP.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the 'RfsQuote' message.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
1751	SecondaryQuoteID	Y	UInt64	Exchange-assigned unique quote ID.
20026	QuoteSize	Y	UInt64	Quote volume.
44	Price	Y	Decimal5	Quote price.
20027	ExternalID	N	UInt64	Quote external ID.
1629	ExposureDuration	Y	UInt64	Quote lifetime in microseconds.

Tag	Field	Mandatory	Type	Details
20215	Flags	Y	FlagsSet	Flags (bitmask): <ul style="list-style-type: none"> • 0x1 - Day • 0x2 - IOC • 0x8000000 - Multi-leg • 0x4000000000000 - Auto match without last look
48	SecurityID	Y	Int32	Instrument ID.
336	TradingSessionID	Y	Int32	Trading session ID.
167	SecurityType	Y	SecurityTypeEnum	Instrument type: <ul style="list-style-type: none"> • '0' - Futures. • '1' - Option. • '2' - Multi-leg.
54	Side	Y	SideEnum	Quote direction: <ul style="list-style-type: none"> • '1' - Buy. • '2' - Sell.
21009	CodeOfLP	Y	String20	LP code.
58	Text	N	String20	Comment from quote.

4.1.12. RfsQuoteReplaceResponse (message id=9016)

Quote is replaced successfully by LP.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the 'RfsQuote' message.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
1751	SecondaryQuoteID	Y	UInt64	Exchange-assigned unique quote ID.
20026	QuoteSize	Y	UInt64	Quote volume.
44	Price	Y	Decimal5	Quote price.
20027	ExternalID	N	UInt64	External quote ID.
20034	PrevSecondaryQuoteID	Y	UInt64	Previous quote ID.
1629	ExposureDuration	Y	UInt64	Quote lifetime in microseconds.
20215	Flags	Y	FlagsSet	Flags (bitmask): <ul style="list-style-type: none"> • 0x1 - Day • 0x2 - IOC • 0x100000 - The record results from replacing the quote • 0x8000000 - Multi-leg • 0x4000000000000 - Auto match without last look
48	SecurityID	Y	Int32	Instrument ID.
336	TradingSessionID	Y	Int32	Trading session ID.
167	SecurityType	Y	SecurityTypeEnum	Instrument type: <ul style="list-style-type: none"> • '0' - Futures. • '1' - Option. • '2' - Multi-leg.

Tag	Field	Mandatory	Type	Details
54	Side	Y	SideEnum	Quote direction: <ul style="list-style-type: none"> '1' - Buy. '2' - Sell.
21009	CodeOfLP	Y	String20	LP code.

4.1.13. RfsQuoteReject (message id=9017)

Add quote request is rejected.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the 'RfsQuote' message.
20204	Timestamp	Y	TimeStamp	Server side date and time.
300	QuoteRejectReason	Y	Int32	Error code.
54	Side	Y	SideEnum	Quote direction: <ul style="list-style-type: none"> '1' - Buy. '2' - Sell.

4.1.14. RfsQuoteCancelResponse (message id=9018)

Quote is cancelled.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the quote cancel request.
20204	Timestamp	Y	TimeStamp	Server side date and time.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
1751	SecondaryQuoteID	Y	UInt64	Exchange-assigned unique quote ID.
20026	QuoteSize	Y	UInt64	Quote volume.
20027	ExternalID	N	UInt64	Quote external ID.
20215	Flags	Y	FlagsSet	Flags (bitmask): <ul style="list-style-type: none"> 0x1 - Day 0x2 - IOC 0x100000 - The record results from replacing the quote 0x200000 - The record results from cancelling the quote 0x400000 - The record results from mass cancelling 0x8000000 - Multi-leg 0x1000000000 - Flag of cancelling the quote because of fine 0x2000000000000 - Delete quote due timeout 0x4000000000000 - Auto match without last look
336	TradingSessionID	Y	Int32	Trading session ID.

4.1.15. RfsQuoteMassCancelAck (message id=9020)

Quote mass cancellation report.

Tag	Field	Mandatory	Type	Details
<Header>		Y		

Tag	Field	Mandatory	Type	Details
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the mass cancellation request.
20204	Timestamp	Y	TimeStamp	Server side date and time.
1168	TotNoCxlIdQuotes	N	Int32	Number of quotes cancelled.
21012	TotNoSpeedBumpQuotes	N	Int32	Number of quotes not cancelled due to 'Speed bump'
300	QuoteRejectReason	Y	Int32	Error code.

4.1.16. RfsBestQuoteUpdate (message id=9021)

The best quote. It is sent to the liquidity consumer when the TOP1 quotes in the stream are updated.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS servis. ID is unique during the trading day.
1751	SecondaryQuoteID	Y	Int64	Exchange-assigned unique quote ID.
20026	QuoteSize	Y	UInt64	Quote volume.
44	Price	Y	Decimal5	Quote price.
54	Side	Y	SideEnum	Quote direction: <ul style="list-style-type: none"> '1' - Buy. '2' - Sell.
574	MatchType	Y	MatchTypeEnum	Quote type: <ul style="list-style-type: none"> '4' - Firm quote (without Last Look). '10' - Indicative quote (with Last Look).

4.1.17. RfsQuoteHitAck (message id=9022)

Acknowledgement for quote hit.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the 'RfsQuoteHit' message.
20204	Timestamp	Y	TimeStamp	Server side date and time.
1751	SecondaryQuoteID	Y	UInt64	Exchange-assigned unique quote ID.
300	QuoteRejectReason	Y	Int32	Error code.

4.1.18. RfsConfirmationAck (message id=9023)

Acknowledgement for the trade confirmation.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	QuoteMsgID that was submitted with the 'RfsConfirmationAck' message.
20204	Timestamp	Y	TimeStamp	Server side date and time.
17	ExecID	Y	UInt64	Indicative trade ID.
300	QuoteRejectReason	Y	Int32	Error code.

4.1.19. RfsExecutionReport (message id=9024)

Two quotes has matched into indicative trade. The message will be sent if:

- an indicative trade has performed in RFS;
- an indicative trade status has changed in RFS.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
1166	QuoteMsgID	Y	UInt64	Client ID of the last request to add/replace quote.
20204	Timestamp	Y	TimeStamp	Server side date and time.
21002	AuctionID	Y	UInt64	The liquidity stream ID assigned by RFS service. ID is unique during the trading day.
1751	SecondaryQuoteID	Y	Int64	Exchange-assigned quote ID.
31	LastPx	Y	Decimal5	Indicative trade price.
32	LastQty	Y	UInt32	Instrument amount in indicative trade.
1629	ExposureDuration	Y	UInt64	Quote lifetime in microseconds.
20027	ExternalID	N	UInt64	Quote external ID.
17	ExecID	Y	UInt64	Indicative trade ID
880	TrdMatchID	Y	UInt64	Trade ID within Spectra.
37	OrderID	Y	UInt64	Order ID within Spectra.
336	TradingSessionID	Y	Int32	Trading session ID.
48	SecurityID	Y	Int32	Instrument ID.
103	OrdRejReason	N	Int32	Rejection reason code on the request to add a negotiated order. For the complete list of codes of field 'OrdRejReason', please see p2gate_en.pdf (ftp://ftp.moex.com/pub/ClientsAPI/Spectra/CGate/docs/). Please note to subtract 100 out of each value of field 'OrdRejReason' before committing your search.
167	SecurityType	Y	SecurityTypeEnum	Instrument type: <ul style="list-style-type: none"> • '0' - Futures. • '1' - Option. • '2' - Multi-leg.
54	Side	Y	SideEnum	Quote direction: <ul style="list-style-type: none"> • '1' - Buy. • '2' - Sell.
20030	Status	Y	StatusEnum	Indicative trade status: <ul style="list-style-type: none"> • '0' - Matched. • '1' - Wait Confirm. • '2' - Confirmed. • '3' - Failed. • '4' - Success.
20033	RejectReason	Y	RejectReasonEnum	Error code from RFS. Possible values: <ul style="list-style-type: none"> • '0' - Not Applicable. • '1' - Not Confirmed. • '2' - Active Side Error. • '3' - Passive Side Error.
21009	CodeOfLP	Y	String20	LP code.
58	Text	N	String20	Comment from quote.

