FIX Bridge for Meta Trader 4

User Guide

v.1.9, December 01, 2016

Document History

Version	Date	Description of change
1.0	2011-0411	Initial version
1.1	2013-0204	Adaptation
1.2	2013-0916	Market data messages added
1.3	2013-0918	Some new settings added
1.4	2013-0926	Added missed tags and Market Data Incremental Refresh(X) FIX message
1.5	2013-0927	Added description of the new configuration parameters
1.6	2015-0204	Added description of the SecurityMaster
1.7	2015-0310	Added description of the new configuration parameters
1.8	2015-0312	Added description of the Market Definition and Custom FIX Rules
1.9	2016-1201	Added description of the new configuration parameters Added block Party to order related messages Added value '6' Good Till Date (GTD) to tag 59 (TimeInForce) Added tag 126 (ExpireTime) to support GTD orders Added description of Buying Power calculation algorithm

Table of Contents

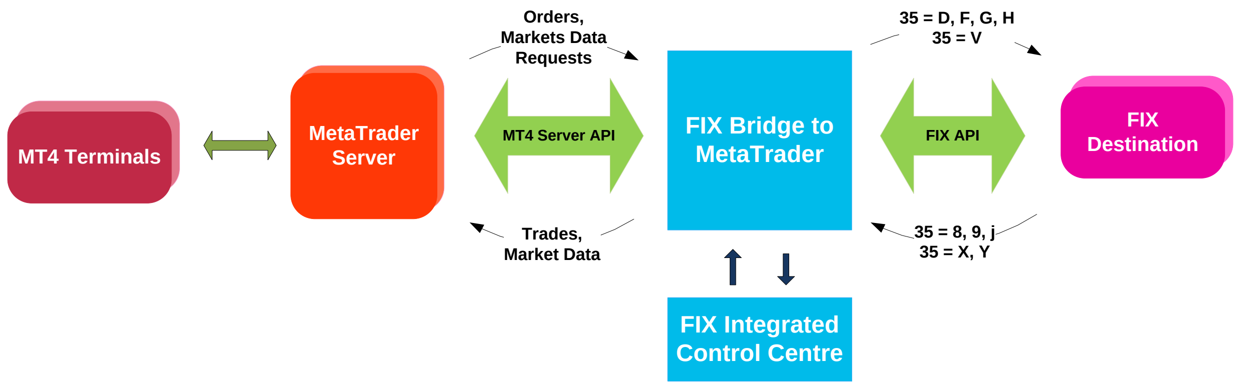
1. Overview

FIX Bridge for MetaTrader 4 (MT4) platform (www.metaquotes.net) allows attaching MetaTrader system to FIX compliant destinations, such as brokers, exchanges, ECN. It supports all asset classes and order types which are available in MetaTrader 4. FIX Bridge also allows customization of order types specific for a FIX destination.

FIX Bridge is actually a plug-in for Meta Trader Server that allows transmitting financial data to FIX sessions (to ECN FIX Servers) and vice-versa.

FIX Bridge plug-in ships Market or Pending orders and actions with these orders (Modify, Cancel, Delete, Stop Loss, Take Profit) to other FIX-systems platforms and vice-versa. FIX Bridge also receives and analyzes Execution Reports from other FIX-system platforms through FIX-protocol.

Scheme of work:



2. Installation

2.1. Requirements

Hardware Requirements

- 2.0 GHz CPU (or faster)
- 2 GB RAM (or more)
- 250 GB of available disk space (required for logs)

Note: As size of files with FIX logs grows, the situation when there is not enough free space on the hard drive can occur. FIX logs should be archived by FIX Bridge user.

Software Requirements

- Microsoft Windows Server 2008 or later
- MetaQuotes MetaTrader 4 Platform (MetaTrader 4 Server, MetaTrader 4 Administrator, etc.)
- Microsoft Visual C++ 2010 Redistributable Package

2.2. Installation package

FIX Bridge installation package contains the following files:

File	Description
additional.xml	The FIX Engine parser extra-configuration file. Is used to declare custom (user-defined) FIX fields and messages.
mtfix.cfg	FIX Bridge main configuration file.
mtfix_st1_vs2010.dll	Main FIX Bridge plugin file.
mtfix.engine	FIX Engine configuration file (see the FIX Antenna documentation for details).
mtfix.license	FIX Engine license file.
fixdicXX.xml	FIX dictionary files for FIX protocols 4.2, 4.3, 4.4, 5.0, 5.0SP2
V12-vc10-MD-x32.dll	FIX Engine library

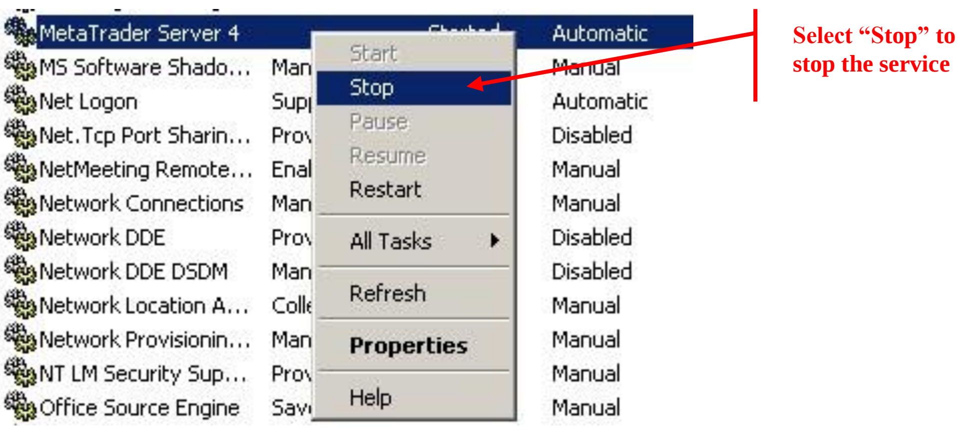
2.3 Installation process

1. Stop MT4 Server and close MT4 applications

Prior to FIX Bridge installation:
- All MT4 Applications must be closed.
- ***MT4 Server*** must be stopped.

To Stop MT4 Server:

Open Windows Services: ***Control Panel → Administrative Tools → Services.***
Select ***MetaTrader Server 4*** in the grid. If the value in "Status" field = Started, right-click on the service and select ***"Stop"*** from the pop-up menu. The Server will be stopped.



