



SPECTRA Plaza-2 gate

version 6.15

Almaty 2022

Table of Contents

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. Introduction | 8 |
| 1.1. Document purpose | 8 |
| 1.2. Target audience | 8 |
| 1.3. Abbreviations | 8 |
| 1.4. Additional Information | 8 |
| 2. SPECTRA system overview | 9 |
| 2.1. Trading Membership | 9 |
| 2.1.1. Clearing firms | 9 |
| 2.1.2. Brokerage firms | 9 |
| 2.1.3. Clients | 9 |
| 2.1.4. System code pattern | 9 |
| 2.1.5. Disclosure of data on participants | 10 |
| 2.1.6. Users. How a user is linked to a trading participant | 10 |
| 2.2. Instruments | 10 |
| 2.2.1. Underlying assets | 10 |
| 2.2.2. Futures | 10 |
| 2.2.3. Identification of instruments | 11 |
| 2.3. Trading operations | 11 |
| 2.3.1. Orders — general information | 11 |
| 2.3.2. Negotiated orders (Negotiated trades mode) | 12 |
| 2.3.3. Trades | 12 |
| 2.3.4. Iceberg orders | 12 |
| 2.3.4.1. Iceberg orders in the system information streams | 13 |
| 2.3.4.2. Iceberg order operations | 15 |
| 2.3.4.3. Change of order ID during iceberg orders operations | 15 |
| 2.4. Delivery of assets and exercise of options (**) | 16 |
| 2.4.1. Deliveries on futures (**) | 16 |
| 2.4.2. Trade types, created upon exercising and expiration of futures and options (**) | 16 |
| 2.5. Trading and clearing schedule | 19 |
| 2.5.1. Trading schedule. Trading sessions. | 19 |
| 2.5.2. Clearing session | 19 |
| 2.5.3. How different entities act on assigning a new trading session | 19 |
| 2.5.3.1. Reference data and session data | 19 |
| 2.5.3.2. Funds and positions | 19 |
| 2.5.3.3. Orders and trades | 19 |
| 2.5.3.4. Instruments | 19 |
| 2.5.3.5. Replication streams | 19 |
| 2.5.4. Event-sensitive scheme for data synchronizing | 20 |
| 2.6. Risk management and limitation of trading operations | 21 |
| 2.6.1. Collaterals | 21 |
| 2.6.2. Trading limits | 22 |
| 2.6.3. Limitations on trading operations and opening positions for clients | 22 |
| 2.6.4. Position (obligation) transfer | 23 |
| 2.6.5. Pausing trading session for extending limits of trading prices fluctuations | 23 |
| 2.6.6. Risk parameters forecast information for trading participants | 23 |
| 2.6.7. Blocking the brokerage part of the client fee | 24 |
| 2.6.8. Negative prices support in SPECTRA (**) | 24 |
| 2.7. SMA Login (Sponsored Market Access) | 26 |
| 2.8. Settlement trades | 27 |
| 2.8.1. Reasons for settlement trades | 27 |
| 2.8.2. Fines and fees | 28 |
| 3. Trading gate description | 29 |
| 3.1. SPECTRA Plaza-2 gateway. Components, installation and setup. | 29 |
| 3.1.1. Components and architecture | 29 |
| 3.1.2. Hardware and software requirements | 29 |
| 3.1.2.1. Hardware requirements | 29 |
| 3.1.2.2. Software requirements | 30 |
| 3.1.3. Installation for Windows | 30 |
| 3.1.4. Installation for Linux | 35 |
| 3.1.5. Developer guidelines | 35 |
| 3.1.5.1. Usage of test examples | 35 |
| 3.1.5.2. Distributed configurations | 37 |
| 3.1.5.3. Recommendations for third-party companies on including the KASE runtimes into user application when distributing the user software | 37 |
| 3.2. Market data structure | 37 |
| 3.2.1. Reference information | 37 |
| 3.2.2. Trade information | 38 |
| 3.2.3. Recovery information | 38 |
| 3.2.4. User's limits information | 38 |

| | |
|--------------------------------------------------------------------------------------------------------|----|
| 3.2.5. Clearing information | 38 |
| 3.2.6. Auxiliary information streams | 39 |
| 3.3. Gateway usage specifics | 39 |
| 3.3.1. Service replication fields | 39 |
| 3.3.2. Commands | 39 |
| 3.3.3. Flood control | 39 |
| 3.3.4. Latency monitoring by the client side | 40 |
| 3.3.5. Cancel On Disconnect | 40 |
| 3.3.6. Replication stream sets for different login subtypes | 40 |
| 3.3.7. Password policy | 42 |
| 3.3.7.1. Basic provisions | 42 |
| 3.3.7.2. Changing user password for the Trading System | 42 |
| 3.3.8. Partitioning of the matching | 43 |
| 3.3.9. Stream types | 44 |
| 3.4. Handling abnormal situations | 44 |
| 3.4.1. Recovery on loss of connection with Exchange servers | 44 |
| 3.4.1.1. Connection loss detection | 45 |
| 3.4.1.2. Recovery algorithm | 45 |
| 3.4.1.3. General recommendations | 47 |
| 3.4.2. Recovery in case of the Exchange infrastructure failure | 47 |
| 3.4.2.1. Data cleanup by streams | 47 |
| 3.4.2.2. Possible data change in case of abnormal work of publishing services | 47 |
| 4. Replication scheme FORTS_PUBLIC | 49 |
| 4.1. Stream FORTS_TRADE_REPL - User's orders and trades (Type=R) | 49 |
| 4.1.1. Data scheme | 49 |
| 4.1.1.1. Table orders_log: Log of operations with orders | 49 |
| 4.1.1.2. Table user_deal: User trades | 50 |
| 4.1.1.3. Table heartbeat: Server times table | 52 |
| 4.1.1.4. Table sys_events: table of events | 52 |
| 4.2. Stream FORTS_ORDLOG_REPL – anonymous orders (Type=R) | 52 |
| 4.2.1. Data scheme | 52 |
| 4.2.1.1. Table orders_log: Log of operations with orders | 52 |
| 4.2.1.2. Table heartbeat: Server times table | 53 |
| 4.2.1.3. Table sys_events: table of events | 53 |
| 4.3. Stream FORTS_DEALS_REPL – anonymous trades (Type=R) | 54 |
| 4.3.1. Data scheme | 54 |
| 4.3.1.1. Table deal: Trades | 54 |
| 4.3.1.2. Table heartbeat: Server times table | 54 |
| 4.3.1.3. Table sys_events: table of events | 55 |
| 4.4. Stream FORTS_FEE_REPL - exchange fees and penalties (Type=AR) | 55 |
| 4.4.1. Data scheme | 55 |
| 4.4.1.1. Table adjusted_fee: exchange fees | 55 |
| 4.4.1.2. Table penalty: penalties | 56 |
| 4.4.1.3. Table sys_events: table of events | 56 |
| 4.5. Stream FORTS_FEERATE_REPL - Precise Exchange fee rates (Type=AR) | 56 |
| 4.5.1. Data scheme | 56 |
| 4.5.1.1. Table futures_rate: fee rates on futures | 57 |
| 4.5.1.2. Table sys_events: table of events | 57 |
| 4.6. Stream FORTS_BROKER_FEE_REPL - Brokerage fees (Type=I) | 57 |
| 4.6.1. Data scheme | 57 |
| 4.6.1.1. Table broker_fee: brokerage fee | 57 |
| 4.6.1.2. Table sys_events: table of events | 58 |
| 4.7. Stream FORTS_BROKER_FEE_PARAMS_REPL - Parameters for calculating the brokerage fee (Type=I) | 58 |
| 4.7.1. Data scheme | 58 |
| 4.7.1.1. Table broker_fee_params: Parameters for calculating the brokerage fee | 58 |
| 4.7.1.2. Table sys_events: table of events | 59 |
| 4.8. Stream FORTS_USERORDERBOOK_REPL - User orders: order-book snapshot (Type=R) | 59 |
| 4.8.1. Data scheme | 59 |
| 4.8.1.1. Table orders: User current order-book | 60 |
| 4.8.1.2. Table info: Order-book snapshots information | 61 |
| 4.9. Stream FORTS_ORDBOOK_REPL - Depersonalized order-book snapshot (Type=R) | 62 |
| 4.9.1. Data scheme | 62 |
| 4.9.1.1. Table orders: Current order-book | 62 |
| 4.9.1.2. Table info: Order-book snapshots information | 63 |
| 4.10. Stream FORTS_COMMON_REPL - Market fundamentals (Type=I) | 63 |
| 4.10.1. Data scheme | 63 |
| 4.10.1.1. Table common: Market fundamentals | 63 |
| 4.11. Aggregated order-book streams (Type=I) | 64 |
| 4.11.1. Data scheme | 64 |
| 4.11.1.1. Table orders_aggr: Aggregated order-books | 64 |
| 4.12. Stream FORTS_POS_REPL - information on positions (Type=I) | 65 |
| 4.12.1. Data scheme | 65 |

| | |
|-----------------------------------------------------------------------------------------------------------------------|----|
| 4.12.1.1. Table position: Client positions | 65 |
| 4.12.1.2. Table position_sa: Settlement Account positions | 66 |
| 4.12.1.3. Table sys_events: table of events | 66 |
| 4.13. Stream FORTS_PART_REPL - information about funds and limits (Type=I) | 66 |
| 4.13.1. Data scheme | 66 |
| 4.13.1.1. Table part: Funds and limits of clients and brokerage firms | 66 |
| 4.13.1.2. Table part_sa: Funds and limits for Settlement Account | 67 |
| 4.13.1.3. Table sys_events: table of events | 67 |
| 4.14. Stream FORTS_PROHIBITION_REPL - Prohibitions (Type=R) | 68 |
| 4.14.1. Data scheme | 68 |
| 4.14.1.1. Table prohibition: Prohibitions | 68 |
| 4.15. Stream FORTS_REFDATA_REPL - Reference and session information (Type=R) | 69 |
| 4.15.1. Data scheme | 69 |
| 4.15.1.1. Table fut_sess_contents: Traded instruments directory (futures) | 70 |
| 4.15.1.2. Table fut_vcb: Traded assets directory (futures) | 71 |
| 4.15.1.3. Table fut_instruments: Instruments dictionary | 72 |
| 4.15.1.4. Table fut_bond_registry: Spot asset parameters directory | 72 |
| 4.15.1.5. Table dealer: Companies directory | 73 |
| 4.15.1.6. Table sys_messages: Trading system messages | 74 |
| 4.15.1.7. Table prohibition: Prohibitions | 75 |
| 4.15.1.8. Table fut_rejected_orders: Register of orders rejected during the clearing (futures) | 76 |
| 4.15.1.9. Table fut_bond_nkd: Accrued interest as of the bond futures contract expiration date | 76 |
| 4.15.1.10. Table fut_bond_nominal: Payment of bonds' face value | 77 |
| 4.15.1.11. Table fut_bond_isin: Guide on bond instruments | 77 |
| 4.15.1.12. Table user: System users | 77 |
| 4.15.1.13. Table investor: Clients directory | 78 |
| 4.15.1.14. Table fut_margin_type: Type of margining | 79 |
| 4.15.1.15. Table fut_settlement_account: Settlement Account | 79 |
| 4.15.1.16. Table session: Information about a trading session | 80 |
| 4.15.1.17. Table sma_master: SMA login binding to MASTER login | 81 |
| 4.15.1.18. Table sma_pre_trade_check: SMA login pre-trade verification settings | 81 |
| 4.15.1.19. Table clearing_members: Clearing Members | 81 |
| 4.15.1.20. Table instr2matching_map: Instrument binding to Matching ID | 82 |
| 4.15.1.21. Table sys_events: table of events | 82 |
| 4.16. Stream FORTS_MM_REPL - information on MM's obligations (Type=I) | 82 |
| 4.16.1. Data scheme | 82 |
| 4.16.1.1. Table fut_MM_info: MM's obligations in futures | 82 |
| 4.16.1.2. Table mm_agreement_filter: Table numbers and types of contracts for the provision of market-making services | 83 |
| 4.17. Stream FORTS_CLR_REPL - clearing information (Type=AR) | 83 |
| 4.17.1. Data scheme | 83 |
| 4.17.1.1. Table limit_clearing: Status of clients' cash limits after clearing | 84 |
| 4.17.1.2. Table clr_rate: Currency and Index rates | 84 |
| 4.17.1.3. Table fut_pos: Open interest in futures as a result of evening clearing session | 84 |
| 4.17.1.4. Table fut_sess_settl: Futures settlement prices | 85 |
| 4.17.1.5. Table money_clearing_sa: Status of clients' cash accounts after clearing | 85 |
| 4.17.1.6. Table fut_pos_sa: Open interest in futures as a result of evening clearing session | 85 |
| 4.17.1.7. Table sys_events: table of events | 86 |
| 4.18. Stream FORTS_VM_REPL - online variational margin stream (Type=I) | 86 |
| 4.18.1. Data scheme | 86 |
| 4.18.1.1. Table fut_vm: Variation margin for futures | 86 |
| 4.18.1.2. Table fut_vm_sa: Variation margin for futures | 86 |
| 4.19. Stream FORTS_INFO_REPL - additional reference information (Type=R) | 87 |
| 4.19.1. Data scheme | 87 |
| 4.19.1.1. Table base_contracts_params: Base contracts parameters | 87 |
| 4.19.1.2. Table futures_params: Futures parameters | 88 |
| 4.19.1.3. Table investor: Clients directory | 88 |
| 4.19.1.4. Table dealer: Companies directory | 89 |
| 4.19.1.5. Table common_params: Collateral calculation parameters | 90 |
| 4.19.1.6. Table sys_events: Table of events | 90 |
| 4.20. Stream FORTS_TNPENALTY_REPL - information about Transaction fees (Type=I) | 90 |
| 4.20.1. Data scheme | 90 |
| 4.20.1.1. Table fee_all: Information on the number of points accrued | 91 |
| 4.20.1.2. Table fee_tn: Detailed information on the number of incorrect transaction | 91 |
| 4.21. Stream FORTS_FORECASTIM_REPL - Risk forecast after limits extension (Type=I) | 91 |
| 4.21.1. Data scheme | 91 |
| 4.21.1.1. Table part_sa_forecast: Free funds for SA volume forecast | 91 |
| 5. Commands description | 92 |
| 5.1. Method AddOrder - Adding orders | 92 |
| 5.2. Method DelOrder - Deletion of orders | 93 |
| 5.3. Method DelUserOrders - Mass cancel orders | 93 |
| 5.4. Method MoveOrder - Modify orders | 94 |

| | |
|----------------------------------------------------------------------------------------------------------------|-----|
| 5.5. Method IcebergAddOrder - Adding iceberg orders | 95 |
| 5.6. Method IcebergDelOrder - Deletion of iceberg orders | 96 |
| 5.7. Method IcebergMoveOrder - Modify iceberg orders | 97 |
| 5.8. Method ChangeClientMoney - Change client limits | 97 |
| 5.9. Method FutChangeClientProhibit - Modify client's restrictions for futures | 98 |
| 5.10. Method TransferClientPosition - Transfer client positions | 99 |
| 5.11. Method ChangeBFParametersNextSession - Change BF's parameters by a clearing member | 100 |
| 5.12. Method ChangeClientParameters - Change parameters of client account | 100 |
| 5.13. Method ChangeClientParametersNextSession - Change parameters of client account in clearing session | 101 |
| 5.14. Method ChangeBFClientDefaultParametersNextSession - Change default parameters of client sections | 102 |
| 5.15. Method ChangeBFLimit - Change BF trading limits | 102 |
| 5.16. Method CODHeartbeat - Heartbeat message for Cancel on Disconnect Service | 103 |
| 5.17. Method SetSmaPreTradeCheck - Enable pre-trade verification mode for SMA login orders | 103 |
| 5.18. Method DelSmaPreTradeCheck - Disable pre-trade verification mode for SMA login orders | 105 |
| 5.19. Method UserKillSwitch - Disable transactions for trading member login | 105 |
| 5.20. Method SetBrokerFeeParamNextSession - Setting parameters for calculating the brokerage fee | 106 |
| 5.21. Method ChangePassword - Change user password for the Trading System | 107 |
| Plaza-2 data types | 107 |
| List of return codes | 108 |

History of changes

| Date | Changes |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11.05.2022 | <p>Changes applied:</p> <ul style="list-style-type: none"> Added section '2.10. Settlement trades'. In the section '2.4.2. Trade types, created upon exercising and expiration of futures and options (**) a description of new flags is added: <ul style="list-style-type: none"> eDontFineRF (0x8000000000000000) - 'No penalty for settlement transactions'. Updating the utility for changing the password (change_password.exe): <ul style="list-style-type: none"> The 'app_name' parameter (application name) has been added to the command string. The 'local_pass' parameter (password for the local connection to the router) has been added to the command string. The 'key' parameter has been removed from the valid command string parameters. The 'FORTS_FUTTRADE_REPL' stream has been deleted. The 'FORTS_TRADE_REPL' stream should be used instead. The 'FORTS_FUTORDERBOOK_REPL' stream has been deleted. The 'FORTS_USERORDERBOOK_REPL' stream should be used instead. The 'FORTS_FUTCOMMON_REPL' stream has been deleted. The 'FORTS_COMMON_REPL' stream should be used instead. The 'FORTS_FUTINFO_REPL' stream has been deleted. The 'FORTS_REFDATA_REPL' stream should be used instead. Removed streams 'FORTS_FUTAGGR5_REPL', 'FORTS_FUTAGGR20_REPL' and 'FORTS_FUTAGGR50_REPL'. Instead, the 'FORTS_FUTAGGR5_REPL', 'FORTS_FUTAGGR20_REPL' and 'FORTS_FUTAGGR50_REPL' streams should be used. Stream 'FORTS_TRADE_REPL': <ul style="list-style-type: none"> Removed 'local_stamp' field from 'orders_log' table. The 'reason' field has been added to the 'orders_log' table. The 'reason_buy' and 'reason_sell' fields have been added to the 'user_deal' table. Stream 'FORTS_FEE_REPL': <ul style="list-style-type: none"> Added table 'penalty'. Stream FORTS_FEERATE_REPL: <ul style="list-style-type: none"> Table 'futures_rate' now contain 'exp_clearing_fee' field. In version 6.15, the field will always contain "0.0". Stream 'FORTS_USERORDERBOOK_REPL': <ul style="list-style-type: none"> Table 'orders' now contain field 'reason'. Stream 'FORTS_PART_REPL': <ul style="list-style-type: none"> Table 'part' now contain field 'penalty'. Table 'part_sa' now contain field 'blocked_tax'. Added a new stream 'FORTS_PROHIBITION_REPL' - Prohibitions. The 'prohibition' table is broadcast in a separate stream. Stream 'FORTS_REFDATA_REPL': <ul style="list-style-type: none"> Added the tables 'fut_bond_registry', 'fut_bond_nkd', 'fut_bond_nominal' and 'fut_bond_is in'. Table 'prohibition' now contain field 'xprohibition_id'. The 'enforce_ims_half_netting' field was added to the 'fut_sess_contents' and 'fut_instruments' tables. Starting with version 6.15, the 'prohibition' table is deprecated and will be removed in version 7.3. Instead of this table, use the 'prohibition' table of the 'FORTS_PROHIBITION_REPL' stream. |

| Date | Changes |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> • Stream 'FORTS_CLR_REPL': <ul style="list-style-type: none"> ◦ Table 'money_clearing_sa' now contain field 'blocked_tax'. • Stream 'FORTS_INFO_REPL': <ul style="list-style-type: none"> ◦ Table 'futures_params' now contains field 'enforce_ims_half_netting'. • Changes applied to command scheme repository: <ul style="list-style-type: none"> ◦ The 'local_stamp' field has been removed from the 'AddOrder', 'MoveOrder', 'DelOrder', 'DelUserOrders', 'IcebergAddOrder', 'IcebergMoveOrder', and 'IcebergDelOrder' messages. Added new message types: 'AddOrder' (msgid = 465), 'MoveOrder' (msgid = 460), 'DelOrder' (msgid = 461), 'DelUserOrders' (msgid = 466), 'IcebergAddOrder' (msgid = 462), 'IcebergMoveOrder' (msgid = 463) and 'IcebergDelOrder' (msgid = 464). • Added new error codes: 80, 81, 3001, 4280-4282. • Changed texts of error codes: 4017, 4160. • Removed error codes: 4120, 4121, 4168. |
| 07.06.2021 | <p>Changes applied:</p> <ul style="list-style-type: none"> • Stream 'FORTS_REFDATA_REPL': <ul style="list-style-type: none"> ◦ Table 'dealer' now contain field 'order_allowed_in_morning_session'. • Stream 'FORTS_INFO_REPL': <ul style="list-style-type: none"> ◦ Table 'dealer' now contain field 'order_allowed_in_morning_session'. • Added new error codes: 4226. • Changed texts of error codes: 31, 4142. |

1. Introduction

1.1. Document purpose

This document is aimed to overview all the details which users may demand to architect and develop software applications for accessing the derivatives market using the SPECTRA Plaza-2 gate. The following parts are available in this document:

- The SPECTRA system general overview, including overview of trading instruments, trading participants, trading operations, risk management, limiting of operations, etc.
- Configuration, installation and setup of the SPECTRA Plaza-2 gate software in the form of user manuals on software installation and setup with information on minimum hardware and software requirements. Also, some general references on using the SPECTRA Plaza-2 gate software are added.
- Information on the structure of transmitted data, including description of replication streams and transmitted tables.
- List of commands.
- Help information.

1.2. Target audience

This document is intended for business-analysts, system architects and developers, taking part in architecting and developing software for accessing the derivatives market using the SPECTRA Plaza-2 gateway.

1.3. Abbreviations

The following abbreviations may be used in the Document:

| Abbreviation | Description |
|--------------|-----------------------------------|
| BF | Brokerage Firm |
| CC | Clearing Centre |
| CCP | Central Counterparty |
| CF | Clearing Firm |
| COD | Service 'Cancel On Disconnect' |
| EDM | Electronic Document Management |
| MM | Market Maker |
| SMA | Service 'Sponsored Market Access' |
| TS | Trading System |
| VM | Variation Margin |

1.4. Additional Information

The current version of the SPECTRA system does not include the following functionality:

- Delivery on futures.
- Options and multi-leg instruments trading.
- Intraday clearing session.
- Synthetic matching.
- Additional trading sessions (in the evening and in the morning).

Information on these topics is marked with a special sign - (**).

2. SPECTRA system overview

2.1. Trading Membership

Trading Membership may be subdivided to the following:

- Clearing Firms
- Brokerage Firms
- Clients

2.1.1. Clearing firms

Clearing firms are firms which incur liabilities for risks and cover risks of their clients and sub-brokers.

Clearing firms are authorized to:

- Perform trades on behalf of themselves and at for their own accounts;
- Perform trades on behalf of themselves and for their clients' accounts;
- Perform settlement directly with Central Counterparty.
- Service their clients, including brokers;
- Exercise control over their clients and brokers during trading sessions.

Clearing firms are obliged to:

- Become members of Derivatives Market Section;
- Provide collaterals for their own trades and for their clients' trades.

2.1.2. Brokerage firms

Unlike clearing firms, brokerage firms do not settle up with exchange directly; instead, they use their clearing firms. Also, brokerage firms are not obliged to obtain licences.

Brokerage firms are authorized to:

- Perform trades on behalf of themselves;
- Perform trades on behalf of their clients;
- Place orders in the Trading system via the client terminal application
- Exercise control over their clients during trading sessions.

Brokerage firms are obliged to:

- Provide guarantees for their own trades and for their clients' trades.

2.1.3. Clients

Any physical or corporate person can participate in the derivatives market as a client on the authority of trading service agreement signed with a brokerage firm or with clearing firm directly.

2.1.4. System code pattern

There is a 7-symbol code pattern (XXYYZZZ) to identify each participant in the system, where

- XX — indicates a clearing firm
- YY — indicates a brokerage firm
- ZZZ — indicates a client

The 00 brokerage firm code indicates state of account of the clearing firm.

Example 1.

Q100 — indicates the Q1 clearing firm

Q1DU — indicates the DU sub-broker of the Q1 clearing firm

The 000 client code indicates state of account of the brokerage firm.

Example 2.

Q1DU000 — indicates state of account of the DU sub-broker of the Q1 clearing firm

2.1.5. Disclosure of data on participants

The list of clearing and brokerage firms is stored in the 'dealer' table of the 'FORTS_REFDATA_REPL' stream, and the list of clients is stored in the 'investor' table of the 'FORTS_REFDATA_REPL' stream. Disclosure of data on brokerage firms and clients is limited in accordance with user access rights.

Streams and tables also contain links to 7-symbol clients' codes and 4-symbol brokerage firms' codes.

2.1.6. Users. How a user is linked to a trading participant

A user (login) can be associated with various levels of participants:

- Clearing firm login. Users connected with this login are allowed to view data and perform trading operations on behalf of any brokerage firm or of any client of the clearing firm (please note that performing trading actions is only allowed when the user has sufficient rights!). Users also allowed to set limits for clients and sub-brokers by calling the appropriate operations.
- Brokerage firm login. Users connected with this login are allowed to view data and perform trading operations on behalf of all broker's clients within the clearing firm, and also set limits for the broker's clients.
- Client login. Users connected with this login are allowed to perform trading operations on behalf of a certain client of a brokerage firm and view data in accordance with the client login rights.

There is a special 4-symbol 'broker_code' field within the scheme of message-command (see Commands description). Every application using the clearing firm account is to fill in this field with a 4-symbol code of a brokerage company registered with SPECTRA when sending messages. Applications which use the client or the brokerage firm account are exempt from this rule.

2.2. Instruments

The SPECTRA instruments are structured hierarchically. Below you will find descriptions of the SPECTRA instruments starting from the root level.

2.2.1. Underlying assets

An underlying asset is an entity related to a certain contract. Therefore, it can be a stock in a stock exchange, a lot of tradable commodity in a commodities exchange or an index/exchange rate/indicator for settling futures. There are certain attributes characterising an underlying asset along with its instruments, which are:

- Trade section name;
- Various commission fees rates and signs of scalping when fees are calculated. If an asset shows a sign of scalping, the commission fee will be only levied on opening trades.
- Delivery type according to the contract (for details, see Delivery of assets and expiration of options):
 - Delivery of the asset itself;
 - Settlement type. The margin between the opening price and the closing price is the single amount of money to be paid after the trade is closed.

An underlying asset IS IN NO WAY A TRADING INSTRUMENT!

Data concerning underlying assets are contained in the 'fut_vcb' table of the 'FORTS_REFDATA_REPL' stream.

2.2.2. Futures

Futures contracts are the main trading instruments in the SPECTRA system.

Each futures contract is linked to a certain underlying asset and has its own unique characteristics of the maturity (the date of delivery), lot characteristics, minimum price step and cost of the price step value.

There can be more than one futures contract for each underlying asset.

Futures contract with various dates of delivery may form a calendar spread. In this case, when risks are calculated, the price correlation is always taken into account. As a result, the total collateral for the spread can be less than sum of collaterals for each futures contract itself.

Futures are quoted in price points. The price in tenge for a contract is calculated as following:

$$\text{Price} = \text{PricePoints} * \frac{\text{step_price}}{\text{min_step}}$$

, where:

- PricePoints — indicates price in points;
- step_price — indicates cost of minimum price step
- min_step — indicates minimum price step in points.

Two more fields are required to fill when it comes about future contracts quoted in foreign currency:

- Cost of price step in initial currency;
- Cost of price step in tenge, which is fixed upon the clearing session opening.

Futures contract data are stored in following tables of trade interface:

- 'FORTS_REFDATA_REPL' stream, 'fut_sess_contents' table. This is the main table, which contains a list of futures contract available on the current trade session;
- 'FORTS_REFDATA_REPL' stream, 'fut_instruments' table. The table contains limited data amount about all future contracts put into the system, including non-tradable contracts.
- 'FORTS_INFO_REPL' stream, 'futures_params' table. This table contains data about futures contracts. According to the data format the table can be loaded by the SpectralM client application for calculating risks.

2.2.3. Identification of instruments

The SPECTRA system has four fields to identify each instrument:

1. 'isin_id' field, which contains the unique numeral code for each instrument.
2. 'isin' field, which contains the instrument's symbol code.
3. 'short_isin' field, which contains short symbol code for using in order books etc.
4. 'name' field, which contains a long 'humanized' instrument's description.

A value in the 'isin_id' field is the primary unique instrument's code, which is used throughout of data structure of the system wherever a corresponded reference exists.

The 'isin' field contains the main symbol futures' code, which is used in order's instructions. The uniqueness and invariability in time of the 'isin' value is guaranteed.

The 'short_isin' field is an alternative symbol contract code. It has been implemented in order to ease access to the SPECTRA system data for news agencies.

2.3. Trading operations

2.3.1. Orders — general information

Order — a command, which is sent into the trading system by a trading participant, aimed to perform an action of buying or selling an instrument at specified price. There are two main types of orders available: negotiated and system.

System orders — a common type of orders available for all users of the system. System orders have to participate in auction along with offsetting orders. If there is an offsetting order available for any system order at a better or equal price, the order itself has to be exercised at the price equal to that of the offsetting order. The unexercised part of the order remains in the system as an order with less amount of instruments.

Orders can be subdivided into three types: quoted, offsetting, and fill-or-kill orders. A quoted order remains in queue after it has been fully or partly exercised. An offsetting orders have to be removed from the system after auction ended, no matter whether it has been exercised fully or partly. At last, the fill-or-kill orders — the offsetting orders which can only be exercised fully.

All orders can be also subdivided into common and multi-day orders, in accordance with their lifetime. Common orders do not have the date of expiration specified; such orders remain in queue until the end of the current trading session. Contrary, the expiration date for multi-day orders is specified, ranged from 1 day up to one year. Such orders are relisted automatically at opening of the next session; additionally each order receives a new ID and a link to the initial order's ID. When relisting, the orders are checked for having sufficient instrument, client and funds. Orders which are out-of-date are automatically removed after the trading session ends.

There are two additional fields added to meet the developers' needs:

- 'comment field' - a 20-symbol string;
- 'ext_id field' - a 4-byte number to store order's ID in the client application.

Note: The SPECTRA system does not check values of the additional fields for being unique.

Orders data are stored in the 'orders_log' tables of the 'FORTS_TRADE_REPL' and 'FORTS_ORDLOG_REPL' streams. The tables contain orders changing log, where every change is recorded as a separate record in the table. The table 'orders_log' of the stream 'FORTS_TRADE_REPL' contains information on the 'own' orders only. The 'own' orders are:

- For a client login - records about all orders, placed on behalf of this client;
- For brokerage firm or clearing firm login - records about all orders placed on behalf of clients of these firms.

Users can view all data on the 'own' orders, including data in service fields and user fields.

Clients are able to be subscribed for receiving the table 'orders_log' of the stream 'FORTS_ORDLOG_REPL'; in this case, they will receive complete history of changes for all orders in the trading system in anonymous mode.

Users can do the following:

- Add an order;
- Delete an order according to its code in the SPECTRA system;
- Move an order (the 'MoveOrder' command). Moving of an order is implemented in two steps: deleting an 'old' order and adding a new one into the system (with a new code, which is sent to user after the order was added). Thus, at least two records (about deleting an order and adding a new one) will be added in the 'orders_log' table. You can move two orders at time by adding parameters ('order_id1', 'order_id2') to the 'MoveOrder' command, which can be useful for market-makers' needs. If you move only one order, then you should specify the 'order_id1' parameter only.
- Delete orders by mask. The following masks can be applied:
 - Direction of operation: buying or selling;
 - Order type: negotiated order or system order;
 - Client's code;
 - Underlying asset's code;
 - Order's ID in the client system ('ext_id');
 - Instrument's code;
 - Instrument's group.

2.3.2. Negotiated orders (Negotiated trades mode)

An order addressed to a certain client are called negotiated order. Unlike system orders, negotiated orders have some limitations for users in managing orders and selecting counterparts, namely the following:

- Negotiated orders may be added only by a BF's login, with the Brokerage Firm as the only allowed counterparty.
- For specifying a counterpart, the counterpart's RTS code is used in orders in 'broker_to' field. The brokerage firms which do not have the RTS code act as counterparts for negotiated orders.
- Instead of moving, negotiated orders can only be deleted and listed anew manually.
- Negotiated orders can only be exercised when the price value, and value of field 'match_ref' of one order exactly matches that of the counterpart order. Negotiated orders can also be exercised partly.