4.1.20. EmptyBook (message id=9009)

Empty order-book at the client side.

Tag	Field	Mandatory	Type	Description
<Header>		Y		

Tag	Field	Mandatory	Type	Description
20204	Timestamp	Y	TimeStamp	Server side date and time.
336	TradingSessionID	Y	Int32	Trading session ID.

4.1.21. SystemEvent (message id=9010)

Trading system events.

Tag	Field	Mandatory	Type	Details
<Header>		Y		
20204	Timestamp	Y	TimeStamp	Server side date and time.
336	TradingSessionID	Y	Int32	Trading session ID.
1368	TradSesEvent	Y	TradSesEventEnum	Event type: <ul style="list-style-type: none"> • '101' SessionDataReady - All data have been successfully uploaded from clearing system into trading system for the new trading session. • '102' IntradayClearingFinished - All intraday clearing procedures have been completed. • '104' IntradayClearingStarted - Intraday clearing session started. • '105' ClearingStarted - Main clearing session started. • '106' ExtensionOfLimitsFinished - Limits extension completed. • '108' BrokerRecalcFinished - Funds recalculation completed after intraday clearing session. • '10100' OtcSessionInited - OTC session is initiated. • '10101' OtcSessionStarted - OTC session is in progress. • '10102' OtcSessionSuspended - OTC session is halted. • '10103' OtcSessionStoped - OTC session is stopped. • '10104' OtcSessionFinished - OTC session is finished.

4.1.22. Unsolicited messages

Unsolicited message is a message which is not associated with any input message. It can be resulted from a third-party transaction (change/cancel a quote added by a TWIME login) via WireGate, P2gate, or a trading terminal. Therefore, an unsolicited message is a standard message, with field **QuoteMsgID** containing 'nullValue', to cancel (**RfsQuoteCancelResponse**), or change (**RfsQuoteReplaceResponse**) a quote.

4.2. Trading Interaction Scenarios

4.2.1. Creating of liquidity stream by a liquidity consumer LC

Liquidity Consumer (LC) sends a request into the system to create a new liquidity stream - **NewStream** message.

On successful creation of a liquidity stream:

- WireGate sends **NewStreamResponse** message containing stream identifier **AuctionID** to LC. The fields **Account** and **Text** in **NewStreamResponse** message are broadcast to LC only;
- WireGate broadcasts unsolicited **NewStreamResponse** message to all liquidity providers (LP). If the LC that created the liquidity stream is simultaneously an LP, then such an unsolicited message is not sent to it.

WireGate sends **NewStreamReject** message to LC in case of error while creating stream.

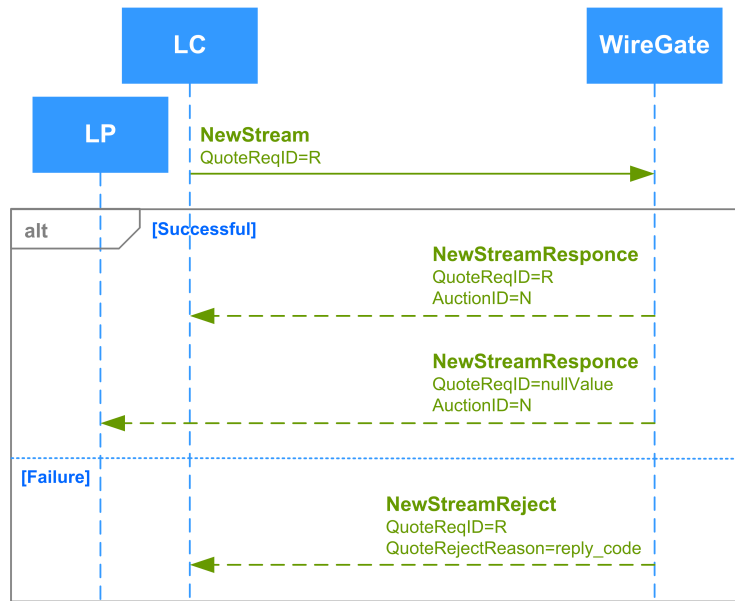


Figure 7. Diagram. Liquidity stream creating

4.2.2. Provider LP1 adds a quote in liquidity stream, WireGate sends update of the best quotes to LC

LP1 add a new quote into an empty liquidity stream, sending **RfsQuote** message to WireGate.

WireGate sends **RfsQuoteResponse** message to LP1 when successful adding quote.

After that WireGate sends two **RfsBestQuoteUpdate** messages with updating the best quotes for buying and selling to LC.

WireGate sends **RfsQuoteReject** message to LP1 in case of error.

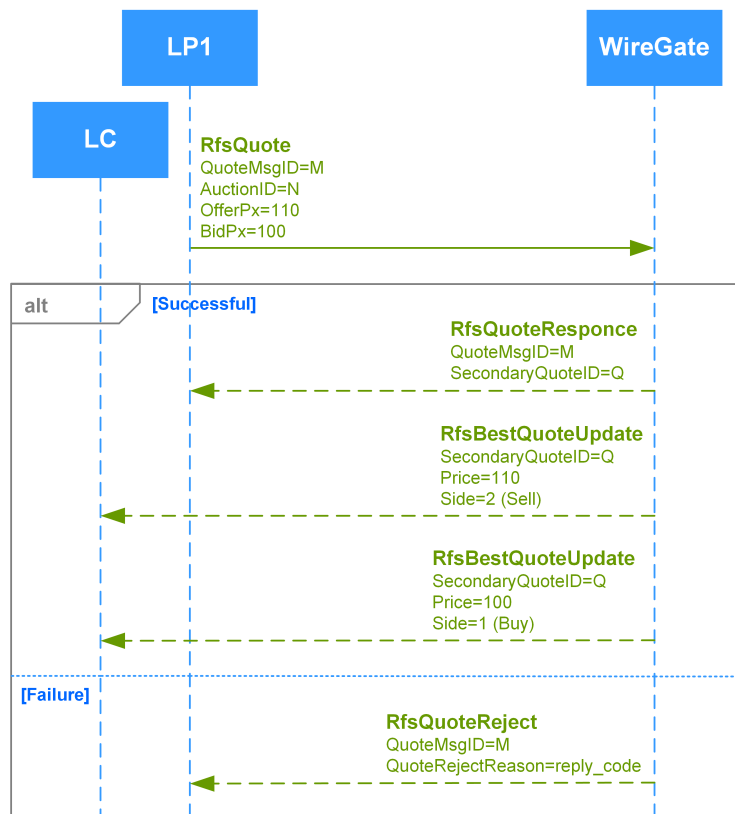


Figure 8. Diagram. Add LP1's quote

4.2.3. Provider LP2 adds a quote into liquidity stream, WireGate sends update of the best quotes to LC

Only one quote from the provider can stand in one liquidity stream (quote may be two-way). Providers cannot see each other's quotes.