2. Copy FIX Bridge files

1) Find the MT4 Server installation directory ***<MT4Root>*** (typically is *C:\MetaTraderServer4*).
2) Copy ***"V12-vc10-MD-x32.dll"*** from the installation package to the ***<MT4Root>*** directory.
3) Copy all other files from the installation package to the ***<MT4Root>\plugins*** directory.
4) Create "logs" subfolder in the ***<MT4Root>\plugins*** directory.
5) Create "backup" subfolder in the ***<MT4Root>\plugins\logs*** directory.
6) Open the file ***<MT4Root>\plugins\mtfix.engine***. Make sure that the ***EngineRoot*** property is set to the ***<MT4Root>\plugins*** value.

Note: To ensure the correct processing of configuration file use slash (**/**) or doubled backslash (****) instead of backslash (****) in file system paths in configuration files.

Note: To ensure the correct start of FIX Bridge the ***<MT4Root>\plugins\logs\backup*** directory **MUST** be created.

Note: You should use the datafeed for quotes receiving.

3. Configure FIX Bridge

Edit the ***mtfix.cfg*** file in the ***<MT4Root>\plugins*** directory with any appropriate text editor. See Page 9 for detailed information about configuration parameters.

4. Start MetaTrader 4 Server

After FIX Bridge has been installed and configured start the MT4 Server.

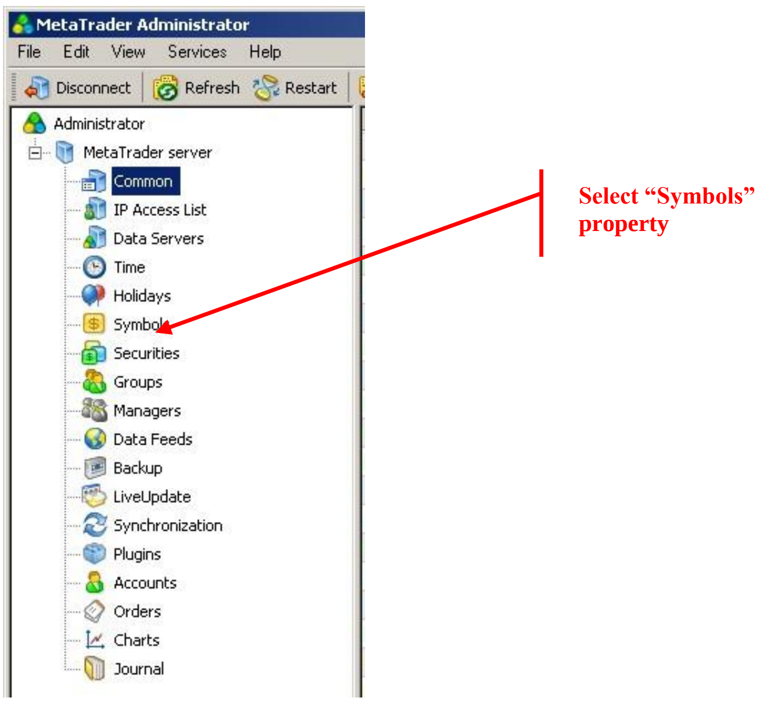
Open Windows Services: Control Panel → Administrative Tools → Services.
Select ***MetaTrader Server 4*** in the grid. Right-click it to open the pop-up menu and select ***"Start"***.

The Server will be started.

5. Change MT4 Server settings

Launch MetaTrader4 Administrator
Connect to Metatrader Server. MT4 Server properties are available for editing.

<u>Change "Symbols" property</u>

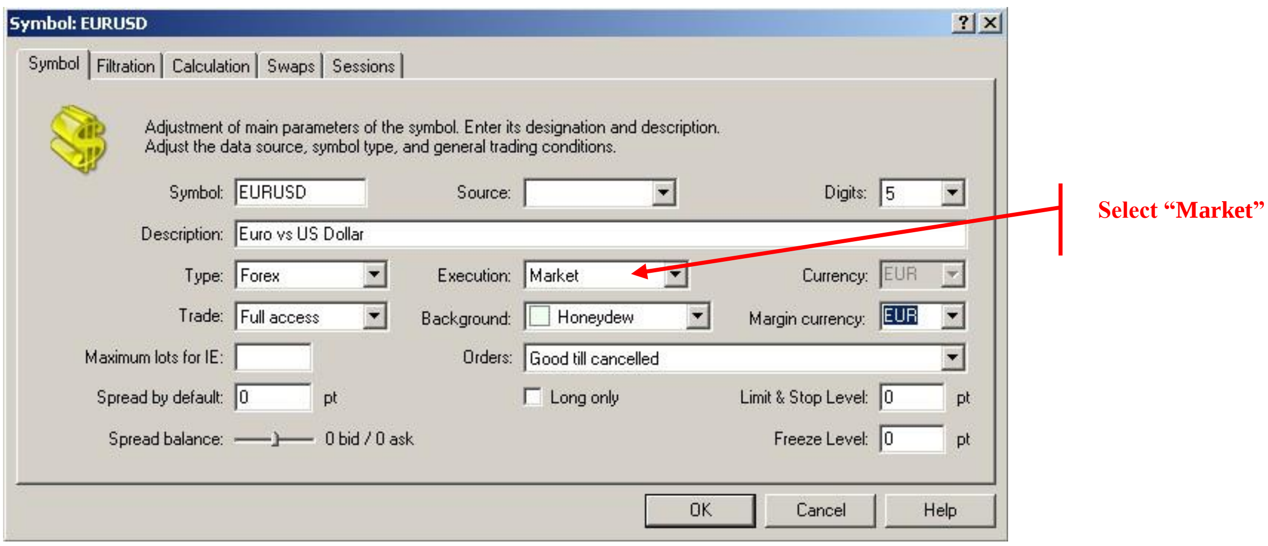


The list of currencies appears.