2.3.3. Trades

Within SPECTRA trading system, a trade will be performed if an instrument price in one order meets the instrument price in an opposite order, i.e. sell or buy one for the same instrument. The price of the order settled first is the price of the trade. There are two types of trades: negotiated and system. Many trade's attributes are equivalent of that of the orders. Trades cannot be edited, or deleted from the system.

Data on own trades are stored in table 'user_deal' of stream 'FORTS_TRADE_REPL'. The data on all trades in the system are distributed in accordance with the following rules: a user gets access only to their own part of the trade (buyer's or seller's). If a user acts as a brokerage firm, or a clearing firm, and both buyer and seller parts are the clients of the same firm, the user gets access to the data concerning both parts of the trade. Data on all trades are available for all users in table 'deal' of stream 'FORTS_DEALS_REPL'. All data in tables are anonymised.

Along with the records containing the trades-related data, there are some additional records stored in tables containing trades data. These records cannot be classified as trades legally, but still they render some transactions within the system, which influence the participant's status. These trades are so called 'technical trades'. One can tell trading trades from the technical ones by values in fields 'xstatus_sell' and 'xstatus_buy' in table 'user_deal' of stream 'FORTS_TRADE_REPL', or by the flag nosystem in table 'deal' of stream 'FORTS_DEALS_REPL' (for details see Trade types, created upon exercising and expiration of futures and options).

2.3.4. Iceberg orders

An Iceberg order is a variation of a quote order. It allows you to hide a part of its volume from the market (that is, in the Order-book window) to minimize the impact on the large orders market price. Iceberg orders appear in the order-book in portions. The next portion "pops up" only after the visible part of the order will be executed. This process can be repeated until the whole hidden part is used.

The iceberg orders main features:

- Iceberg order can be on-exchange only. In terms of lifetime, iceberg orders can be ordinary and multi-day.

- When adding an iceberg order, it additionally indicates the parameters for calculating the size of the pop-up (visible) part. The pop-up part consists of a constant part ('disclose_const_amount') and a randomly calculated addition. The addition value is a random variable with a uniform distribution from the range $[-\text{Round}(\text{disclose_const_amount} * \text{variance_amount}/100, 0); \text{Round}(\text{disclose_const_amount} * \text{variance_amount}/100, 0)]$, where 'variance_amount' - variance amplitude from the constant part volume. Accordingly, when adding an iceberg order, two parameters are indicated in it:
 - 'disclose_const_amount' - volume of the constant popup part. This parameter cannot be larger than the entire iceberg. A disclose volume cannot be less than the minimum lot for this instrument (values are published on the exchange website).
 - 'variance_amount' - the value of the random deviation of the visible iceberg part (optional). The parameter value can be from zero to the value published on the exchange website. By default, the parameter is not set.

All specified parameters can take only integer positive values.

- Collateral is blocked for the full volume of the iceberg, not for the visible part only.
- When changing an iceberg order, only the price can be changed, the volume is not available for change.
- When deleting or changing an iceberg order, the entire iceberg is deleted or changed, including the visible part.
- In the tables of your orders and trades, iceberg orders and trades are marked with a special attribute in the fields
- In the tables of your orders and trades, iceberg orders and trades are marked with a special sign "eiceberg" (0x800000000000) in the 'xstatus' and 'xstatus_sell / xstatus_buy' fields.

2.3.4.1. Iceberg orders in the system information streams

An iceberg order consists of two parts: public - this is the visible part of the iceberg order, and private - the entire iceberg order, including the visible part. Accordingly, there are two sets of fields in the tables of your orders and trades (order ID, quantity in operation, rest, action, etc.):

1. Public data are broadcast in the fields with the prefix "public_":

- Table 'orders_log':
 - public_order_id - ID of the visible part of the iceberg order.
 - public_amount - The number of contracts in the operation for the visible part of the iceberg order.
 - public_amount_rest - The remaining number of contracts in the visible part of the iceberg order.
 - public_action - Type of operation with the visible part of the iceberg order.
- Table 'user_deal':
 - public_order_id_buy - ID of the visible part of the buyer's iceberg order.
 - public_order_id_sell - ID of the visible part of the seller's iceberg order.

2. Private data are broadcast in the fields with the prefix "private_":

- Table 'orders_log':
 - private_order_id - ID of the entire iceberg order.
 - private_amount - The number of contracts in the operation for the entire iceberg order.
 - private_amount_rest - The remaining number of contracts in the entire iceberg order.
 - private_action - Type of operation with the entire iceberg order.
- Table 'user_deal':
 - private_order_id_buy - ID of the entire buyer's iceberg order.
 - private_order_id_sell - ID of the entire seller's iceberg order.

Please note that for iceberg orders, data on the entire iceberg order are broadcast in the existing fields: 'id_ord', 'xamount', 'xamount_rest', 'action', 'id_ord_buy' and 'id_ord_sell', of the tables of your orders and trades.

Below is an example of entry in the stream for adding and matching of iceberg order with amount=1000 and visible amount=100 (without filtering):

| public_order_id | public_amount | public_amount_rest | public_action | price | moment | dir | client_code | private_order_id | private_amount | private_amount_rest | private_action | comment |
|-----------------|---------------|--------------------|---------------|-------|---------------------|-----|-------------|------------------|----------------|---------------------|----------------|-------------|
| 101 | 100 | 100 | 1 | 312 | 2019-01-11 11:55:58 | 1 | OD01123 | 101 | 1000 | 1000 | 1 | Add Iceberg |

| public_order_id | public_amount | public_amount_rest | public_action | price | moment | dir | client_code | private_order_id | private_amount | private_amount_rest | private_action | comment |
|-----------------|---------------|--------------------|---------------|-------|---------------------|-----|-------------|------------------|----------------|---------------------|----------------|----------------------|
| 102 | 1 | 1 | 1 | 312 | 2019-01-11 14:56:58 | 1 | PJ99888 | 102 | 1 | 1 | 1 | Add standard Order |
| 103 | 250 | 250 | 1 | 310 | 2019-01-11 16:58:58 | 2 | FS01020 | 103 | 250 | 250 | 1 | Add standard Order |
| 101 | 100 | 0 | 2 | 312 | 2019-01-11 16:58:58 | 1 | OD01123 | 101 | 100 | 900 | 2 | Match Iceberg |
| 103 | 100 | 150 | 2 | 310 | 2019-01-11 16:58:58 | 2 | FS01020 | 103 | 100 | 150 | 2 | Match standard Order |
| 102 | 1 | 0 | 2 | 312 | 2019-01-11 16:58:58 | 1 | PJ99888 | 102 | 1 | 0 | 2 | Match standard Order |
| 103 | 1 | 149 | 2 | 310 | 2019-01-11 16:58:58 | 2 | FS01020 | 103 | 1 | 149 | 2 | Match standard Order |
| 104 | 100 | 100 | 1 | 312 | 2019-01-11 16:58:58 | 1 | OD01123 | 101 | 100 | 900 | 3 | Pop-up Iceberg |
| 104 | 100 | 0 | 2 | 312 | 2019-01-11 16:58:58 | 1 | OD01123 | 101 | 100 | 800 | 2 | Match Iceberg |
| 103 | 100 | 49 | 2 | 310 | 2019-01-11 16:58:58 | 2 | FS01020 | 103 | 100 | 49 | 2 | Match standard Order |
| 105 | 100 | 100 | 1 | 312 | 2019-01-11 16:58:58 | 1 | OD01123 | 101 | 100 | 800 | 3 | Pop-up Iceberg |
| 105 | 49 | 51 | 2 | 312 | 2019-01-11 16:58:58 | 1 | OD01123 | 101 | 49 | 751 | 2 | Match Iceberg |
| 103 | 49 | 0 | 2 | 310 | 2019-01-11 16:58:58 | 2 | FS01020 | 103 | 49 | 0 | 2 | Match standard Order |
| 105 | 51 | 0 | 0 | 312 | 2019-01-11 17:00:58 | 1 | OD01123 | 101 | 751 | 0 | 0 | Cancel Iceberg |

Explanations for the table:

- Client OD01123 adds iceberg order with entire amount 1000 and visible part amount 100. New order with 'private_order_id'=101 (order ID), 'private_amount'=1000 (entire iceberg amount) and 'public_amount'=100 (visible part) is added to the system ('private_action'=1).
- Clients PJ99888 and FS01020 consistently add their standard orders to the system. Moreover, the order of client FS01020 is an opposite order that satisfies the price of two previous orders.
- Visible part of iceberg order and opposite order of client FS01020 are matched ('private_action'=2), size of the remaining iceberg 'private_amount_rest'=900.
- Then the standard orders of clients PJ99888 and FS01020 are matched.
- The next portion of the iceberg order pops up ('private_action'=3), which immediately is matched ('private_action'=2) with the remaining part of the order of client FS01020, size of the remaining iceberg 'private_amount_rest'=800.
- The next portion of the iceberg order pops up ('private_action'=3) and is matched with the remaining part of the order of client FS01020, size of the remaining iceberg 'private_amount_rest'=751.
- Then client OD01123 canceled iceberg order.
- Please note that when the next portion of the iceberg order pops up, it's visible part has number ('public_order_id') different from the identifier of the iceberg order itself ('private_order_id')!

For standard orders, private and public fields are filled with the same values and contain the usual values for ID, quantity in operation, remaining quantity and order operation code. Explanations for the filling of the fields from the example above:

| public_order_id | public_amount | public_amount_rest | public_action | id_ord | xamount | xamount_rest | action | private_order_id | private_amount | private_amount_rest | private_action | comment |
|-----------------|---------------|--------------------|---------------|--------|---------|--------------|--------|------------------|----------------|---------------------|----------------|---------------|
| 101 | 100 | 100 | 1 | 101 | 1000 | 1000 | 1 | 101 | 1000 | 1000 | 1 | Add Iceberg |
| 102 | 1 | 1 | 1 | 102 | 1 | 1 | 1 | 102 | 1 | 1 | 1 | Add Order |
| 103 | 250 | 250 | 1 | 103 | 250 | 250 | 1 | 103 | 250 | 250 | 1 | Add Order |
| 101 | 100 | 0 | 2 | 101 | 100 | 900 | 2 | 101 | 100 | 900 | 2 | Match Iceberg |
| 103 | 100 | 150 | 2 | 103 | 100 | 150 | 2 | 103 | 100 | 150 | 2 | Match Order |

| public_order_id | public_amount | public_amount_rest | public_action | id_ord | xamount | xamount_rest | action | private_order_id | private_amount | private_amount_rest | private_action | comment |
|-----------------|---------------|--------------------|---------------|--------|---------|--------------|--------|------------------|----------------|---------------------|----------------|----------------|
| 102 | 1 | 0 | 2 | 102 | 1 | 0 | 2 | 102 | 1 | 0 | 2 | Match Order |
| 103 | 1 | 149 | 2 | 103 | 1 | 149 | 2 | 103 | 1 | 149 | 2 | Match Order |
| 104 | 100 | 100 | 1 | 101 | 100 | 900 | 3 | 101 | 100 | 900 | 3 | Pop-up Iceberg |
| 104 | 100 | 0 | 2 | 101 | 100 | 800 | 2 | 101 | 100 | 800 | 2 | Match Iceberg |
| 103 | 100 | 49 | 2 | 103 | 100 | 49 | 2 | 103 | 100 | 49 | 2 | Match Order |
| 105 | 100 | 100 | 1 | 101 | 100 | 800 | 3 | 101 | 100 | 800 | 3 | Pop-up Iceberg |
| 105 | 49 | 51 | 2 | 101 | 49 | 751 | 2 | 101 | 49 | 751 | 2 | Match Iceberg |
| 103 | 49 | 0 | 2 | 103 | 49 | 0 | 2 | 103 | 49 | 0 | 2 | Match Order |
| 105 | 51 | 0 | 0 | 101 | 751 | 0 | 0 | 101 | 751 | 0 | 0 | Cancel Iceberg |

Anonymous streams of orders and trades contain only public fields, in which there is always the visible part of icebergs only.

Please note that the old fields 'id_ord', 'xamount', 'xamount_rest', 'action', 'id_ord_buy' and 'id_ord_sell' are left in the streams of your orders and trades to support backward compatibility, and will be deleted after two releases.

2.3.4.2. Iceberg order operations

The following operations are possible for iceberg orders.

- Add order (command IcebergAddOrder).
- Delete order (command IcebergDelOrder). The command can work both on 'public_order_id' and on 'private_order_id'.
- Move order (command IcebergMoveOrder). The command can work both on 'public_order_id' and on 'private_order_id'.

Please note, that the 'IcebergMoveOrder' and 'IcebergDelOrder' commands will work on 'public_order_id' 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 'private_order_id'.

2.3.4.3. Change of order ID during iceberg orders operations

When an iceberg order is added, its ID of the visible part ('public_order_id') is the same as ID of the entire iceberg order ('private_order_id'). When a new part pops up, a new ID ('public_order_id') is assigned to it, the ID of the entire iceberg order does not change. When an iceberg order is changed (move), a new 'private_order_id' is set for it.

When multi-day (GTD) iceberg orders are replaced in the evening clearing, a new iceberg order with a new 'private_order_id' is set. This new order has 'the private_order_id' of the first iceberg order as the initial order (field 'id_ord1').

An example of changing order IDs during an iceberg order operations:

| Operation | public_order_id | public_action | private_order_id | private_action | id_ord1 |
|---------------------------------|-----------------|---------------|------------------|----------------|---------|
| Add | 100 | 1 | 100 | 1 | |
| Matching into a trade | 100 | 2 | 100 | 2 | |
| New part pops up | 105 | 1 | 100 | 3 | |
| Replace in the evening clearing | 1106 | 1 | 1106 | 1 | 100 |
| Move | 1106 | 0 | 1106 | 0 | 100 |
| | 1200 | 1 | 1200 | 1 | |

Explanations for the table:

- **Add** - iceberg order is added ('private_action'=1) with ID 'private_order_id'=100 and ID of visible part 'public_order_id'=100.
- **Matching into a trade** - matching the visible part of the iceberg order ('private_action'=2) with the counter direction order.
- **New part pops up** - when a new part pops up ('private_action'=3), a new ID 'public_order_id'=105 is assigned to it, the ID of the entire iceberg order does not change.
- **Replace in the evening clearing** - a new iceberg order ('private_action'=1) with a new 'private_order_id'=1106 is set in the evening clearing. This new order has 'the private_order_id' of the first iceberg order as the initial order (field 'id_ord1'=100).
- **Move** - old iceberg order is deleted (private_action=0) and new order with new 'private_order_id'=1200 is added (private_action=1).

Values of 'public_action':

- 0 - Order cancelled

- 1 - Order added
- 2 - Order exercised in a trade

Values of 'private_action':

- 0 - Order cancelled
- 1 - Order added
- 2 - Order exercised in a trade
- 3 - New visible part pops up

2.4. Delivery of assets and exercise of options (**)

2.4.1. Deliveries on futures (**)

There are three types of futures exist in terms of deliveries:

- Non-deliverable futures: upon expiration, only the amount of difference between the contract price and the current price of the asset will be delivered. The delivery is performed as technical closing of the position, and is marked with a special sign in the 'xstatus_sell' and 'xstatus_buy' fields of the appropriate table containing trades data (for details see Trade types, created upon exercising and expiration of futures and options).
- Commodity futures: upon expiration, the assets and money are delivered. The delivery is performed as technical closing of the position, and is marked with a special sign in the 'xstatus_sell' and 'xstatus_buy' fields of the appropriate table containing trades data.
- Stock futures: upon settlement, the position for futures turns into a position on the T+ market (Main Market). The settlement is processed as a technical position closing trade on the derivatives market (the trade is marked with special flag in the 'xstatus_sell' and 'xstatus_buy' fields of the appropriate table containing trades data), and position opening trade on T+ market (added into the ASTS system of derivatives market).

2.4.2. Trade types, created upon exercising and expiration of futures and options (**)

Flags applied to orders and trades:

| Flag name | Bit mask | Description |
|---------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Market orders | | |
| eAuctionStatus | 0x1 | Day order. |
| eOppositeStatus | 0x2 | Immediate-or-Cancel order (IOC). |
| eFOKStatus | 0x80000 | Fill-or-Kill order. |
| Clearing trades | | |
| eNonQuoteStatus | 0x4 | Applied to negotiated orders/trades, technical trades, clearing trades and multi-leg trades. |
| eExecStatus | 0x20 | A trade resulting from the exercise of an option. The flag is set for option trades and for futures trades that appeared as a result of the exercise of options. |
| eExpirationStatus | 0x80 | Instrument, futures, or option expiration. |
| eDUFlowStatus | 0x800 | Technical trade of position transfer between trust managing BFs of Clearing Firms. |
| eOptionLapse | 0x800000 | Option expiration trade |
| eClearingTrade | 0x2000000 | Off-book clearing trade. Applied to all clearing trades. |
| eFuturesExecution | 0x40000000 | Futures exercise trade. |
| Negotiated orders/trades | | |
| eTransferClientPosition | 0x8 | Position transfer between BFs. |
| eAddressStatus | 0x4000000 | Negotiated order, negotiated trade. |
| eTransferSource | 0x200000000 | Donor BF side in position transfer between BFs. |
| Multi-leg orders/trades | | |
| eREPOBackStatus | 0x4000 | Far leg transaction. |
| eStrategy | 0x8000000 | Multi-leg order/trade. Applied to all multi-leg transactions. |

| Flag name | Bit mask | Description |
|-----------------------------------------|-----------------|---------------------------------------------------------------------------------------------|
| Other | | |
| eDontCheckMoney | 0x10 | Do not verify collateral for client accounts. |
| eDontCheckLimits | 0x200 | Do not verify limits for options. |
| eLastRecStatus | 0x1000 | End Of Transaction bit. |
| eOppositeOrderTailDeleteDueToCrossTrade | 0x20000000 | Order cancellation due to a cross-trade. |
| eCODBulkDeleteOperationStatus | 0x100000000 | Order cancellation due to COD. |
| eUKSBulkDeleteOperationStatus | 0x2000000000 | Orders are cancelled by UserKillSwitch |
| eLiqNettingRF | 0x10000000000 | Orders and trades formed in the process of liquidation netting |
| eActiveSide | 0x20000000000 | The active side in the trade. The order that led to the trade when added to the order-book. |
| ePassiveSide | 0x40000000000 | The passive side in the trade. The order from the order-book involved in the trade. |
| elceberg | 0x800000000000 | Iceberg order/trade. |
| eOperatorInputSA | 0x1000000000000 | A sign of an order or trade formed in SA blocking mode. |
| eDontFineRF | 0x8000000000000 | No penalty for settlement transactions. |

Trade types, created upon execution and expiration of futures and options are listed in the table below:

| Operation type | Position closing trade | Position opening trade | Date and time of trades availability in the gateway |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Exercise of deliverable futures | <ul style="list-style-type: none"> id in reports is 0, id in gateways is nonzero. Trade price is rounded to minimal step price. Technical trade is not a trade legally. Signs in gateways and reports (bitmask): eNonQuoteStatus, eExpirationStatus, eFuture-sExecution, eClearingTrade. | No. | On the execution day, in the morning. |
| Exercise of cash-settled futures | <ul style="list-style-type: none"> id in reports is 0, id in gateways is nonzero. Trade price is rounded to the minimal price step. Technical trade is not a trade legally. Signs in gateways and reports (bitmask): eNonQuoteStatus, eExpirationStatus, eFuture-sExecution, eClearingTrade. | No. | On the execution day, in the evening. |
| Exercise of option | <ul style="list-style-type: none"> id in gateways is nonzero. id in reports is 0 (trade at the evening clearing session), nonzero id (trade at the intermediate clearing session). Trade price is 0. Technical trade is not a trade legally. Signs in gateways and reports (bitmask): eNonQuoteStatus, eExecStatus, eClearingTrade. | <ul style="list-style-type: none"> id in gateways is nonzero. id in reports is 0. Trade price is rounded to 5 places. This trade is a trade legally. Signs in gateways and reports (bitmask): eNonQuoteStatus, eExecStatus, eClearingTrade. | <ul style="list-style-type: none"> At the intermediate clearing session At the evening clearing session <p>Depending on time of applying the option (the next clearing session after applying).</p> |
| Expiration of option | <ul style="list-style-type: none"> id in gateways is nonzero. id in reports is 0. | No. | On the futures execution day, in the evening. |

| Operation type | Position closing trade | Position opening trade | Date and time of trades availability in the gateway |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| | <ul style="list-style-type: none"> Trade price is 0. Technical trade is not a trade legally. Signs in gateways and reports (bitmask): eNonQuoteStatus, eExpirationStatus, eClearingTrade, eOptionLapse. | | |
| Segregated Brokerage Firm position transfer | <ul style="list-style-type: none"> id in gateways is nonzero. Trade price is 0. Technical trade is not a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus, eTransferClientPosition, eClearingTrade. | <ul style="list-style-type: none"> id in gateways is nonzero. Trade price is 0. Technical trade is not a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus, eTransferClientPosition, eTransferSource, eClearingTrade. | During the evening clearing session. |

Trades are shown as following:

| Operation type | Operations info |
|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Stock futures trade based on a negotiated order | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to the minimal price step. This trade is a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus. |
| Stock futures trade based on a system order | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to the minimal price step. This trade is a trade legally. Signs in gateways (bitmask):bits value is 0. |
| Stock futures option trade based on a negotiated order | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to the minimal price step. This trade is a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus. |
| Stock futures option trade based on a system order | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to the minimal price step. This trade is a trade legally. Signs in gateways (bitmask):bits value is 0. |
| Position transfer trade | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to 5 places. This trade is not a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus, eTransferClientPosition, eTransferSource. |
| Technical trade based on the negotiated multi-leg order (near leg) | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to 5 places. This trade is a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus, eStrategy. |
| Technical trade based on the negotiated multi-leg order (far leg) | <ul style="list-style-type: none"> id in gateways is unique and nonzero. |

| Operation type | Operations info |
|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> Trade price is rounded to 5 places. This trade is a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eAddressStatus, eStrategy, eREPOBackStatus. |
| Technical trade based on the system multi-leg order (near leg) | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to 5 places. This trade is a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eStrategy. |
| Technical trade based on the system multi-leg order (far leg) | <ul style="list-style-type: none"> id in gateways is unique and nonzero. Trade price is rounded to 5 places. This trade is a trade legally. Signs in gateways (bitmask): eNonQuoteStatus, eStrategy, eREPOBackStatus. |

2.5. Trading and clearing schedule

2.5.1. Trading schedule. Trading sessions.

Trading in the SPECTRA system is carried out within a trading session. During a trading session, the same trading instruments are traded and the same parameters are used to calculate the collateral to pledge. There are very important operations taking place in the SPECTRA system between the two sessions: clearing, contracts expirations and many others.

2.5.2. Clearing session

The clearing session is taking place in the end of the trading session. The following operations are performed:

- Calculation and fixation of settling prices in accordance with the trading session results.
- Calculation and transferring of varying margins between participants.
- Deletion of expired instruments and adding new ones.
- Renewing information on clients, brokerage and clearing firms by deleting obsolete data and loading newly calculated data.

2.5.3. How different entities act on assigning a new trading session

2.5.3.1. Reference data and session data

When a new trading session is assigned, the data in the tables linked to the session number are loaded anew. For the tables that are not linked to the session number, new records are added in accordance with the new data available in the trading session; the records which do not correspond the actual trading session data will be deleted. The reference data are sent out within the tables of the 'FORTS_REFDATA_REPL' stream. As a result, the new record with a new session number is added into the 'session' table.

2.5.3.2. Funds and positions

When a trade session changes, the data on funds, limitations and clients positions are updated as following: only the records which have been modified are subject to change (including the 'FORTS_PART_REPL', 'FORTS_POS_REPL' and 'FORTS_INFO_REPL' streams).

2.5.3.3. Orders and trades

The main trading data (the FORTS_TRADE_REPL, 'FORTS_ORDLOG_REPL' and 'FORTS_DEALS_REPL' streams) is saved i.e. the orders and trades which were made of the current trading session are available in the system till 12:00 PM on the current day.

Upon changing the trading session, the multi-day orders are relisted automatically except those which are expired. The relisting is made by deleting an old order and adding a new one with a new number, with no data added into the 'orders_log' table. Therefore, the client system should act as following: after finding a new trading session number in the 'session' table, the client system should 'forget' all the orders stored in memory by the moment and 'listen' to the replication stream for new orders with the new trading session number.

2.5.3.4. Instruments

When switching the trading sessions, the system deletes expired trading instruments and adds new.

2.5.3.5. Replication streams

The replication streams can be closed and then reopen again by the trading system servers, yet some streams may transmit notification about changing the life number of a scheme.

For now, the following streams can be reopen without changing life numbers:

- 'FORTS_COMMON_REPL' — general market data.
- 'FORTS_VM_REPL' — the current varying margin value

The following streams are not subjects to reopen:

- 'FORTS_REFDATA_REPL' — reference data
- 'FORTS_TRADE_REPL' — trading data
- 'FORTS_USERORDERBOOK_REPL' — snapshots of order books
- Streams with aggregated order books.
- 'FORTS_PART_REPL', 'FORTS_POS_REPL', 'FORTS_INFO_REPL'
- 'FORTS_CLR_REPL'.

2.5.4. Event-sensitive scheme for data synchronizing

If a developed system demands the possibility of synchronizing the consistent states of data, then the event-sensitive scheme should be used. The following events are used to start synchronization:

- All data for a new trading session are loaded and calculated
- Main clearing session has started
- All data after the main clearing session are recalculated
- Limits have been extended

The new 'sys_events' table is added to the replication streams in order to inform outer systems about the events occurred:

| Field | Type | Description |
|------------|------|-------------------------------------|
| replID | i8 | Replication subsystem service field |
| replRev | i8 | Replication subsystem service field |
| replAct | i8 | Replication subsystem service field |
| event_id | i8 | Unique event ID |
| sess_id | i4 | Trading session ID |
| event_type | i4 | Event type |
| message | c64 | Text description |

The table is added into the following replication streams:

- 'FORTS_TRADE_REPL'
- 'FORTS_INFO_REPL'
- 'FORTS_PART_REPL'
- 'FORTS_POS_REPL'
- 'FORTS_REFDATA_REPL'
- 'FORTS_ORDLOG_REPL'
- 'FORTS_CLR_REPL'
- 'FORTS_DEALS_REPL'
- 'FORTS_FEE_REPL'
- 'FORTS_FEERATE_REPL'
- 'FORTS_BROKER_FEE_REPL'
- 'FORTS_BROKER_FEE_PARAMS_REPL'

The rules of the synchronization are the following: when a global event occurs in the system, and when all the data regarding this event are generated by all the subsystems, the new record is added to the 'sys_event' table containing the same 'event_id' value, with the 'event_type' value corresponding to the following event occurred:

- 1 (session_data_ready) - all data from the clearing system have been loaded into the trading system; this type of event is transmitted in all streams containing sys_events, except the 'FORTS_CLR_REPL' stream

- 3 (clearing_data_ready) - data are ready after the main clearing session; this type of event is transmitted only in the 'FORTS_CLR_REPL' stream
- 5 (clearing_started) - main clearing session has started; this type of event is transmitted in all streams containing sys_events, except the 'FORTS_CLR_REPL' stream
- 6 (extension_of_limits_finished) - limits have been extended; this type of event is transmitted in all streams containing sys_events, except the 'FORTS_CLR_REPL' stream

An outer system may subscribe to receive the event table via all the available replication streams; when the data are ready, a notification will be sent to the outer system. The 'sys_event' table records, relating to the same event, will have the same 'event_id' field value in every replication stream. There are additional data available in the 'sess_id' and 'message' fields: the number of the current or upcoming trading session and a text message, respectively. Please also note that:

- The identity of service fields values (the 'replID' and 'replRev' fields) cannot be guaranteed for the same event in the different replication streams. You should view the 'event_id' value instead.
- The notification for the 'sys_event' table arrives AFTER all other data. It means that working in on-line mode, the system receives the newest data available, for example, instruments or the multi-day orders rolled over from the previous session, before adding records into the 'sys_events' table.
- Data consistency is not guaranteed in snapshot mode. Replication protocol does not remember the order of retrieving records between different tables in a stream and data are distributed in the order of description of tables in the schema, that is why snapshot recordings will not arrive in the order in which they arrived in on-line mode. For example, the 'session_data_ready' event at the moment of receiving a snapshot does not mean at all that the data is ready, because the 'session_data_ready' event may refer to the previous trading session. Therefore, you can use the notifications received in 'sys_events' to assess the consistent state of data in the system in online mode only.

2.6. Risk management and limitation of trading operations

2.6.1. Collaterals

The Risk Management System implemented into SPECTRA allows to dramatically reduce risks of non-fulfilment of obligations by permanent evaluation of market risks for every participant's position. The core of the system is the initial margin (hereinafter also referred to as IM) calculation algorithm, applied to open positions and orders recognized on clearing and trading participants' position accounts.

One of the key features of the SPECTRA Risk Management System is the calculating initial margin on orders and positions per one trading transaction in online mode. Therefore, it is almost impossible for non-pledged orders and trades to appear in the system, because the initial margin is always verified before any relating order appears in the system.

Another important feature of the SPECTRA Risk Management System is the three-level calculating scheme, in accordance to which the position accounts are subdivided into three groups:

Settlement Account - the upper-level account of a clearing participant (i.e. Clearing Firm). The Settlement Account is an independent account for recognizing collateral assets margined by a trading participant (and/or their clients), orders added for all lower-level accounts (sub-accounts) belonging to the Settlement Account, trades performed basing on these orders, and resultant positions. Therefore, a position for an instrument recognized under the Settlement Account is equal to the net amount of all positions for the given instrument which are recognized under the sub-accounts.

The amount of initial margin for a Settlement Account is calculated independently of the other settlement account. All settings of SPECTRA Risk Management System are specified by the Central Counterparty (clearing firm).

During a clearing session, the system calculates the clearing participant's obligations and requirements (variation margin, commission fees, etc.). Also, the system calculates collateral sufficiency to meet initial margin requirements.

Settlement accounts are subdivided into three sections:

- own - trades are covered with clearing participant's assets;
- client - trades are covered with direct clients and clearing participant's 2nd-level clients' assets;
- Asset management - trades are covered with assets managed by a clearing participant.

For each clearing participant (Clearing Firm), there are at least two Settlement Accounts assigned: own and client.

Each Settlement Account will be identified by trading system SPECTRA in accordance with its unique 5-digit code.

Brokerage Firm - a sub-account of a Settlement account, which can be set up upon application by the clearing participant (Clearing Firm). Each Brokerage Firm belongs to a single Settlement Account, where the Settlement Account is subject to change upon the clearing participant application applied to the clearing entity. To make a Settlement Account available in SPECTRA, there should be at least one Brokerage Firm bound to it.

By default, initial margin value is calculated in half nett mode (margin_type =3 in command ChangeBFPParametersNextSession) in accordance with risk values for positions recognised at client sections of Brokerage Firm. Nett mode is also available for Brokerage Firm for initial margin calculation (margin_type =4 in command ChangeBFPParametersNextSession); using this mode, the initial margin value of a Brokerage Firm will be calculated according to nett sum of all positions for the given instrument at all sections of the Brokerage Firm, and total amount of orders added for sections of the Brokerage Firm.

All margining settings of a Brokerage Firm can be changed by a clearing participant (Clearing Firm) using the command `ChangeBFParametersNextSession`.

Separate Brokerage Firm (SBF) - a separated sub-account of a Settlement account, similar to common Brokerage Firms, purposed for recognizing collateral assets deposited by a client, and/or their clients, and not recognized at any section of common Brokerage Firms.

Also, each SBF contains a special account called liquidation account, which is purposed to recognize positions based upon trades performed by Clearing Centre in order to handle obligations unperformed by a clearing participant (for example, an unperformed Margin Call requirement for the Settlement Account). None of clearing participants (a Clearing Firm) is able to add orders with the liquidation account specified; excluding the orders aimed to lower the amount of an opened position for the given account. Also, clearing participants (Clearing Firms) are able to transfer positions from the liquidation account to other Brokerage Firms' accounts (command `TransferClientPosition`).