LC stream owner cannot see all offers from providers. He sees only the best price to buy and sell - TOP1. LP's quote in the stream is replaced with the best, when a new quote with a better price appears in the stream.

LP2 adds a new better quote than LP1's quote in the stream - send **RfsQuote** message to WireGate.

WireGate sends **RfsQuoteResponse** message to LP2.

WireGate sends one **RfsBestQuoteUpdate** message with updating the best quote to LC. There is only one message, because LP2 improved Bid only (in the diagram).

WireGate sends **RfsQuoteReject** message to LP2 in case of error.

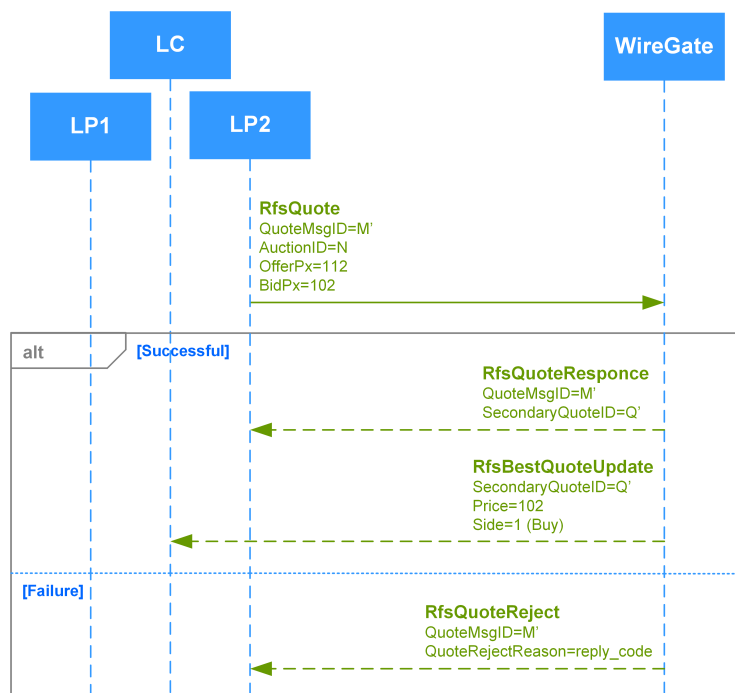


Figure 9. Diagram. Adding of the best quote

4.2.4. Liquidity provider changes the quote

When a quote is changed, the old quote is deleted and a new one is put in its place.

The liquidity provider can change its quotes no more often than the minimum quote lifetime specified in the stream. Regardless of the limitation of the rate of change of quotes, the new quote will be accepted by the system if it contains a better price than the quote set earlier.

LP2 changes his quote by sending **RfsQuote** message to WireGate.

WireGate sends **RfsQuoteReplaceResponse** message to LP2 upon successful change of quote.

WireGate sends **RfsQuoteReject** message to LP2 in case of error.

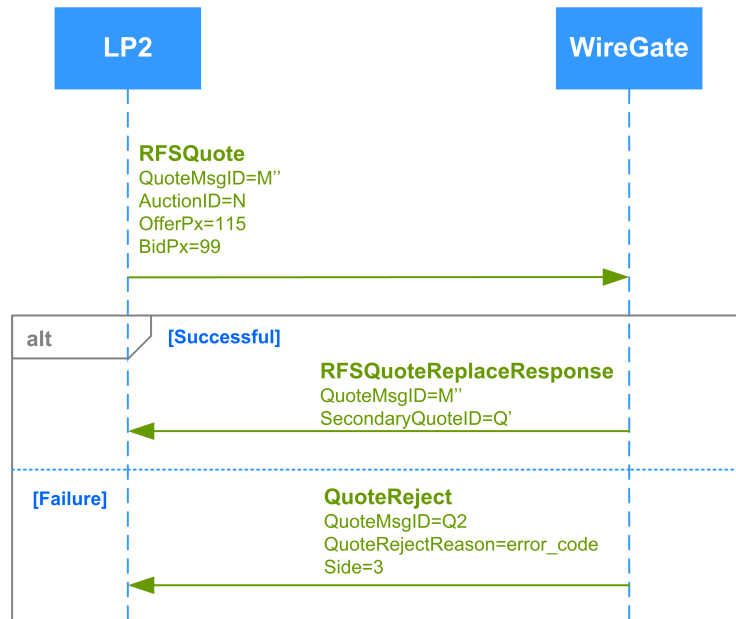


Figure 10. Diagram. Change of LP2's quote LP2

4.2.5. Cancellation of quotes by a liquidity provider LP

LP can mass cancel quotes by sending **RfsQuoteMassCancel** message to WireGate. Mass cancellation is performed by type:

- by instrument code - **SecurityID** field;
- by client code - the last three characters in the seven-digit code from the **Account** field;
- by external number - **ExternalID** field.

LP can cancel one quote by sending **RfsQuoteMassCancel** message with filled **AuctionID** and **Side** fields to WireGate.

If the best quotes in the stream have deleted, as a result of cancellation quotes, then WireGate sends **RfsBestQuoteUpdate** message (to LC) with field values:

- AuctionID = stream_id;
- SecondaryQuoteID = 0;
- QuoteSize = 0;
- Price = 0;
- Side = dir;
- MatchType - value not defined.

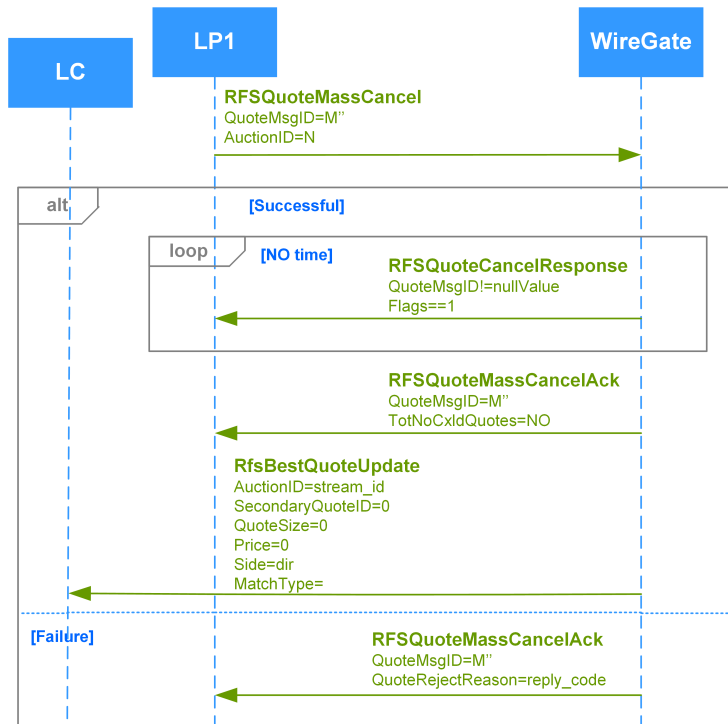


Figure 11. Diagram. Cancellation of LP quote

4.2.6. The quote is cancelled due to the end of its lifetime

When LP adds a quote, hi can send a positive value to **ExposureDuration** field in **RfsQuote** message - this is the quote life time. A quote is deleted from the trading system after its lifetime is ended and WireGate sends **RfsQuoteCancelResponse** (QuoteMsgID!=nulValue) message to LP.