Symbol	Type	Execution	Filter	Spread	Stops	Long	Short	Digits	Trade
EURUSD	Forex	Market	No	No	0	0.1500	-0.5100	5	Full
GBPUSD	Forex	Market	No	No	0	0.8000	-1.5200	5	Full
USDCHF	Forex	Market	No	No	0	0.0400	-0.3700	5	Full
USDJPY	Forex	Market	No	No	0	0.6100	-1.2900	5	Full
USDCAD	Forex	Market	No	No	0	-0.3600	0.0600	5	Full
AUDUSD	Forex	Market	No	No	0	0.6700	-0.9700	5	Full
NZDUSD	Forex	Market	No	No	0	0.8700	-1.0900	5	Full
EURAUD	Forex	Market	No	No	0	-1.3600	0.8200	5	Full
EURCHF	Forex	Market	No	No	0	0.4700	-0.9100	5	Full
EURGBP	Forex	Market	No	No	0	-0.4900	0.2500	5	Full
EURJPY	Forex	Market	No	No	0	1.2000	-1.9800	5	Full
EURCAD	Forex	Market	No	No	0	-0.3000	-0.2700	5	Full
GBPCHF	Forex	Market	No	No	0	1.4000	-2.0300	5	Full
GBPJPY	Forex	Market	No	No	0	2.1000	-3.2100	5	Full
AUDCHF	Forex	Market	No	No	0	0.9900	-1.3100	5	Full
AUDJPY	Forex	Market	No	No	0	1.3900	-1.8300	5	Full
AUDNZD	Forex	Market	No	No	0	-0.7000	0.3400	5	Full
AUDCAD	Forex	Market	No	No	0	0.5100	-0.7800	5	Full
CADCHF	Forex	Market	No	No	0	0.3000	-0.5900	5	Full
CADJPY	Forex	Market	No	No	0	0.8000	-1.2800	5	Full
NZDJPY	Forex	Market	No	No	0	1.6500	-2.0500	5	Full
USDNOK	Forex	Market	No	No	0	-4.4600	2.0700	5	Full
USDZAR	Forex	Market	No	No	0	-16.4700	11.5100	5	Full
USDSEK	Forex	Market	No	No	0	-2.5000	0.5000	5	Full
USDDKK	Forex	Market	No	No	0	-1.7800	0.2400	5	Full
USDSGD	Forex	Market	No	No	0	0.4000	-0.8900	5	Full
CHFJPY	Forex	Market	No	No	0	0.3100	-0.7400	5	Full
EURNZD	Forex	Market	No	No	0	-2.5000	1.8600	5	Full
GOLD	Commodities	Market	No	No	0	-3.3500	1.0600	5	Full
SILVER	Commodities	Market	No	No	0	-3.3500	1.0600	5	Full

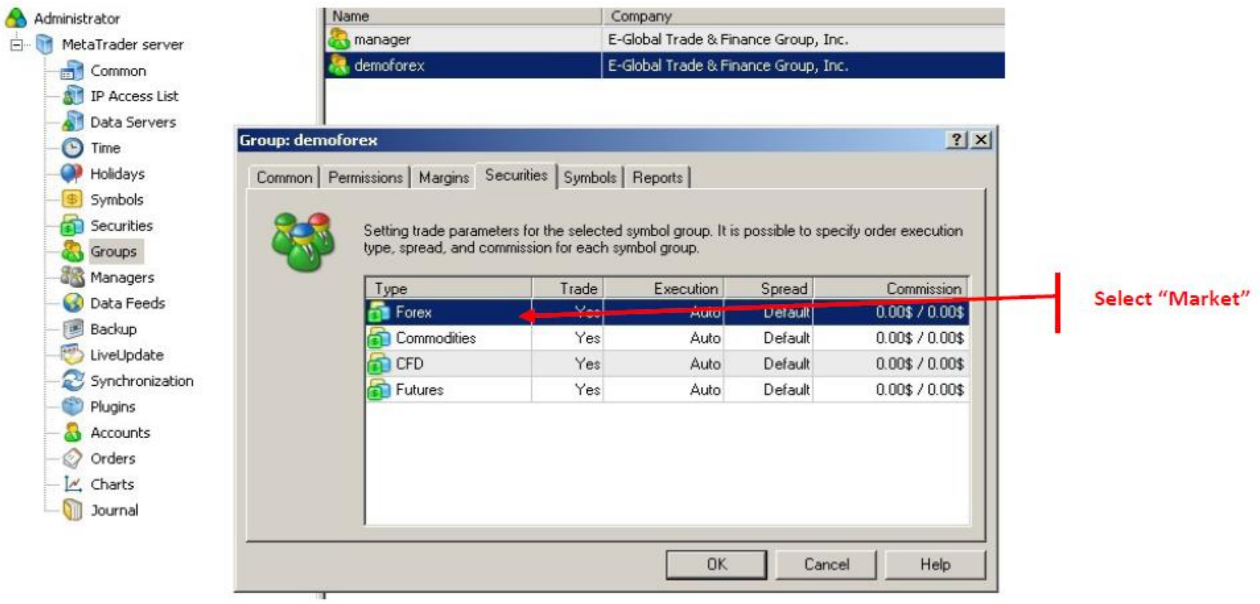
For each currency pair Execution should be set to “Market”:

1. Select a currency pair
2. Double click on the currency
3. The window with Symbol Settings appears (see the picture bellow)
4. Select “Market” value in the **Execution** drop-down
5. Click the “Ok” button to save changes
6. Go to the next currency pair

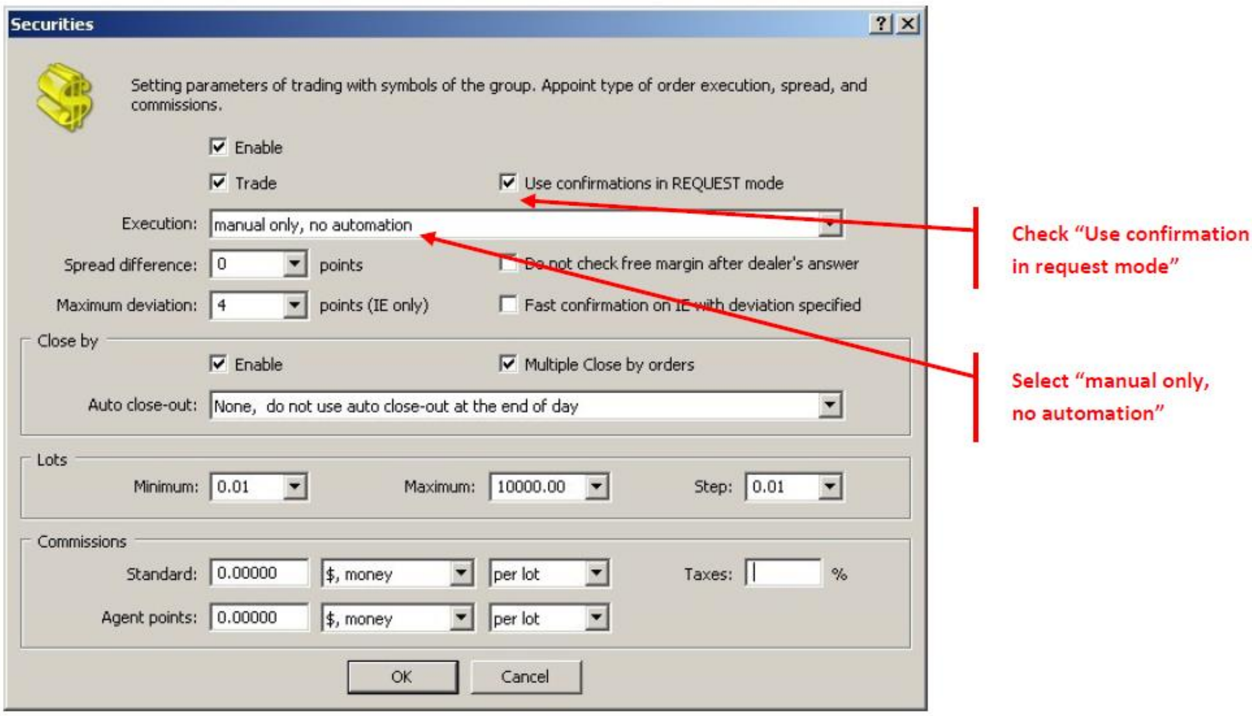


<u>Change "Groups" property</u>

Select "Securities" tab. Select "Forex" in the tab and double click on it.