Client Account - a sub-account of Brokerage Firm. The low-level account, which can be opened upon the application by client, and specified as the 'client code' in order adding transaction. This is the primary account to recognize orders added by participant and/or client, trades performed upon these orders, and open positions; the initial margin value will be calculated using these orders and positions. One can change Client Account margining settings via commands `ChangeClientParameters`, `ChangeClientParametersNextSession`, `ChangeBFClientDefaultParametersNextSession`.

2.6.2. Trading limits

Trading limits are aimed to restrict a participant, and/or their clients, from adding orders and open positions for position accounts.

Trading limit for a Settlement Account can be calculated based upon total imputed value of IM recognized for the given Settlement Account, i.e. total value of IM recognized for all sub-accounts of the given Settlement Account. The Kazakh tenge is used as collateral.

Trading limit for a Settlement Account can be changed by depositing, withdrawal, or transferring collaterals, based upon requests applied to the clearing entity, or clearing depository (as well as to other settlement entities, once there have been any collateral deposited) by participant via the appropriate EDM systems.

Trading limits are used to reserve negative varying margins, withdraw fees and premiums, accrue premiums and reserve collaterals.

The trading limit for a Brokerage Firm does not depend on the size of the collateral and is set using the `ChangeBFLimit` command. Also, the trading limit of the Brokerage Firm is changed to the amount of profit or loss determined during the evening clearing session (variation margin and fees).

Trading limit for Client Account does not depend on IM value recognized for the given account. To manage trading limits for Client Accounts, please use command `ChangeClientMoney`, with the following possibilities:

- Set up/change/delete trading limits;
- The client's trading results will be automatically applied for limits in the next trading session.

As a rule, one is able to add an order only if there are sufficient limits to cover the required IM for all three levels (Client Account, Brokerage Firm, Settlement Account). It is possible to switch off the limit sufficiency verification for Brokerage Firm and Client Account using commands `ChangeBFParametersNextSession` and `ChangeClientMoney`, appropriately.

Please note that it is not possible to switch off the limit sufficiency verification for a Settlement Account.

2.6.3. Limitations on trading operations and opening positions for clients

The SPECTRA system allows to set up the additional limits on client trading operations, i.e. prohibitions, when it is possible to prohibit opening positions and placing orders for a certain client (or for all clients), a certain instrument (or for all instruments) or a certain underlying asset (or for all underlying assets). To perform such actions, the gateway provides the `FutChangeClientProhibit` method.

Also, the system provides possibilities of automatic prohibition on opening positions and adding orders in case of occurrence of a large negative trading limit. The following parameters are used to manage the prohibition settings:

- `Pr_state` - Automatic prohibition application; 0 - do not apply prohibition, 1 - apply prohibition.
- `Pr_type` - Prohibition type; 0 - prohibition on opening positions, 1 - prohibition on adding orders.
- `Pr_coeff` - Multiplying coefficient; a positive fractional number with an accuracy of 2 decimal places.
- `Del_ord` - Action on applying the prohibition; 0 - do not cancel orders, 1 - cancel orders.

The settings can be applied by a clearing participant acting on behalf of a Brokerage Firm. There are two sets of parameters available for applying prohibitions, with one set is applicable for the Brokerage Firm's clients, and the other one is applicable for the Brokerage Firm as a whole.

The prohibition settings are applied based on a clearing participant's request sent to the Clearing Firm using the appropriate electronic document flow systems. The new settings will be applied at the next clearing session. Please note that the prohibition request should be sent not later than 1 hour prior to the clearing session.

Apply prohibitions. After the limits have been extended during the evening and intraday clearing sessions, the prohibition will be automatically applied for the appropriate 7-symbol client section, or Brokerage Firm, when all the following conditions are met simultaneously:

$$\begin{cases} Limits_set = 1 \\ Pr_state = 1 \\ FreeMoney < - Pr_coeff * max(Trade_limit, 0) \end{cases}$$

, where

- Limits_set – Client limit verification setting;
- Trade_limit – Trading limit, i. e. funds and collateral, including liquidity ratio.
- FreeMoney – Free funds available for the client section/Brokerage Firm

The prohibition type is specified via parameter 'Pr_type'. If 'Del_ord=1', then all active orders will be automatically cancelled at applying the prohibition. Limit verifications are held independently for the Brokerage Firm and for the clients.

Cancel prohibitions. The applied prohibitions cannot be cancelled by the Brokerage Firm; instead, the prohibitions will be cancelled automatically following the every minute verification result, when there is at least one condition met:

$$\begin{cases} FreeMoney \geq 0 \\ Pr_state = 0 \\ Limits_set = 0 \end{cases}$$

Example. If a prohibition is set on opening positions, a client is able to cancel the orders, or close the position which causes the increased collateral requirements. The prohibition then will be automatically cancelled in not more than a minute.

One is unable to cancel prohibitions during the night sessions, despite the availability of trading system.

By default, prohibitions are disabled from being set for all Brokerage Firms (Pr_state=0).

2.6.4. Position (obligation) transfer

The SPECTRA system provides possibility to transfer a positions from one Brokerage Firm client to another client of the same Brokerage Firm.

To transfer a position from one clearing register to another, a clearing participant should add a transaction TransferClientPosition into the Trading system.

Verification procedures on position transfer are the same as that of adding an order. Additionally, it is verified that volume of the position to be transferred does not exceed that of the donor account. Also, the UIN (including separated brokerage firms accounts) must be equal for both accounts.

Technically, the position transfer is a trade, performed between a donor account and a recipient account. Juridically, it is not a trade (for details see Trade types, created upon exercising and expiration of instruments).

2.6.5. Pausing trading session for extending limits of trading prices fluctuations

Technically, the following actions take place in the SPECTRA system when pausing trading:

- When the condition is set to pause trading for a certain underlying asset, then the trading pauses for this asset.
- The trading administrators calculate the new extended limits of prices fluctuations.
- The amount of collateral is recalculated for every position, which includes the underlying asset (if the limits extend, then the amount grows)
- After the collateral is recalculated, the trading still pauses, allowing participants to delete orders.
- The trading resumes in the standard mode.

The corresponding notifications are sent on every action listed above (see the 'sys_message' table of the 'FORTS_REFDATA_REPL' stream):

- Warning about the upcoming trading pausing for a certain instrument if the prices remain unchanged.
- Notification about pausing the trading.
- Notification about successful recalculation of collateral (orders can now be deleted).
- Notification about resuming the trading.

2.6.6. Risk parameters forecast information for trading participants

The Trading System provides risk parameters forecast information to the trading participants (via service ForecastIM). The service recalculates the Initial Margin value at a specified time interval, forecasting a new most probable value which would occur after the limits were extended. Then, the new data will be transmitted to the trading participants.

This will be done in the following steps:

- At the time interval of 1 minute, the market condition is being analyzed for instruments due to which the limit extension procedure may, or will be performed (once an instrument's price value stays close to a specified limit for more than X minutes).
- Once such instruments are detected, the Initial Margin will be recalculated for client portfolios. The new risk parameters will be applied for the instruments, according to the new limit values set after extension.
- The recalculated funds are transmitted within the table `part_sa_forecast` of the replication stream `FORTS_FORECASTIM_REPL`.
- Once the limit extension risk is over due to the market condition change, or when the limits have been already extended before, the service halts recalculation and transmitting of the Initial Margin value. All the previously received data will be declared as non-valid (all forecasted data in the appropriate table will be cleared at receiving command `CLEARDELETED` with the maximum possible revision value).

If the limits have been already extended twice for the same instrument during a single trading session, there will be no more risk parameters forecasted and forecast information transmitted for this given instrument during this trading session.

2.6.7. Blocking the brokerage part of the client fee

The SPECTRA system allows the broker to block part of the client fee on the exchange side. The blocked fee is taken into the client account, reducing the client's free funds (`money_free`) by the amount of the blocked part. The blocking is carried out during the trading session and is reset in the evening clearing.

Brokerage fee is calculated based on the exchange fee. The brokerage part of the fee is calculated according to the following formula for each trade:

$$broker_fee = N \times \max\left(lower_fee, \min\left(upper_fee, \text{round}\left(\frac{multiplier \times ex_fee}{N}, 2\right) + additive\right)\right)$$

- *N* – number of contracts per trade;
- *lower_fee* – minimum brokerage fee per contract;
- *upper_fee* – maximum brokerage fee per contract;
- *multiplier* – multiplier to the amount of exchange and clearing fees;
- *ex_fee* – clearing and exchange fees per trade with scalper discounts;
- *additive* – constant addition per contract.

The broker can set '*lower_fee*', '*upper_fee*', '*multiplier*' and '*additive*' parameters using the command `SetBrokerFeeParamNextSession`. Parameters can be set for an individual client and for the entire brokerage firm. The parameters set for the BF are used in the calculation for all of its clients. The set parameters will be applied in the next trading session. The set parameters are broadcast in the gateway in the stream `FORTS_BROKER_FEE_PARAMS_REPL`.

For example, the broker always takes half the exchange fee - then '*multiplier*' = 0.5, '*additive*' = 0, '*lower_fee*' = 0.01, '*upper_fee*' = inf. Or the broker always takes 2 rubles for any contract - then '*multiplier*' = 0, '*additive*' = 2, '*lower_fee*' = 2, '*upper_fee*' = 2.

The brokerage fee is broadcast in the gateway in the table '*part*' of the stream `FORTS_PART_REPL` (total by client) and in the stream `FORTS_BROKER_FEE_REPL` (by trades).

2.6.8. Negative prices support in SPECTRA (**)

Starting from this release SPECTRA supports negative prices, i. e. correct system behavior if futures prices and options strikes fall below zero during trades or as the result of clearing. There are two different modes for every underlying asset:

- Mode in which futures prices and options strikes are not limited. Negative and zero futures prices and options strikes are allowed in this mode. In this case options prices, volatility and risks are calculated based on Bachelier model or modified Black-Scholes model, which takes into account only intrinsic value of an option in negative range.
- Mode in which futures prices and options strikes are limited to be positive. In this mode prices cannot fall below zero during or as the result of trades. Option prices are calculated based on Black-Scholes model (Bachelier model may be used as alternative). However, this mode allows manual negative exercise price and /or indicative current market price setting in case of corresponding CCP (KASE) decision. Nevertheless futures prices and options strikes have positive limits.

Mode and option pricing model are set on underlying contract level and are effective for all instruments of this underlying contract. Modes and option pricing models can be switch during the clearing session. To set the mode and risk model the following parameter of underlying contract is used:

- *negative_prices*: 1 – futures prices and options strikes are not limited; 0 - futures prices and options strikes are limited to be positive only.
- *option_model*: 1 – Bachelier model; 0 - Black-Scholes model.

Current parameter values is published in the '*fut_vcb*'/'*opt_vcb*' tables of `FORTS_REFDATA_REPL` data stream in the gateway.

In the prohibition mode of negative prices (negative_prices = 0), if the CCP (KASE) decides accordingly, there is a possibility to manually set indicative current market price, which is published in FORTS_COMMON_REPL data stream. This price affects indicative current variation margin translated in FORTS_VM_REPL stream and current theoretical options price published in FORTS_VOLAT_REPL stream. To indicate that current market price for futures is set manually the following parameter is used:

- price_assigned_by_admin – attribute of manual current market price setting by trades Administrator.

Fields of the trading interface tables, where negative values may appear in the negative prices mode ('negative_prices' = 1):

| Stream | Table | Field | Description |
|----------------------------------------------------|---------------------|-----------------------|------------------------------------------------------|
| FORTS_TRADE_REPL | orders_log | price | Price of the order |
| FORTS_TRADE_REPL | orders_log | deal_price | Price of the trade |
| FORTS_TRADE_REPL | user_deal | price | Price of the trade |
| FORTS_ORDLOG_REPL | orders_log | price | Price of the order |
| FORTS_ORDLOG_REPL | orders_log | deal_price | Price of the trade |
| FORTS_USERORDERBOOK_REPL | orders | price | Price of the order |
| FORTS_ORDBOOK_REPL | orders | price | Price of the order |
| FORTS_COMMON_REPL | common | best_buy | Best bid |
| FORTS_COMMON_REPL | common | best_sell | Best offer |
| FORTS_COMMON_REPL | common | open_price | Opening price |
| FORTS_COMMON_REPL | common | close_price | Closing price |
| FORTS_COMMON_REPL | common | price | Price of the last trade |
| FORTS_COMMON_REPL | common | min_price | The low price |
| FORTS_COMMON_REPL | common | max_price | The high price |
| FORTS_COMMON_REPL | common | avr_price | Average weighted price |
| FORTS_COMMON_REPL | common | settlement_price_open | Settlement price of the previous session |
| FORTS_COMMON_REPL | common | market_price | Current market price |
| Streams of the aggregated order-books FORTS_AGGRXX | orders_aggr | price | Price level |
| FORTS_POS_REPL | position | waprice | Volume-weighted average price |
| FORTS_POS_REPL | position_sa | waprice | Volume-weighted average price |
| FORTS_REFDATA_REPL | fut_sess_contents | limit_up | Upper price limit |
| FORTS_REFDATA_REPL | fut_sess_contents | limit_down | Lower price limit |
| FORTS_REFDATA_REPL | fut_sess_contents | settlement_price_open | Settlement price at the start of the session |
| FORTS_REFDATA_REPL | fut_sess_contents | settlement_price | Settlement price after the last clearing session |
| FORTS_REFDATA_REPL | fut_instruments | settlement_price_open | Settlement price at the start of the session |
| FORTS_REFDATA_REPL | fut_instruments | settlement_price | Settlement price after the last clearing session |
| FORTS_REFDATA_REPL | fut_rejected_orders | price | Price of the order |
| FORTS_REFDATA_REPL | opt_sess_contents | strike | Exercise price |
| FORTS_MM_REPL | fut_MM_info | price_edge_sell | Price of the worst sell order included in the spread |
| FORTS_MM_REPL | fut_MM_info | price_edge_buy | Price of the worst buy order included in the spread |
| FORTS_CLR_REPL | fut_sess_settl | settl_price | Settlement price |
| FORTS_INFO_REPL | futures_params | risk_range_center | Risk calculation center |
| FORTS_INFO_REPL | futures_params | settlement_price | Settlement price after the last clearing session |
| FORTS_INFO_REPL | options_params | strike | Exercise price |

In the positive prices mode ('negative_prices' = 0), in the case of an appropriate decision by the CCP (KASE), it is possible:

- use of negative futures exercise price;

- broadcast of a negative value as an indicative current market price set by the Trading Administrator ('price_assigned_by_admin' = 1) in the 'market_price' field.

Negative and zero values in the instruments trading codes are displayed as follows:

The codes with strike "-10":

- Contract short code ('short_isin'): "BR-10BF0".
- Long contract code ('isin'): "BR-7.20M250620CA-10".

The codes with strike "0":

- Contract short code ('short_isin'): "BR0BF0".
- Long contract code ('isin'): "BR-7.20M250620CA0".

2.7. SMA Login (Sponsored Market Access)

SMA, i.e. Sponsored Market Access is a method of providing trading participants' clients with technical access to the trading and clearing system of the Derivatives Market of Moscow Exchange. Using SMA, a client is able to send requests to the trading participant (i.e. the sponsoring entity) for adding orders directly into the trading system under control and responsibility of the client.

In order to gain access to the trading system, each trading participant's client is granted with a personal ID, i.e. SMA login, using which a client is able to add orders directly to the trading system via gateways Plaza2, FIX, and TWIME.

In order to control the client's transactions performed via their SMA login, each SMA login is bound to the trading participant's login called MASTER login. Using their MASTER login, a trading participant is able to connect to the trading system, add orders and carry control of performing orders. A trading participant is able to use a single MASTER login for more than one SMA login. Also, each SMA login can be bound to more than one MASTER login. All logins are available via gateways in table 'user' of stream FORTS_REFDATA_REPL, where SMA logins are flagged with 1 in 3rd bit of bitmask 'sma_flags'. All SMA-MASTER login bindings are available via gateways in table 'sma_master' of stream FORTS_REFDATA_REPL.

In order to obtain their SMA login, a trading participant should file an application containing their MASTER login into the Client Centre of Moscow Exchange in order to carry control over transactions performed via SMA login.

The following risk management services are provided by the Moscow Exchange in order to prevent erroneous orders from being added into the trading system:

- Pre-Trade control - some settings to carry additional control over adding orders;
- Cancel On Drop-Copy Disconnect - the service guarantees that all orders added from an SMA login are available in the trading system only when the appropriate MASTER login is connected and active. Every order added via an SMA login contain references to the MASTER login (field 'asprel' of tables 'orders_log' and 'multileg_orders_log');
- UserKillSwitch - forced deactivation of SMA login by a trading participant.

Pre-Trade control service provides some additional restrictions/verifications which can be imposed/applied on adding orders from an SMA login. The verifications can be applied for selected SMA logins, instruments, or client codes, where instruments are:

- <Underlying asset>: <Derivative type>, where <Derivative type> = {Futures, Option, Calendar Spread} - Instrument*
- <Underlying asset>: <Derivative type>, where <Derivative type> = {Futures, Option} - Instrument**

The following verifications can be applied:

| Verification number | Verification details | Binding | Unit of measure | Applied |
|---------------------|------------------------------------------------------|----------------------------------------|---------------------|-------------|
| 1 | Price fluctuation against the current price | SMA login or SMA login x Instrument** | Percent | Immediately |
| 2 | Maximum volume of order | SMA login or SMA login x Instrument* | Number of contracts | Immediately |
| 3 | Disallow negotiated mode | SMA login | Yes/No | Immediately |
| 4 | Maximum volume of order in Kazakhstani Tenge | SMA login or SMA login x Instrument* | KZT | Immediately |
| 5 | Maximum volume of all orders per trading day (gross) | SMA login or SMA login x Instrument* | KZT | Immediately |
| 6 | Maximum position (long) in contracts | SMA login x Instrument** x Client code | Number of contracts | Immediately |
| 7 | Maximum position (short) in contracts | SMA login x Instrument** x Client code | Number of contracts | Immediately |

To apply/cancel the verifications, the gateway methods SetSmaPreTradeCheck and DelSmaPreTradeCheck can be used, appropriately. Information on the already applied verifications is available via gateways in table 'sma_pre_trade_check' of field FORTS_REFDATA_REPL.

Cancel On Drop-Copy Disconnect - the service guarantees that all orders added from an SMA login are available in the trading system only when the appropriate MASTER login is connected and active.

When an order is being added from an SMA login, the service verifies whether there is at least a single MASTER login bound to the given SMA login. If there is no MASTER login detected, the order will be rejected, and the appropriate error message will be issued. If there is at least one bound MASTER login connected and alive, the order will be processed, and a reference to the given MASTER login (MASTER login ID) will be added into field 'aspref'.

The service constantly verifies in real-life mode (similar to that of the Cancel On Disconnect service) if MASTER logins are alive. If there is no transactional activity detected from a MASTER login, the MASTER login will be deactivated. Once there is no any MASTER login remains connected for an SMA login, all orders added via the given SMA login will be cancelled.

All active orders added via an SMA login will be cancelled automatically on technological break in the end of the trading day if the Cancel On Drop-Copy Disconnect service is enabled for these SMA logins.

In order to get access to the Cancel On Drop-Copy Disconnect service, the appropriate application should be filed to the Client Centre of Moscow Exchange.

Method UserKillSwitch allows trading participants manually deactivate (and activate again) an SMA login, with the supported possibility to automatically cancel all active orders of the given SMA login. Once deactivated, an SMA login is no more able to perform trading transactions till the end of the trading day. The SMA login will be activated again after the trading system restart due to technological break, or due to a failure.

2.8. Settlement trades

Settlement trades are concluded by NCC on behalf of and at for settlement account (SA) of the Clearing Member.

If a Clearing Member fails to fulfill its obligations in time, then NCC considers such a participant to be a Defaulting Clearing Member (DCM). NCC, on behalf of and for settlement account of the Defaulting Clearing Member, concludes trades that lead to a reduction in position and fulfillment of obligations. The purpose of such trades is to eliminate the insufficient of collateral for obligations with a maturing and non-maturing execution date. The procedure is described in more detail in the Clearing Rules in the "Procedure for Margin Calls and default funds Margin Calls" section.

NCC concludes settlement trade on behalf of and according to the pre-agreed settlement account of the Non-defaulting Clearing Member, if the settlement trade with Defaulting Clearing Member cannot be concluded via the order book. The procedure is described in more detail in the Clearing Rules in the "Procedure for the closing and/or balancing trades execution" section. Commissions (fines) are not charged for trades with Non-defaulting Clearing Member.

2.8.1. Reasons for settlement trades

The flag of settlement trades is broadcast in the gateway in the tables of yours orders 'orders_log' and 'multileg_orders_log' ('reason' field) and trades 'user_deal' and 'user_multileg_deal' ('reason_buy' and 'reason_sell' fields), and also in reports: 'f04', 'f04cl', 'o04', 'o04cl'.

| Field value 'reason/ reason_buy/ reason_sell' | Reason | Member |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| 0 | Common trade. | Clearing Member |
| 4 | Balancing Derivatives Contracts entered into with the Non-defaulting Clearing Member without submitting orders. | Non-defaulting Clearing Member |
| 6 | Closing Derivatives Contracts entered into under the cross-default procedure. | Defaulting Clearing Member |
| 7 | Closing Derivatives Contracts entered into upon non-fulfillment of the Margin Call. | Defaulting Clearing Member |
| 8 | Closing Derivatives Contracts entered into in into upon non-fulfillment of the Delivery Obligation on the deliverable Derivatives Contracts for precious metals. | Defaulting Clearing Member |
| 100 | Other | Defaulting Clearing Member |

In the 'f04', 'f04cl', 'o04', 'o04cl' reports, the reason for settlement trades is in the 'Type' field.

Futures trades reports 'f04', 'f04cl':

- "3" - for balancing Derivatives contracts entered into with the Non-defaulting Clearing Member without submitting orders;
- "21" - for closing Derivatives contracts entered into under the cross-default procedure;
- "22" - for closing Derivatives contracts entered into upon non-fulfillment of the Margin Call;
- "23" - for closing Derivatives contracts entered into in into upon non-fulfillment of the Delivery Obligation on the deliverable Derivatives contracts for precious metals.

Options trades reports 'o04', 'o04cl':

- "3" - for balancing Derivatives contracts entered into with the Non-defaulting Clearing Member without submitting orders;
- "6" - for closing Derivatives contracts entered into under the cross-default procedure;
- "7" - for closing Derivatives contracts entered into upon non-fulfillment of the Margin Call.

2.8.2. Fines and fees

For settlement trades, a penalty is charged from the Defaulting Clearing Member instead of a commission. The amount of the penalty for concluding the closing Derivatives contracts is equal to the sum of 5 exchange fees established by Moscow Exchange and 5 clearing commissions from the amount of the closing Derivatives contracts, without applying a scalper discount. The penalty is calculated for each settlement trade and is accounted for the 7-digit section of the Clearing Member specified in the settlement trade.

Information about penalties is broadcast in the gateway in the 'penalty' field of the 'part' table in the FORTS_PART_REPL stream (in total for 7-digit section), and also in the 'penalty' table of the FORTS_FEE_REPL stream (in the context of trades).

Fines and commissions for concluding settlement trades on behalf of the Non-Defaulting Clearing Member (balancing Derivatives contracts entered into with the Non-defaulting Clearing Member without submitting orders) are not charged.

Fines for the conclusion of closing Derivatives contracts with Defaulting Clearing Member are not charged:

- if the Clearing Member is under the Liquidation Netting procedure;
- if the Clearing Member is in the status "Suspension of clearing services for the Clearing Member due to cancellation of the license to carry out professional activities in the Securities Market".

Information on these locks is broadcast in the gateway in the 'clearing_members' table of the FORTS_REFDATA_REPL stream.

Settlement trades for which no fines were charged due to cancellation of the license to carry out professional activities in the Securities Market are marked in the tables of trades in the 'xstatus_sell' and 'xstatus_buy' fields with a special sign:

- eDontFineRF (0x8000000000000000) - No penalty for settlement trades.

Information on the amount of the fine is included as a new type of payment in 'pay' reports on the date of debiting. Fines are taken into account in the report in the state of the current money position. Reports:

- payXXYY.csv;
- payclXXYY.csv.

3. Trading gate description

3.1. SPECTRA Plaza-2 gateway. Components, installation and setup.

3.1.1. Components and architecture

The SPECTRA Plaza-2 gate consists of the following software components:

- The 'P2MQRouter' module. This module provides the following services:
 - Establishing TCP-connections to the Exchange servers.

Normally, the SPECTRA Plaza-2 gate uses four TCP-connections to the Exchange servers: one outgoing default connection and three outgoing direct connections. This structure is used as the standard to establish connection directly to the Exchange server farm, but connection via a Brokerage Firm server server may require a different structure; in this case, clients should apply to the server's owners for more details about connection.

- Receiving/sending P2-messages.
- Encrypting data sent by participants and decrypting data received from the Exchange.
- Authentication of participants in the Exchange network.
- 'cgate' - the gateway library.

The library is the official software interface, provided to trading participants along with their clients as well as to software developers. The interface provides availability to create and send messages into the trading system and receive trading data from the trading system (data replication). There are x32 and x64 versions available for Windows systems, as well as a version for Linux OS.

- Plaza-2 system libraries.
- Software development kit, including additional utilities, command files, documentation and test examples.

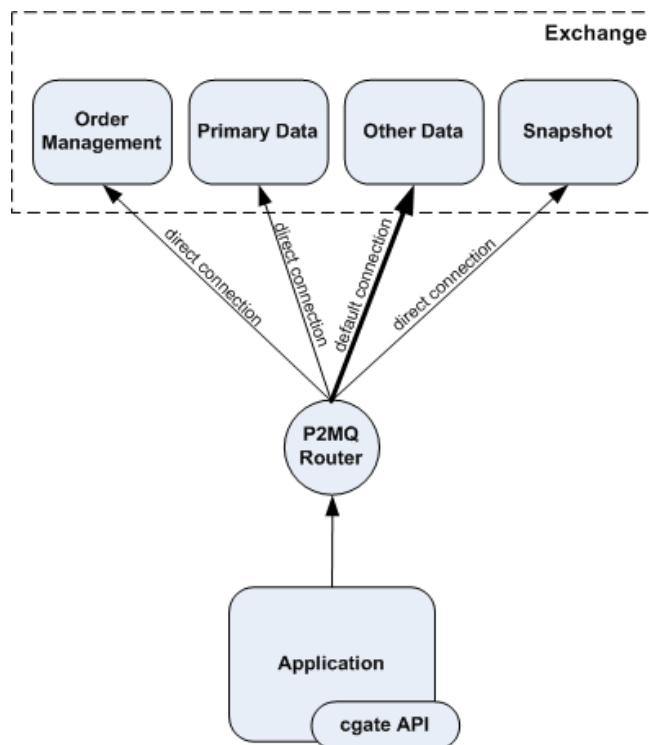


Figure 1. Gateway architecture

3.1.2. Hardware and software requirements

3.1.2.1. Hardware requirements

Hardware requirements may vary depending on usage of the Plaza-2 gate.

The minimum system requirements for individual login without disk saving option are as follows:

- CPU: Core 2 duo 1 Ghz or better
- Memory: 2 GB or more for x32 systems, 4 GB or more for x64 systems.

The minimum system requirements for brokerage firm login without disk saving option are as follows:

- CPU: Intel Xeon 53xx 2 cores or better (or a similar AMD CPU 2 cores or better)
- Memory: 24 GB or more
- Separate SAS controller. Minimum 2 hard drives in RAID1. Two partitions, 30 GB each.

The minimum system requirements for brokerage firm login with disk saving option are as follows:

- CPU: Intel Xeon 53xx 2 cores or better (or a similar AMD CPU 2 cores or better)
- Memory: 4 GB or more
- Separate SAS controller powered with the write-back cache policy. Minimum 4 hard disks in RAID10. Two partitions, 30 GB each.

3.1.2.2. Software requirements

The following operation systems are supported by the gateway software:

- Microsoft Windows 10 (both 32 bit and 64 bit OS versions are supported)
- Microsoft Windows Server 2016/2019 (both 32 bit and 64 bit OS versions are supported)
- Linux RedHat/CentOS 7 and newer (only 64 bit versions). It is also possible to use other distributions

3.1.3. Installation for Windows

Download the latest gateway software version from. The installation file's name is 'setup_SpectraCGate_x64_vx.x.x.x.exe', where x.x.x.x is the software version number, for example 6.6.0.186

Run 'setup_SpectraCGate_x64_vx.x.x.x.exe'. The installation wizard will guide you through the installation process:

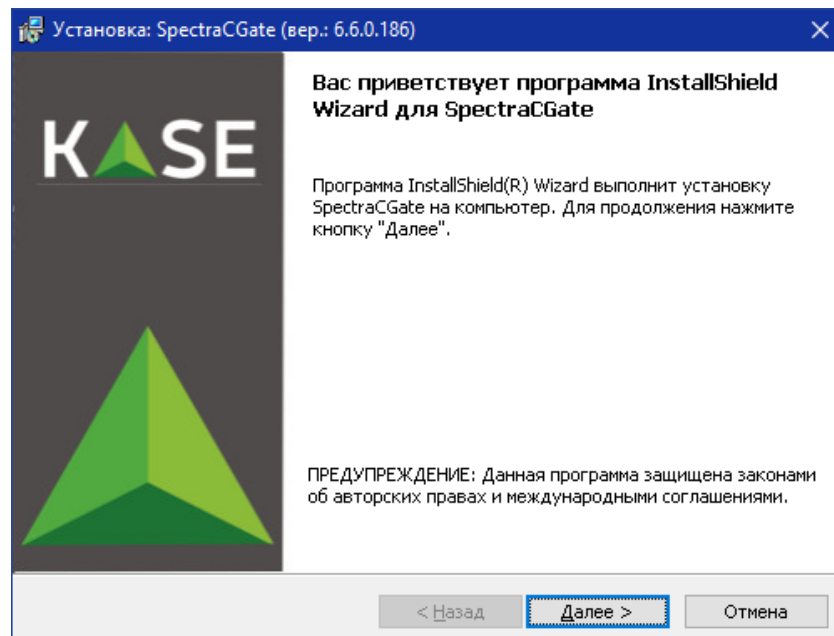


Figure 2. Installation start

Click the 'Далее' button to continue with installation:

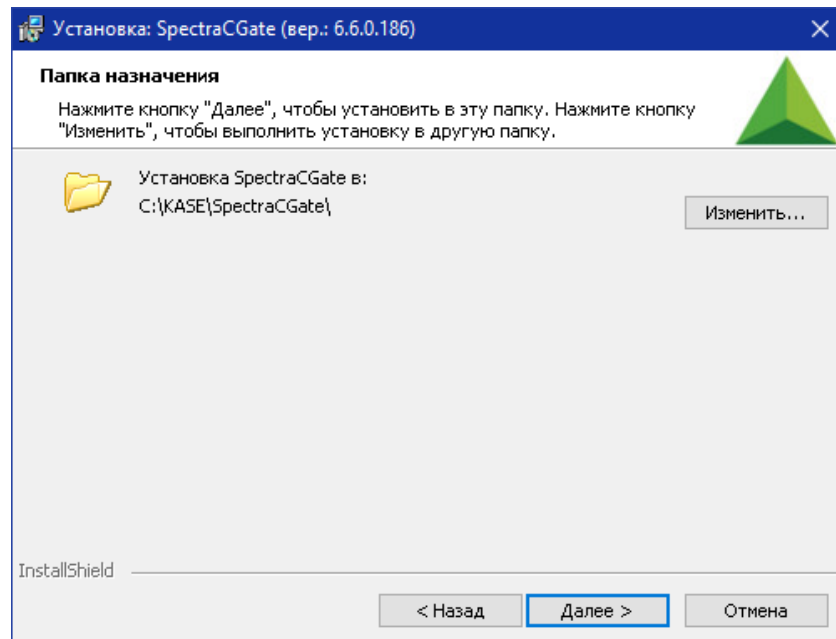


Figure 3. Destination folder

The default destination folder is C:\KASE\SpectraCGate\. To confirm installation using the default folder and continue to the next step, click the 'Далее' button.