4.2.7. Liquidity stream closing

Liquidity stream closing can occur in several ways:

- if liquidity consumer does not have enough funds;
- if trade completed - the stream closes automatically;
- automatic closing at the end of the stream lifetime. The stream duration is set by the liquidity consumer at its opening. By default, the stream duration is determined by the trading session - at its end, the stream closes;
- at the liquidity consumer request. LC can close only its own liquidity streams. Streams that are open from own login and from other logins of LC clients belong to own.

4.2.7.1. Liquidity stream closing by the command from LC

LC sends **CancelStream** message to WireGate.

On successful stream closing, WireGate sends:

- **CancelStreamResponse** message to LC. The fields **Account** and **Text** are broadcast to LC only.
- unsolicited message **CancelStreamResponse** - to all LP clients. If LC and LP is the same, then WireGate does not send an unsolicited message to him. WireGate broadcasts blank rows in **Account** and **Text** fields.
- **RfsQuoteCancelResponse** message - to LP-quote owner (if there are quotes in the stream).

WireGate sends **CancelStreamReject** message to LC in case of error.

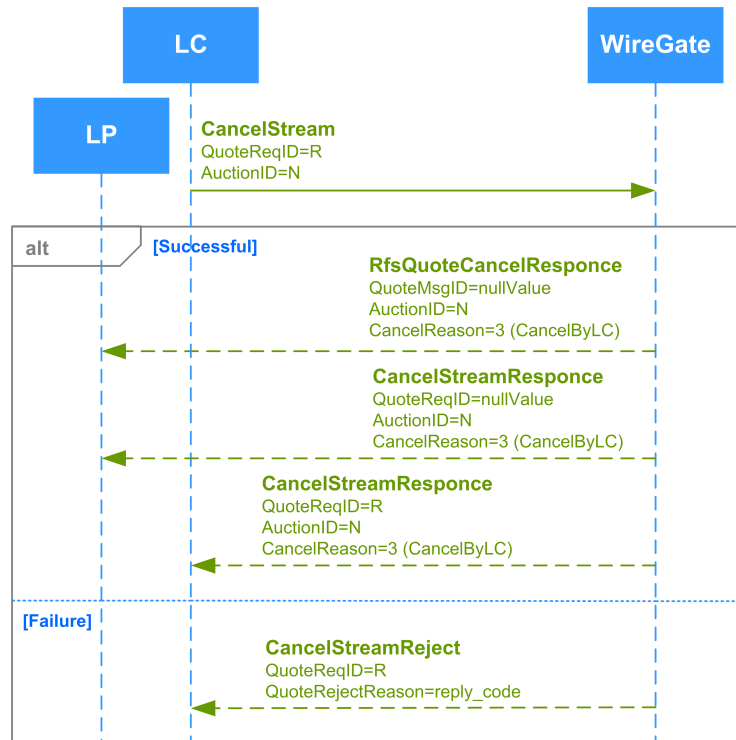


Figure 12. Diagram. Stream closing by LC command

4.2.7.2. Automatic liquidity stream closing

Reasons for automatic closing of liquidity stream

- at the trading session end, unless the stream lifetime is specified;
- at the stream lifetime end (StreamExposureDuration);
- if trade completed;
- if liquidity consumer does not have enough funds.

WireGate sends **CancelStreamResponse** message to LC when stream closes.

WireGate sends unsolicited message **CancelStreamResponse** to all LPs.

WireGate sends **RfsQuoteCancelResponse** message to LP-quote owner, if there are quotes in the stream.

When the stream closes at the trading session end or at the stream lifetime end, a bit **0x200000000000** is fired in **Flags** field in **RfsQuoteCancelResponse** message.

CancelReason field in **CancelStreamResponse** message have one of the following values:

- 6 (EndOfTradingSession) - the trading session end;
- 4 (TimeOut) - the stream lifetime end;
- 1 (Deal) - trade is executed in the stream;
- 2 (LCDoesntHaveEnoughMoney) - LC does not have enough funds;
- 5 (CancelByAdministrator) - trading administrator's decision.

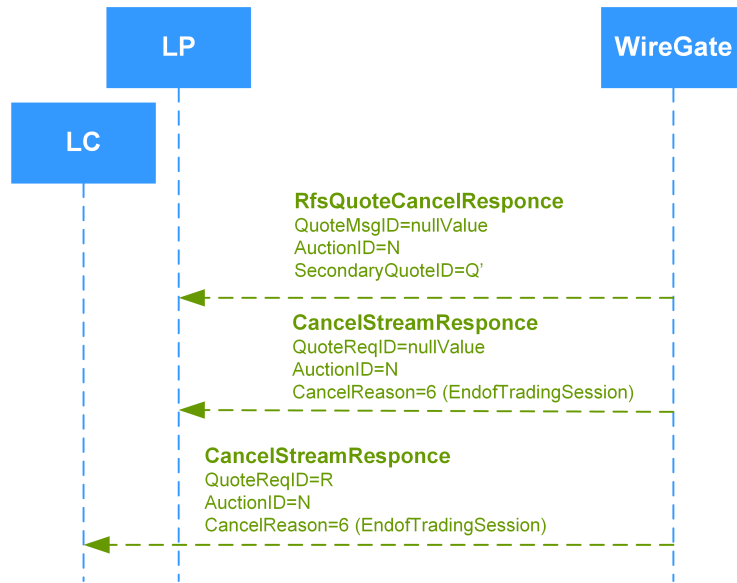


Figure 13. Diagram. Automatic stream closing

4.2.8. Liquidity consumer hits the quote

LC cannot see all offers from providers. He sees only the best price to buy and sell, and if it suits him, he can accept this offer.

Accepting the best quote, LC sends **RfsQuoteHit** message to WireGate.

WireGate send **RfsQuoteHitAck** message to LC (**QuoteRejectReason = 0**), If quotes are matched into indicative trade.

WireGate sends **RfsQuoteHitAck (QuoteRejectReason != 0)** message to LC in case of error.