- Check the "Use confirmations in REQUEST mode" box



- Execution: Manual only, no automation

3. Configuration settings

FIX Bridge main configuration file ***mtfix.cfg*** has a simple structure similar to standard *.**ini** configuration files. All lines in the file begin with # are treated as comments and don't affect FIX Bridge.

3.1 General configuration settings

Option	Accepted Values	Description
ManagerLogin	string	MT4 manager's login
ManagerPass	string	MT4 manager's password
ShutdownTime	time of day (MT4 server time)	Time when FIX Bridge stops accepting user requests before reset of FIX sessions sequence numbers. ***Note***: MUST BE less than ***SeqNumResetTime*** *For example:* 22:59
SeqNoResetTime	time of day (MT4 server time)	Time of FIX sessions sequence numbers reset at the end of the day. *For example:* 23:00
EndOfDay	time of day (MT4 server time)	Time when end of day activities are performed (backup database etc.). *For example:* 23:02
DestinationType	string	Specifies ECN type. Supported values are: 'ADS' for ADS securities (http://www.ads-securities.com) 'CME' for CME iLink (http://www.cmegroup.com/) 'CNX' for Currenex (http://www.currenex.com) 'FXCM' for FXCM (http://www.fxcm.com) 'GENERIC' for generic FIX ECN 'LMAX' for LMAX (http://www.lmax.com/) 'MBT' for MB Trading (http://www.mbtrading.com) 'OANDA' for Oanda (http://www.currenex.com) 'SAXO' for Saxo Bank (http://saxobank.com) 'SMART' for SmartTrade (http://www.smart-trade.net/)
Account	number	Client's account on the destination. ***Note***: MUST BE changed for your system!
AccountWithoutCheckSession	number	If empty – then Bridge waiting for 35=h

		messages (session opened)
TradingSessionStatusOpenAlways	true | false	If true then disable the Trading session status checking functionality
LogFolder	string	Contains path to the folder, where bridge will create log files. Folder must exist
FIXProcessingLog	true | false	If true, bridge creates additional log file and writes information about what FIX messages were received
OrderQtyInNominalUnits	true | false	If true, Bridge uses orderQty in nominal units at FIX messages, otherwise orderQty is in absolute units (multiplied by contractSize value)
AccountConfigXMLFile	string	Path and file name with ECN account configuration settings
UseMT4Currency	true | false	By default is false. If true, bridge will currency which is configured in MT4 in the FIX messages
ProcessMissedOrdersImmediatelly	true | false	By default is false. If true, bridge submits missed TP/SL orders again when TP/SL price is reached.
LastSeqNoReset	number	Last sequence before rollover (SeqNumResetTime)
AgentPercentage	value	Agent commission
AlertsRecipients	MT4 server managers accounts	FIX session state information for managers
FilterGroupsByManager	true | false	FilterGroupsByManager excludes messages from users not included in the manager group
ForceAvgPx	true | false	Ability to calculate real average price
Mode	Net | Hedge	Execution mechanism
Mode24x7	true | false	Ability to work 24h7d. If true then daily EOD procedure will be disabled (FIX session's disconnect, reset seqNums, etc)
OlderDaysRepost	value	This parameter specifies the number of days for send new orders in place of old
TPAlwaysLimit	true | false	TakeProfit with StopLoss always will send Limit order to Destination
Time2CheckGTC	time of day (MT4 server time)	Time to check GoodTillCancel orders. If expired reposted due next setting
OlderDaysRepost	value in day	The GTC orders repost
ExecutionReport.commission	true | false	If true then takes commission from 12 tag of ExecutionReport (8) message
SymbolAsIs	true | false	I true then mt4 symbol correspond to Symbol (55) tag of FIX message as is (for example: EURUSD.b (mt4) will be 55=EURUSD.b (FIX)
TakeCommission	true | false	If false then swith on the standart mt4 commission functionality, otherwise Bridge calculates commission
MarketOrderTIFValue	string	By default is 1(GTC). It could be used to

		define custom TIF value for the outgoing Market orders
AcceptRejectsNoSuchOrder	true | false	When bridge receives OrderCancelReject FIX message with RejectNoSuchOrderReason reason, OrderCancel request will be processed as accepted and order will be cancelled. This functionality is used to remove order, if it doesn't exist on the FIX Exchange side (for example it was deleted at EOD). By default is false
RejectNoSuchOrderReason	string	Contains reject reason used by the AcceptRejectsNoSuchOrder option. Default value is "No such order"
TransformMT4FIXSymbols	true | false	When false, bridge doesn't change name of symbols (from FIX to MT4 and back). Default value is true
CollateralInquiryEnabled	true | false	When false, bridge doesn't send CollateralInquiry(BB) FIX message into the FIX session (when new MT4 user is logged in). Default value is true
CheckHighLowPriceLimit	true | false	By default is false. If true, bridge checks order’s price for the limits configured for this symbol (at SecurityMaster). Order is rejected if order’s price violates configured limits
UseManagerTraderId4ManagerOrders	true | false	By default is false. If true and order was submitted for the trader by manager account, bridge adds Manager’s TraderId and TraderRole to the FIX Parties group
EmailFinalSettlementPrice	true | false	By default is false. If true, bridge sends final Settlement price by email to the MT4
WriteAuxMessages2MT4Log	true | false	By default is false. If true, bridge writes description of the incoming PositionReport, SecurityList, DerivativeSecurityList, MarketDefinition, TradingSessionList, SecurityStatus and News FIX messages to the MT4 log
WriteMarketDataDetail2Log	true | false	By default is false. If true, Bridge prepares detailed marketData report using data from incoming FIX message
CustomFIXRules	string	Path and filename of the Custom FIX rules XML configuration
ExecType4Trade	string	By default is ‘F’. Value of the ExecType(150) FIX tag that means Trade
SendTradeServerStatusByEmail	true | false	By default is false. If true, Bridge sends TradeSessionStatusReq FIX message after each FIX session reconnect
TradingSessionStatus.PreOpenStatusAsOpen	true | false	By default is false. If true, Bridge process PreOpen tradeSessionStatus like Open