To change the destination folder, click the 'Изменить...' button. A new window appears in the screen; in this window, select a new destination folder using the "Поиск в папке" button; to move one level up in the folder tree use the button. Also, you can create a new destination folder using the button, or select an already existing one by manually typing the path in the "Имя папки" entry box in the lower part of the window. Click 'OK' to apply the changes you have made and close the current window. In the "Папка назначения" window, press the "Далее" button to continue to the next step.

Please be known that you will only be able to change the destination folder upon the initial installation, or when you are re-installing the software anew. In all other cases, you will not be able to change the destination folder (the 'Изменить...' button will be disabled).

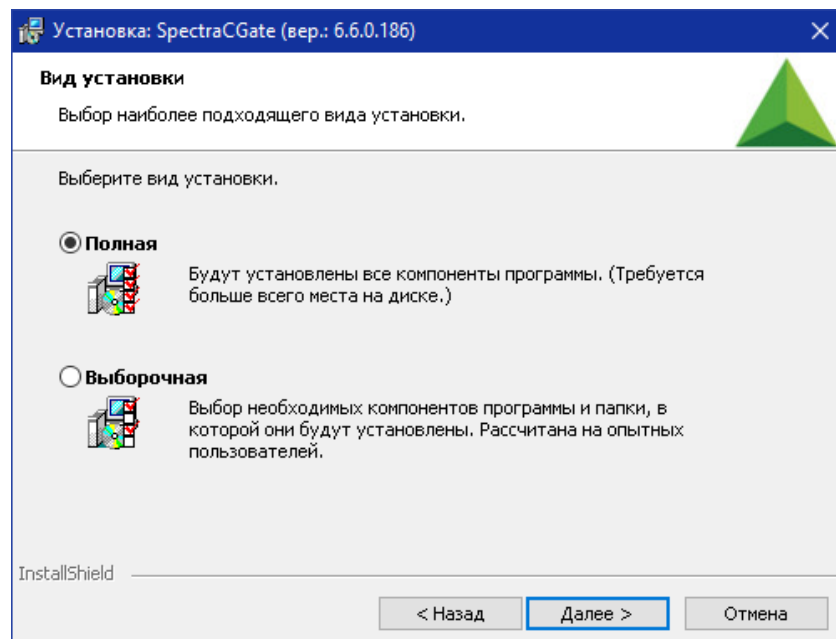


Figure 4. Select components to install

Select the installation mode you want to use, full or custom. The full install mode will install all the gateway components including module P2MQRouter, library cgate, additional utilities and the software development kit. The custom install mode allows you to manually select software components to install.

Click the 'Далее' button to continue with installation:

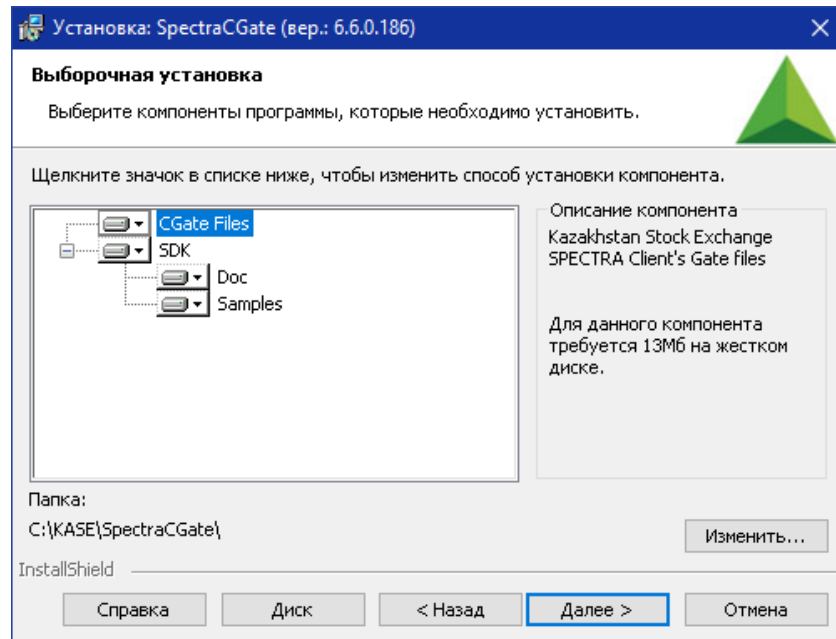


Figure 5. Custom install

Select the software components you need to install and the destination folder. The destination folder should be selected in accordance with the administrative recommendations.

Click the 'Далее' button to continue with installation:

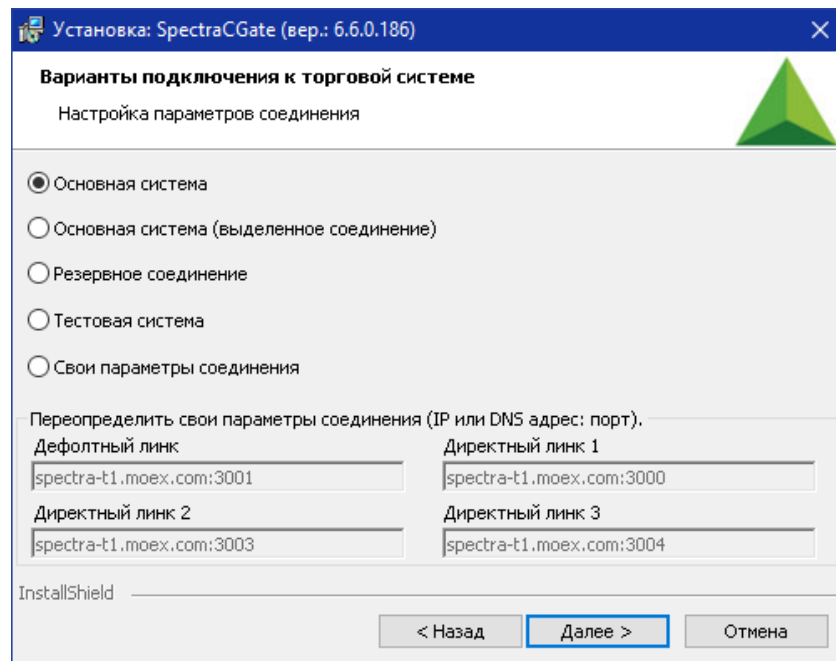


Figure 6. Select an address to connect

Select the trading system to which you want to connect (production, test, etc), or enter your parameters for connection to the exchange servers. The connection parameters are stored in a separate configuration file for each connection option. A configuration files are in the 'links' directory of the installation directory.

| Connection option | Configuration file | Details |
|-------------------|-----------------------|----------------------------------------|
| Основная система | links_public.prod.ini | Connect to Spectra - production system |

| Connection option | Configuration file | Details |
|---------------------------|-------------------------|--------------------------------------------|
| Резервное соединение | links_public.rezerv.ini | Connect to Spectra - reserve system |
| Тестовая система | links_public.t0.ini | Connect to Spectra - public testing system |
| Свои параметры соединения | links_public.custom.ini | User defined connection |

After the installation complete, a link to the corresponding file with connection parameters will be added to the ini-file of the 'P2MQRouter' module in the 'connections_ini' parameter. To change the connection type, just restart the installer and select the desired option. Please note that if you reinstall or uninstall the system, the 'links' directory and the file with user connection settings (links_public.custom.ini) are not deleted.

The fields in the user settings section display:

- if initial installation - default values (for example, parameters from links_public.t0.ini).
- if reinstalling - user connection options from links_public.custom.ini or client_router.ini. If there are no files, the default values are displayed.

Figure 7. User connection settings section

For selecting the proper connection type, you should contact your brokerage firm and/or the Exchange technical support service.

Click the 'Далее' button to continue with installation:

Figure 8. Enter username and password

Enter username and password for the connection selected in the previous step.

After the installation complete, the username and password will be added to a separate configuration file 'auth_client.ini', which will be created in the 'auth' directory of the installation directory, and a link to this file will be added to the 'auth_ini' parameter of the 'P2MQRouter' module ini-file.

When reinstalling, the username and password values specified in 'auth_client.ini' or 'client_router.ini' files are displayed in the form fields. Please note that if you reinstall or uninstall the system, the 'auth' directory and the file with identification data (auth_client.ini) are not deleted.

Click the 'Далее' button to continue with the installation:

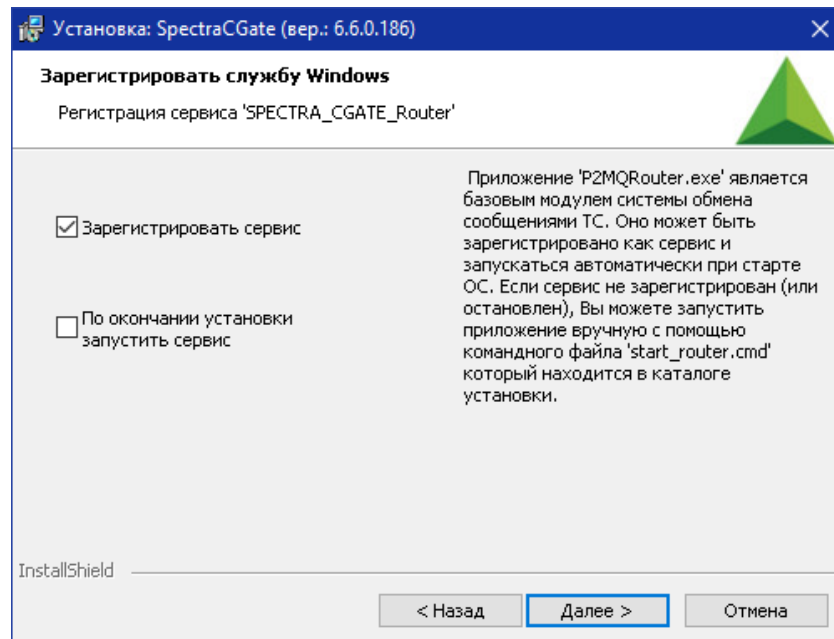


Figure 9. Registering router as OS service

If you need to install the Router as Windows service, check the appropriate checkbox and click the 'Далее' button to continue with the installation.

If you do not register the P2MQRouter as an OS service, you can do it later manually using the command file 'install_router.cmd (uninstall_router.cmd)'. The file is a part of distribution kit.

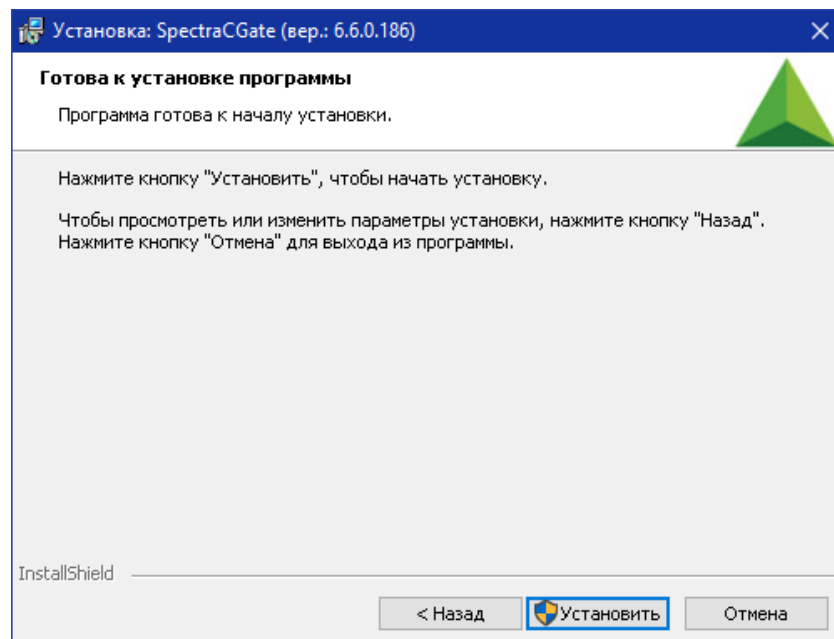


Figure 10. Starting installation

Click 'Установить' to begin installation.

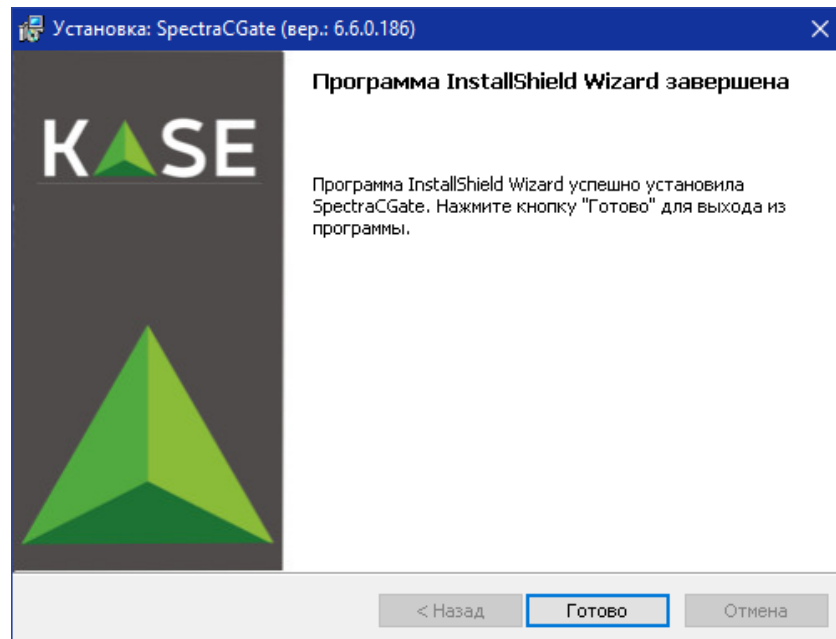


Figure 11. Finishing installation

Click 'Готово' to finish the installation.

3.1.4. Installation for Linux

The distribution kit for Linux OS consists of installation script ('install.sh') and archive file 'tar.gz' (for example, 'cgate-1.3.9.8.x86_64.tar.gz'); the archive file contains loadable modules 'cgate' and 'cgate_java', files 'include', documentation files and test examples.

Installation order:

1. Execute the command:

```
chmod 755 ./install.sh
```

2. Execute the command:

```
./install.sh ./cgate_linux_amd64-5.3.6.11.zip
```

Please note that the archive file name depends on the software version, and may differ from the one shown above!

3. When you receive 'Please, enter cgate install path', specify the full path to the folder to decompress the cgate software.
4. When you receive 'Please, enter P2 login', specify the user's login.
5. When you receive 'Please, enter P2 password', specify the user's password.
6. The next installation steps may vary depending on the Linux OS software version installed:

- Debian 6:
 - Install 'ant'
 - Install 'openjdk-6-jdk' (java examples compilation)
 - Install g++ (C++ examples compilation).
- CentOS 6:
 - Install 'gcc'
 - Install 'gcc-c++' (C++ examples compilation)
 - Install 'ant' (java examples compilation).

3.1.5. Developer guidelines

3.1.5.1. Usage of test examples

In order to verify the installation accuracy you can compile and run the test examples included into the distribution kit.

The examples can be found either in folder KASE\SpectraCGate\SDK\samples for Windows OS, or /usr/share/doc/cgate-examples for Linux OS. To compile examples, you should run the special scripts, which may vary depending on the operation system and programming language used. For Linux OS, it is recommended to copy the examples into your login directory to compile them.

Description of examples:

1. Example 'aggrspy'

'aggrspy' is an example which is used to build aggregated orderbook for to buy and sell a fixed instrument for the stream 'FORTS_FUTAGGR50_REPL'. Press 'Enter' to display the orderbook snapshot.

Execute:

```
aggrspy ISIN_ID depth outfile [r]
```

Input arguments:

- 'isin_id' - instrument's ID;
- 'depth' - depth of orderbook (up to 50);
- 'outfile' - orderbook file for printing;
- 'r' - reverse the sorting order (for instrument with reversed sorting order).

2. Example 'repl'

'repl' is an example which is used for receiving replication data and printing all incoming messages into a log-file. When disconnected, the replica transfer process starts anew from the beginning. No input parameters required.

3. Example 'repl_resume'

'repl_resume' is an example very similar to 'repl'. When disconnected, it allows to resume replica transfer process starting from the last message 'TN_COMMIT'. No input parameters required.

4. Example 'send'

'send' is used to add order into the SPECTRA and write incoming replies into the trading system log. No input parameters required.

5. Example 'orderbook'

'orderbook' is an example which is used to build aggregated orderbook for to buy and sell a fixed instrument for the online stream 'FORTS_ORDLOG_REPL' along with the snapshot stream 'FORTS_USERORDERBOOK_REPL'. It is recommended to use it for developing 'late join', and also for minimizing inactivity time when archival data is being downloaded. Press 'Enter' to display the orderbook snapshot.

Execute:

```
orderbook ISIN_ID depth outfile [r]
```

Input arguments:

- 'isin_id' - instrument's ID;
- 'depth' - depth of orderbook (up to 50);
- 'outfile' - orderbook file for printing;
- 'r' - reverse the sorting order (for instrument with reversed sorting order).

6. Example 'p2sys'

'p2sys' is an example, which is used for authorising the Router from cgate side. The following actions are executed cyclically:

- Send erroneous login/pwd pair, get the 'logon failed' in reply;
- Send the correct login/pwd pair;
- Receive an 'authorisation successful' message, send 'logout' request;
- Go back to the beginning.

7. Example 'send_mt'

'send_mt' is an example of multi-thread order adding. (Please note, that only C++11 compilers are supported!). Thread 1 is used for adding orders, while thread 2 is used for processing 'reply' messages received.

Before executing the examples, please make sure that 'P2MQRouter' has started and connected to the Plaza-2 network (with router messages analyzed), the INI files are accessible for the example file, as well as the Plaza-2 libraries (it may be necessary to add 'KASE\SpectraCGate\bin' folder into the PATH environment variable or specify KASE\SpectraCGate\bin for your development environment).

Note: The examples above are not intended to be used with data other than test data! It is strictly prohibited to use these examples for working with the real logins!

3.1.5.2. Distributed configurations

The 'cgate' application and the 'P2MQRouter' module can be distributed to different computers. To distribute the modules in the brokerage firm network, you should do the following: a) install the 'Router' module to the computer connected to the Exchange network; b) install 'cgate' to the client computer with the client application installed; c) specify the following settings:

- On the client side:
 - Specify the 'Host' and 'Port' settings in accordance with that of your corporate network router.
 - Specify the Password settings (the local AppName application password for the router, which must be applied every time the application connects router from outside of the same computer). Please note that the local connection password is not the same as the Plaza-2 authentication password!
- On the router side:
 - Add the '<AppName>=<local password>' string into the 'router.ini' file, [AS:Local] section, where 'AppName' (the application name) and 'Password' (the local password) should match the parameters transmitted by the client application.

3.1.5.3. Recommendations for third-party companies on including the KASE runtimes into user application when distributing the user software

Users should copy the file set from the installation folder (KASE\SpectraCGate\bin), as well as data and messages schemes (KASE\SpectraCGate\SDK\scheme) into the folder containing user application. All these software parts should be distributed together.

It is not allowed to use different versions of 'P2MQRouter' and 'cgate' due to incompatibility. Before installing user application, please make sure that the 'P2MQRouter' version matches the one used in developing.

3.2. Market data structure

This section describes the structure of information sent by Plaza-2 gateway.

All transmitted data is divided into the following logical groups:

- Reference information
- Trade information
- Recovery information
- Funds and limits information
- Clearing information
- Auxiliary data streams

3.2.1. Reference information

The reference information includes the following data:

- Trading session status and schedule

Trading session time information are available in 'session' table of the 'FORTS_REFDATA_REPL' stream. You can find trading session status in the same table, that helps to track current session status.

- Instruments and underlying assets dictionary, properties

Futures Instruments assigned to the trading session are available in the 'fut_sess_contents' table of the 'FORTS_REFDATA_REPL' stream. Dictionary of the futures' underlying assets is represented by the 'fut_vcb' table of the 'FORTS_REFDATA_REPL' stream.

These directories can be updated during the trading session, for example, as a result of the suspension of trading on any instrument or during the price limit enlargement procedure

- Companies and clients references

Are sent in the 'dealer' and 'investor' tables from the 'FORTS_REFDATA_REPL' stream. Personal clients' information is available in these references.

- Parametric volatility curve parameters

Are sent in the 'volat_coeff' table of the 'FORTS_MISCINFO_REPL' stream.

To carry out operations on all of the SPECTRA trading system markets user's system should receive at least the following reference information on-line:

- Sessions' schedule (session)
- Instruments dictionary (fut_sess_contents)

3.2.2. Trade information

Trade information includes:

- Aggregate orderbooks

Are generated on the basis of user system requests by adding up the volume for each instrument, the price level and the direction of an order. Updated online and comes to be the main way to get information by current prices and volumes. User can select the desired depth of an orderbook from 5, 20 or 50 of price levels in each direction; this choice is made when configuring a login and can not be changed during the trading session.

Orderbooks are sent by multiple Plaza-2 replication streams: 'FORTS_AGGR5_REPL', 'FORTS_AGGR20_REPL' and 'FORTS_AGGR50_REPL'.

- Market activity

The best bid/ask price, opening price, closing price, current settlement prices, etc are sent as a part of market activity information. This information is sent in the 'FORTS_COMMON_REPL' stream.

- User's orders log (and full orders log in the trade system)

The entire history of user's operations with orders is sent in user's orders log. User's orders log are available in 'orders_log' table of the 'FORTS_TRADE_REPL' stream.

In case the user configures his login with option to receive "full orders log", he will receive the complete log of all operations with orders on market (including own operations with orders) in anonymous mode. The log will be available in the table 'orders_log' of the stream FORTS_ORDLOG_REPL.

- User's deals log

It contains a list of user's committed deals in the current session. User's deals log are available in the 'user_deal' table of the 'FORTS_TRADE_REPL' stream.

- All trade system deals log

It contains a list of all committed deals from all users in the current session. Information of somebody else's deals is presented in anonymous mode. System's deals log are available in the 'deal' table of the 'FORTS_DEALS_REPL' stream.

3.2.3. Recovery information

To ensure fast recovery of trade information receiving after losing connection with SPECTRA, and same with late start scenario connecting to exchange, Plaza-2 gateway receives periodic snapshots from recent orderbooks in a non-aggregated form. This helps to receive the recent status of personal orders (in case when the 'full orders log' option is set - all orders in the trade system) at the current time.

Snapshots of active orders are sent with 2 minute interval in 'FORTS_USERORDERBOOK_REPL' stream.

3.2.4. User's limits information

Includes the following:

Position information

- Positions information

Is sent in form of time snaps in the 'FORTS_POS_REPL' stream and last deal ID, included in position calculation by each position value, is available.

- User's limits information

Is sent in form of time snaps in the 'FORTS_PART_REPL' stream. Limits at the beginning of the trade session, also current and reserved limits - all of them are available for each value of the client's account.

3.2.5. Clearing information

Clearing information, sent by Plaza-2 gateway, includes the following data:

- Clearing settlement prices

Are formed by the time of evening clearing and available in the 'fut_sess_settl' table of the 'FORTS_CLR_REPL' stream. The table with settlement prices also includes the instruments whose validity period has ended allowing this table to be used to receive right prices when delivery comes.

- Registries, containing orders rejected during the clearing session.

Contain the orders, which were not replaced during the clearing session due to lack of funds. The futures registry is transmitted in the 'fut_rejected' table of the 'FORTS_REFDATA_REPL' stream.

- Rejected during clearing orders' registries

Include information on the amount of funds in the accounts, account activity, fees, total initial and variation margin by the time of clearing. Are sent in the 'FORTS_CLR_REPL' stream.

3.2.6. Auxiliary information streams

That group includes the streams providing the following additional functions:

- Current values of variation margin

are sent in the 'FORTS_VM_REPL' in the context of client's positions.

3.3. Gateway usage specifics

3.3.1. Service replication fields

Each replicated table contains three fields of the fixed type i8 in the top, which are used for replication purpose:

- **replID** — the unique record ID. When a new record appears in the table, the record is assigned with a unique ID. Even though a table may already have a primary ID-key, the one and only ID for replication purpose is the ID contained in the field 'replID'.
- **replRev** — when there is a change made in the table such as record insert, record edit or record deletion, this record will be assigned with a new value in the field 'replRev' (maximum previous 'replRev' value + 1).
- **replAct** — flag of a deleted record. If 'replAct' contains a value other than 0, then the record has been deleted. If 'replAct' contains 0, then the record is active.

3.3.2. Commands

To send a command, you should create a publisher with parameters 'NAME = FORTS_SRV', 'category = FORTS_MSG'. If you need to receive replies to the messages sent, you should specify the flag 'CG_PUB_NEEDREPLY' within the message sending function and create a type 'p2mqreply' listener.

In case of the message delivery and handling errors, the client receives either sending message function error or the 'system error' (msgid=100) message in return.

| Field | Type | Description |
|---------|------|--------------|
| code | i4 | Return code |
| message | c255 | Message body |

Please note that the 'system error' message can be received in reply to any business-logic command.

3.3.3. Flood control

The control system of clients' application flood control is functioning in the SPECTRA trade system. It restricts client's application to send more transactions per time unit (for single login on SPECTRA) than it is stated in the connection agreement. At present moment you can acquire login on SPECTRA with 30, 60, 90, etc. (but not more than 3000) trading transactions per second. Trade operations are all transactions associated with order managing. Amount of non-trade (all the rest) operations for any type of login is limited in 1000 transactions per second.

If you exceed the limit of messages, the control system does not transmit a message into the trade system core, and sends the user a reply message with the notification of denial of service. It is msgid=99 and has the following structure:

| Field | Type | Description |
|----------------|------|-------------------------------------------------------------------------------------------------|
| queue_size | i4 | Number of messages for a single user |
| penalty_remain | i4 | Time in milliseconds after which the next message from this user will be successfully received. |
| message | c128 | Error text message |

Please pay attention to the two details:

1. The number of messages for the elapsed second is estimated while receiving *every single* message. Thus, if a user constantly sends requests with the frequency greater than it is allowed, then his messages will not be processed at all.
2. A reject message with 99 type can be sent in a reply to any user's message.

3.3.4. Latency monitoring by the client side

Trading system SPECTRA provides a possibility to automatically monitor data distribution latencies by marking a period of time between sending a message and receiving a reply message or a replication record; the time difference between two marks allows to calculate the latency. The data collected are available for further analysis by the SPECTRA centralized monitoring system. Please note that you should install the Plaza2 software and use the new messages scheme versions compatible with SPECTRA 3.8.2 and later; otherwise, there will be no possibility of usage the latency monitoring feature. The string below (can be found in the message description) points to the new message schemes:

```
LocalTimeField=<field
name>
```

Please also note that using the new message schemes with old Plaza2 binary modules will cause problems, and is strictly not recommended!

3.3.5. Cancel On Disconnect

The Trading System SPECTRA provides a client connection control feature ('Cancel On Disconnect' or 'COD'). This option allows to automatically cancel some client's orders (anonymous orders without specified expiration time) on disconnect.

To enable/disable the 'COD' option, a trading participant should apply the appropriate request to the Client Center. The 'COD' option will be enabled for the ID (p2login) belonging to the trading participant.

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:

- If the 'COD' mode is enabled for the client, the system monitors the client's activity on transaction layer. Each and every client's command or message registered by the Trading System is considered as activity, no matter whether it was complete or not.
- If the client does not send a single message, or does not reconnect to the Trading System after losing connection within the time period specified (now is 20 seconds), all they active orders are automatically cancelled.

The order cancellation conditions are as following:

- A client has not sent any transaction within the specified time limit.
- Client application has lost connection to the Router. Orders will be cancelled after reaching the specified time limit.
- Router has lost connection to the Access Server. Active orders will be cancelled after reaching the specified time limit.
- Access Server has lost connection to the Trading System or become unable to operate properly due to an error. All active orders of all clients connected to this Access Server will be cancelled after reaching the specified time limit.
- There may occur an issue when FIX server or an API clients access server connected to the Trading System via gateway becomes unable to operate properly: it loses connections to a client but does not inform the Trading System about it. 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.

The orders cancelled via the 'Cancel On Disconnect' option are marked with a special status (field 'xstatus') in the table.

If clients need to simulate their transaction activity, they should send command 'CODHeartbeat (msgid=10000)' into the Trading System at least once per at least once per 10 seconds. The command structure is as following:

| Field | Type | Details |
|------------|------|-------------------------------------------------------------|
| seq_number | i4 | Heartbeat-message number (not used in the current version). |

The command is not included into transaction fee.

The connection control service does not send replies to the Heartbeat messages, so that clients should set 0 (no reply expected) when calling the message sending function: (cg_pub_post(pub, msgptr, 0)). Any attempt to call the function 'cg_pub_post' with 'CG_PUB_NEEDREPLY' on sending a Heartbeat message will result the error 'CG_MSG_P2MQ_TIMEOUT'.

3.3.6. Replication stream sets for different login subtypes

Depending on the user login subtype (main, viewing, transactional), there are different sets of replication streams each login receives.

Replication streams set for the main login subtype:

- FORTS_COMMON_REPL - Market fundamentals
- FORTS_REFDATA_REPL - Reference and session information
- FORTS_TRADE_REPL - User's orders and trades
- FORTS_USERORDERBOOK_REPL - User orders: order book snapshot
- FORTS_CLR_REPL - Clearing information
- FORTS_DEALS_REPL - Anonymous trades
- FORTS_FEE_REPL - Exchange fees
- FORTS_FEERATE_REPL - Precise Exchange fee rates
- FORTS_BROKER_FEE_REPL - Brokerage fees
- FORTS_BROKER_FEE_PARAMS_REPL - Parameters for calculating the brokerage fee
- FORTS_INFO_REPL - Reference information
- FORTS_MM_REPL - Information on MM's obligations
- FORTS_PART_REPL - Information about funds and limits
- FORTS_POS_REPL - Information on positions
- FORTS_VM_REPL - Online variational margin stream
- FORTS_TNPENALTY_REPL - Information about Transaction fees
- FORTS_FORECASTIM_REPL - Risk forecast after limits extension

Additional replication streams received (market data source: aggregated orderbook):

- FORTS_AGGR5_REPL, FORTS_AGGR20_REPL, FORTS_AGGR50_REPL - Aggregated order-book streams

Additional replication streams received (market data source: full orderbook):

- FORTS_ORDLOG_REPL - Anonymous orders
- FORTS_ORDBOOK_REPL - Depersonalized order book snapshot

Replication streams set for viewing login subtype:

- FORTS_CLR_REPL - Clearing information
- FORTS_DEALS_REPL - Anonymous trades
- FORTS_FEE_REPL - Exchange fees
- FORTS_FEERATE_REPL - Precise Exchange fee rates
- FORTS_BROKER_FEE_REPL - Brokerage fees
- FORTS_BROKER_FEE_PARAMS_REPL - Parameters for calculating the brokerage fee
- FORTS_COMMON_REPL - Market fundamentals
- FORTS_REFDATA_REPL - Reference and session information
- FORTS_TRADE_REPL - User's orders and trades
- FORTS_MM_REPL - Information on MM's obligations

Additional replication streams received (market data source: aggregated orderbook):

- FORTS_AGGR5_REPL, FORTS_AGGR20_REPL, FORTS_AGGR50_REPL - Aggregated order-book streams

Additional replication streams received (market data source: full orderbook):

- FORTS_ORDLOG_REPL - Anonymous orders
- FORTS_USERORDERBOOK_REPL - User orders: order book snapshot
- FORTS_ORDBOOK_REPL - Depersonalized order book snapshot

Replication streams set for the transactional login subtype:

- FORTS_CLR_REPL - Clearing information
- FORTS_FEE_REPL - Exchange fees
- FORTS_FORECASTIM_REPL - Risk forecast after limits extension
- FORTS_REFDATA_REPL - Reference and session information
- FORTS_USERORDERBOOK_REPL - User orders: order book snapshot
- FORTS_TRADE_REPL - User's orders and trades
- FORTS_INFO_REPL - Reference information
- FORTS_MM_REPL - Information on MM's obligations
- FORTS_PART_REPL - Information about funds and limits
- FORTS_POS_REPL - Information on positions
- FORTS_TNPENALTY_REPL - Information about Transaction fees
- FORTS_VM_REPL - Online variational margin stream

3.3.7. Password policy

3.3.7.1. Basic provisions

Main provisions of the supported password policy:

1. Requirements for password complexity and originality:

- The password must be at least 8 characters long and meet three of the four requirements:
 - contain latin lowercase letters (from 'a' to 'z');
 - contain latin capital letters (from 'A' to 'Z');
 - contain numerals (from '0' to '9');
 - contain special or non-alphanumeric characters (@,+,_, etc).
- Password must not contain 3 or more repeating characters.
- The password must not match one of the previously used passwords.