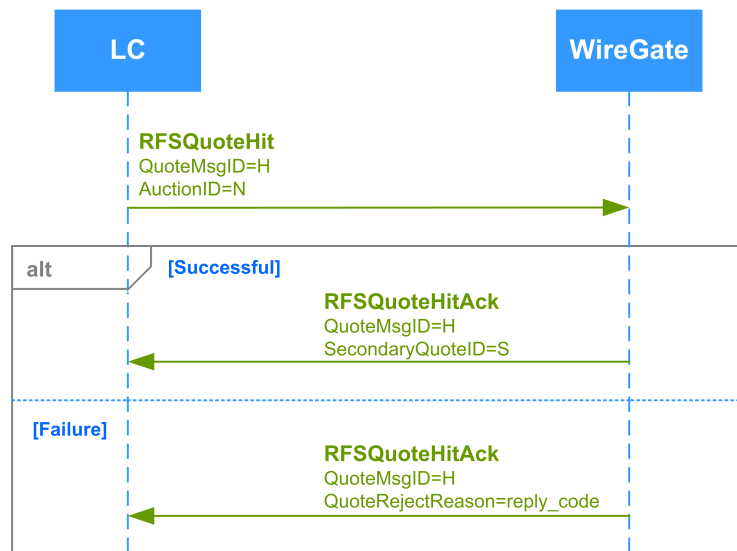


Figure 14. Diagram. LC's quote

4.2.9. Match quotes into a trade and Last Look

Accepting LP offer LC sets a matching quote. In this case, an indicative trade is executed in the system.

If the provider quote is with auto-confirmation, the trade will be made automatically without asking for any confirmation. If the provider quote is without auto-confirmation, then an additional iteration is added to the process of making trade - confirmation from the provider - Last Look (see Section 4.2.9.4, "Provider indicative trade confirmation - Last Look"). When LC opens liquidity stream he can set it up so that quotes with auto-confirmation will be accept only.

Two negotiated orders are then added into SPECTRA, one by one, that perform a trade. Finally, the process of matching two quotes into trade is considered to be completed and liquidity stream closes.

Every stage of matching indicative quotes is indicated by indicative trade status change.

Below is the quote matching algorithm, step by step:

1. After two RFS quotes are matched, an indicative trade is performed within RFS, and **RfsExecutionReport** message with **Status=0** (being processed) will be sent to the client side.
2. A negotiated order from LC based on the indicative trade parameters is added into SPECTRA. Upon adding the order, all necessary checks and verifications are applied, including the one for collateral sufficiency. After the order is added, **RfsExecutionReport** message with **Status=1** (wait for confirmation from provider) will be sent to the client side.
3. Within a certain time frame (specified by administrator), RFS is awaiting for trade confirmation from the provider.
4. After confirmation is received, **RfsExecutionReport** message with **Status=2** (confirmation from provider received) will be sent to the client side. A negotiated order from liquidity provider (LP) based on the indicative trade parameters is added into SPECTRA.
5. After two orders are matched into trade within SPECTRA, the **RfsExecutionReport** message with **Status=4** (trade is performed) will be sent to the client side.
6. Once there any error occurs on matching the quotes, or, for any reason, the trade cannot be confirmed, the **RfsExecutionReport** message with **Status=3** (trade not performed due to an error) will be sent to the client side, containing the appropriate error code in **RejectReason** field.

4.2.9.1. LC hits the quote without Last Look

When LC accept provider quote, he sends **RfsQuoteHit** message to LC.

Upon successful execution of quotes WireGate sends:

- **RfsExecutionReport** message to LC;
- **RfsExecutionReport** message to LP, whose quote is executed;
- **CancelStreamResponse** message (**CancelReason = 1 (Deal)**) message to all LPs.

In case of failure WireGate sens **RfsQuoteHitAck** message to LC with non-zero error code in **QuoteRejectReason** field.

The diagram shows the case of successful execution of RFS quotes and successful making trade in the Spectra system.

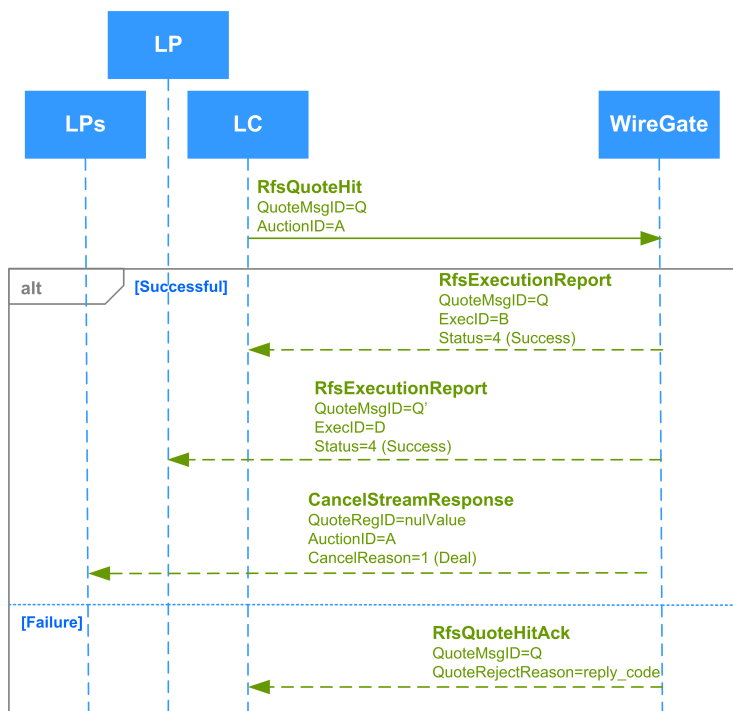


Figure 15. Diagram. Accept from LC without Last Look

4.2.9.2. LP negotiated order is rejected by the Spectra

Negotiated order from LP cannot be put up to the Spectra system if LP does not have enough funds.

In this case:

- liquidity stream goes on;
- LP quote is deleted from the liquidity stream;
- WireGate sends **RfsExecutionReport** message to LC with **Status = 3 (Failed)**;
- WireGate sends **RfsExecutionReport** message to LP, whose quote was involved in indicative trade, with **Status = 3 (Failed)**;
- if the best quotes in the stream have changed, then WireGate sends **RfsBestQuoteUpdate** message with the update of the best quotes to LC (the diagram does not show this case).

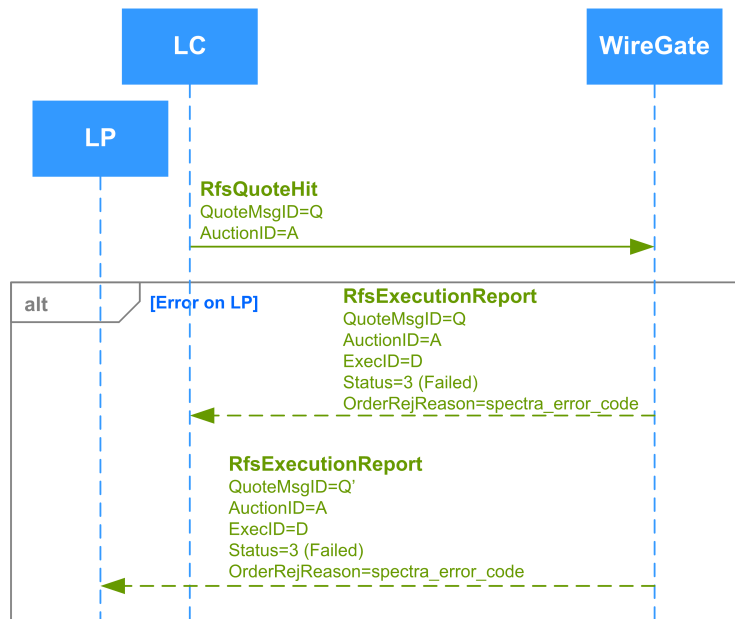


Figure 16. Diagram. LP negotiated order rejection in Spectra

4.2.9.3. LC negotiated order is rejected by the Spectra

Negotiated order from LC cannot be put up to the Spectra system if LC does not have enough funds.