TradingSessionStatus.PreCloseStatusAsOpen	true | false	By default is false. If true, Bridge process PreClose tradeSessionStatus like Open
AccountOverride.Count AccountOverride.<id>.SourceAccounts AccountOverride.<id>.TargetAccount		These parameters allow configuring external accounts to be used in outgoing FIX messages. SourceAccounts contains list of MT4 accounts. TargetAccount contains ECN account
PassConvRateOnCheck	true | false	By default is false. If true, Bridge skips conversion rate when cost of trade is calculated
isOrderQtyInteger	true | false	By default is false. If true, Bridge sends OrderQty value as integer, otherwise OrderQty value as float
CalculateCommAtClose	true | false	By default is false. If true, Bridge calculates commission when position is closing
DefaultAgentAccount	number	By default is not configured. If configured, Bridge will calculate Agency commission for this MT4 account.
AdminPort	number	Destination connection Admin port for ShowState application
CorporateAction.ReloadTime	time in local timezone	Time in local time zone when Corporate Actions starts
ForceSeqNumReset	never onetime always	Defines SeqNum reset mode
BackupFolder	string	Indicates path to backup folder for moving log files upon rollover procedure
MT4AccountIdFIXTag	number	FIX tag number used for indicating. MT4 account ID in News(B) message
MT4PositionIdFIXTag	number	FIX tag number used for indicating MT4 position ID in News(B) message
MT4PositionNewsClOrderIdFIXTag	number	FIX tag number used for indicating ClientOrdID in News(B) FIX message
MT4PositionNewsOrigClOrderIdFIXTag	number	FIX tag number used for indicating OriginalClientOrdID in News(B) FIX message.
SendMT4PositionNewsAfterRequest	true | false	By default is false. If true, Bridge will send News(B) message after each order
SendMT4PositionNewsAfterTrade	true | false	By default is false. If true, Bridge will send News(B) message after each trade
TakeMT4AgentCommissionTPSL	true | false	If true, Bridge will take agent commission when position is closed due to TP or SL
TradeAmendment.ReloadTime	time in local timezone	Time for Trade Amendment

3.2. Order flow session's configuration settings

Option	Accepted Values	Description
Host	hostname | ip-address	Destination connection hostname or ip-address.
Port	number	The destination connection port number.

BackupHost.<number>	hostname | ip-address	Destination connection hostname or ip-address for backup connection. Bridge allows to configure multiple backup connections
BackupPort.<number>	number	The destination connection port number for backup connection. Bridge allows to configure multiple backup connections
Sender	string	Specifies SenderCompId identifier for the FIX session.
Target	string	Specifies TargetCompId identifier for the FIX session.
User	string	User name
Password	string	User password. Can be omited if the destination doesn't require authorization.
HBI	number of seconds	Specifies Heart Beat Interval common for all FIX sessions. This value can be overridden by a session settings.
StartSeqNum	number	Overrides first expected incoming message sequence number with specified one.
OrderFlowSsnDisabled	true | false	If true, OrderFlow session is disabled and bridge doesn't create it
HandleNextExpectedSeqNumInLogon	true | false	By default is false. If true, bridge process NextExpectedMsgSeqNum(789) tag in Logon(A) FIX message
SuppressDoubleResendRequest	true | false	By defauls is false. If true, bridge doesn't send sequence of the ResendReq(2) FIX messages, in case of the sequenceGap problem.
ThrottlingMsgAmount	number	By default is 0 – disabled. If values > 0, bridge doesn't allow to send more than ThrottlingMsgAmount FIX messages into the FIX session
ClearingId	string	Clearing Firm ID, transmitted with PartyGroup as PartyID (448) when PartyRole (452) = '4'(Clearing Firm). By default is empty
ClearingRole	string	Corresponds to PartyRole (452) tag. By default is empty
DayOrder.TradeSsnID	string	TradingSessionID (336) for Time in force type Day order
DayOrder.TradeSsnSubID	string	TradingSessionSubID (625) for Time in force type Day order
GTDateOrder.ExpireDateTime	UTC Timestamp	Order Expiration date in format: YYYYMMDD or YYYYMMDD-HH:MM:SS
LimitOrderTIFValue	number	By default is 1(GTC). Used to define custom TIF value for the outgoing Limit orders
StopOrderTIFValue	number	TimeInForce (59) value for stop orders. By default is empty
StopSupportedByECN	true | false	By default is false. Parameter indicates if ECN supports stop orders. If set to false, tag OrdType (40) will have value "2", otherwise - current order value
StopsDeleteAtNoMargin	true | false	By default is false. Defines if Stop orders should be deleted at NoMargin. Used in Buying Power

		check
PartyId	string	Corresponds to PartyID (448) FIX tag
PartyIdSource	string	Corresponds to PartyIDSource (477) FIX tag
PartyRole	number	Corrspond to PartyRole (452) FIX tag
TraderId	string	Correspond to PartyID (448) tag for Trader. By default is empty
TraderRole	number	Correspond to PartyRole (452). By default is empty
PendingsCloseAtMarginCall	true | false	Defines if Pending Orders should be closed on MarginCall
SpreadMode	true | false	Used for correction price of spread orders - decrease for Sell and increase for Buy
StopOrderResendEveryDay	true | false	By default is false. If true, resends Stop Orders every day