2. Password expiration supported. The user cannot access the system after the password expires. The password expiration date is displayed in the Moex Spectra Terminal and broadcast in the gateway in the 'user' table of the FORTS_REFDATA_REPL stream.

3. The number of failed password attempts is limited. The user will be blocked for a certain time when this limit will be exhausted.

4. The password policy applies to all system logins.

Attention, please! At the gateway, an attempt to connect with an invalid password suggests the following system behavior. The gateway (router) will make a one-time attempt to connect to each of the four connections specified in the ini-file of the router, receive "BAD_CRYPTO" errors, write them to the log and will not make any more connection attempts. The administrator of user system should track these errors, correct the password, and restart the router.

3.3.7.2. Changing user password for the Trading System

A user is able to change their authentication password for the Trading System by one of the following methods:

- using utility **change_password** (described below);
- create their own application for changing authentication password (for details, see the appropriate API object description in cgate_en.pdf, section 'Password change protocol objects') and send message **ChangePassword** (for details, see here) into the Trading System.

Utility 'change_password'

The 'change_password' utility is designed to change the user's password in the trading system. The utility receives the old and new password of the user, sends them to the TS Spectra, and receives a response about the successful (or not) change of the user's password in the trading system. The utility uses a protocol that provides secure data transmission, the password and user login in clear text are not transmitted over the network.

The utility is a console application launched from the command line using the 'change_password.exe' executable file. Possible parameters to run the utility:

--app_name Application name. Optional setting;
 --local_pass Password for the local connection to the router. Optional setting;
 --host Router IP address. Optional setting, the default value is 127.0.0.1
 --port Router port. Optional setting, the default value is 4001
 --ini INI file containing logging settings. Optional setting. If no INI file specified, the data will be output to console.

Command line example:

```
C:\Moscow Exchange\SpectraCGate\bin\change_password.exe --port=4001
```

To change the password, follow these steps:

- Run the utility.
- Enter old and new password from console.
- Press 'Enter'.

The utility returns '0' if the password change succeed, and '1' in case of any error.

Please note that receiving a successful response means changing the user's password in the trading system, while the authorization of the current router connection does not change. To authorize the router with a new password, you need to change the password in the ini-file of the router and restart the router.

3.3.8. Partitioning of the matching

TS SPECTRA supports division (partitioning) of the traded instruments into groups and trading them separately on several independent orders matching modules. Wherein each matching module processes its own group of instruments. The belonging of an instrument to a group (matching) is determined by the underlying asset code (base_contract_code) of the instrument.

Broadcast trading data is also separate and independent, own replication streams are assigned to each of the matching modules. Matching affiliation of the replication streams is determined by the postfix _MATCH \$ {id} in the stream name, where \$ {id} is the ID of the matching module. For example, the 'FORTS_TRADE_REPL_MATCH1' stream is the user's orders and trades on futures instruments that are processed on MATCH1.

These streams are broadcast separately for each matching (have postfix _MATCH\${id}):

- FORTS_DEALS_REPL
- FORTS_FEE_REPL
- FORTS_FORECASTIM_REPL
- FORTS_COMMON_REPL
- FORTS_TRADE_REPL
- FORTS_ORDBOOK_REPL
- FORTS_USERORDERBOOK_REPL
- FORTS_AGGR5_REPL, FORTS_AGGR20_REPL, FORTS_AGGR50_REPL
- FORTS_ORDLOG_REPL
- FORTS_POS_REPL
- FORTS_VM_REPL

The table 'instr2matching_map' is broadcasted to determine the correspondence between the instrument and the matching on which it is processed, in 'FORTS_REFDATA_REPL' stream. The table 'instr2matching_map' has following fields:

- base_contract_id - underlying contract ID;
- matching_id - matching ID.

Binding of instruments to matchings may change when trading session changes.

New algorithm for receiving trade data

- Define 'base_contract_code' for 'isin_id' according to the tables 'fut_sess_contents' / 'opt_sess_contents'.
- Define 'base_contract_id' for 'base_contract_code' according to the tables 'fut_vcb' / 'opt_vcb'.
- Define matching ID by 'base_contract_id' in the table 'instr2matching_map'.

- Open streams with matching `_MATCHES(id)` for getting instrument trade data.

There is only one orders matching module In the TS SPECTRA version 6.3, and old replication streams (without partitioning by matchings) are left for backward compatibility. But old streams will be deleted in future versions of the system. So, we recommend that users rebuild their systems to work with new data streams. Also two new commands 'MoveOrder' (msgid=438) and 'DelOrder' (msgid=436) was added to TS version 6.3. These commands should be used to move and delete orders for futures and options in the trading system with several matchings.

3.3.9. Stream types

Types of data streams:

- **'Reliable' (R)** - The data published in such streams is relevant, reliable and not subject to change. Any change is a force majeure related to an emergency situation on the Exchange. The trading member can fully rely on data from such flows when making decisions.
- **'Almost Reliable' (AR)** - Data requires reconciliation with reports. Usually data in such streams is not subject to change, but there may be rare situations where the final values published in the reports differ from online data. For example, the settlement price can be adjusted by the CCP (this situation is provided for by regulatory documents). The trading member can rely on data from such flows, taking into account that it may be necessary to adjust the data obtained on the basis of automatic reconciliation with reports.
- **'Informational' (I)** - Data that the trading member cannot rely on as the sole source when he makes a decision. Data from such flows should be used with caution, if possible, by conducting a weighted comparison with similar data obtained in another way. An example of such data is volatility data, which is estimates and depends on the model used and calculation methodology.

The table below shows the gradation of streams by type:

| Stream name | Description | Type |
|------------------------------|----------------------------------------------|------|
| FORTS_TRADE_REPL | User's orders and trades | R |
| FORTS_ORDLOG_REPL | Anonymous orders | R |
| FORTS_DEALS_REPL | Anonymous trades | R |
| FORTS_USERORDERBOOK_REPL | User orders: order-book snapshot | R |
| FORTS_ORDBOOK_REPL | Depersonalized order-book snapshot | R |
| FORTS_REFDATA_REPL | Reference and session information | R |
| FORTS_INFO_REPL | Reference information | R |
| FORTS_FEE_REPL | Exchange fees | AR |
| FORTS_FEERATE_REPL | Precise Exchange fee rates | AR |
| FORTS_CLR_REPL | Clearing information | AR |
| FORTS_BROKER_FEE_REPL | Brokerage fees | I |
| FORTS_BROKER_FEE_PARAMS_REPL | Parameters for calculating the brokerage fee | I |
| FORTS_COMMON_REPL | Market fundamentals | I |
| FORTS_AGGR##_REPL | Aggregated order-book | I |
| FORTS_POS_REPL | Information on positions | I |
| FORTS_PART_REPL | Information about funds and limits | I |
| FORTS_MM_REPL | Information on MM's obligations | I |
| FORTS_VM_REPL | Online variational margin stream | I |
| FORTS_TNPENALTY_REPL | Information about transaction fees | I |
| FORTS_FORECASTIM_REPL | Risk forecast after limits extension | I |

3.4. Handling abnormal situations

3.4.1. Recovery on loss of connection with Exchange servers

In the standard configuration of Plaza 2 gate, there are four TCP-connections to the Exchange servers:

- Connection for sending requests and commands
- Connection for receiving the main market data such as aggregated order-books streams and the streams 'FORTS_ORDLOG_REPL', 'FORTS_DEALS_REPL', 'FORTS_TRADE_REPL' and 'FORTS_COMMON_REPL'.
- Connection for receiving auxiliary and reference streams
- Connection for receiving snapshots (at the first connection or when recovering after loss of connection)

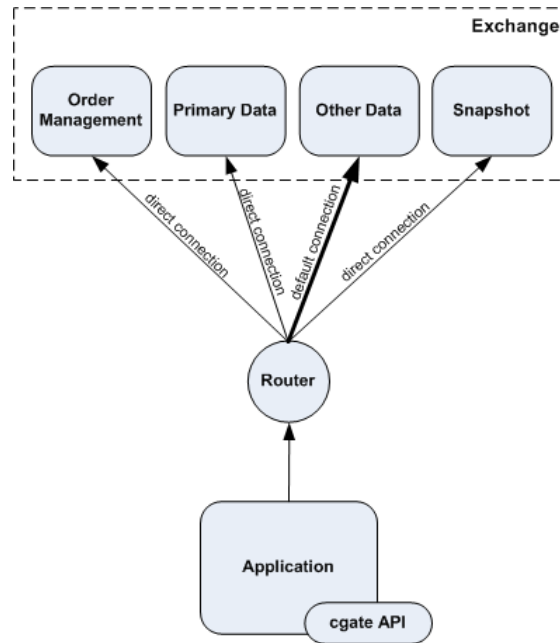


Figure 12. Connection scheme

In order to obtain stability, the trading system uses load balancing method to connect clients to the least loaded server at the moment.

3.4.1.1. Connection loss detection

P2MQRouter software handles all TCP-connections, with settings specified in the INI file where connection 'Other Data' is specified as the default outgoing connection and the other connections are specified as outgoing direct connections. This structure is used as the standard to establish connection directly to the Exchange server farm, but connection via a Brokerage Firm server may require a different structure; in this case, clients should apply to the server's owners for more details about connection.

The P2MQRouter software also handles connection recovery in case of loss of connection. After disconnecting, P2MQRouter starts attempting to reestablish the connection periodically in accordance with the specified time value, while the client software is not able to interfere with the process. P2MQRouter status then changes from 'ROUTER_CONNECTED' to 'ROUTER_RECONNECTING' by receiving the appropriate notifications from object 'connection', and this is a way for client to check whether connection is still active or not.

The CGate library acts in the following way:

- When the loss of connection to the incoming request processing gateway occurs, it is detected directly on the moment of receiving the TCP-connection error. All the 'publisher' objects concerned go to the error state.
- When the loss of connection for receiving the main market data occurs, it is detected within 30 seconds. All the 'listener' objects concerned go to the error state.

All object in error state should be released. After that, it is necessary to try to reopen them anew periodically, for example, once in a few seconds.

3.4.1.2. Recovery algorithm

In general, the connection recovery algorithm is as follows:

- After start-up, try to open connection to P2MQRouter periodically;
- When the router is reconnected to the Plaza 2 network, the object 'connection' will go to the ACTIVE state;
- Open the necessary streams. To make it faster, it is recommended to receive data starting from the last update. When opening a stream, you should use the 'repl state' value received on closing the stream; also, you can directly specify revision numbers for tables and scheme life number by using that of the last received data.
- Recover the list of active orders (see below)
- Register 'publisher' for orders and commands.

The table below contains the recommended methods for recovering data depending on the stream:

| Stream (table) name | Information type | Recovery method |
|----------------------------------|-----------------------------------------------|------------------------|
| FORTS_TRADE_REPL • orders_log | Own orders activity log (futures and options) | List of active orders: |

| Stream (table) name | Information type | Recovery method |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <ul style="list-style-type: none"> use stream 'FORTS_USERORDERBOOK_REPL' to receive snapshot, then open stream 'FORTS_TRADE_REPL' using the revision number specified in snapshot <p>Orders activity log:</p> <ul style="list-style-type: none"> open 'FORTS_TRADE_REPL' starting from the last received revision number |
| FORTS_TRADE_REPL <ul style="list-style-type: none"> multileg_orders_log | Own orders activity log (multileg orders) | <p>Orders activity log:</p> <ul style="list-style-type: none"> open 'FORTS_TRADE_REPL' starting from the last received revision number |
| FORTS_ORDLOG_REPL <ul style="list-style-type: none"> orders_log | Complete anonymous orders activity log (futures and options) | <p>List of active orders:</p> <ul style="list-style-type: none"> use stream 'FORTS_ORDRBOOK_REPL' to receive snapshot, then open stream 'FORTS_ORDLOG_REPL' using the revision number specified in the snapshot <p>Orders activity log:</p> <ul style="list-style-type: none"> open 'FORTS_ORDLOG_REPL' starting from the last received revision number |
| FORTS_ORDLOG_REPL <ul style="list-style-type: none"> multileg_orders_log | Complete anonymous orders activity log (multileg orders) | <p>Orders activity log:</p> <ul style="list-style-type: none"> open 'FORTS_ORDLOG_REPL' starting from the last received revision number |
| FORTS_DEALS_REPL <ul style="list-style-type: none"> deal multileg_deal FORTS_TRADE_REPL <ul style="list-style-type: none"> user_deal multileg_deal | Orders log (futures, options, multileg instruments) | Reopen the appropriate stream using the last received revision number or 'repl state' value received on closing the stream. |
| FORTS_COMMON_REPL | General market information (futures and options) | Reopen the stream anew |
| FORTS_AGGR##_REPL | Order books for futures and options. (### - depth of order book) | Reopen the appropriate stream anew |
| FORTS_REFDATA_REPL | Reference and session information | <p>Quick method:</p> <ul style="list-style-type: none"> Reopen the stream using the last received revision number or 'repl state' value received on closing the stream. <p>Allowable method:</p> <ul style="list-style-type: none"> Reopen the stream anew |
| FORTS_PART_REPL | Information on limits | Reopen the stream anew |
| FORTS_POS_REPL | Information on positions | Reopen the stream anew |
| FORTS_VM_REPL | Information on variation margin | Reopen the stream anew |

Upon recovery, it is very important to receive the lists of the client's current orders:

1. List of orders which are active during the recovery procedure period
2. Orders activity log during the connection loss period.

For the first case, you should receive the order-book snapshot ('FORTS_USERORDERBOOK_REPL'). The orders missed in the snapshot have been either already matched or cancelled during the connection loss period.

For the second case, you should receive your own orders activity log (the table 'orders_log' of the streams 'FORTS_TRADE_REPL', also, the table 'multileg_orders_log' of the stream 'FORTS_TRADE_REPL') covering the connection loss period. To do this, you should open the

appropriate stream using revision number of the last record actually received before the loss of connection occurred. Every order activity happened during the connection loss period will be recorded in these tables. Changing the stream state to 'ONLINE' indicates that all orders activity data have been successfully received.

Note: The recovering procedure described above can be also used for the late start connection.

3.4.1.3. General recommendations

In general case, to minimize possibility of loss of connection, the Exchange recommends to do the following:

- establish alternative connections
- obtain two client's IDs for the gateway, with the same user rights in order to have possibility to receive the same data by running two client applications simultaneously. Therefore, in case of any failure, you will be able to switch between two applications.

Alternatively, it is recommended to enable a feature in your application allowing to switch to another connection (a P2MQRouter connected to the Exchange servers using an alternative connection) in case of any failure.

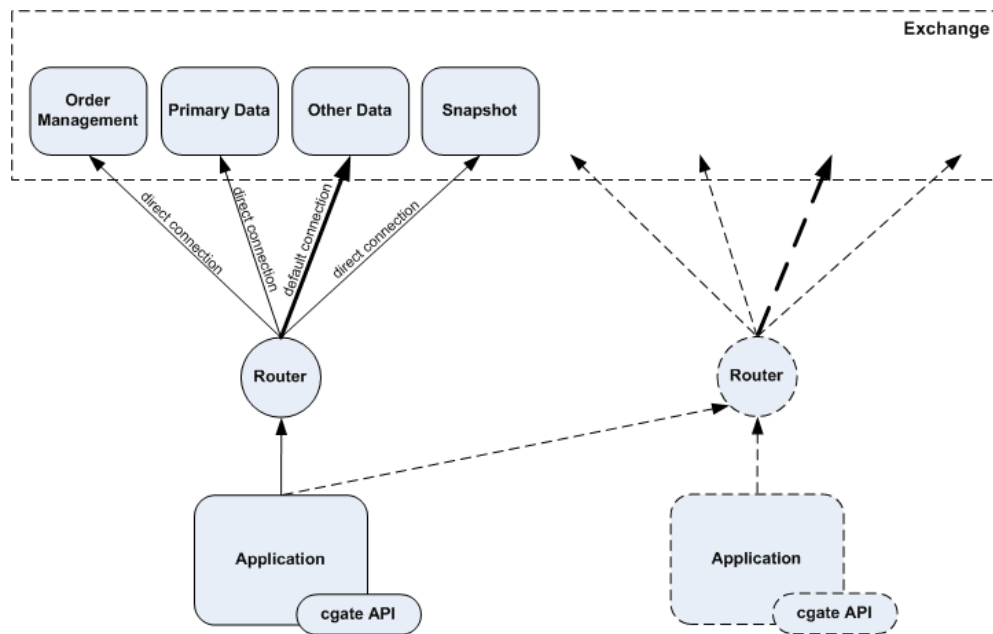


Figure 13. Channel duplication scheme

3.4.2. Recovery in case of the Exchange infrastructure failure

By the Exchange infrastructure failure we mean failures on the Exchange side caused by the Trading System kernel errors, or by errors in market data generating services. Then, as a rule, the services halt and restart.

3.4.2.1. Data cleanup by streams

In case of any routine maintenance, normal or abnormal service restarts on the Exchange side, or after reestablishing connection to a client, the publishing services send out notifications about obsolete data cleanup before sending the current snapshot to clients.

There are two types of data cleanup notifications:

- `CG_MSG_P2REPL_CLEARDELETED` - by every table, with use of revision number. The notification gives client the order to cleanup all records with the 'replRev' value smaller than the one in notification. In order to optimise data transfer, the notification may have a revision number value as 'MAX(int64)'. This means that client should cleanup all data from the specified table, as the entire table will be transferred anew.
- `CG_MSG_P2REPL_LIFENUM` - for the entire replication stream, using the new stream life number. This notification means, that data have been significantly changed since the last connection. Client should cleanup all data in all tables. All data will be transferred anew.

3.4.2.2. Possible data change in case of abnormal work of publishing services

In normal work mode, including routine works at non-trading time, when opening or reopening any replication stream except those related to history of orders and trades ('FORTS_TRADE_REPL', 'FORTS_ORDLOG_REPL' and 'FORTS_DEALS_REPL'), a client may receive both 'CG_MSG_P2REPL_CLEARDELETED' or 'CG_MSG_P2REPL_LIFENUM' notification types, and should process them correctly.

In normal work mode, for the streams related to history of orders and trades (see above), the notification `CG_MSG_P2REPL_LIFENUM` is sent only in case of system version change, after the testing-mode trades, in order to make clients cleanup the user data. The notification 'CG_MSG_P2REPL_CLEARDELETED' has the 'replRev' value for the first available order or trade at the moment.

A 'CG_MSG_P2REPL_LIFENUM' with a new stream life number during a trading session indicates a severe failure in the Trading System, so the system is to resend data on orders and trades which could be already delivered to clients.

4. Replication scheme FORTS_PUBLIC

4.1. Stream FORTS_TRADE_REPL - User's orders and trades (Type=R)

4.1.1. Data scheme

Tables:

- orders_log - Log of operations with orders
- user_deal - User trades
- heartbeat - Server times table
- sys_events - table of events

4.1.1.1. Table orders_log: Log of operations with orders

Table 1. Fields of table orders_log

| Field | Type | Description |
|--------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| public_order_id | i8 | Order ID (for iceberg order – ID of its visible part) |
| sess_id | i4 | Trading session ID |
| isin_id | i4 | Instrument unique ID |
| public_amount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for its visible part) |
| public_amount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in its visible part) |
| id_deal | i8 | Deal ID for this operation |
| xstatus | i8 | Extended order's status |
| price | d16.5 | Price |
| moment | t | Order status changing time |
| moment_ns | u8 | Order status changing time, nanoseconds since Unix epoch, UTC |
| dir | i1 | Direction |
| public_action | i1 | Type of operation with the order (for iceberg order – type of operation with its visible part) |
| deal_price | d16.5 | Price of the deal |
| client_code | c7 | Client code |
| login_from | c20 | Login of the user who has entered the order |
| comment | c20 | Trader's comment |
| ext_id | i4 | External ID number. It is added to orders, trades |
| broker_to | c7 | SPECTRA code of the company to whom the negotiated order is addressed |
| broker_to_rts | c7 | RTS code of the company to whom the negotiated order is addressed |
| broker_from_rts | c7 | RTS code of the company who has entered the order |
| date_exp | t | Order's expiration date |
| id_ord1 | i8 | ID number of the first order |
| aspref | i4 | Client ID. For orders added by SMA login - MASTER login ID. |
| id_ord | i8 | Order ID (for iceberg order – ID of the entire order). The field will be deleted in version 6.7, see value private_order_id |
| xamount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for the entire order). The field will be deleted in version 6.7, see value private_amount. |
| xamount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in the entire order). The field will be deleted in version 6.7, see value private_amount_rest. |

| Field | Type | Description |
|-----------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| variance_amount | i8 | Variance amplitude for a random addition for the pop-up part (in contracts) |
| disclose_const_amount | i8 | Number of instrument units in the pop-up part of the iceberg order |
| action | i1 | Type of operation with the order (for iceberg order – type of operation with the entire order). The field will be deleted in version 6.7, see value private_action |
| private_order_id | i8 | Order ID (for iceberg order – ID of the entire order) |
| private_amount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for the entire order) |
| private_amount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in the entire order) |
| private_action | i1 | Type of operation with the order (for iceberg order – type of operation with the entire order) |
| reason | i4 | The flag (reason) of the order submitted for the making of the settlement trade of obligations. |

Notes:

- Field xstatus is a bit mask. For the complete list of all possible values of field 'status' please refer to section Trade types, created upon exercising and expiration of futures and options.
- Field dir can take the following values:
 - 1 Buy
 - 2 Sell
- Field public_action can take the following values
 - 0 Order cancelled
 - 1 Order added
 - 2 Order is exercised in the trade
- Field id_ord1 contains the initial order ID number, i.e. the ID number which was assigned to order before the order has once been relisted
- Field 'private_action' ('action') can take the following values:
 - 0 Order cancelled
 - 1 Order added
 - 2 Order is exercised in the trade
 - 3 The order was added by appearance of a new visible part of the iceberg
- Field 'reason' can take the following values:
 - 0 Regular order
 - 4 Balancing Derivatives contracts entered into with the Non-defaulting Clearing Member without submitting orders
 - 6 Closing Derivatives contracts entered into under the cross-default procedure
 - 7 Closing Derivatives Contracts entered into upon non-fulfillment of the Margin Call
 - 8 Closing Derivatives contracts entered into in into upon non-fulfillment of the Delivery Obligation on the deliverable Derivatives contracts for precious metals.
 - 100 Other

4.1.1.2. Table user_deal: User trades

Table 2. Fields of table user_deal

| Field | Type | Description |
|---------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |

| Field | Type | Description |
|-----------------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| sess_id | i4 | Trading session ID |
| isin_id | i4 | Instrument unique ID |
| id_deal | i8 | Deal ID number |
| id_deal_multileg | i8 | Deal ID number for multileg deals |
| id_repo | i8 | Deal ID number of the other leg |
| xpos | i8 | Number of positions in the instrument in the market after the trade |
| xamount | i8 | Volume, number of units of the instrument |
| public_order_id_buy | i8 | The buyer's order ID (for iceberg order – ID of its visible part) |
| public_order_id_sell | i8 | The seller's order ID (for iceberg order – ID of its visible part) |
| price | d16.5 | Price |
| moment | t | Time when the deal was made |
| moment_ns | u8 | Time when the deal was made, nanoseconds since Unix epoch, UTC |
| nosystem | i1 | Sign of non-system deal |
| xstatus_buy | i8 | Status of the trade from the buyer's side |
| xstatus_sell | i8 | Status of the trade from the seller's side |
| ext_id_buy | i4 | External ID number from the buyer's order |
| ext_id_sell | i4 | External ID number from the seller's order |
| code_buy | c7 | Buyer's code |
| code_sell | c7 | Seller's code |
| comment_buy | c20 | Comment from the buyer's order |
| comment_sell | c20 | Comment from the seller's order |
| fee_buy | d26.2 | Fee of the buyer's deal |
| fee_sell | d26.2 | Fee of the seller's deal |
| login_buy | c20 | Login of the buyer user |
| login_sell | c20 | Login of the seller user |
| code_rts_buy | c7 | RTS code of the buyer company |
| code_rts_sell | c7 | RTS code of the seller company |
| id_ord_buy | i8 | The buyer's order ID (for iceberg order – ID of the entire order). The field will be deleted in version 6.7, see value private_order_id_buy |
| id_ord_sell | i8 | The seller's order ID (for iceberg order – ID of the entire order). The field will be deleted in version 6.7, see value private_order_id_sell |
| private_order_id_buy | i8 | The buyer's order ID (for iceberg order – ID of the entire order) |
| private_order_id_sell | i8 | The seller's order ID (for iceberg order – ID of the entire order) |
| reason_buy | i4 | The flag(reason) of the buyer's settlement trade. |
| reason_sell | i4 | The flag(reason) of the seller's settlement trade. |

Notes:

- Fields code_sell, comment_sell, ext_id_sell, login_sell, code_rts_sell, fee_sell, code_buy, comment_buy, ext_id_buy, login_buy, code_rts_buy, fee_buy, are filled with info only for "own" deals.
- Fields xstatus_sell and xstatus_buy are bit masks (for details see Trade types, created upon exercising and expiration of futures and options)
- For all other (not client-related) trades, fields 'xstatus_buy' and 'xstatus_sell' may contain flags 'eNonQuoteStatus', 'eClearingTrade', 'eAddressStatus', 'eStrategy'.
- The fields fee_buy and fee_sell contain the estimated size of the limit blocked for the trade fee. Fee size must be viewed in the FORTS_FEE_REPL stream.
- The reason_buy and reason_sell fields can take the following values:

| | |
|---|----------------------------------------------------------------------------------------------------------------|
| 0 | Regular trade |
| 4 | Balancing Derivatives contracts entered into with the Non-defaulting Clearing Member without submitting orders |
| 6 | Closing Derivatives contracts entered into under the cross-default procedure |

- 7 Closing Derivatives Contracts entered into upon non-fulfillment of the Margin Call
- 8 Closing Derivatives contracts entered into in into upon non-fulfillment of the Delivery Obligation on the deliverable Derivatives contracts for precious metals.
- 100 Other

4.1.1.3. Table heartbeat: Server times table

Records in this table are added periodically by the trading system's core. It can be used for synchronization purposes (e.g. to check whether all the trades were received at specified moment of time). The table is insert-only, no modifications or deletions occur during trading session.

Table 3. Fields of table heartbeat

| Field | Type | Description |
|-------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| server_time | t | Server date and time |

4.1.1.4. Table sys_events: table of events

Table 4. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events
 - event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system
 - event_type = 5
message = "clearing_started"
Main clearing session has started
 - event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.2. Stream FORTS_ORDLOG_REPL – anonymous orders (Type=R)

4.2.1. Data scheme

Tables:

- orders_log - Log of operations with orders
- heartbeat - Server times table
- sys_events - table of events

4.2.1.1. Table orders_log: Log of operations with orders

Table 5. Fields of table orders_log

| Field | Type | Description |
|--------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |

| Field | Type | Description |
|--------------------|-------|------------------------------------------------------------------------------------------------------------------------------|
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| public_order_id | i8 | Order ID (for iceberg order – ID of its visible part) |
| sess_id | i4 | Trading session ID |
| isin_id | i4 | Instrument unique ID |
| public_amount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for its visible part) |
| public_amount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in its visible part) |
| id_deal | i8 | Deal ID for this operation |
| xstatus | i8 | Extended order's status |
| price | d16.5 | Price |
| moment | t | Order status changing time |
| moment_ns | u8 | Order status changing time, nanoseconds since Unix epoch, UTC |
| dir | i1 | Direction |
| public_action | i1 | Type of operation with the order (for iceberg order – type of operation with its visible part) |
| deal_price | d16.5 | Price of the deal |

Notes:

- Field xstatus is a bit mask. For the complete list of all possible values of field 'status' please refer to section Trade types, created upon exercising and expiration of futures and options.
- Field dir can take the following values:
 - 1 Buy
 - 2 Sell
- Field public_action can take the following values
 - 0 Order cancelled
 - 1 Order added
 - 2 Order is exercised in the trade

4.2.1.2. Table heartbeat: Server times table

Records in this table are added periodically by the trading system's core. It can be used for synchronization purposes (e.g. to check whether all the trades were received at specified moment of time). The table is insert-only, no modifications or deletions occur during trading session.

Table 6. Fields of table heartbeat

| Field | Type | Description |
|-------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| server_time | t | Server date and time |

4.2.1.3. Table sys_events: table of events

Table 7. Fields of table sys_events

| Field | Type | Description |
|----------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |

| Field | Type | Description |
|------------|------|--------------------------|
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.3. Stream FORTS_DEALS_REPL – anonymous trades (Type=R)

4.3.1. Data scheme

Tables:

- deal - Trades
- heartbeat - Server times table
- sys_events - table of events

4.3.1.1. Table deal: Trades

Table 8. Fields of table deal

| Field | Type | Description |
|----------------------|-------|---------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Trading session ID |
| isin_id | i4 | Instrument unique ID |
| id_deal | i8 | Deal ID number |
| xpos | i8 | Number of positions in the instrument in the market after the trade |
| xamount | i8 | Volume, number of units of the instrument |
| public_order_id_buy | i8 | The buyer's order ID (for iceberg order – ID of its visible part) |
| public_order_id_sell | i8 | The seller's order ID (for iceberg order – ID of its visible part) |
| price | d16.5 | Price |
| moment | t | Time when the deal was made |
| moment_ns | u8 | Time when the deal was made, nanoseconds since Unix epoch, UTC |
| nosystem | i1 | Sign of non-system deal |
| xstatus_buy | i8 | Status of the trade from the buyer's side |
| xstatus_sell | i8 | Status of the trade from the seller's side |

Notes:

- Fields xstatus_sell and xstatus_buy are bit masks (for details see Trade types, created upon exercising and expiration of futures and options)

4.3.1.2. Table heartbeat: Server times table

Records in this table are added periodically by the trading system's core. It can be used for synchronization purposes (e.g. to check whether all the trades were received at specified moment of time). The table is insert-only, no modifications or deletions occur during trading session.