In this case:

- WireGate sends **RfsExecutionReport** message to LC with **Status = 3 (Failed)**;
- WireGate sends **RfsExecutionReport** message to LP, whose quote was involved in indicative trade, with **Status = 3 (Failed)**;
- liquidity stream is closed;
- WireGate sends **CancelStreamResponse** message to LC with **CancelReason = 2 (LCDoesntHaveEnoughMoney)**;
- WireGate sends **CancelStreamResponse** message to all LPs with **CancelReason = 2 (LCDoesntHaveEnoughMoney)**;
- all providers quotes of this stream are deleted from the trading system.

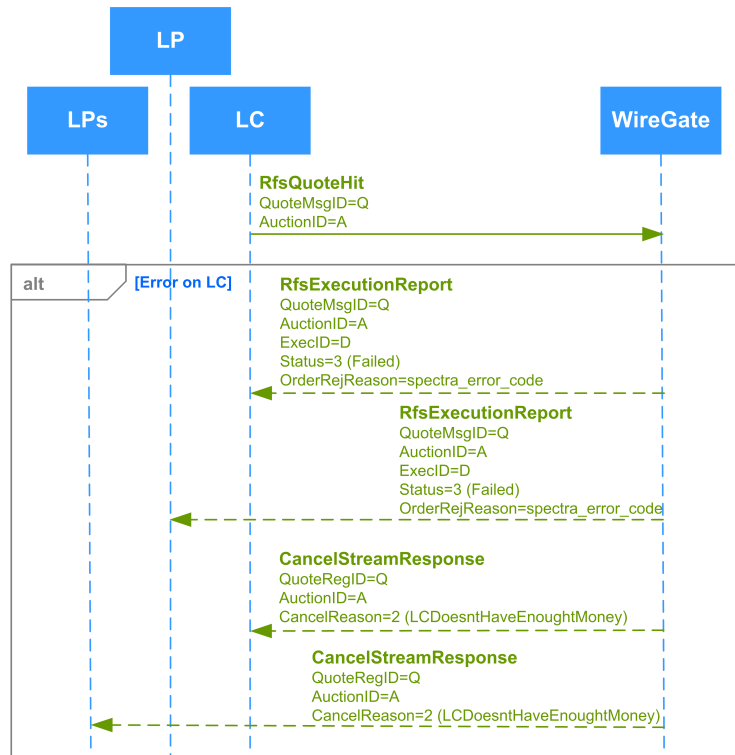


Figure 17. Diagram. LC negotiated order rejection in Spectra

4.2.9.4. Provider indicative trade confirmation - Last Look

LC accepts provider offer and sends **RfsQuoteHit** message to WireGate.

If all checks for the LC quote are successful then WireGate sends **ExecutionReport** message to LP with **Status = 1** (WaitConfirm).

Further, there are three possible scenarios:

1. Provider confirms indicative trade:

- LP sends **RfsConfirmation** message to WireGate;
- WireGate sends **RfsExecutionReport** message to LC with **Status = 2** (Confirmed);
- WireGate sends **RfsExecutionReport** message to LP with **Status = 2** (Confirmed);

2. Provider does not confirm indicative trade:

- LP sends **RfsConfirmation** message to WireGate;
- WireGate sends **RfsConfirmationAck** message to LP with non-zero error code in **QuoteRejectReason** field.

3. If LP does not send a message to WireGate during the timeout, the RFS system considers that LP refused the trade. In this case:

- WireGate sends **RfsExecutionReport** message to LC with **Status = 3** (NotConfirmed);
- WireGate sends **RfsExecutionReport** message to LP with **Status = 3** (NotConfirmed).

If the indicative trade is not confirmed by the provider and the time is out, then in this case:

- liquidity stream goes on;
- LP quote is deleted from the liquidity stream;
- if the best quotes in the stream have changed, then WireGate sends **RfsBestQuoteUpdate** message with the update of the best quotes to LC (the diagram does not show this case).

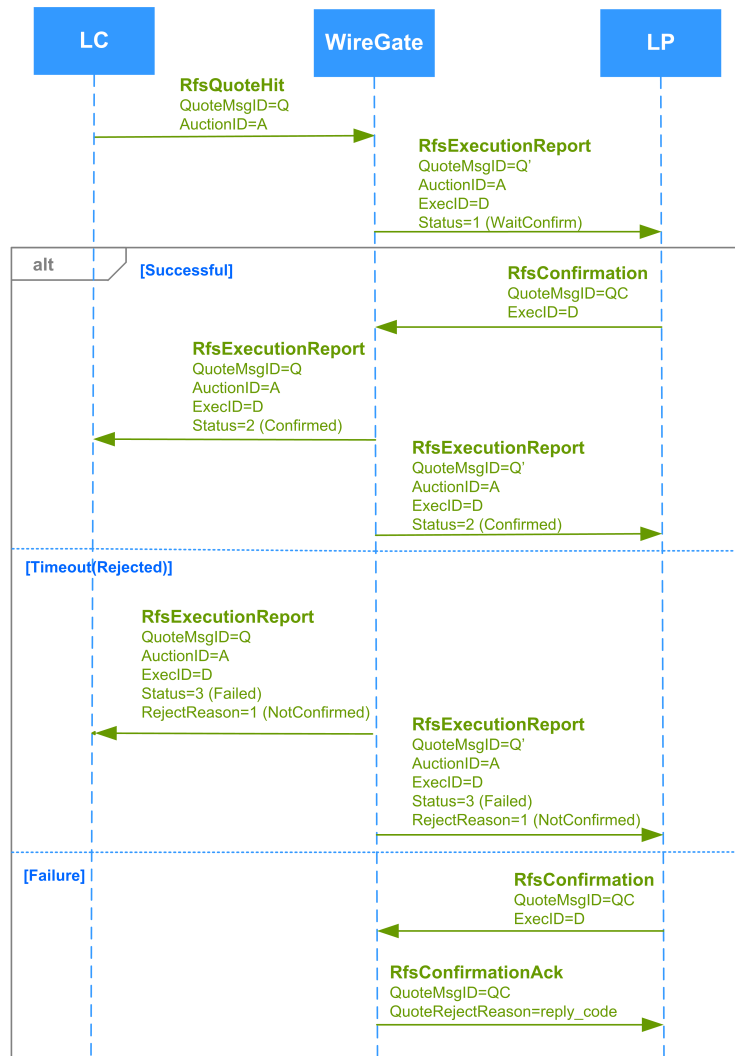


Figure 18. Diagram. Procedure Last Look

4.2.10. Client quote ID

The WireGate checks the identifiers uniqueness. A client is required to provide unique 'QuoteMsgID' for quotes with the session-long lifetime during a single trading session. Once there is a non-unique QuoteMsgID sent by the client side, it will be rejected by WireGate, and message **SessionReject** with error code 101 (QuoteMsgIDIsNotUnique) will be sent to the client side.