3.3 Market data configuration settings

Option	Accepted Values	Description
FIXMarketData	true | false	Main parameter for swith on the market data receiving
FIXProtocolMarketData	FIX44	FIX protocol version for market data
ParallelMarketDataProcessing	true | false	Ability use the additional threads for market data processing
MarketDataProcessorWorkers	value	Number of additional threads
MarketDataLog	true | false	Ability to have the additional market data log (created in \plugins directory)
FIXMarketData.Sessions	string	Contains list of the names for marketData FIX sessions. Parameter is required, if several marketData FIX sessions need to be defined.
FIXMarketData.<name>.Protocol FIXMarketData.<name>.Host FIXMarketData.<name>.Port FIXMarketData.<name>.BackupHost.<id> FIXMarketData.<name>.BackupPort.<id> FIXMarketData.<name>.Sender FIXMarketData.<name>.Target FIXMarketData.<name>.User FIXMarketData.<name>.Password FIXMarketData.<name>.SenderSubId FIXMarketData.<name>.SenderLocationId FIXMarketData.<name>.TargetSubId FIXMarketData.<name>.TargetLocationId FIXMarketData.<name>.Persistent	Values	Parameters for <name> market data FIX session. Parameters are applied if FIXMarketData.Sessions is defined. User need to define parameters for every <name> in the FIXMarketData.Sessions list
FIXMarketData.<name>.SendTradeSsnStatusRequest	true | false	If true, Bridge will send TradeSessionStatusReq FIX

		message to this marketData FIX session.
FIXMarketData.<name>.SendSecListRequest	true | false	If true, Bridge will send SecurityListReq FIX message to this marketData FIX session.
FIXMarketData.<name>.SendSecurityDefinitionRequest	true | false	If true, Bridge will send SecurityDefinitionReq FIX message to this marketData FIX session.
FIXMarketData.<name>.SendDerivativeSecurityListRequest	true | false	If true, Bridge will send DerivativeSecurityListReq FIX message to this marketData FIX session.
FIXMarketData.<name>.SendSecurityStatusRequest	true | false	If true, Bridge will send SecurityStatusReq FIX message to this marketData FIX session.
FIXMarketData.<name>.SendNewsRequest	true | false	If true, Bridge will send NewReq FIX message to this marketData FIX session.
FIXMarketData.<name>.SendMDSubscriptions	true | false	If true, Bridge will send MDSubscriptionReq FIX messages to this marketData FIX session.
FIXMarketData.<name>.SendMarketDefinitionRequest	true | false	If true, Bridge will send MarketDefinitionReq FIX message to this marketData FIX session.
FIXMarketData.<name>.SymbolList	values	Parameter contain list of the MT4 symbols that could be used for this marketData FIX session. '*' means any MT4 symbol.
FIXMarketData.<name>.ThrottlingMsgAmount	number	By default is 0 – disabled. If values > 0, bridge doesn't allow to send more than ThrottlingMsgAmount FIX messages into the FIX session
FIXMarketData.SeparateSession	true | false	If true, bridge uses separate FIX session for trading and market data. Parameter is applied if FIXMarketData.Sessions is not defined.
FIXMarketData.Host FIXMarketData.Port FIXMarketData.Sender FIXMarketData.Target FIXMarketData.User	values	Parameters for market data FIX session. Parameters are applied if FIXMarketData.Sessions is not defined.

FIXMarketData.Password		
MDPersistent	true | false	If true, bridge will create persistent session's logs for marketData session. Parameter is applied if FIXMarketData.Sessions is not defined.
FIXProtocolMarketData	FIX44	FIX version for market data session. Parameter is applied if FIXMarketData.Sessions is not defined.
SendSnapshotMDSubscription	true | false	By default is 'false'. When true bridge will send MarketDataSubscription(V) FIX message with MDUpdateType=0(Full refresh).
WriteMarketData2MT4Log	true | false	By default is false. When true, bridge writes marketdata description to the MT4 log
FIXMarketData.SeparateSession	true | false	Separate trading and market data FIX session

3.4. Outgoing FIX requests configuration settings

Option	Accepted Values	Description
UseSecurityListRequest	true | false	By default is false. If true, bridge sends SecurityListRequest (x) FIX message to the FIX session configured at SendSecurityListRequest2Ssn parameter
SecurityListRequestType	string	Defines value for the SecurityListRequestType(559) tag in SecurityListRequest (x) FIX message.
SecurityListRequestAllSecurities	true | false	By default is false. If true, bridge creates one SecurityListRequest (x) FIX message to subscribe for all instruments, therwise bridge will create separate SecurityListRequest (x) FIX message for each market.
SendSecurityListRequest2Ssn	string	Name of the FIX session
UseDerivativeSecurityListRequest	true | false	By default is false. If true, Bridge prepares and sends DerivativeSecurityListReq FIX message
DerivSecurityListRequestType	string	Defines value for the SecurityListRequestType(559) tag in DerivativeSecurityListReq FIX message.
DerivSecurityListRequestAllSecurities	true | false	By default is false. If true, bridge creates one DerivativeSecurityListReq FIX message to subscribe for all instruments, therwise bridge will create separate DerivativeSecurityListReq FIX message for each market.
SendDerivativeSecurityListRequest2Ssn	string	Name of the FIX session
UseMarketDefinitionRequest	true | false	By default is false. If true, Bridge prepares and sends MarketDefinitionReq FIX message

SendMarketDefinitionRequest2Ssn	string	Name of the FIX session
UseTradingSessionListRequest	true | false	By default is false. If true, Bridge prepares and sends TradingSessionListReq FIX message
UseSecurityStatusRequest	true | false	By default is false. If true, Bridge prepares and sends SecurityStatusReq FIX message
UseNewsRequest	true | false	By default is false. If true, Bridge prepares and sends NewsReq FIX message
UsePositionRequest	true | false	By default is false. If true, Bridge prepares and sends PositionReq(AN) FIX message
PositionRequestType	string	Parameter contains value for the PosReqType FIX tag, whis is used in the PositionReq(AN) FIX message.
PublishSecStatusAsNews	true | false	By default is false. If true, Bridge publishes SecStatus description as MT4 News
PublishTradeSsnStatusAsNews	true | false	By default is false. If true, Bridge publishes TradeSessionStatus description as MT4 News
SendOrderQtyInOrderCancelReq	true | false	By default is false. If true, Bridge publishes OrderQty value ay OrderCancelReq FIX message
UseOrderStatusReq4Recovery	true | false	By default is false. If true, Bridge sends OrderStatusReq(H) FIX messages when session is established

3.5. SecurityMaster configuration settings

Option	Accepted Values	Description
SecurityMasterXMLFile	string	Path and file name with SecurityMaster XML configuration settings
SecMasterUpdateTime	time in local timezone	Time to update the securityDef.xml

3.6. MarketDefinition configuration settings

Option	Accepted Values	Description
MarketDefinition.XMLFile	string	Path and file name with MarketDefinition XML configuration settings

Notes:
For safe HDD space switch to “false” following settings:
MDPersistent=false
MarketDataLog=false

4. FIX Component Blocks

FIX protocol and set of the FIX messages, used by the MTFIXBridge, is depend on the configured value of DestinationType parameter. MTFIXBridge supports customizations for the different ECNs and Exchanges, like CME, MBT, SAXO, etc. Generic FIX protocol is following:

4.1 Standard Message Header

Tag	Field name	Req'd	Comments
8	BeginString	Y	Identifies beginning of new message and protocol version. Always first field in message. Supported values: 'FIX.4.2', 'FIX.4.3', 'FIX.4.4', 'FIXT.1.1'
9	BodyLength	Y	Message length, in bytes, forward to the CheckSum field. Always second field in message
35	MsgType	Y	Defines message type. Supported values: • 'D' – New Order Single • 'F' – Order Cancel Request • 'G' – Order Cancel/Replace Request • '8' - Execution Report • '9' – Order Cancel Reject
49	SenderCompID	Y	Assigned value used to identify firm sending message
56	TargetCompID	Y	Assigned value used to identify receiving firm
34	MsgSeqNum	Y	Integer message sequence number
50	SenderSubID	N	Assigned value used to identify specific message originator
57	TargetSubID	N	Assigned value used to identify specific individual or unit intended to receive message
43	PossDupFlag	N	Indicates possible retransmission of message with this sequence number. Supported values: 'Y', 'N'
97	PossResend	N	Indicates that message may contain information that has been sent under another sequence number. Supported values: 'Y', 'N'
52	SendingTime	Y	Time of message transmission (expressed in UTC). Format: YYYYMMDD-HH:MM:SS.sss
122	OrigSendingTime	N	Original time of message transmission when transmitting messages as the result of resend request (expressed in UTC). Format: YYYYMMDD-HH:MM:SS.sss
116	OnBehalfOfSubID	N	Contains MT4 account

4.2 Standard Message Trailer

Tag	Field name	Req'd	Comments
10	CheckSum	Y	Three byte, simple checksum. Always last field in message

4.3 Component Block <Parties>

Tag	Field name	Req'd	Comments
453	NoPartyIDs	N	Repeating group below should contain unique combinations of PartyID (448) , PartyIDSource (447) , and PartyRole (452)
448	PartyID	C	Used to identify source of PartyID. Required if PartyIDSource (447) is specified. Required if NoPartyIDs (453) > 0
447	PartyIDSource	C	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID (448) is specified. Required if NoPartyIDs (453) > 0
452	PartyRole	C	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs (453) > 0

5. FIX Session Level Messages

5.1 Logon

The Logon (A) message authenticates a user establishing a connection. The Logon (A) message must be the first message sent by the application requesting to initiate a FIX session.

From FIX Client to MT4 Server Plug-in

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'A'
98	EncryptMethod	Y	Method of encryption. Supported value: ‘0’ (None)
108	HeartBtInt	Y	Heartbeat interval (seconds)
553	Username	N	Username assigned by MT4
554	Password	N	Password.
<StandardMessageTrailer>		Y	

From MT4 Server to FIX Client Plug-in. Successful Logon

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'A'
98	EncryptMethod	Y	Method of encryption. Supported value: ‘0’ (None)
108	HeartBtInt	Y	Heartbeat interval (seconds)
<StandardMessageTrailer>		Y	

5.2 Logout

The Logout (5) message is used to close established session or to notify about failed authentication.

From FIX Client to MT4 Server Plug-in

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = '5'
58	Text	N	Free format text.
<StandardMessageTrailer>		Y	

From MT4 Server to FIX Client Plug-in. Successful Logout

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = '5'
58	Text	N	‘Logged out’

<StandardMessageTrailer>		Y	

From MT4 Server to FIX Client Plug-in. Failed Logon

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = '5'
58	Text	N	‘User authentication failed’
<StandardMessageTrailer>		Y	

5.3 Session Level Reject

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = '3'
45	RefSeqNum	N	MsgSeqNum of rejected message
372	RefMsgType	Y	MsgType of rejected message
373	SessionRejectReason	N	Code to identify reason for a session-level Reject (3) message. Valid values: ‘1’ (Required field is missing), ‘6’ (Incorrect data format for value)
58	Text	N	Reason of rejection.
<StandardMessageTrailer>		Y	

6. FIX Application Level Messages

6.1 New Order — Single

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'D'
1	Account	Y	Account mnemonic
11	ClOrdID	Y	Unique identifier for order as assigned by institution.
526	SecondaryClOrdID	N	Position ID assigned by MetaTrader. **Note**: Field is required to outline ID of position which should be closed. (Can be used only when MT4 Plug-In is configured to treat all new orders as different as different positions)
21	HandlInst	Y	Instructions for order handling on broker trading floor. Supported values: '1'
<OrderQtyData>			
38	OrderQty	Y	Number of lots
</OrderQtyData>			
40	OrdType	Y	Order type. Supported values: '1' (Market), '2' (Limit)
44	Price	C	Limit price. Required if OrdType (40) = '2'
54	Side	Y	Side of order. Supported values: '1' (Buy), '2' (Sell)
<Instrument>			
55	Symbol	Y	Currency pair
</Instrument>			
58	Text	N	Free format text string
<Parties/>			
59	TimeInForce	N	Specifies how long the order remains in effect. Supported values: '0' - Day (or session) ‘1’ - Good Till Cancel (GTC) ‘6’ – Good Till Date (GTD) Absence of this field is interpreted as '0'
126	ExpireTime	N	Time/Date of order expiration. Conditionally required if TimeInForce (59) = ‘6’ (GTD)
60	TransactTime	Y	Time of execution/order creation (expressed in UTC). Format: YYYYMMDD-HH:MM:SS.sss or YYYYMMDD-HH:MM:SS
21	HandlInst		Instructions for order handling Supported values: ‘1’
100	ExDestination	Y	Contains value of Destination, configured in the

			mtfix.cfg file
<StandardMessageTrailer>		Y	

6.2 Order Cancel Request

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'F'
1	Account	Y	Account mnemonic
11	ClOrdID	Y	Unique identifier for order as assigned by institution. Must be unique across all sessions/clients!
526	SecondaryClOrdID	N	Position ID assigned by MetaTrader. **Note**: Field is can be used only by Extended FIX Client
41	OrigClOrdID	Y	ClOrdID of the previous order as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests
37	OrderID	N	Unique identifier for order as assigned by broker
<OrderQtyData>			
38	OrderQty	Y	Number of lots
</OrderQtyData>			
54	Side	Y	Side of order. Supported values: '1' (Buy), '2' (Sell)
<Instrument>			
55	Symbol	Y	Currency pair
</Instrument>			
60	TransactTime	C	Time of execution/order creation (expressed in UTC). Format: YYYYMMDD-HH:MM:SS.sss
<Parties/>			
<StandardMessageTrailer>		Y	