Table 9. Fields of table heartbeat

| Field | Type | Description |
|-------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| server_time | t | Server date and time |

4.3.1.3. Table sys_events: table of events**Table 10. Fields of table sys_events**

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
 message = "session_data_ready"
 All data from the clearing system have been loaded into the trading system

event_type = 5
 message = "clearing_started"
 Main clearing session has started

event_type = 6
 message = "extension_of_limits_finished"
 Limits have been extended

4.4. Stream FORTS_FEE_REPL - exchange fees and penalties (Type=AR)**4.4.1. Data scheme**

Tables:

- adjusted_fee - exchange fees
- penalty - penalties
- sys_events - table of events

4.4.1.1. Table adjusted_fee: exchange fees**Table 11. Fields of table adjusted_fee**

| Field | Type | Description |
|------------------|-------|----------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| id_deal | i8 | Deal ID number |
| moment | t | Time when the deal was made |
| moment_ns | u8 | Time when the deal was made, nanoseconds since Unix epoch, UTC |
| code_buy | c7 | Buyer's code |
| code_sell | c7 | Seller's code |
| initial_fee_buy | d26.2 | Initial fee of the buyer's deal |
| initial_fee_sell | d26.2 | Initial fee of the seller's deal |

| Field | Type | Description |
|-------------------|-------|-----------------------------------|
| adjusted_fee_buy | d26.2 | Adjusted fee of the buyer's deal |
| adjusted_fee_sell | d26.2 | Adjusted fee of the seller's deal |
| id_deal_multileg | i8 | Deal ID number for multileg deals |

4.4.1.2. Table penalty: penalties

Table 12. Fields of table penalty

| Field | Type | Description |
|------------------|-------|----------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Session number |
| id_deal | i8 | Deal ID number |
| id_deal_multileg | i8 | Deal ID number for multileg deals |
| moment | t | Time when the deal was made |
| moment_ns | u8 | Time when the deal was made, nanoseconds since Unix epoch, UTC |
| code_buy | c7 | Buyer's code |
| code_sell | c7 | Seller's code |
| penalty_buy | d26.2 | Penalty of the buyer's deal |
| penalty_sell | d26.2 | Penalty of the seller's deal |

4.4.1.3. Table sys_events: table of events

Table 13. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.5. Stream FORTS_FEERATE_REPL - Precise Exchange fee rates (Type=AR)

4.5.1. Data scheme

Tables:

- futures_rate - fee rates on futures

- sys_events - table of events

4.5.1.1. Table futures_rate: fee rates on futures

Table 14. Fields of table futures_rate

| Field | Type | Description |
|----------------------|-------|----------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| sess_id | i4 | Session number |
| exchange_fee_negdeal | d26.2 | Precise exchange fee rate for negotiated trade |
| exchange_fee | d26.2 | Precise exchange fee rate for anonymous trade |
| clearing_fee_negdeal | d26.2 | Precise clearing fee rate for negotiated trade |
| clearing_fee | d26.2 | Precise clearing fee rate for anonymous trade |
| exp_clearing_fee | d26.2 | Precise clearing fee rate for contract exercising. |

4.5.1.2. Table sys_events: table of events

Table 15. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.6. Stream FORTS_BROKER_FEE_REPL - Brokerage fees (Type=I)

4.6.1. Data scheme

Tables:

- broker_fee - brokerage fee
- sys_events - table of events

4.6.1.1. Table broker_fee: brokerage fee

Table 16. Fields of table broker_fee

| Field | Type | Description |
|--------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |

| Field | Type | Description |
|------------------|-------|----------------------------------------------------------------|
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Session number |
| id_deal | i8 | Deal ID number |
| id_deal_multileg | i8 | Deal ID number for multileg deals (**) |
| moment | t | Time when the deal was made |
| moment_ns | u8 | Time when the deal was made, nanoseconds since Unix epoch, UTC |
| code_buy | c7 | Buyer's code |
| code_sell | c7 | Seller's code |
| broker_fee_buy | d26.2 | Brokerage fee of the buyer's deal |
| broker_fee_sell | d26.2 | Brokerage fee of the seller's deal |

4.6.1.2. Table sys_events: table of events

Table 17. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.7. Stream FORTS_BROKER_FEE_PARAMS_REPL - Parameters for calculating the brokerage fee (Type=I)

4.7.1. Data scheme

Tables:

- broker_fee_params - Parameters for calculating the brokerage fee
- sys_events - table of events

4.7.1.1. Table broker_fee_params: Parameters for calculating the brokerage fee

Table 18. Fields of table 'broker_fee_params'

| Field | Type | Description |
|---------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |

| Field | Type | Description |
|-------------|-------|--------------------------------------------------------|
| sess_id | i4 | Session number |
| client_code | c7 | Client code (broker code) |
| lower_fee | d26.2 | Minimum possible amount of brokerage fee per contract |
| upper_fee | d26.2 | Maximum possible amount of brokerage fee per contract |
| multiplier | d26.2 | Multiplier to the amount of exchange and clearing fees |
| additive | d26.2 | Constant addition per contract |

Notes:

- The 'client_code' field may contain the client code or BF code. If a client code is specified, then the specified parameters are used to calculate the brokerage fee for the trades of this client. If a broker code is specified, then the parameters are used to calculate the brokerage fee for all BF clients.

- Field 'sess_id' can take the following values:

sess_id Current calculation parameters.

-1 Adding new calculation parameters. Parameters will be applied in the next trading session.

-2 Deletion of current calculation parameters. Parameters will be deleted in the next trading session.

4.7.1.2. Table sys_events: table of events

Table 19. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.8. Stream FORTS_USERORDERBOOK_REPL - User orders: order-book snapshot (Type=R)

The following data is published in the stream every 2 minutes: snapshot of active orders and a record in the 'info' table with the revision of the last processed transaction from 'orders_log', the stream's life number and the publication state of the snapshot ('publication_state' field). The 'publication_state' field is set to '0' in snapshot publication moment. After the snapshot is published, 'publication_state' field is set to '1'. The data in the 'orders' table may be inconsistent until 'publication_state' = 1.

4.8.1. Data scheme

Tables:

- orders - User current order-book
- info - Order-book snapshots information

4.8.1.1. Table orders: User current order-book

Table 20. Fields of table orders

| Field | Type | Description |
|-----------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| public_order_id | i8 | Order ID (for iceberg order – ID of its visible part) |
| sess_id | i4 | Trading session ID |
| client_code | c7 | Client code |
| moment | t | Order status changing time |
| moment_ns | u8 | Order status changing time, nanoseconds since Unix epoch, UTC |
| xstatus | i8 | Extended order's status |
| public_action | i1 | Type of operation with the order (for iceberg order – type of operation with its visible part) |
| isin_id | i4 | Instrument unique ID |
| dir | i1 | Direction |
| price | d16.5 | Price |
| public_amount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for its visible part) |
| public_amount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in its visible part) |
| comment | c20 | Trader's comment |
| ext_id | i4 | External ID number. It is added to orders, trades |
| login_from | c20 | Login of the user who has entered the order |
| broker_to | c7 | SPECTRA code of the company to whom the negotiated order is addressed |
| broker_to_rts | c7 | RTS code of the company to whom the negotiated order is addressed |
| date_exp | t | Order's expiration date |
| id_ord1 | i8 | ID number of the first order |
| broker_from_rts | c7 | RTS code of the company who has entered the order |
| aspref | i4 | Client ID. For orders added by SMA login - MASTER login ID. |
| id_ord | i8 | Order ID (for iceberg order – ID of the entire order). The field will be deleted in version 6.7, see value private_order_id |
| xamount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for the entire order). The field will be deleted in version 6.7, see value private_amount. |
| xamount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in the entire order). The field will be deleted in version 6.7, see value private_amount_rest. |
| variance_amount | i8 | Variance amplitude for a random addition for the pop-up part (in contracts) |
| disclose_const_amount | i8 | Number of instrument units in the pop-up part of the iceberg order |
| action | i1 | Type of operation with the order (for iceberg order – type of operation with the entire order). The field will be deleted in version 6.7, see value private_action |
| public_init_moment | t | Placement order time (for iceberg order – placement time of its visible part) |
| public_init_amount | i8 | The initial number of contracts in the order (for iceberg order - the initial number of contracts in its visible part) |
| init_moment | t | Placement order time (for iceberg order – placement time of the entire order). The field will be deleted in version 6.7, see value private_init_moment |
| xinit_amount | i8 | The initial number of contracts in the order (for iceberg order - the initial number of contracts in the entire order). The field will be deleted in version 6.7, see value private_init_amount |
| private_order_id | i8 | Order ID (for iceberg order – ID of the entire order) |
| private_amount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for the entire order) |

| Field | Type | Description |
|---------------------|------|----------------------------------------------------------------------------------------------------------------------------|
| private_amount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in the entire order) |
| private_action | i1 | Type of operation with the order (for iceberg order – type of operation with the entire order) |
| private_init_moment | t | Placement order time (for iceberg order – placement time of the entire order) |
| private_init_amount | i8 | The initial number of contracts in the order (for iceberg order - the initial number of contracts in the entire order) |
| reason | i4 | The flag (reason) of the order submitted for the making of the settlement trade of obligations. |

Notes:

- Field xstatus is a bit mask. For the complete list of all possible values of field 'status' please refer to section Trade types, created upon exercising and expiration of futures and options.
- Field dir can take the following values:
 - Buy
 - Sell
- Field public_action can take the following values:
 - Order cancelled
 - Order added
 - Order is exercised in the trade
- Field 'private_action' ('action') can take the following values:
 - Order cancelled
 - Order added
 - Order is exercised in the trade
 - The order was added by appearance of a new visible part of the iceberg
- Field 'reason' can take the following values:
 - Regular order
 - Balancing Derivatives contracts entered into with the Non-defaulting Clearing Member without submitting orders
 - Closing Derivatives contracts entered into under the cross-default procedure
 - Closing Derivatives Contracts entered into upon non-fulfillment of the Margin Call
 - Closing Derivatives contracts entered into in into upon non-fulfillment of the Delivery Obligation on the deliverable Derivatives contracts for precious metals.
 - Other

4.8.1.2. Table info: Order-book snapshots information

Table 21. Fields of table info

| Field | Type | Description |
|-------------------|------|----------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| infoID | i8 | Unique key |
| logRev | i8 | Last processed revision at the time of snapshot creation |
| lifeNum | i4 | Stream life number |
| moment | t | Snapshot time |
| publication_state | i1 | State of the snapshot publication |

Notes:

- Field `publication_state` can take the following values:

0 in progress

1 done

4.9. Stream FORTS_ORDBOOK_REPL - Depersonalized order-book snapshot (Type=R)

The following data is published in the stream every 2 minutes: snapshot of active orders and a record in the 'info' table with the revision of the last processed transaction from 'orders_log', the stream's life number and the publication state of the snapshot ('publication_state' field). The 'publication_state' field is set to '0' in snapshot publication moment. After the snapshot is published, 'publication_state' field is set to '1'. The data in the 'orders' table may be inconsistent until 'publication_state' = 1.

4.9.1. Data scheme

Tables:

- orders - Current order-book
- info - Order-book snapshots information

4.9.1.1. Table orders: Current order-book

Table 22. Fields of table orders

| Field | Type | Description |
|--------------------|-------|------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| public_order_id | i8 | Order ID (for iceberg order – ID of its visible part) |
| sess_id | i4 | Trading session ID |
| moment | t | Order status changing time |
| moment_ns | u8 | Order status changing time, nanoseconds since Unix epoch, UTC |
| xstatus | i8 | Extended order's status |
| public_action | i1 | Type of operation with the order (for iceberg order – type of operation with its visible part) |
| isin_id | i4 | Instrument unique ID |
| dir | i1 | Direction |
| price | d16.5 | Price |
| public_amount | i8 | The number of contracts in the operation (for iceberg order - the number of contracts in the operation for its visible part) |
| public_amount_rest | i8 | The remaining number of contracts in the order (for iceberg order – the remaining number of contracts in its visible part) |
| public_init_moment | t | Placement order time (for iceberg order – placement time of its visible part) |
| public_init_amount | i8 | The initial number of contracts in the order (for iceberg order - the initial number of contracts in its visible part) |

Notes:

- Field `xstatus` is a bit mask. For the complete list of all possible values of field 'status' please refer to section Trade types, created upon exercising and expiration of futures and options.
- Field `dir` can take the following values:
 - Buy
 - Sell
- Field `public_action` can take the following values
 - Order added
 - Order is exercised in the trade

4.9.1.2. Table info: Order-book snapshots information

Table 23. Fields of table info

| Field | Type | Description |
|-------------------|------|----------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| infoID | i8 | Unique key |
| logRev | i8 | Last processed revision at the time of snapshot creation |
| lifeNum | i4 | Stream life number |
| moment | t | Snapshot time |
| publication_state | i1 | State of the snapshot publication |

Notes:

- Field publication_state can take the following values:

0 in progress

1 done

4.10. Stream FORTS_COMMON_REPL - Market fundamentals (Type=I)

4.10.1. Data scheme

Tables:

- common - Market fundamentals

4.10.1.1. Table common: Market fundamentals

The table contains market fundamentals data (best buy/sell orders, opening/closing price values, etc).

Table 24. Fields of table common

| Field | Type | Description |
|---------------------|-------|--------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Trading session ID |
| isin_id | i4 | Instrument unique ID |
| best_buy | d16.5 | Best bid |
| xamount_buy | i8 | Size of the best bid |
| orders_buy_qty | i4 | Number of bid orders |
| xorders_buy_amount | i8 | Total number of contracts in bid |
| best_sell | d16.5 | Best offer |
| xamount_sell | i8 | Size of the best offer |
| orders_sell_qty | i4 | Number of offer orders |
| xorders_sell_amount | i8 | Total number of contracts in offer |
| open_price | d16.5 | Opening price |
| close_price | d16.5 | Closing price |
| price | d16.5 | Price of the last trade |
| trend | d16.5 | Price trend (difference between the prices of the last two trades) |
| xamount | i8 | Size of the last trade |
| deal_time | t | Date and time of the last trade |
| deal_time_ns | u8 | Date and time of the last trade, nanoseconds since Unix epoch, UTC |
| min_price | d16.5 | The low price |
| max_price | d16.5 | The high price |

| Field | Type | Description |
|----------------------------|-------|-------------------------------------------------------------------------------------|
| avr_price | d16.5 | Average weighted price |
| xcontr_count | i8 | Total number of contracts in the trades |
| capital | d26.2 | Total volume of trades in Russian rubles |
| total_premium_volume | d26.2 | Total volume of option premium |
| deal_count | i4 | Number of trades |
| settlement_price_open | d16.5 | Settlement price of the previous session. |
| xpos | i8 | Current open interest |
| mod_time | t | Date and time of changing the entry in the table |
| mod_time_ns | u8 | Date and time of changing the entry in the table, nanoseconds since Unix epoch, UTC |
| market_price | d16.5 | Current market price. |
| best_buy_native | d16.5 | Best bid |
| xamount_buy_native | i8 | Size of the best bid |
| xorders_buy_amount_native | i8 | Total number of contracts in bid |
| best_sell_native | d16.5 | Best offer |
| xamount_sell_native | i8 | Size of the best offer |
| xorders_sell_amount_native | i8 | Total number of contracts in offer |
| local_time | t | Time stamp for monitoring purposes |
| price_assigned_by_admin | i1 | Attribute of manual current market price setting by trades Administrator. |

Notes:

- Field open_price contains the price of the first transaction in the current session, and if not, then 0.
- Field close_price contains a price value of the last trade in the appropriate trading session. Before the trading session closes, the field contains 0. After the session closes (7 PM till 10 AM), the field close_price contains a price value of the last trade, or 0, if there were no trades during the last trading session.
- Field 'price_assigned_by_admin' can take the following values:
 - The value of the current market price in the 'market_price' field is set by the Trading Administrator.
 - The value of the current market price in the 'market_price' field is calculated by the system.

4.11. Aggregated order-book streams (Type=I)

There are several streams of aggregated order-books with different depths.

- FORTS_AGGR50_REPL – with a depth of 50 price levels
- FORTS_AGGR20_REPL – with a depth of 20 price levels
- FORTS_AGGR5_REPL – with a depth of 5 price levels

The ability to receive particular stream depends on user account rights.

4.11.1. Data scheme

Tables:

- orders_aggr - Aggregated order-books

4.11.1.1. Table orders_aggr: Aggregated order-books

Aggregated order-books are formed by summing up volumes of active orders with the same instrument, price and direction.

Table 25. Fields of table orders_aggr

| Field | Type | Description |
|---------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |

| Field | Type | Description |
|--------------|-------|---------------------------------------------------------------------|
| price | d16.5 | Price level |
| volume | i8 | Volume |
| moment | t | Moment of the last record update |
| moment_ns | u8 | Moment of the last record update, nanoseconds since Unix epoch, UTC |
| dir | i1 | Direction |
| synth_volume | i8 | The volume of synthetic liquidity (**) |

Note:

- The order-book for an instrument may contain records with zero values. This means that the number of orders for the instrument (price levels) is not enough to fill the entire fixed depth of the order-book. Such records should be ignored. Records with zeros can be updated with values with a new price level, when new orders for the instrument appearance in the system.
- The records in the order-book for an instrument can be updated (change price/volume/dir). This means that the previous price level "out" the order-book, and the new one "come in" the order-book.
- Zeroing (volume = 0) of an existing record in the order-book means that the price level has "out" the order-book (for example, the only order that formed the price level was deleted), and there are no other hidden price levels (orders) for the instrument in the system.
- The 'moment' (moment_ns) field value in the table is not monotonically increasing. As 'replRev' increases, records with an earlier value of the 'moment' field may appear in the stream of aggregated order-books. This behavior of the system is expected and can occur in different situations, when the previously formed price level was hidden for some reason, but then began to be displayed. The 'moment' field contains the time of the event that led to the formation of the price level (adding, canceling, exercising an order). Example of similar system behavior:
 - The simplest case: in the streams of the aggregated order-books, a price-limited number of liquidity levels is displayed. For example, in FORTS_AGGR20_REPL the top 20 price levels is shown only. A hidden, but already formed level with a price outside the displayed range may appear if one of the displayed price levels "disappeared" (for example, the only order that formed the visible price level was deleted).

4.12. Stream FORTS_POS_REPL - information on positions (Type=I)

4.12.1. Data scheme

Tables:

- position - Client and BRokerage Firm positions
- position_sa - Settlement Account positions
- sys_events - table of events

4.12.1.1. Table position: Client positions

The table contains information on clients and BF positions.

Table 26. Fields of table position

| Field | Type | Description |
|----------------|-------|-----------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client code |
| isin_id | i4 | Instrument's unique ID |
| xpos | i8 | Current position |
| xbuys_qty | i8 | Number of contracts bought during the session |
| xsells_qty | i8 | Number of contracts sold during the session |
| xopen_qty | i8 | Number of positions at the start of the session |
| waprice | d16.5 | Volume-weighted average price |
| net_volume_rur | d26.2 | Nett volume per trading session, in Rubles. Positive value indicates credited funds, negative value indicates debited funds |
| last_deal_id | i8 | ID of the last deal |
| account_type | i1 | <ul style="list-style-type: none"> • 1 - BF's account • 2 - client's account |

4.12.1.2. Table position_sa: Settlement Account positions

The table contains information on Settlement Account positions.

Table 27. Fields of table position_sa

| Field | Type | Description |
|----------------|-------|-----------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c12 | Settlement Account code |
| isin_id | i4 | Instrument's unique ID |
| xpos | i8 | Current position |
| xbuys_qty | i8 | Number of contracts bought during the session |
| xsells_qty | i8 | Number of contracts sold during the session |
| xopen_qty | i8 | Number of positions at the start of the session |
| waprice | d16.5 | Volume-weighted average price |
| net_volume_rur | d26.2 | Nett volume per trading session, in Rubles. Positive value indicates credited funds, negative value indicates debited funds |
| last_deal_id | i8 | ID of the last deal |

4.12.1.3. Table sys_events: table of events

Table 28. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.13. Stream FORTS_PART_REPL - information about funds and limits (Type=I)

4.13.1. Data scheme

Tables:

- part - Funds and limits of clients and brokerage firms
- part_sa - Funds and limits for Settlement Account
- sys_events - table of events

4.13.1.1. Table part: Funds and limits of clients and brokerage firms

The table contains information about funds, limits, and settings for automatic limit changes for clients and brokerage firms.

Table 29. Fields of table part

| Field | Type | Description |
|----------------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client or brokerage code |
| money_free | d26.2 | Amount of free cash, available for opening positions. (money_free=money_amount + vm_intercl – money_blocked – vm_reserve – fee – broker_fee) |
| money_blocked | d26.2 | Assets pledged as initial margin. |
| vm_reserve | d26.2 | Variation margin on closed positions, and FX risk. |
| fee | d26.2 | Debited fee |
| limits_set | i1 | Flag of set limits: 1 - limit is set (checked), 0 - limit is not set (not checked) |
| money_old | d26.2 | Total amount of funds at the end of the previous session |
| money_amount | d26.2 | Total amount of funds |
| vm_intercl | d26.2 | Variation margin debited or credited during the intraday clearing (**) |
| is_auto_update_limit | i1 | Flag of automatic adjustment of the limit by the amount of income during downloading after clearing: 0-no, 1-adjust. |
| broker_fee | d26.2 | Assets blocked as brokerage fees. |
| penalty | d26.2 | Penalty for settlement trades made into during the procedure of forced closing of positions of the Defaulting Clearing Member |

4.13.1.2. Table part_sa: Funds and limits for Settlement Account**Table 30. Fields of table part_sa**

| Field | Type | Description |
|--------------------|-------|---------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| settlement_account | c12 | Settlement Account |
| money_old | d26.2 | Total amount of funds at the end of the previous session |
| money_amount | d26.2 | Total amount of funds |
| money_free | d26.2 | Amount of free cash, available for opening positions. (money_free=money_amount + vm_intercl – money_blocked – vm_reserve – fee) |
| money_blocked | d26.2 | Assets pledged as initial margin. |
| vm_reserve | d26.2 | Variation margin on closed positions, and FX risk. |
| vm_intercl | d26.2 | Variation margin withdrawn or deposited during the intraday clearing session (**) |
| fee | d26.2 | Debited fee |
| blocked_tax | d26.2 | Assets blocked for tax payments. |

4.13.1.3. Table sys_events: table of events**Table 31. Fields of table sys_events**

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_type | i4 | Type of the event |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| message | c64 | Description of the event |

Notes:

- Possible types of events

```
event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system
```

```
event_type = 5
message = "clearing_started"
Main clearing session has started
```

```
event_type = 6
message = "extension_of_limits_finished"
Limits have been extended
```

4.14. Stream FORTS_PROHIBITION_REPL - Prohibitions (Type=R)

4.14.1. Data scheme

Tables:

- prohibition - Prohibitions

4.14.1.1. Table prohibition: Prohibitions

Table 32. Fields of table prohibition

| Field | Type | Description |
|--------------------|------|----------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| xprohibition_id | i8 | Number of prohibition |
| client_code | c7 | Client code |
| initiator | i4 | Prohibition originator |
| section | c50 | Section |
| base_contract_code | c25 | Underlying asset code. |
| isin_id | i4 | Instrument unique ID |
| priority | i4 | Priority of prohibition |
| group_mask | i8 | Bitmask of groups for which there is a prohibition |
| type | i4 | Type of prohibition |
| is_legacy | i4 | Prohibition originator type |

Notes:

- Field 'initiator' - Initiator of the prohibition:
 - 0 BF;
 - 1 CF Chief trader;
 - 2 CC Administrator;
 - 3 TS Administrator.
- Field 'type' - Prohibition type
 - 0 No prohibitions (when cancelling a previous prohibition with lower priority, otherwise simply delete the line);
 - 1 prohibited to open positions;
 - 2 prohibited to perform all trading operations;
 - 3 prohibited to open sell positions;
 - 8 BF prohibition to add orders for exercising;
 - 16 Chief Trader prohibition to add orders for exercising;
- Field 'group_mask' - Instrument type bitmask:
 - 0x40000000 futures

0x80000000 options

- Field 'priority' - From high to low

| | |
|-------------------------|---|
| Client code, instrument | 9 |
| Client code, UA | 8 |
| Client code, all UAs | 7 |
| BF code, instrument | 6 |
| BF code, UA | 5 |
| BF code, all UAs | 4 |
| CF code, instrument | 3 |
| CF code, UA | 2 |
| CF code, all UAs | 1 |

- Field 'section' - Section name:

- 1 Securities
- 2 Commodities
- 3 FX

- Field 'is_legacy' - Prohibition originator type:

- 0 indicates the prohibition set by the Trading Administrator/Clearing Administrator; these prohibitions cannot be changed by traders.
- 1 indicates the prohibition set by a trader; these prohibitions can be changed by traders.

4.15. Stream FORTS_REFDATA_REPL - Reference and session information (Type=R)

4.15.1. Data scheme

Tables:

- fut_sess_contents - Traded instruments directory (futures)
- fut_vcb - Traded assets directory (futures)
- fut_instruments - Instruments dictionary
- fut_bond_registry - Spot asset parameters directory
- dealer - Companies directory
- sys_messages - Trading system messages
- prohibition - Prohibitions
- fut_rejected_orders - Register of orders rejected during the clearing (futures)
- fut_bond_nkd - Accrued interest as of the bond futures contract expiration date
- fut_bond_nominal - Payment of bonds' face value
- fut_bond_isin - Guide on bond instruments
- user - System users
- investor - Clients directory
- fut_margin_type - Type of margining
- fut_settlement_account - Settlement Account
- session - Information about a trading session
- sma_master - SMA login binding to MASTER login
- sma_pre_trade_check - SMA login pre-trade verification settings.

- clearing_members - Clearing Members
- instr2matching_map - Instrument binding to Matching ID
- sys_events - Table of events

4.15.1.1. Table fut_sess_contents: Traded instruments directory (futures)

The table contains dictionary of instruments which are traded in specified trading session.

Table 33. Fields of table fut_sess_contents

| Field | Type | Description |
|--------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Trading session ID |
| isin_id | i4 | Instrument unique ID |
| short_isin | c25 | Short symbol code of the instrument for information systems |
| isin | c25 | Symbol code of the instrument |
| name | c75 | Instrument name |
| inst_term | i4 | Shift from spot instruments |
| base_contract_code | c25 | Underlying asset code. |
| limit_up | d16.5 | Upper price limit |
| limit_down | d16.5 | Lower price limit |
| settlement_price_open | d16.5 | Settlement price at the start of the session. |
| buy_deposit | d16.2 | Collateral of the buyer |
| sell_deposit | d16.2 | Collateral of the seller |
| roundto | i4 | Number of decimal places after the decimal point for the price |
| min_step | d16.5 | Minimum price increment |
| lot_volume | i4 | Lot, i.e. number of units of the underlying asset in the instrument |
| step_price | d16.5 | Value of the minimum price increment |
| last_trade_date | t | Expiration date. |
| is_spread | i1 | Flag of the futures contract's being part of an intermonth spread 1 – spread; 0 – no spread. (**) |
| d_exp_start | t | Opening date of instrument exercise |
| is_percent | i1 | Flag of futures contract. 1 – interest rate futures, 0 – common futures, 2 - weather and electricity futures, 3 - Eurobonds futures, 4 - RUONIA (**) |
| percent_rate | d6.2 | Variation margin rate for interest rate futures (**) |
| settlement_price | d16.5 | Settlement price after the last clearing session. |
| signs | i4 | Flags field |
| is_trade_evening | i1 | Flag of being traded during the additional trading session (evening/morning) (**) |
| ticker | i4 | Unique ID number of the primary RTS standard instruments |
| state | i4 | State of trading in the instrument |
| multileg_type | i4 | Type of multileg instrument (**) |
| legs_qty | i4 | Number of instruments for multileg instrument (**) |
| step_price_clr | d16.5 | Value of the minimum increment for the clearing session |
| step_price_interclr | d16.5 | Value of the minimum increment for the intraday clearing session (**) |
| step_price_curr | d16.5 | Value of the minimum increment in USD |
| pctyield_coeff | d16.5 | Coef. for yield calculation on percent rates futures (**) |
| pctyield_total | d16.5 | Sum of rates for yield calculation on percent rates futures (**) |
| d_exp_end | t | Closing date of instrument exercise |
| enforce_ims_half_netting | i1 | Flag - consider the risks of intermonth spread according to the "half-netto" rule: "1" - yes; "0" - no. |

Notes:

- Trading session state has priority over instrument state. That is, if a session is in "suspended" or "finished" state, then all instruments can't be traded regardless their states.
- Field state can take the following values:
 - 0 Session for this instrument is scheduled. One can cancel orders for this instrument
 - 1 Session for this instrument is running. One can both add and cancel orders for this instrument
 - 2 Trading in all instruments has been suspended. One can cancel orders for each instrument.
 - 3 Session for this instrument has been closed compulsorily. Orders can be neither added nor cancelled
 - 4 Session for this instrument has been completed because the time is up. Orders can be neither added nor cancelled
 - 5 Trading in this instrument has been suspended. One can cancel orders for this instrument
- Field signs is a bit mask and defines the following values:
 - 0x10 Sign of anonymous trading
 - 0x20 Sign of non-anonymous trading
- Field multileg_type can take the following values:
 - 0 Ordinary instrument, not the multileg one
 - 3 The instrument is calendar futures spread

4.15.1.2. Table fut_vcb: Traded assets directory (futures)

The table contains directory of base contracts for instruments.

Table 34. Fields of table fut_vcb

| Field | Type | Description |
|--------------------|------|------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| base_contract_code | c25 | Underlying asset code. |
| name | c75 | Name |
| exec_type | c1 | Settlement type |
| curr | c3 | Payment currency |
| trade_scheme | c1 | Trading mode |
| section | c50 | Market section. 'Securities', 'Commodities', 'Money' |
| rate_id | i4 | Rate ID (**) |
| base_contract_id | i4 | Underlying contract ID |
| signs | i4 | Flags field |
| negative_prices | i1 | Sign of restriction of negative prices. |
| option_model | i1 | Options pricing model. (**) |

Notes:

- Field exec_type can take the following values:
 - D Settlement
 - I Index
- Field trade_scheme can take the following values:
 - F With 100% collateral
- Field signs is a bit mask and defines the following values:
 - 0x2 Foreign instrument: 0 - not foreign; 1 - foreign
- Field negative_prices can take the following values:

0 Futures prices, price limits and options strikes are limited to be positive only.

1 Futures prices and options strikes are not limited.

4.15.1.3. Table fut_instruments: Instruments dictionary

Table 35. Fields of table fut_instruments

| Field | Type | Description |
|--------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| short_isin | c25 | Short symbol code of the instrument for information systems |
| isin | c25 | Symbol code of the instrument |
| name | c75 | Instrument name |
| inst_term | i4 | Shift from RTS standard instruments |
| base_contract_code | c25 | Underlying asset code. |
| settlement_price_open | d16.5 | Settlement price at the start of the session. |
| roundto | i4 | Number of decimal places after the decimal point for the price |
| min_step | d16.5 | Minimum price increment |
| lot_volume | i4 | Lot, i.e. number of units of the underlying asset in the instrument |
| step_price | d16.5 | Value of the minimum price increment |
| last_trade_date | t | Expiration date. |
| is_spread | i1 | Flag of the futures contract's being part of an intermonth spread 1 – spread; 0 – no spread. (**) |
| d_exp_start | t | Start date of instrument exercise. |
| is_percent | i1 | Flag of futures contract. 1 – interest rate futures, 0 – common futures, 2 - weather and electricity futures, 3 - Eurobonds futures, 4 - RUONIA (**) |
| percent_rate | d6.2 | Variation margin rate for interest rate futures (**) |
| settlement_price | d16.5 | Settlement price after the last clearing session. |
| signs | i4 | Flags field |
| multileg_type | i4 | Type of multileg instrument (**) |
| legs_qty | i4 | Number of instruments for multileg instrument (**) |
| step_price_clr | d16.5 | Value of the minimum increment for the clearing session |
| step_price_interclr | d16.5 | Value of the minimum increment for the intraday clearing session (**) |
| step_price_curr | d16.5 | Value of the minimum increment in USD |
| pctyield_coeff | d16.5 | Coef. for yield calculation on percent rates futures (**) |
| pctyield_total | d16.5 | Sum of rates for yield calculation on percent rates futures (**) |
| series_type | c1 | Flag of dated option. D-daily, W-weekly, M-monthly. (**) |
| enforce_ims_half_netting | i1 | Flag - consider the risks of intermonth spread according to the "half-netto" rule: "1" - yes; "0" - no. |

Notes:

- Field roundto. For this field, the number of decimal places in its value may differ for expiration technical trades. The number of decimal places for expiration price value is determined according to contract specification.