5. Message scheme

The current message scheme is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="sbe_schema.xsl" type="text/xsl"?>
<sbe:messageSchema package="moex_spectra_rfs_twime" byteOrder="littleEndian" id="20809" version="1"
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="String64" primitiveType="char" length="64"/>

    <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="BooleanEnum" encodingType="uint8">
      <validValue name="False" >0</validValue>
      <validValue name="True" >1</validValue>
    </enum>

    <enum name="TerminationCodeEnum" encodingType="uint8">
      <validValue name="Finished" >0</validValue>
      <validValue name="UnspecifiedError" >1</validValue>
      <validValue name="ReRequestOutOfBounds" >2</validValue>
      <validValue name="ReRequestInProgress" >3</validValue>
      <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>
      <validValue name="SequenceReset" >11</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>
  </types>
</sbe:messageSchema>
```

```

    <validValue name="SystemIsUnavailable" >100</validValue>
    <validValue name="QuoteMsgIDIsNotUnique">101</validValue>
  </enum>

  <enum name="TimeInForceEnum" encodingType="uint8">
    <validValue name="Day" >0</validValue>
    <validValue name="IOC" >3</validValue>
  </enum>

  <enum name="SideEnum" encodingType="uint8">
    <validValue name="Unavailable" >0</validValue>
    <validValue name="Buy" >1</validValue>
    <validValue name="Sell" >2</validValue>
    <validValue name="BothSides" >89</validValue>
  </enum>

  <enum name="ModeEnum" encodingType="uint8">
    <validValue name="DontChangeQuoteQty" >0</validValue>
    <validValue name="ChangeQuoteQty" >1</validValue>
    <validValue name="CheckQuoteQtyAndCancelQuote" >2</validValue>
    <validValue name="FixStyleReplace" >3</validValue>
  </enum>

  <enum name="MatchTypeEnum" encodingType="uint8">
    <validValue name="AutoMatch" >4</validValue>
    <validValue name="AutoMatchWithLastLook" >10</validValue>
  </enum>

  <enum name="TradSesEventEnum" encodingType="uint16">
    <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>
    <validValue name="OtcSessionInited" >10100</validValue>
    <validValue name="OtcSessionStarted" >10101</validValue>
    <validValue name="OtcSessionSuspended" >10102</validValue>
    <validValue name="OtcSessionStoped" >10103</validValue>
    <validValue name="OtcSessionFinished" >10104</validValue>
  </enum>

  <enum name="SecurityTypeEnum" encodingType="uint8">
    <validValue name="Future" >0</validValue>
    <validValue name="Option" >1</validValue>
    <validValue name="Multileg" >2</validValue>
  </enum>

  <enum name="StatusEnum" encodingType="uint8">
    <validValue name="Matched" >0</validValue>
    <validValue name="WaitConfirm" >1</validValue>
    <validValue name="Confirmed" >2</validValue>
    <validValue name="Failed" >3</validValue>
    <validValue name="Success" >4</validValue>
  </enum>

  <enum name="RejectReasonEnum" encodingType="uint8">
    <validValue name="NotApplicable" >0</validValue>
    <validValue name="NotConfirmed" >1</validValue>
    <validValue name="ActiveSideError" >2</validValue>
    <validValue name="PassiveSideError" >3</validValue>
  </enum>

  <enum name="StreamExposureDurationEnum" encodingType="uint8">
    <validValue name="NotApplicable" >0</validValue>
    <validValue name="Duration30sec" >1</validValue>
    <validValue name="Duration60sec" >2</validValue>
    <validValue name="Duration90sec" >3</validValue>
    <validValue name="Duration120sec" >4</validValue>
  </enum>

  <enum name="SpeedBumpTypeEnum" encodingType="uint8">

```

```

    <validValue name="NotApplicable" >0</validValue>
    <validValue name="Duration200ms" >1</validValue>
    <validValue name="Duration500ms" >2</validValue>
    <validValue name="Duration1000ms" >3</validValue>
    <validValue name="Duration3000ms" >4</validValue>
  </enum>

  <enum name="CancelReasonEnum" encodingType="uint8">
    <validValue name="Deal" >1</validValue>
    <validValue name="LCDoesntHaveEnoughMoney" >2</validValue>
    <validValue name="CancelByLC" >3</validValue>
    <validValue name="TimeOut" >4</validValue>
    <validValue name="CancelByAdministrator" >5</validValue>
    <validValue name="EndOfTradingSession" >6</validValue>
  </enum>

  <set name="FlagsSet" encodingType="uint64">
    <choice name="Day" description="Quotes: Day" >0</choice>
    <choice name="IOC" description="Quotes: IOC" >1</choice>
    <choice name="Replace" description="Quotes:
      The record results from replacing the quote" >20</choice>
    <choice name="Cancel" description="Quotes:
      The record results from cancelling the quote" >21</choice>
    <choice name="MassCancel" description="Quotes:
      The record results from mass cancelling" >22</choice>
    <choice name="MultiLeg" description="Quotes: Multi leg"
      >27</choice>
    <choice name="FineOperation" description="Quotes:
      Flag of cancelling the quote because of fine" >36</choice>
    <choice name="ActiveSide" description="Trades: Quote initiator is aggressor"
      >41</choice>
    <choice name="PassiveSide" description="Trades: Quote initiator is passive"
      >42</choice>
    <choice name="TimeOut" description="Quotes: Delete quote due timeout"
      >49</choice>
    <choice name="AutoMatch" description="Quotes and trades:
      Auto match without last look" >50</choice>
  </set>

  <set name="StreamFlagsSet" encodingType="uint64">
    <choice name="AutoMatch" description="Auto match without last look">0</choice>
    <choice name="ClosedStream" description="Stream is closed">1</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>

  <!-- do not change this one as it is hardcoded in serializer -->
  <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>

<!-- Session layer messages -->
<sbe:message name="Establish" id="5100">
  <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="5101">
  <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="5102">
  <field name="RequestTimestamp" id="20207" type="TimeStamp" />
  <field name="EstablishmentRejectCode" id="20209" type="EstablishmentRejectCodeEnum" />
</sbe:message>

<sbe:message name="Terminate" id="5103">
  <field name="TerminationCode" id="20210" type="TerminationCodeEnum" />
</sbe:message>

<sbe:message name="RetransmitRequest" id="5104">
  <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="5105">
  <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="5106">
  <field name="NextSeqNo" id="20208" type="UInt64" />
</sbe:message>

<sbe:message name="FloodReject" id="5107">
  <field name="QuoteMsgID" id="1166" type="UInt64"/>
  <field name="QueueSize" id="20213" type="UInt32"/>
  <field name="PenaltyRemain" id="20214" type="UInt32"/>
</sbe:message>

<sbe:message name="SessionReject" id="5108">
  <field name="QuoteMsgID" id="1166" type="UInt64"/>
  <field name="RefTagID" id="371" type="UInt32"/>
  <field name="SessionRejectReason" id="373" type="SessionRejectReasonEnum"/>
</sbe:message>

<!-- Application layer messages -->
<sbe:message name="NewStream" id="8007">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="MinQty" id="110" type="UInt64" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="StreamExposureDuration" id="21011" type="StreamExposureDurationEnum"/>
  <field name="MatchType" id="574" type="MatchTypeEnum" />
  <field name="SpeedBumpType" id="21000" type="SpeedBumpTypeEnum" />
  <field name="Account" id="1" type="String7" />
  <field name="TextToLP" id="21001" type="String20" />
  <field name="Text" id="58" type="String20" />
</sbe:message>