4.15.1.4. Table fut_bond_registry: Spot asset parameters directory

Table 36. Fields of table fut_bond_registry

| Field | Type | Description |
|---------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| bond_id | i4 | ID of the bond |

| Field | Type | Description |
|--------------|-------|---------------------------|
| small_name | c25 | Trading code of the bond |
| short_isin | c25 | Bonds issue |
| name | c75 | Bond's name |
| date_redempt | t | Bond's maturity date |
| nominal | d16.5 | Bond's face value |
| bond_type | i4 | Type: share/bond/currency |
| year_base | i2 | Day-count basis |

Notes:

- Field bond_type is a bit mask and defines the following values:

| | |
|----------|--------------------------------------|
| 0 | not set |
| 0x1 | Share |
| 0x2 | Bond (not amortized, actual formula) |
| 0x4 | Amortized bond |
| 0x8 | Bond, virtual American formula |
| 0x10 | Bond, virtual European formula |
| 0x800000 | Currency |

4.15.1.5. Table dealer: Companies directory

Table 37. Fields of table dealer

| Field | Type | Description |
|----------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client code |
| name | c200 | Company name |
| rts_code | c50 | RTS code of the company |
| signs | i4 | Lock mode. 4 - locked by the Trading System Administrator. 8 - locked by Clearing Firm's Chief Trader. |
| status | i4 | Sign of segregated account |
| transfer_code | c7 | Account code for position transfer |
| exp_weight | d3.2 | Expiration scenario weight for BF, in total collateral. Applied during the evening clearing session. |
| num_clr_2delivery | i4 | Number of clearing sessions before expiration to start BF expiration scenarios calculation. Applied during the evening clearing session. |
| margin_type | i1 | Margin type, according to BF's sections, applied during the evening clearing session: <ul style="list-style-type: none"> 3 - half nett 4 - nett |
| calendar_spread_margin_type | i1 | Margin type for calendar spreads, for BF portfolio, applied during the evening clearing session: <ul style="list-style-type: none"> 3 - half nett 4 - nett (**) |
| num_clr_2delivery_client_default | i4 | Number of clearing sessions before expiration to start clients expiration scenarios calculation (default value). Applied during the evening clearing session. |
| exp_weight_client_default | d3.2 | Expiration scenario weight for clients, in total collateral (default value). Applied during the evening clearing session. |

| Field | Type | Description |
|-----------------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coeff_im | d16.5 | Total collateral ratio value, for BF. Applied during the evening clearing session. |
| limits_set | i1 | Verify collateral sufficiency, for BF, upon adding orders: <ul style="list-style-type: none"> • 1 - Verify • 0 - Do not verify |
| no_fut_discount | i1 | Discount on futures for BF portfolio, applied during the evening clearing session: <ul style="list-style-type: none"> • 1 - Discount prohibited • 0 - Discount allowed |
| no_fut_discount_client_default | i1 | Discount on futures for BF's clients, default value: <ul style="list-style-type: none"> • 1 - Discount prohibited • 0 - Discount allowed Applied during the evening clearing session. |
| firm_id | c12 | Trading Member's code for Derivatives Market (**) |
| tm_name | c200 | Trading Member's name |
| short_option_minimum_charge_ratio | d5.3 | Individual coefficient of SOMC scenario weight. (**) |
| ics_margin_type | i1 | Margin type for cross-contract spreads: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett (**) |
| order_allowed_in_morning_session | i1 | Access to trading during the morning trading session. (**) |

Notes:

- Status field is a bit mask:
 - 0x01 - Brokerage Firm (Trust Management type)
 - 0x02 - Segregated Brokerage Firm
 - 0x100 - BF for a client - legal entity
 - 0x200 - BF for non-resident client
 - 0x20000 – Own Brokerage Firm
 - 0x40000 – Client Brokerage Firm
 - 0x80000 - Special Brokerage Firm
Other bits contain technical information
- Field order_allowed_in_morning_session can take the following values:
 - 0 Access to trading during the morning trading session is limited. Trading operations are prohibited, except for orders cancellation operations.
 - 1 Access to trading during the morning trading session is allowed.

4.15.1.6. Table sys_messages: Trading system messages

Table 38. Fields of table sys_messages

| Field | Type | Description |
|---------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| msg_id | i4 | Unique message ID |
| moment | t | Message date and time |

| Field | Type | Description |
|--------------|-------|------------------|
| lang_code | c8 | Message language |
| urgency | i1 | Urgency |
| status | i1 | Message status |
| text | c255 | Short text |
| message_body | c4000 | Full text |

4.15.1.7. Table prohibition: Prohibitions

Table 39. Fields of table prohibition

| Field | Type | Description |
|--------------------|------|----------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| prohib_id | i4 | Number of prohibition |
| xprohibition_id | i8 | Number of prohibition |
| client_code | c7 | Client code |
| initiator | i4 | Prohibition originator |
| section | c50 | Section |
| base_contract_code | c25 | Underlying asset code. |
| isin_id | i4 | Instrument unique ID |
| priority | i4 | Priority of prohibition |
| group_mask | i8 | Bitmask of groups for which there is a prohibition |
| type | i4 | Type of prohibition |
| is_legacy | i4 | Prohibition originator type |

Notes:

- Field 'initiator' - Initiator of the prohibition:
 - 0 BF;
 - 1 CF Chief trader;
 - 2 CC Administrator;
 - 3 TS Administrator.
- Field 'type' - Prohibition type
 - 0 No prohibitions (when cancelling a previous prohibition with lower priority, otherwise simply delete the line);
 - 1 prohibited to open positions;
 - 2 prohibited to perform all trading operations;
 - 3 prohibited to open sell positions;
 - 8 BF prohibition to add orders for exercising;
 - 16 Chief Trader prohibition to add orders for exercising;
- Field 'group_mask' - Instrument type bitmask:
 - 0x40000000 futures
 - 0x80000000 options
- Field 'priority' - From high to low
 - Client code, instrument 9
 - Client code, UA 8
 - Client code, all UAs 7

| | |
|---------------------|---|
| BF code, instrument | 6 |
| BF code, UA | 5 |
| BF code, all UAs | 4 |
| CF code, instrument | 3 |
| CF code, UA | 2 |
| CF code, all UAs | 1 |

• Field 'section' - Section name:

- 1 Securities
- 2 Commodities
- 3 FX

• Field 'is_legacy' - Prohibition originator type:

- 0 indicates the prohibition set by the Trading Administrator/Clearing Administrator; these prohibitions cannot be changed by traders.
- 1 indicates the prohibition set by a trader; these prohibitions can be changed by traders.

4.15.1.8. Table fut_rejected_orders: Register of orders rejected during the clearing (futures)

Table 40. Fields of table fut_rejected_orders

| Field | Type | Description |
|---------------|-------|--------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| order_id | i8 | Order ID number |
| sess_id | i4 | Trading session ID |
| moment | t | Order status changing time |
| isin_id | i4 | Instrument unique ID |
| client_code | c7 | Client code |
| dir | i1 | Direction |
| xamount | i8 | Volume, in units of the instrument |
| price | d16.5 | Price |
| date_exp | t | Order's expiration date |
| id_ord1 | i8 | ID number of the first order |
| moment_reject | t | Time when the order was rejected |
| ret_code | i4 | Return code of the re-entering procedure |
| ret_message | c255 | Text of the message containing the reason for rejection of the order when it is re-entered |
| comment | c20 | Trader's comment |
| login_from | c20 | Login of the user who has entered the order |
| ext_id | i4 | External ID number. It is added to orders, trades |

4.15.1.9. Table fut_bond_nkd: Accrued interest as of the bond futures contract expiration date

Table 41. Fields of table fut_bond_nkd

| Field | Type | Description |
|---------|-------|------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| bond_id | i4 | ID of the bond |
| date | t | Coupon payment date |
| nkd | d16.7 | Accrued interest as of the coupon payment date |

| Field | Type | Description |
|----------|------|----------------------------------------------------------------------------------------------------------------------------------|
| is_cupon | i1 | Flags: 0 - accrued interest as of the bond futures contract settlement date, 2 - accrued interest as of the bond settlement date |

4.15.1.10. Table fut_bond_nominal: Payment of bonds' face value

Table 42. Fields of table fut_bond_nominal

| Field | Type | Description |
|----------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| bond_id | i4 | ID of the bond |
| date | t | Coupon payment date |
| nominal | d16.5 | payment of bonds' face value |
| face_value | d16.5 | Payment of bonds' rest face value |
| coupon_nominal | d8.5 | Coupon value in % of face value |
| is_nominal | i1 | Type of record in the table: 0 - Remaining face value as of bond futures contract expiration date, 2 - Remaining face value as of bond settlement date |

4.15.1.11. Table fut_bond_isin: Guide on bond instruments

Table 43. Fields of table fut_bond_isin

| Field | Type | Description |
|------------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| bond_id | i4 | ID of the bond |
| coeff_conversion | d5.4 | Conversion ratio |

4.15.1.12. Table user: System users

Table 44. Fields of table user

| Field | Type | Description |
|----------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| login | c20 | Trading participant's login |
| start_date | t | Login start time |
| end_date | t | Login end time |
| client_code | i4 | 7-symbol client code |
| operation_mask | i4 | Bitmask: <ul style="list-style-type: none"> • 2 - Limit open positions for BF • 8 - Limit BF on funds transfer. The setting is available to Clearing Firm operator or Trading Administrator only. • 16 - Money back. • 32 - Limit client positions. • 128 - Client restrictions. • 1024 - Orders-related restrictions for SMA logins. |
| lang | i2 | Message language code |
| sma_flags | i4 | Bitmask (see Notes below): |

| Field | Type | Description |
|--------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <ul style="list-style-type: none"> 1st bit - Cancel on Disconnect 2nd bit - Cancel on DropCopy Disconnect 3rd bit - SMA login. |
| sma_status | i4 | Bitmask (see Notes below): <ul style="list-style-type: none"> 1st bit - enable/disable trading transactions for the login. 2nd bit - cancel/do not cancel orders when trading transactions are disabled for the login. |
| aspref | i4 | Client ID. For orders added by SMA login - MASTER login ID. |
| password_expiration_date | t | Password expiration date. |

Notes:

- Field 'sma_flags' is bitmask:
 - 1st bit: 0 - Cancel on Disconnect is disabled for the login, 1 - Cancel on Disconnect is enabled for the login
 - 2nd bit: 0 - Cancel on Drop-Copy Disconnect is disabled for the login, 1 - Cancel on Drop-Copy Disconnect is enabled for the login
 - 3rd bit: 0 - SMA mode is disabled for the login, 1 - SMA mode is enabled for the login.
- Field 'sma_status' is bitmask::
 - 1st bit: 0 - trading transactions are enabled for the login, 1 - trading transactions are disabled for the login
 - 2nd bit: 0 - do not cancel orders when trading transactions are disabled for the login, 1 - cancel orders when trading transactions are disabled for the login.

4.15.1.13. Table investor: Clients directory

Table 45. Fields of table investor

| Field | Type | Description |
|-----------------------------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client code |
| name | c200 | Client name |
| status | i4 | Client's flags |
| calendar_spread_margin_type | i1 | Margin type for client calendar spread, applied during the evening clearing session: <ul style="list-style-type: none"> 3 - half nett 4 - nett (**) |
| short_option_minimum_charge_ratio | d5.3 | Individual coefficient of SOMC scenario weight. (**) |
| ics_margin_type | i1 | Margin type for cross-contract spreads: <ul style="list-style-type: none"> 3 - half nett 4 - nett (**) |
| coeff_im | d16.5 | Total collateral ratio value. |
| no_fut_discount | i1 | Discount on futures: <ul style="list-style-type: none"> 1 - Discount prohibited 0 - Discount allowed |
| num_clr_2delivery | i4 | Number of clearing sessions before expiration to start expiration scenarios calculation. |
| exp_weight | d3.2 | Expiration scenario weight, in total collateral. |

Notes:

- Status field is a bit mask:
 - 0x1 - Trust Management
 - 0x2 - Separated
 - 0x4 - Brokerage Firm (Trust Management type)
 - 0x80 - Private entity
 - 0x100 - Legal entity
 - 0x200 - Non-resident
 - 0x4000 - Cross-trades permission flag. 1 - cross-trades allowed; 0 - cross-trades prohibited
 - 0x8000 - Stateless person
 - 0x20000 - Own
 - 0x40000 - Client
 - 0x80000 - Special BF
 - 0x10000000 - Additional own account

4.15.1.14. Table fut_margin_type: Type of margining

Table 46. Fields of table fut_margin_type

| Field | Type | Description |
|-------------------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| code | c12 | Settlement Account or Brokerage Firm Code |
| type | i1 | Type of Code. Settlement Account - 0, Brokerage Firm - 1. |
| margin_type | i1 | Type of margining. 2 - Gross, 3 - Half nett, 4 - Nett. |
| prohibit_coeff | d16.2 | Debt coefficient value for Settlement Account/Brokerage Firm/section. The value defines the maximum correlation between negative free limit volume and trading limit volume. As the value exceeded, the system prohibits operations. The prohibition mode is specified by field 'prohibit_type'. |
| prohibit_type | i4 | Type of automatic prohibition for Settlement Account: <ul style="list-style-type: none"> • 1 - prohibited to open positions • 2 - prohibited to add orders. |
| settlement_account_type | i1 | Settlement Account Type. 0 - own SA, 1 - client SA, 2 - SA (Trust Management type). |
| operator_input | i1 | Settlement account blocking set by the TS Administrator: <ul style="list-style-type: none"> • 0 - blocking off • 1 - blocking on. |

Notes:

- Possible 'operator_input' field values: 0 - blocking off, 1 - blocking on. When the blocking mode is turned on, orders placed from all BF clearing accounts linked to the blocked SA are automatically cancelled. The cancelled orders in the 'xstatus' field are marked with a special sign - 'eOperatorInputSA' (0x10000000000000). In the blocking mode, any trading commands with the indication of BF clearing accounts linked to this SA are prohibited, and the positions transfer between BFs is also prohibited. Orders and trades formed for SA by the Trading Administrator in blocking mode, have a special sign in the 'xstatus' field (in orders) and 'xstatus_sell' or 'xstatus_buy' fields (in trades) - 'eOperatorInputSA' (0x10000000000000).

4.15.1.15. Table fut_settlement_account: Settlement Account

Table 47. Fields of table fut_settlement_account

| Field | Type | Description |
|--------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |

| Field | Type | Description |
|--------------------|------|--------------------------------------------|
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| code | c7 | Brokerage Firm Code or Client Code |
| type | i1 | Brokerage Firm - 1, Client - 2 |
| settlement_account | c12 | Settlement Account |

4.15.1.16. Table session: Information about a trading session

The table contains trading sessions timetable.

Table 48. Fields of table session

| Field | Type | Description |
|--------------------|------|-------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Trading session ID |
| begin | t | Opening time |
| end | t | Closing time |
| state | i4 | Session status |
| opt_sess_id | i4 | ID number of the relevant session for options (**) |
| inter_cl_begin | t | Time when the intraday clearing begins (**) |
| inter_cl_end | t | Time when the intraday clearing is over (**) |
| inter_cl_state | i4 | Status of the intraday clearing (**) |
| eve_on | i1 | Flag of holding an additional evening session (**) |
| eve_begin | t | Time when the additional evening session starts (**) |
| eve_end | t | Time when the additional evening session is over (**) |
| mon_on | i1 | Flag of holding an additional morning session (**) |
| mon_begin | t | Time when the additional morning session starts (**) |
| mon_end | t | Time when the additional morning session is over (**) |
| pos_transfer_begin | t | Time when the special period for position transfer starts |
| pos_transfer_end | t | Time when the special period for position transfer finishes |

Notes:

- Fields pos_transfer_begin and pos_transfer_end specify the period of trading session during which special mode of concluding trades with instruments that are delivered during this current trading day is in power. During this special mode all orders with this certain instrument are prohibited excluding negotiated trades within one Clearing member.
- Field state can take the following values:
 - 0 Session is scheduled. Orders can't be placed but can be cancelled.
 - 1 Session is running. Orders can be both placed and cancelled.
 - 2 Trading with all instruments is suspended. Orders can't be placed but can be cancelled.
 - 3 Session is closed compulsorily. Orders can be neither placed nor cancelled.
 - 4 Session is completed because the time is up. Orders can be neither added nor cancelled.
- Field inter_cl_state is a bit mask:
 - 0x0 It is not defined. Orders can be both placed and cancelled.
 - 0x01 It is scheduled today. Orders can be placed and cancelled.
 - 0x02 It is cancelled. Orders can be placed and cancelled.
 - 0x04 Current, i.e. it is running, nothing can be done. Orders can't be placed and cancelled.
 - 0x08 Current, i.e. it is running (due to time schedule), but actually it is over and intraday clearing data is already available. Orders can't be placed but can be cancelled.

0x10 It is successfully over (due to time schedule as well). Orders can be placed and cancelled.

4.15.1.17. Table sma_master: SMA login binding to MASTER login

The table contains information on how SMA login is binding to MASTER login.

Table 49. Fields of table sma_master

| Field | Type | Description |
|---------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sma_asp | c20 | SMA login |
| sma_aspref | i4 | SMA login ID |
| master_asp | c20 | MASTER login |
| master_aspref | i4 | MASTER login ID |

4.15.1.18. Table sma_pre_trade_check: SMA login pre-trade verification settings

The table contains information on SMA login pre-trade verification settings.

Table 50. Fields of table sma_pre_trade_check

| Field | Type | Description |
|--------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| check_id | i8 | Unique record ID. |
| sma_asp | c20 | SMA login |
| sma_aspref | i4 | SMA login ID |
| check_number | i1 | Verification number (1 - 7). |
| base_contract_code | c25 | Underlying asset code. |
| instrument_type | i1 | Instrument type: <ul style="list-style-type: none"> • 0 - Futures • 1 - Option • 3 - Calendar spread |
| client_code_check | c7 | Client code under verification. |
| value | d26.2 | Verification number. |

4.15.1.19. Table clearing_members: Clearing Members

The table contains information about blocking of members.

Table 51. Fields of table clearing_members

| Field | Type | Description |
|-----------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| code | c2 | Member code |
| lock_type | i1 | Blocking type |
| lock_date | t | Blocking date |
| name | c200 | Member name |

Notes:

- Field lock_type can take the following values:
 - 0 No blocking.

2 Liquidation netting in respect of the Clearing Member

4.15.1.20. Table instr2matching_map: Instrument binding to Matching ID

The table contains information on how instrument is binding to Matching ID.

Table 52. Fields of table instr2matching_map

| Field | Type | Description |
|------------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| base_contract_id | i4 | Underlying contract ID |
| matching_id | i1 | Matching ID |

4.15.1.21. Table sys_events: table of events**Table 53. Fields of table sys_events**

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events

event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system

event_type = 5
message = "clearing_started"
Main clearing session has started

event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.16. Stream FORTS_MM_REPL - information on MM's obligations (Type=I)**4.16.1. Data scheme**

Tables:

- fut_MM_info - MM's obligations in futures
- mm_agreement_filter - Table numbers and types of contracts for the provision of market-making services

4.16.1.1. Table fut_MM_info: MM's obligations in futures**Table 54. Fields of table fut_MM_info**

| Field | Type | Description |
|---------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| sess_id | i4 | Trading session ID |

| Field | Type | Description |
|-------------------|-------|-----------------------------------------------------------------------------|
| spread | d16.5 | Spread in points |
| price_edge_sell | d16.5 | Price of the worst sell order included in the spread |
| xamount_sells | i8 | Number of contracts in the sell order included in the spread |
| price_edge_buy | d16.5 | Price of the worst buy order included in the spread |
| xamount_buys | i8 | Number of contracts in the buy order included in the spread |
| mm_spread | d16.5 | Agreed spread |
| xmm_amount | i8 | Number in accordance with the agreement |
| spread_sign | i1 | Sign: 1-spread is not maintained, 0-spread is maintained |
| amount_sign | i1 | Sign: 1- number is not maintained, 0- number is maintained |
| percent_time | d6.2 | % of fulfilled obligations |
| period_start | t | Start of the period of MM rules coming into force |
| period_end | t | End of the period of MM rules coming into force |
| client_code | c7 | Client code |
| active_sign | i4 | Sign: 1-note is deleted (stopped being active), 0-is active |
| fulfil_min | d6.2 | Minimum percentage of the liabilities for the trading session |
| fulfil_partial | d6.2 | Percentage of partial fulfillment of the obligations of the trading session |
| fulfil_total | d6.2 | Percentage of fulfillment of obligations of the trading session |
| is_fulfil_min | i1 | Minimum sign of the liabilities for the trading session |
| is_fulfil_partial | i1 | Sign of partial fulfillment of the obligations of the trading |
| is_fulfil_total | i1 | Sign of fulfillment of obligations of the trading session |
| agmt_id | i4 | Identifier of the MM agreement |
| is_rf | i1 | Sign of clearing member market-maker requirement |
| id_group | i4 | ID of market-maker group of instrument |

Notes: The 'fut_MM_info' table of the 'FORTS_MM_REPL' stream contains market-makers obligations accurate to 7-symbol client code.

4.16.1.2. Table mm_agreement_filter: Table numbers and types of contracts for the provision of market-making services

Table 55. Fields of table mm_agreement_filter

| Field | Type | Description |
|-------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| agmt_id | i4 | Identifier of the agreement |
| is_fut | i1 | Type of obligation |
| agreement | c50 | Number of the agreement |
| client_code | c7 | Client code |

4.17. Stream FORTS_CLR_REPL - clearing information (Type=AR)

4.17.1. Data scheme

Tables:

- limit_clearing - Status of clients' cash limits after clearing
- clr_rate - Currency and Index rates
- fut_pos - Open interest in futures as a result of evening clearing session
- fut_sess_settl - Futures settlement prices
- opt_sess_settl - Options settlement prices
- money_clearing_sa - Status of clients' cash accounts after clearing

- fut_pos_sa - Open interest in futures as a result of evening clearing session
- sys_events - table of events

4.17.1.1. Table limit_clearing: Status of clients' cash limits after clearing

Table 56. Fields of table limit_clearing

| Field | Type | Description |
|--------------|-------|----------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client code |
| account_type | i1 | Account type. 1 - BF, 2 - Client. |
| amount_beg | d16.2 | Limits at the beginning of the day |
| vm | d16.2 | Variation margin including variation margin on futures-style options |
| fee_fut | d16.2 | Exchange fee on futures |
| fee_opt | d16.2 | Exchange fee on options |
| go | d16.2 | Total collateral on futures and options |
| amount_end | d16.2 | Limits at the end of the day |
| free | d16.2 | Available funds |

4.17.1.2. Table clr_rate: Currency and Index rates

Table 57. Fields of table clr_rate

| Field | Type | Description |
|---------|-------|---------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| rate | d16.5 | Index value |
| moment | t | Date and time value was fixed |
| signs | i1 | Sign, that corresponds to the current value |
| sess_id | i4 | Trading session ID |
| rate_id | i4 | Rate ID |

4.17.1.3. Table fut_pos: Open interest in futures as a result of evening clearing session

Table 58. Fields of table fut_pos

| Field | Type | Description |
|--------------|-------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| sess_id | i4 | Trading session ID |
| isin | c25 | Symbol code of the instrument |
| client_code | c7 | Client code |
| account_type | i1 | Account type (0 - CF; 1 - BF; 2 - client). |
| xpos_beg | i8 | Position on trading session start |
| xpos_end | i8 | Position on trading session end |
| vm | d16.2 | Total variation margin at clearing time |
| fee | d16.2 | Total fee |
| accum_go | d16.2 | Accumulated Collateral Deposit |
| fee_ex | d16.2 | Exchange fee |
| vat_ex | d16.2 | VAT included in exchange fee |
| fee_cc | d16.2 | Clearing fee |

| Field | Type | Description |
|--------|-------|------------------------------|
| vat_cc | d16.2 | VAT included in clearing fee |

4.17.1.4. Table fut_sess_settl: Futures settlement prices

Table 59. Fields of table fut_sess_settl

| Field | Type | Description |
|-------------|-------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| sess_id | i4 | Trading session ID |
| date_clr | t | Clearing date |
| isin | c25 | Symbol code of the instrument |
| isin_id | i4 | Instrument unique ID |
| settl_price | d16.5 | Settlement price |

4.17.1.5. Table money_clearing_sa: Status of clients' cash accounts after clearing

Table 60. Fields of table money_clearing_sa

| Field | Type | Description |
|--------------------|-------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| settlement_account | c12 | Settlement Account |
| asset_type | i1 | Account type. 0 - roubles, 1 - pledge. |
| amount_beg | d26.2 | Money at the beginning of the day |
| vm | d26.2 | Variation margin |
| premium | d26.2 | Premium on options |
| pay | d26.2 | Account operations |
| fee_fut | d26.2 | Exchange fee on futures |
| fee_opt | d26.2 | Exchange fee on options |
| go | d26.2 | Total collateral on futures and options |
| amount_end | d26.2 | Money at the end of the day |
| free | d26.2 | Available funds |
| blocked_tax | d26.2 | Funds blocked for tax payments. |

4.17.1.6. Table fut_pos_sa: Open interest in futures as a result of evening clearing session

Table 61. Fields of table fut_pos_sa

| Field | Type | Description |
|--------------------|-------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| sess_id | i4 | Trading session ID |
| isin | c25 | Symbol code of the instrument |
| settlement_account | c12 | Settlement Account |
| xpos_beg | i8 | Position on trading session start |
| xpos_end | i8 | Position on trading session end |
| vm | d26.2 | Total variation margin at clearing time |
| fee | d26.2 | Total fee |
| fee_ex | d26.2 | Exchange fee |
| vat_ex | d26.2 | VAT included in exchange fee |

| Field | Type | Description |
|--------|-------|------------------------------|
| fee_cc | d26.2 | Clearing fee |
| vat_cc | d26.2 | VAT included in clearing fee |

4.17.1.7. Table sys_events: table of events

Table 62. Fields of table sys_events

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| event_type | i4 | Type of the event |
| message | c64 | Description of the event |

Notes:

- Possible types of events:
event_type = 3
message = "clearing_data_ready"
Data are ready after main clearing session

4.18. Stream FORTS_VM_REPL - online variational margin stream (Type=I)

4.18.1. Data scheme

Tables:

- fut_vm - Variation margin for futures
- fut_vm_sa - Variation margin for futures

4.18.1.1. Table fut_vm: Variation margin for futures

Table 63. Fields of table fut_vm

| Field | Type | Description |
|-------------|-------|---------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| client_code | c7 | Client code |
| sess_id | i4 | Trading session ID |
| vm | d16.5 | The accumulated variation margin on trades calculated according to the current market price |

4.18.1.2. Table fut_vm_sa: Variation margin for futures

Table 64. Fields of table fut_vm_sa

| Field | Type | Description |
|--------------------|-------|---------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin_id | i4 | Instrument unique ID |
| settlement_account | c12 | Settlement Account |
| sess_id | i4 | Trading session ID |
| vm | d26.2 | The accumulated variation margin on trades calculated according to the current market price |

4.19. Stream FORTS_INFO_REPL - additional reference information (Type=R)

4.19.1. Data scheme

Tables:

- base_contracts_params - Base contracts parameters
- futures_params - Futures parameters
- investor - Clients directory
- dealer - Companies directory
- common_params - Collateral calculation parameters
- sys_events - Table of events

4.19.1.1. Table base_contracts_params: Base contracts parameters

Table 65. Fields of table base_contracts_params

| Field | Type | Description |
|--------------------|-------|--------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| base_contract_code | c25 | Code of the underlying contract. |
| code_mcs | c25 | Intercontract spread ID (**) |
| volat_num | i1 | Number of volatility curves (**) |
| subrisk_step | f | Number of risk subpoints (**) |
| is_percent | i1 | Flag of futures contract (**) |
| has_options | i1 | Option on futures for given underlying asset. 0 - none, 1 - yes. (**) |
| percent_rate | d16.5 | Variation margin rate for interest rate futures (**) |
| somc | f | Collateral rate for uncovered sells (**) |
| misp_type | i1 | Price pitch value type. 0 - fixed, 1 - taken from FX indicator value |
| currency_id | i4 | FX ID taken from directory 'rates' of stream 'FORTS_REFDATA_REPL' (**) |
| spot_price | f | The settlement price of the underlying asset, expressed in rubles, as determined by the results of the clearing session. |
| mr1 | f | Market risk rate value |
| mr2 | f | Market risk rate value (Concentration Limit 1) |
| mr3 | f | Market risk rate value (Concentration Limit 2) |
| lk1 | i8 | Amount of underlying asset, in units (Concentration Limit 1) |
| lk2 | i8 | Amount of underlying asset, in units (Concentration Limit 2) |
| risk_points_n | i4 | Number of contract price fluctuation scenarios near risk calculation point. |
| window_size | f | Coefficient of determination smoothing window size for cross-contract spread margining (**) |
| option_model | i1 | Options pricing model (**) |

Notes:

- Field is_percent may contain the following values:
 - 0 common futures
 - 1 interest rate futures
 - 2 weather and electricity futures
 - 3 eurobonds futures
 - 4 RUONIA
- Field option_model can take the following values:

0 Black-Scholes model.

1 Bachelier model.

4.19.1.2. Table futures_params: Futures parameters

Table 66. Fields of table futures_params

| Field | Type | Description |
|--------------------------|-------|---------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| isin | c25 | Symbol code of the instrument |
| isin_id | i4 | Instrument unique ID |
| base_contract_code | c25 | Code of the underlying contract. |
| risk_range_center | d16.5 | Risk calculation center |
| spread_aspect | i1 | Flag of making up futures spread (**) |
| subrisk | i1 | Sign of accounting risks in risks subpoints |
| step_price | f | Value of the minimum price increment |
| exp_date | t | Date of expiration |
| settlement_price | d16.5 | Settlement price after the last clearing session |
| min_step | f | Minimal price increment |
| lot | i4 | Number of underlying asset in instrument, in units |
| attribute | i4 | Bit flags defining futures type |
| interest_rate_risk_up | f | Interest risk variable rate on rate up scenario |
| interest_rate_risk_down | f | Interest risk variable rate on rate down scenario |
| time_to_expiration | f | Time before instrument expiration, in fraction of year |
| normalized_spot | f | Theoretical price value of underlying asset on spot market, in points, reduced to dimension of the primary one |
| mr_addon_up | f | Up-addition for NormalizedSpot to control initial margin on futures level, specified in NormalizedSpot fractions. |
| mr_addon_down | f | Down-addition for NormalizedSpot to control initial margin on futures level, specified in NormalizedSpot fractions. |
| enforce_ims_half_netting | i1 | Flag - consider the risks of intermonth spread according to the "half-netto" rule: "1" - yes; "0" - no. |

Notes:

- Field spread_aspect can take the following values:

0 It is not included in spread

2 It is included into calendar spread

- Field 'attribute' can take the following values:

0 Ordinary futures

4.19.1.3. Table investor: Clients directory

Table 67. Fields of table investor

| Field | Type | Description |
|-----------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client code |
| calendar_spread_margin_type | i1 | Margin type for client calendar spread, current value: <ul style="list-style-type: none"> 3 - half nett 4 - nett |

| Field | Type | Description |
|-----------------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------|
| | | (**) |
| num_clr_2delivery | i4 | Number of clearing sessions before expiration to start expiration scenarios calculation. |
| exp_weight | d3.2 | Expiration scenario weight, in total collateral. |
| coeff_im | d16.5 | Total collateral ratio value. |
| no_fut_discount | i1 | Discount on futures: <ul style="list-style-type: none"> • 1 - Discount prohibited • 0 - Discount allowed |
| short_option_minimum_charge_ratio | d5.3 | Individual coefficient of SOMC scenario weight. (**) |
| ics_margin_type | i1 | Margin type for cross-contract spreads: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett (**) |

4.19.1.4. Table dealer: Companies directory

Table 68. Fields of table dealer

| Field | Type | Description |
|-----------------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| client_code | c7 | Client code |
| margin_type | i1 | Margin type, according to BF's sections, current value: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett |
| calendar_spread_margin_type | i1 | Margin type for calendar spreads, for BF portfolio, current value: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett (**) |
| num_clr_2delivery | i4 | Number of clearing sessions before expiration to start BF expiration scenarios calculation. Current value. |
| exp_weight | d3.2 | Expiration scenario weight for BF, in total collateral. Current value. |
| coeff_im | d16.5 | Total collateral ratio value, for BF. Current value. |
| no_fut_discount | i1 | Discount on futures for BF portfolio, current value: <ul style="list-style-type: none"> • 1 - Discount prohibited • 0 - Discount allowed |
| num_clr_2delivery_client_default | i4 | Number of clearing sessions before expiration to start clients expiration scenarios calculation (default value). Current value. |
| exp_weight_client_default | d3.2 | Expiration scenario weight for clients, in total collateral (default value). Current value. |
| no_fut_discount_client_default | i1 | Discount on futures for BF's clients, default value: <ul style="list-style-type: none"> • 1 - Discount prohibited • 0 - Discount allowed Current value. |
| short_option_minimum_charge_ratio | d5.3 | Individual coefficient of SOMC scenario weight. (**) |
| ics_margin_type | i1 | Margin type for cross-contract spreads: <ul style="list-style-type: none"> • 3 - half nett |

| Field | Type | Description |
|----------------------------------|------|-----------------------------------------------------------------|
| | | <ul style="list-style-type: none"> 4 - nett (**) |
| order_allowed_in_morning_session | i1 | Access to trading during the morning trading session. (**) |

Notes:

- Field order_allowed_in_morning_session can take the following values:
 - 0 Access to trading during the morning trading session is limited. Trading operations are prohibited, except for orders cancellation operations.
 - 1 Access to trading during the morning trading session is allowed.

4.19.1.5. Table common_params: Collateral calculation parameters

Table 69. Fields of table common_params

| Field | Type | Description |
|------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| common_rev | i4 | Revision number - surrogate key |
| edge_coeff | f | Marginal risk factor ratio |

4.19.1.6. Table sys_events: Table of events

Table 70. Fields of table sys_events

| Field | Type | Description |
|-------------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| event_type | i4 | Type of the event |
| event_id | i8 | Unique ID of the event |
| sess_id | i4 | Session number |
| message | c64 | Description of the event |
| server_time | t | Server date and time |

Notes:

- Possible types of events
 - event_type = 1
message = "session_data_ready"
All data from the clearing system have been loaded into the trading system
 - event_type = 5
message = "clearing_started"
Main clearing session has started
 - event_type = 6
message = "extension_of_limits_finished"
Limits have been extended

4.20. Stream FORTS_TNPENALTY_REPL - information about Transaction fees (Type=I)

4.20.1. Data scheme

Tables:

- fee_all - Information on the number of points accrued
- fee_tn - Detailed information on the number of incorrect transaction

4.20.1.1. Table fee_all: Information on the number of points accrued

Table 71. Fields of table fee_all

| Field | Type | Description |
|---------|-------|--------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| time | i8 | Time value in 'YYYYMMddhhmmssSSS' format |
| p2login | c64 | Login |
| sess_id | i4 | Session number |
| points | i4 | Number of points assessed for a second time from |
| fee | d16.2 | Incorrect transaction fee at the time of time |

4.20.1.2. Table fee_tn: Detailed information on the number of incorrect transaction

Table 72. Fields of table fee_tn

| Field | Type | Description |
|----------|------|--------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| time | i8 | Time value in 'YYYYMMddhhmmssSSS' format |
| p2login | c64 | Login |
| sess_id | i4 | Session number |
| tn_type | i4 | Transaction type |
| err_code | i4 | Error code |
| count | i4 | Number of invalid transactions |

4.21. Stream FORTS_FORECASTIM_REPL - Risk forecast after limits extension (Type=I)

4.21.1. Data scheme

Tables:

- part_sa_forecast - Free funds for SA volume forecast

4.21.1.1. Table part_sa_forecast: Free funds for SA volume forecast

Table 73. Fields of table part_sa_forecast

| Field | Type | Description |
|--------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replID | i8 | Service field of the replication subsystem |
| replRev | i8 | Service field of the replication subsystem |
| replAct | i8 | Service field of the replication subsystem |
| settlement_account | c12 | Settlement account |
| money_free | d26.2 | Funds available |
| MarketDataRev | i8 | Revision number (field 'replRev' value) of the most recent data change (for streams transmitting orders and trades data) included into risk parameters forecast. Orders and trades with the 'replRev' revision value less than field 'MarketDataRev' value will be included into the forecast. Orders and trades with the 'replRev' revision value greater than field 'MarketDataRev' value will NOT be included into the forecast. For more information about field 'replRev' see section 3.3.1. Service replication fields. |

5. Commands description

5.1. Method AddOrder - Adding orders

Message type: 465

Reply message type: 179

Used to add orders for futures.

Table 74. Input parameters

| Name | Type | Default value | Description |
|------------------|------|---------------|----------------------------------------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| isin_id | i4 | | Instrument unique ID |
| client_code | c3 | | Client code |
| dir | i4 | | Order direction |
| type | i4 | | Order type |
| amount | i4 | | Amount |
| price | c17 | | Price |
| comment | c20 | "" | Order comment |
| broker_to | c20 | "" | RTS code of the company to whom the negotiated order is addressed |
| ext_id | i4 | 0 | External ID |
| is_check_limit | i4 | 0 | Flag of checking price limits |
| date_exp | c8 | "" | Order's expiration date |
| dont_check_money | i4 | 0 | Whether to calculate client risks for given order |
| match_ref | c10 | "" | Identical text values entered by both trading parties to match negotiated orders |

Table 75. Execution result

| Name | Type | Default value | Description |
|----------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| order_id | i8 | | Order's ID |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The '**type**' field may contain the following values:
 - quotation order (remains in queue after being partly matched)
 - counter-order (removed after auction end)
 - Fill-or-Kill order
- The '**dir**' field may contain the following values:
 - buy order
 - sell order
- The '**price**' field contains the order price as string: 'nnnnnnnnnn.mmmmm'.
- The '**is_check_limit**' field may contain the following values:
 - Do not verify limits
 - Verify limits

- The '**date_exp**' field contains order expiration date as 'YYYYMMDD'. Empty string indicates a common order. If there is certain date set in the string, the order are automatically relisted in the next session with a new number and a new time, until the date expires (multiday order). Orders with the expired date are removed automatically after the end of the evening session (if there is any on this day). When relisted, the orders are verified for instrument availability, client details and funds availability. Date may vary in the range from >= today to <= 1 year ahead.
- The '**dont_check_money**' order parameter may contain the following values:
 - 0 – verify collaterals for client section
 - 1 – do not verify collaterals for client section

The parameter is eligible for using by a login with the appropriate right. All other logins using this parameter will have their orders rejected.

5.2. Method DelOrder - Deletion of orders

Message type: 461

Reply message type: 177

Used to delete orders for futures.

Table 76. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|----------------------|
| broker_code | c4 | | Brokerage Firm code |
| order_id | i8 | | Order ID to delete |
| client_code | c3 | | Client code |
| isin_id | i4 | | Instrument unique ID |

Table 77. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|-----------------------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| amount | i4 | | Order's amount on deletion moment |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The return code = 14 (order is not found for removing) indicates that there is no such order in queue. Possible reasons: wrong order number, or the order has not been placed today. It does not make sense to continue sending removal requests for the same order number (may be useful for automatic systems).

5.3. Method DelUserOrders - Mass cancel orders

Message type: 466

Reply message type: 186

Mass cancellation of all orders under a criteria.

Table 78. Input parameters

| Name | Type | Default value | Description |
|--------------------|------|---------------|---------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| buy_sell | i4 | | Whether to cancel orders on their directions |
| non_system | i4 | | Whether to cancel orders on their negotiated sign |
| code | c3 | | Client code |
| base_contract_code | c25 | | Underlying asset code |
| ext_id | i4 | 0 | External ID |
| isin_id | i4 | | Instrument unique ID |

| Name | Type | Default value | Description |
|-----------------|------|---------------|-----------------------|
| instrument_mask | i1 | | Instrument group mask |

Table 79. Execution result

| Name | Type | Default value | Description |
|------------|------|---------------|----------------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| num_orders | i4 | | Number of cancelled orders |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The '**buy_sell**' parameter may contain the following values:
 - 1 Buy orders
 - 2 Sell orders
 - 3 All orders
- The '**non_system**' parameter may contain the following values:
 - 0 Common orders
 - 1 Negotiated orders
 - 2 All orders
- The '**instrument_mask**' parameter are bit mask:

0x1 Futures
- If the '**code**' parameter is not set or is '%%%', then all orders for all clients' accounts are removed.
- If the '**base_contract_code**' parameter is not set or is '%', then all orders for all contracts are removed.
- If the '**ext_id**' parameter value is not 0, then all orders with the corresponding '**ext_id**' are removed. The values of the '**buy_sell**', '**non_system**', '**base_contract_code**', and '**isin_id**' parameters are ignored, but their values must be within the allowed range.

5.4. Method MoveOrder - Modify orders

Message type: 460

Reply message type: 176

Used to modify orders for futures.

Table 80. Input parameters

| Name | Type | Default value | Description |
|----------------|------|---------------|-----------------------------------|
| broker_code | c4 | | Brokerage Firm code |
| regime | i4 | | Mode |
| order_id1 | i8 | | ID of the 1st order to remove |
| amount1 | i4 | | New amount for the 1st order |
| price1 | c17 | | New price for the 1st order |
| ext_id1 | i4 | | New external ID for the 1st order |
| order_id2 | i8 | | ID of the 2nd order to remove |
| amount2 | i4 | | New amount for the 2nd order |
| price2 | c17 | | New price for the 2nd order |
| ext_id2 | i4 | | New external ID for the 2nd order |
| is_check_limit | i4 | | Flag of checking limits |
| client_code | c3 | | Client code |

| Name | Type | Default value | Description |
|---------|------|---------------|----------------------|
| isin_id | i4 | | Instrument unique ID |

Table 81. Execution result

| Name | Type | Default value | Description |
|-----------|------|---------------|----------------------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| order_id1 | i8 | | New ID of the 1st modified order |
| order_id2 | i8 | | New ID of the 2nd modified order |

Return codes:

0 operation completed successfully

Any other value error

Notes (in this note, the term 'amount' means the number of units of the instrument):

- The '**regime**' parameter defines the command work mode. It may contain the following values:
 - 0 Do not change amount of orders. The current amounts of orders remains unchanged, the newly sent amounts are ignored.
 - 1 Change amounts of orders. If there is any order found, it will be replaced with the new order with new price and amount.
 - 2 Remove old orders. If any order volume does not coincide with the newly sent one, both orders are removed. Otherwise, the orders will be modified.
 - 3 Set orders amounts to that of received, excluding the matched part (not less than 0). If the amount received is less than the amount of the matched part, both orders will be removed.
- The '**is_check_limit**' may contain the following values:
 - 0 Do not verify limits
 - 1 Verify limits
- All new orders will be auctioned.
- Orders can be shifted only within the same trading instrument and only within the same client register.
- Negotiated orders are not shifted.
- When shifting, the direction of order is not changed.
- Once an order has been removed (or shifted, or fully matched), it is not relisted, and the error message appears.
- If one order of a pair cannot be shifted, then another order is not shifted, too, and the error message appears.
- If two orders with opposite directions are shifted in the way their prices coincide, then the parameters are considered as incorrect, shifting is not performed, and the error message appears.
- If, when shifting a pair of orders, one order meets a cross-trade (matching an order sent from either the same UIN or the same client register), then it is rejected, and another order of the pair is shifted.
- Upon moving orders, the '**date_exp**' parameters are transferred into new orders.
- After the command has been processed, the '**order_id1**' field and '**order_id2**' field are filled with new orders numbers. If no order has been placed, the corresponding field is set to 0.

5.5. Method IcebergAddOrder - Adding iceberg orders

Message type: 462

Reply message type: 180

Used to add iceberg orders.

Table 82. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|---------------------|
| broker_code | c4 | "" | Brokerage Firm code |

| Name | Type | Default value | Description |
|-----------------------|------|---------------|--------------------------------------------------------------------------------------------------------|
| isin_id | i4 | | Instrument unique ID |
| client_code | c3 | | Client code |
| dir | i4 | | Order direction |
| disclose_const_amount | i4 | | Number of instrument units in the visible part of the iceberg order |
| iceberg_amount | i4 | | Total amount of instruments in iceberg order |
| variance_amount | i4 | 0 | Amplitude of deviation (in contracts) of the random allowance to the visible part of the iceberg order |
| price | c17 | | Price |
| comment | c20 | "" | Order comment |
| ext_id | i4 | 0 | External ID |
| is_check_limit | i4 | 0 | Flag of checking price limits |
| date_exp | c8 | "" | Order's expiration date |
| dont_check_money | i4 | 0 | Whether to calculate client risks for given order |

Table 83. Execution result

| Name | Type | Default value | Description |
|------------------|------|---------------|------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| iceberg_order_id | i8 | | Iceberg order ID |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The '**dir**' field may contain the following values:
 - 1 buy order
 - 2 sell order
- The '**price**' field contains the order price as string: 'nnnnnnnnn.mmmmm'.
- The '**is_check_limit**' field may contain the following values:
 - 0 Do not verify limits
 - 1 Verify limits
- The '**date_exp**' field contains order expiration date as 'YYYYMMDD'. Empty string indicates a common order. If there is certain date set in the string, the order are automatically relisted in the next session with a new number and a new time, until the date expires (multiday order). Orders with the expired date are removed automatically after the end of the evening session (if there is any on this day). When relisted, the orders are verified for instrument availability, client details and funds availability. Date may vary in the range from >= today to <= 1 year ahead.
- The '**dont_check_money**' order parameter may contain the following values:
 - 0 – verify collaterals for client section
 - 1 – do not verify collaterals for client section

The parameter is eligible for using by a login with the appropriate right. All other logins using this parameter will have their orders rejected.

5.6. Method IcebergDelOrder - Deletion of iceberg orders

Message type: 464

Reply message type: 182

Used to delete iceberg orders. The command can work both on 'public_order_id' and on 'private_order_id'. That the command will work on 'public_order_id' 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 'private_order_id'.

Table 84. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|----------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| order_id | i8 | | Order ID to delete |
| isin_id | i4 | | Instrument unique ID |

Table 85. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|-----------------------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| amount | i4 | | Order's amount on deletion moment |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The return code = 14 (order is not found for removing) indicates that there is no such order in queue. Possible reasons: wrong order number, or the order has not been placed today. It does not make sense to continue sending removal requests for the same order number (may be useful for automatic systems).

5.7. Method IcebergMoveOrder - Modify iceberg orders

Message type: 463

Reply message type: 181

Used to modify iceberg orders. The command can work both on 'public_order_id' and on 'private_order_id'. That the command will work on 'public_order_id' 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 'private_order_id'.

Table 86. Input parameters

| Name | Type | Default value | Description |
|----------------|------|---------------|--------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| order_id | i8 | | ID of the order to be modified |
| isin_id | i4 | | Instrument unique ID |
| price | c17 | "0" | New price of the order |
| ext_id | i4 | | New external ID of the order |
| is_check_limit | i4 | 0 | Flag of checking limits |

Table 87. Execution result

| Name | Type | Default value | Description |
|----------|------|---------------|------------------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| order_id | i8 | | New ID of the modified order |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The '**is_check_limit**' may contain the following values:

0 Do not verify limits

1 Verify limits

5.8. Method ChangeClientMoney - Change client limits

Message type: 458

Reply message type: 187

The command allows to change funds limits for a client's account.

Table 88. Input parameters

| Name | Type | Default value | Description |
|----------------------|------|---------------|----------------------------------------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| mode | u1 | 0 | Mode |
| code | c3 | | Client code |
| coeff_im | c17 | "" | Clients collateral ratio |
| is_auto_update_limit | i4 | -1 | Flag of automatic adjustment of the limit by the amount of income after clearing |
| check_limit | i4 | 1 | Funds sufficiency verification flag |
| limit_money | c17 | "" | Funds limit |

Table 89. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- Command work mode (the '**mode**' field):
 - 11 Remove the limit, disable checking for sufficient funds
 - 12 Set the funds limit in the amount of '**limit_money**'
 - 13 Change the funds limit by the amount of '**limit_money**'
- The '**is_auto_update_limit**' flag, being set to '1', allows to automatize the limit changing process in accordance with the previous day results. The value "-1" for the '**is_auto_update_limit**' parameter means that the value is not set by the user.
- To change parameter '**is_auto_update_limit**', the mode '13' must be used. The '**limit_money**' parameter value must be set to '0'.
- The following values are set in the **check_limit** parameter:
 - 0 Do not verify funds sufficiency. Change limit unconditionally.
 - 1 Verify funds sufficiency. Do not change limit if there are insufficient funds
- For the field type 'c17', it is possible to specify empty string in order to prevent changing the parameter value, which had been sent into the trading system before.

5.9. Method FutChangeClientProhibit - Modify client's restrictions for futures

Message type: 15

Reply message type: 115

Table 90. Input parameters

| Name | Type | Default value | Description |
|--------------------|------|---------------|-------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| mode | i4 | | Mode |
| code | c3 | | Code of the client's account or '%%%' – for all |
| base_contract_code | c25 | | Code of the underlying asset or '%' - for all |
| isin | c25 | | Futures or '%' - for all |
| state | i4 | 0 | Restriction |

| Name | Type | Default value | Description |
|------------|------|---------------|----------------------------|
| state_mask | i4 | 3 | Mask for parameter 'state' |

Table 91. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- The '**mode**' field specifies the command work mode:
 - 11 remove
 - 12 set
- The '**state**' field may contain the following values:
 - 0 no prohibitions (when cancelling a previous prohibition with lower priority, otherwise simply delete the line);
 - 1 prohibited to open positions;
 - 2 prohibited to perform all trading operations;
 - 3 prohibited to open sell positions;s
- The 'state_mask' parameter values are defined by the bit mask. At the moment, the parameter value must be '3'.
- When setting a certain instrument in the '**isin**' field, the code of the corresponding underlying asset must be set in the '**base_contract_code**' field.

5.10. Method TransferClientPosition - Transfer client positions

Message type: 430

Reply message type: 173

The command allows to transfer positions between your brokerage firms' accounts.

Table 92. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|--------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| code_from | c7 | | Donor code |
| code_to | c7 | | Recipient code |
| isin | c25 | | Instrument ID |
| amount | i8 | | Amount of position to transfer |

Table 93. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

To get access to the procedure, a clearing firm's login must obtain the appropriate right from the Trading Administrator.

5.11. Method ChangeBFParametersNextSession - Change BF's parameters by a clearing member

Message type: 442

Reply message type: 162

The command allows a Clearing member to change BF's parameters. Please note that the Clearing member must belong to a Clearing Firm to use the command. All changes made will be applied during the evening clearing session.

Table 94. Input parameters

| Name | Type | Default value | Description |
|-----------------------------|------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| code_bf | c2 | | BF code |
| margin_type | i4 | -1 | Margin type, according to BF's sections: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett |
| calendar_spread_margin_type | i1 | -1 | Margin type for calendar spreads, for BF portfolio: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett |
| num_clr_2delivery | i4 | -1 | Number of clearing sessions before expiration to start BF expiration scenarios calculation. |
| exp_weight | c17 | "" | Expiration scenario weight for BF, in total collateral. |
| go_ratio | c17 | "" | Total collateral ratio value, for BF. |
| check_limit_for_orders | i4 | -1 | Verification of collateral sufficiency upon adding orders, for BF: <ul style="list-style-type: none"> • 1 - Verify • 0 - Do not verify |
| no_fut_discount | i4 | -1 | Discount on futures for BF portfolio: <ul style="list-style-type: none"> • 1 - Discount prohibited • 0 - Discount allowed |
| ics_margin_type | i1 | -1 | Margin type for cross-contract spreads: <ul style="list-style-type: none"> • 3 - half nett • 4 - nett |

Table 95. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- For the field type 'c17', it is possible to specify empty string in order to prevent changing the parameter value, which had been sent into the trading system before.
- For the fields type 'i4' and 'i1', it is possible to specify '-1' in order to prevent changing the parameter value, which had been sent into the trading system before.

5.12. Method ChangeClientParameters - Change parameters of client account

Message type: 443

Reply message type: 178

The command allows to change parameters for client accounts by a Clearing member. Please note that the Clearing member must belong to a Brokerage Firm/Clearing Firm to use the command.

Table 96. Input parameters

| Name | Type | Default value | Description |
|-----------------|------|---------------|------------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| code | c3 | | Client code |
| coeff_go | c17 | "" | Clients collateral ratio |
| no_fut_discount | i4 | -1 | Flag of prohibition to provide discounts for futures |

Table 97. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- For the field type 'c17', it is possible to specify empty string in order to prevent changing the parameter value, which had been sent into the trading system before.
- For the field type 'i4', it is possible to specify '-1' in order to prevent changing the parameter value, which had been sent into the trading system before.

5.13. Method ChangeClientParametersNextSession - Change parameters of client account in clearing session

Message type: 441

Reply message type: 163

The command allows to change parameters for client accounts by a Clearing member. Please note that the Clearing member must belong to a Brokerage Firm/Clearing Firm to use the command.

Table 98. Input parameters

| Name | Type | Default value | Description |
|-----------------------------|------|---------------|---------------------------------------------------------------------------------------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| code | c3 | | Client code |
| calendar_spread_margin_type | i1 | -1 | Margin type for calendar spreads, for client: <ul style="list-style-type: none"> 3 - half nett 4 - nett |
| ics_margin_type | i1 | -1 | Margin type for cross-contract spreads: <ul style="list-style-type: none"> 3 - half nett 4 - nett |

Table 99. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- For the field type 'i1', it is possible to specify '-1' in order to prevent changing the parameter value, which had been sent into the trading system before.

5.14. Method ChangeBFClientDefaultParametersNextSession - Change default parameters of client sections

Message type: 402

Reply message type: 602

The command allows to change default parameters for client sections of a single BF. Please note that the login must belong to a Brokerage Firm/Clearing Firm to use the command. All changes made will be applied during the evening clearing session.

Table 100. Input parameters

| Name | Type | Default value | Description |
|----------------------------------|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| code_bf | c2 | | BF code |
| num_clr_2delivery_client_default | i4 | -1 | Number of clearing sessions before expiration to start clients expiration scenarios calculation. |
| exp_weight_client_default | c17 | "" | Expiration scenario weight for client, in total collateral. |
| no_fut_discount_client_default | i4 | -1 | Discount on futures for client section portfolios: <ul style="list-style-type: none"> 1 - Discount prohibited 0 - Discount allowed |

Table 101. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- For the field type 'c17', it is possible to specify empty string in order to prevent changing the parameter value, which had been sent into the trading system before.
- For the field type 'i4', it is possible to specify '-1' in order to prevent changing the parameter value, which had been sent into the trading system before.

5.15. Method ChangeBFLimit - Change BF trading limits

Message type: 428

Reply message type: 161

The command allows to change BF trading limits

Table 102. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|---------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| mode | i4 | | Mode |
| code | c2 | | Brokerage firm code |
| limit_money | c17 | | Funds limit |

| Name | Type | Default value | Description |
|-------------|------|---------------|-----------------------------|
| check_limit | i4 | | Verify BF funds sufficiency |

Table 103. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- Command work mode (the '*mode*' field):
 - 12 Set limits value to that of *limit_money*
 - 13 Change limits value to that of *limit_money*
- The following values are set in the *check_limit* parameter:
 - 0 Do not verify
 - 1 Verify

5.16. Method CODHeartbeat - Heartbeat message for Cancel on Disconnect Service

Message type: 10000

The heartbeat message informs the client connection monitoring service that this client login is active.

Table 104. Input parameters

| Name | Type | Default value | Description |
|------------|------|---------------|---------------------------------------------------|
| seq_number | i4 | 0 | Sequence number of heartbeat (currently not used) |

A client of COD (Cancel on Disconnect) service is should send heartbeat messages to the trading system not less than once per 10 second. If the user stays inactive (sends no messages to the trading system) within 20 seconds, all their orders will be automatically cancelled.

Note:

Only the COD service clients are obliged to send heartbeat messages.

The monitoring service does not send any replies on heartbeat messages. Please set flag value to '0' (no reply expected) when calling the heartbeat message sending function (cg_pub_post(pub, msgptr, 0)).

Calling function 'cg_pub_post' with flag 'CG_PUB_NEEDREPLY' for sending heartbeat messages will cause a notification error 'CG_MSG_P2MQ_TIMEOUT'.

5.17. Method SetSmaPreTradeCheck - Enable pre-trade verification mode for SMA login orders

Message type: 406

Reply message type: 166

The command enables pre-trade verification mode for SMA login orders.

Table 105. Input parameters

| Name | Type | Default value | Description |
|--------------|------|---------------|---------------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| sma_asp | c20 | "" | SMA login ID |
| check_number | i1 | | Verification number (1-7) |

| Name | Type | Default value | Description |
|--------------------|------|---------------|--------------------------------------------------------------------------------------------------------------------------------------|
| base_contract_code | c25 | "" | Underlying asset code |
| instrument_type | i1 | 0 | Instrument type: <ul style="list-style-type: none"> • 0 -Futures • 1 - Option • 2 - Calendar spread |
| client_code_check | c3 | "" | Client code under verification |
| value | c29 | | Verification number |

Table 106. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

other value error

Note:

The command is exclusively available to the logins obtained the appropriate right from the Trading Administrator.

Below is the table containing verification number description for field 'check_number':

Table 107. Verification numbers

| Number | What to verify | Fields involved |
|--------|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Price fluctuation against the current price. | Field 'value' contains price value in order fluctuated against the current price value, in percent. Field 'sma_asp' contains an SMA login to be verified. Fields 'instrument_type' and/or 'base_contract_code' will be specified if there is a certain instrument/all instruments to be verified for a selected underlying asset. |
| 2 | Maximum volume of order, in contracts. | Field 'value' contains maximum number of contracts allowed in a single order. Field 'sma_asp' contains an SMA login to be verified. Fields 'instrument_type' and/or 'base_contract_code' will be specified if there is a certain instrument/all instruments to be verified for a selected underlying asset. |
| 3 | Allow/disallow negotiated mode | Field 'value' contains: <ul style="list-style-type: none"> • 0 - allow negotiated mode • 1 - disallow negotiated mode. |
| 4 | Maximum volume of order, in KZT. | Field 'value' contains maximum amount in KZT allowed in a single order. Field 'sma_asp' contains an SMA login to be verified. Fields 'instrument_type' and/or 'base_contract_code' will be specified if there is a certain instrument/all instruments to be verified for a selected underlying asset. |
| 5 | Maximum number of orders (gross) allowed per trading day. | Field 'value' contains maximum number of orders (gross) allowed per trading day. Field 'sma_asp' contains an SMA login to be verified. Fields 'instrument_type' and/or 'base_contract_code' will be specified if there is a certain instrument/all instruments to be verified for a selected underlying asset. |
| 6 | Maximum number of contracts in a single buy order. | Field 'value' contains maximum number of contracts in a single buy order for trading member specified in field 'client_code_check'. Fields 'instrument_type' and/or 'base_contract_code' will be specified if there is a certain instrument/all instruments to be verified for a selected underlying asset. |
| 7 | Maximum number of contracts in a single sell order. | Field 'value' contains maximum number of contracts in a single sell order for trading member specified in field 'client_code_check'. Fields 'instrument_type' and/or 'base_contract_code' will be specified if there is a certain instrument/all instruments to be verified for a selected underlying asset. |

5.18. Method DelSmaPreTradeCheck - Disable pre-trade verification mode for SMA login orders

Message type: 407

Return message type: 167

The command disables pre-trade verification mode for SMA login orders.

Table 108. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|---------------------|
| broker_code | c4 | "" | Brokerage Firm code |
| check_id | i8 | | |

Table 109. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

other value error

Note:

The command is exclusively available to the logins obtained the appropriate right from the Trading Administrator.

5.19. Method UserKillSwitch - Disable transactions for trading member login

Message type: 408

Return message type: 168

The command disables trading transactions for a trading member login.

Table 110. Input parameters

| Name | Type | Default value | Description |
|---------------|------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| login | c20 | | Trading member's login to enable/disable trading transactions for. |
| disable | i1 | | Transaction allowance settings for the login: <ul style="list-style-type: none"> • 0 - trading transactions are enabled for the login • 1 - trading transactions are disabled for the login. |
| cancel_orders | i1 | 0 | Order cancellation settings if trading transactions are disabled for the login: <ul style="list-style-type: none"> • 0 - cancel orders • 1 - do not cancel orders. |

Table 111. Output results

| Name | Type | Default values | Description |
|------------|------|----------------|---------------------------|
| code | i4 | | Return code |
| message | c255 | | Message text |
| num_orders | i4 | | Number of canceled orders |

Return codes:

0 operation completed successfully

other value error

Notes:

The command is exclusively available to the logins obtained the appropriate right from the Trading Administrator.

Flag 'cancel_orders = 1' is only available if 'disable =1'.

5.20. Method SetBrokerFeeParamNextSession - Setting parameters for calculating the brokerage fee

Message type: 453

Reply message type: 183

The command allows to add, change and delete parameters that are used in calculating the brokerage fee for clients trades. Parameters can be set for an individual client and for the entire brokerage firm. The parameters set for the BF are used in the calculation for all of its clients. The command is available for the login CF and BF levels only, to which the Trading Administrator has set the necessary rights. The set parameters will be applied in the next trading session.

Table 112. Input parameters

| Name | Type | Default value | Description |
|-------------|------|---------------|--------------------------------------------------------|
| broker_code | c4 | | Brokerage Firm code |
| mode | i4 | | Mode |
| client_code | c3 | "" | Client code |
| lower_fee | c27 | | Minimum possible amount of brokerage fee per contract |
| upper_fee | c27 | | Maximum possible amount of brokerage fee per contract |
| multiplier | c27 | | Multiplier to the amount of exchange and clearing fees |
| additive | c27 | | Constant addition per contract |

Table 113. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 operation completed successfully

Any other value error

Notes:

- Command work mode (the 'mode' field):
 - 1 Add / Edit
 - 2 Delete
- You have to specify the client code in the command ('client_code' field) if you set parameters for him only. If you set parameters for the entire BF, the 'client_code' field should be empty.
- Acceptable values for 'lower_fee' parameter from 0.00 to +100.00.
- Acceptable values for 'upper_fee' parameter from 0.00 to +10 000.00.
- Acceptable values for 'multiplier' parameter from 0.00 to +100.00.
- Acceptable values for 'additive' parameter from 0.00 to +1 000.00.
- When adding (changing) client parameters ('mode=1' in the command), a new record with 'sess_id=-1' is added to the 'broker_fee_params' table. New parameters will be applied in the next trading session.
- When deleting parameters ('mode=2' in the command):
 - If the client has the parameters added today only (entry in the 'broker_fee_params' table with 'sess_id=-1'), then they are deleted from the table.
 - If the client has the current parameters only, then these parameters are marked for deletion in the 'broker_fee_params' table. For this, a new record with current parameters is added to the table and 'sess_id=-2' is set for it. But parameters will be deleted when the trading session changes.

- If the client has both current and parameters added today, then the newly added parameters (entries with 'sess_id=-1') are deleted from the 'broker_fee_params' table, and the current parameters are marked for deletion (new record with 'sess_id=-2') and will be deleted when the trading session changes.

5.21. Method ChangePassword - Change user password for the Trading System

Message type: 421

Return message type: 169

The command allows to change user password for the Trading System. The command requires a dedicated protocol 'p2mqpwd' provided with the CGate API.

Table 114. Input parameters

| Name | Type | Default value | Description |
|---------|------|---------------|------------------|
| old_pwd | c65 | | Current password |
| new_pwd | c65 | | New password |

Table 115. Execution result

| Name | Type | Default value | Description |
|---------|------|---------------|--------------|
| code | i4 | | Return code |
| message | c255 | | Message text |

Return codes:

0 Success

other value error

Note:

Once any error occurs during the password change procedure, the user password will not be changed.

Plaza-2 data types

| Plaza-2 | C++ | ODBC | Details |
|-----------|-----------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| u1 | UINT8 | SMALLINT | Integer, size: 1 byte |
| u2 | UINT16 | INTEGER | Integer, size: 2 bytes |
| u4 | UINT32 | NUMERIC,10 | Integer, size: 4 bytes |
| u8 | UINT64 | NUMERIC,20 | Integer, size: 8 bytes |
| i1 | INT8 | SMALLINT | Integer with sign, size: 1 byte |
| i2 | INT16 | SMALLINT | Integer with sign, size: 2 bytes |
| i4 | INT32 | INTEGER | Integer with sign, size: 4 bytes |
| i8 | INT64 | BIGINT | Integer, size: 8 bytes |
| a | CHAR | VARCHAR | Symbol string, size: 1 byte. |
| cN | CHAR[N+1] | VARCHAR,N | Symbol string, ended with zero. |
| dN.M sN.M | P2BCDII | NUMERIC,N,M | Fixed-point decimal number coded in binary system, where: <ul style="list-style-type: none"> • N is the whole quantity of digits • M is quantity of digits in the fractional part |
| t | P2TIME | TIMESTAMP | Date and time. |
| f | DOUBLE | REAL | Double-precision number with floating point, size: 8 bytes. |
| bN | | VARBINARY,N | Data unit. |
| zN | | VARBINARY,N | Data unit., where the buffer length is set by the first 4 bytes. |

Note:

Win1251 is used to encode symbol strings.

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 restriction: no restriction 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. |