<sbe:message name="CancelStream" id="8008">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="Account" id="1" type="String7" />
</sbe:message>

<sbe:message name="RfsQuote" id="8009">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="OfferPx" id="133" type="Decimal5" />
  <field name="OfferExternalID" id="20025" type="UInt64" />
  <field name="BidPx" id="132" type="Decimal5" />
  <field name="BidExternalID" id="20024" type="UInt64" />
  <field name="ExposureDuration" id="1629" type="UInt64" />
  <field name="MatchType" id="574" type="MatchTypeEnum" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="Account" id="1" type="String7" />
  <field name="OfferText" id="21003" type="String20" />
  <field name="BidText" id="21004" type="String20" />

```

```

</sbe:message>

<sbe:message name="RfsQuoteMassCancel" id="8011">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="Account" id="1" type="String7" />
</sbe:message>

<sbe:message name="RfsQuoteHit" id="8012">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="Price" id="44" type="Decimal5" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="Text" id="58" type="String20" />
</sbe:message>

<sbe:message name="RfsConfirmation" id="8013">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="ExecID" id="17" type="UInt64" />
</sbe:message>

<sbe:message name="NewStreamResponse" id="9011">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="MinQty" id="110" type="UInt64" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="TradingSessionID" id="336" type="Int32" />
  <field name="StreamFlags" id="21005" type="StreamFlagsSet" />
  <field name="SecurityType" id="167" type="SecurityTypeEnum" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="StreamExposureDuration" id="21011" type="StreamExposureDurationEnum" />
  <field name="SpeedBumpType" id="21000" type="SpeedBumpTypeEnum" />
  <field name="TextToLP" id="21001" type="String20" />
  <field name="Text" id="58" type="String20" />
  <field name="TagOfLC" id="21007" type="String64" />
</sbe:message>

<sbe:message name="NewStreamReject" id="9012">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="QuoteRejectReason" id="300" type="Int32" />
</sbe:message>

<sbe:message name="CancelStreamResponse" id="9013">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="MinQty" id="110" type="UInt64" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="ExecID" id="17" type="UInt64" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="TradingSessionID" id="336" type="Int32" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="StreamExposureDuration" id="21011" type="StreamExposureDurationEnum" />
  <field name="StreamFlags" id="21005" type="StreamFlagsSet" />
  <field name="SecurityType" id="167" type="SecurityTypeEnum" />
  <field name="SpeedBumpType" id="21000" type="SpeedBumpTypeEnum" />
  <field name="CancelReason" id="21008" type="CancelReasonEnum" />
  <field name="TextToLP" id="21001" type="String20" />
  <field name="Text" id="58" type="String20" />
  <field name="TagOfLC" id="21007" type="String64" />
</sbe:message>

<sbe:message name="CancelStreamReject" id="9014">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="QuoteRejectReason" id="300" type="Int32" />

```

```

</sbe:message>

<sbe:message name="RfsQuoteResponse" id="9015">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="SecondaryQuoteID" id="1751" type="UInt64" />
  <field name="QuoteSize" id="20026" type="UInt64" />
  <field name="Price" id="44" type="Decimal5" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="ExposureDuration" id="1629" type="UInt64" />
  <field name="Flags" id="20215" type="FlagsSet" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="TradingSessionID" id="336" type="Int32" />
  <field name="SecurityType" id="167" type="SecurityTypeEnum" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="CodeOfLP" id="21009" type="String20" />
  <field name="Text" id="58" type="String20" />
</sbe:message>

<sbe:message name="RfsQuoteReplaceResponse" id="9016">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="SecondaryQuoteID" id="1751" type="UInt64" />
  <field name="QuoteSize" id="20026" type="UInt64" />
  <field name="Price" id="44" type="Decimal5" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="PrevSecondaryQuoteID" id="20034" type="UInt64" />
  <field name="ExposureDuration" id="1629" type="UInt64" />
  <field name="Flags" id="20215" type="FlagsSet" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="TradingSessionID" id="336" type="Int32" />
  <field name="SecurityType" id="167" type="SecurityTypeEnum" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="CodeOfLP" id="21009" type="String20" />
</sbe:message>

<sbe:message name="RfsQuoteReject" id="9017">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="QuoteRejectReason" id="300" type="Int32" />
  <field name="Side" id="54" type="SideEnum" />
</sbe:message>

<sbe:message name="RfsQuoteCancelResponse" id="9018">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="SecondaryQuoteID" id="1751" type="UInt64" />
  <field name="QuoteSize" id="20026" type="UInt64" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="Flags" id="20215" type="FlagsSet" />
  <field name="TradingSessionID" id="336" type="Int32" />
</sbe:message>

<sbe:message name="RfsQuoteMassCancelAck" id="9020">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="TotNoCxldQuotes" id="1168" type="Int32" />
  <field name="TotNoSpeedBumpQuotes" id="21012" type="Int32" />
  <field name="QuoteRejectReason" id="300" type="Int32" />
</sbe:message>

<sbe:message name="RfsBestQuoteUpdate" id="9021">
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="SecondaryQuoteID" id="1751" type="UInt64" />
  <field name="QuoteSize" id="20026" type="UInt64" />
  <field name="Price" id="44" type="Decimal5" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="MatchType" id="574" type="MatchTypeEnum" />
</sbe:message>

```

```

<sbe:message name="RfsQuoteHitAck" id="9022">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="SecondaryQuoteID" id="1751" type="UInt64" />
  <field name="QuoteRejectReason" id="300" type="Int32" />
</sbe:message>

<sbe:message name="RfsConfirmationAck" id="9023">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="ExecID" id="17" type="UInt64" />
  <field name="QuoteRejectReason" id="300" type="Int32" />
</sbe:message>

<sbe:message name="RfsExecutionReport" id="9024">
  <field name="QuoteMsgID" id="1166" type="UInt64" />
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="AuctionID" id="21002" type="UInt64" />
  <field name="SecondaryQuoteID" id="1751" type="UInt64" />
  <field name="LastPx" id="31" type="Decimal5" />
  <field name="LastQty" id="32" type="UInt64" />
  <field name="ExposureDuration" id="1629" type="UInt64" />
  <field name="ExternalID" id="20027" type="UInt64" />
  <field name="ExecID" id="17" type="UInt64" />
  <field name="TrdMatchID" id="880" type="Int64" />
  <field name="OrderID" id="37" type="Int64" />
  <field name="TradingSessionID" id="336" type="Int32" />
  <field name="SecurityID" id="48" type="Int32" />
  <field name="OrdRejReason" id="103" type="Int32" />
  <field name="SecurityType" id="167" type="SecurityTypeEnum" />
  <field name="Side" id="54" type="SideEnum" />
  <field name="Status" id="20030" type="StatusEnum" />
  <field name="RejectReason" id="20033" type="RejectReasonEnum" />
  <field name="CodeOfLP" id="21009" type="String20" />
  <field name="Text" id="58" type="String20" />
</sbe:message>

<sbe:message name="EmptyBook" id="9009">
  <field name="Timestamp" id="20204" type="TimeStamp" />
  <field name="TradingSessionID" id="336" type="Int32" />
</sbe:message>

<sbe:message name="SystemEvent" id="9010">
  <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>

```