6.3 Order Cancel/Replace Request

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'G'
1	Account	Y	Account mnemonic
11	ClOrdID	Y	Unique identifier for order as assigned by institution. Must be unique across all sessions/clients!
526	SecondaryClOrdID	N	Position ID assigned by MetaTrader. **Note**: Field is can be used only by Extended FIX Client
41	OrigClOrdID	Y	ClOrdID of the previous order as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests
21	HandlInst	Y	Instructions for order handling on broker trading floor. Supported values: '1'

37	OrderID	N	Unique identifier for order as assigned by broker
<OrderQtyData>			
38	OrderQty	Y	Number of lots
</OrderQtyData>			
40	OrdType	Y	Order type. As specified in original order Supported values: '1' (Market), '2' (Limit)
44	Price	C	Limit price. Required if OrdType (40) = '2'
54	Side	Y	Side of order. As specified in original order Supported values: ‘1’ (Buy), ‘2’ (Sell)
<Instrument>			
55	Symbol	Y	Currency pair
</Instrument>			
58	Text	N	Free format text string
<Parties/>			
59	TimeInForce	N	Specifies how long the order remains in effect. Supported values: '0' - Day (or session) ‘1’ - Good Till Cancel (GTC) ‘6’ – Good Till Date (GTD) Absence of this field is interpreted as '0'
126	ExpireTime	N	Time/Date of order expiration. Conditionally required if TimeInForce (59) = ‘6’ (GTD)
60	TransactTime	Y	Time of execution/order creation (expressed in UTC). Format: YYYYMMDD-HH:MM:SS.sss
<StandardMessageTrailer>		Y	

6.4 Execution Report

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = '8'
1	Account	Y	Account mnemonic
37	OrderID	Y	Unique identifier for order as assigned by MT4 Plug-in
11	ClOrdID	Y	Unique identifier for order as assigned by institution **Note**: *ClOrdID (11) should have following format for the case when Execution Report is sent as a result of unsolicited position closing on MT4 side: “ClosedByMT4_N_YYYYMMDD:HH:MM:SS.sss”, where N – number of unsolicited position closing; YYYYMMDD:HH:MM:SS.sss – timestamp.*

526	SecondaryClOrdID	N	Unique position ID. As assigned by MT4. **Note**: *Field is can be used for Extended FIX Client only*
41	OrigClOrdID	N	ClOrdID of the previous order as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests
17	ExecID	Y	Unique identifier of execution message as assigned by MT4 Plug-in
<Parties/>			
6	AvgPx	Y	Calculated average price of all fills on this order
14	CumQty	Y	Total number of shares filled
19	ExecRefID	C	ExecID of the trade that was canceled. Required if ExecType (150) = ‘H’
21	HandlInst	N	Instructions for order handling on broker trading floor. Supported values: '1'
31	LastPx	C	Price of this (last) fill. Required if ExecType (150) = 'F'
32	LastQty	C	Quantity of shares bought/sold on this (last) fill. Required if ExecType (150) = 'F'
38	OrderQty	Y	Number of lots Not required if ExecType (150) = 'I'
39	OrdStatus	Y	Current status of order. Supported values: '0', '1', '2', '4', '8'~~, 'C'~~
40	OrdType	N	Order type. Supported values: ‘1’, ‘2’, ‘3’, ‘J’
44	Price	C	Limit price. Required if OrdType (40) = '2', 'J’
54	Side	Y	Side of order. Supported values: ‘1’, ‘2’
55	Symbol	Y	Currency pair
58	Text	N	Free format text string
59	TimeInForce	N	Specifies how long the order remains in effect. Supported values: '0' - Day (or session) ‘1’ - Good Till Cancel (GTC) ‘6’ – Good Till Date (GTD) Absence of this field is interpreted as '0'
126	ExpireTime	N	Time/Date of order expiration. Conditionally required if TimeInForce (59) = ‘6’ (GTD)
60	TransactTime	N	Time of execution/order creation (expressed in UTC). Format: YYYYMMDD-HH:MM:SS.sss
75	TradeDate	N	Used when reporting other than current day trades
77	OpenClose	N	Indicates whether the resulting position after a trade should be an opening position or closing position. Supported values: 'O', 'C'

103	OrdRejReason	C	Code to identify reason for order rejection. Supported values: ‘0’, ‘4’, ‘5’, '6' Required if ExecType (150) = '8'
150	ExecType	Y	Identifies the type of execution report. Supported values: '0', ‘1', ‘4’, '5', '6', ‘F’, ‘H’
151	LeavesQty	Y	Amount of shares open for further execution
336	TradingSessionID	C	Is used to indicate that message is sent to DropCopy session. Valid value: ‘DropCopy’ Required for all messages sent to drop copy session(s).
625	TradingSessionSubID	C	Is used to indicate name of DropCopy session. Format: ‘<MainSessionTargetCompID, MainSessionSenderCompID >’. Required for all messages sent to drop copy session(s).
99	StopPx	N	Stop price
6	AvgPx	N	Calculated average price of all fills on this order.
12	Commission	N	Commission
<StandardMessageTrailer>		Y	

6.5 Order Cancel Reject

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = '9'
11	ClOrdID	Y	Unique identifier for order as assigned by institution
526	SecondaryClOrdID	N	Unique position ID assigned by MT4. **Note**: Field is can be used for Extended FIX Client only
37	OrderID	C	Unique identifier for order as assigned by MT4 Plug-in.
39	OrdStatus	Y	Current status of order. Supported values: '0', '1', '2', '4', '8', 'C', 'E'.
41	OrigClOrdID	Y	ClOrdID of the previous order as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests
58	Text	N	Free format text string
102	CxlRejReason	N	Code to identify reason for cancel rejection. Supported values: '0', '1', '2', '3', '6'
434	CxlRejResponseTo	Y	Supported values: '1' (corresponds to Order Cancel Request (F)) '2' (corresponds to Order Cancel/Replace Request (G))
<StandardMessageTrailer>		Y	

6.6 Market Data Request

Tag	Field name	Req'd	Comments

<StandardMessageHeader>		Y	MsgType = 'V'
262	MdReqID	Y	Unique identifier of Market Data Request message
263	SubscriptionRequestType	Y	Subscription Request Type Supported values: ‘1’ (Subscribe), ‘2’ (Unsubscribe)
264	MarketDepth	Y	Depth of market for Book Snapshot Supported value: ‘1’ (Top of the book)
265	MDUpdatetype	C	Specifies the type of Market Data update. Conditionally required if SubscriptionRequestType (263) = ‘1’ Supported value: ‘0’ (Full refresh)
267	NoMDEntryTypes	Y	Supported value: ‘2’
=>269	MDEntryType	Y	Type of market data entry. Supported values: ‘0’ (Bid); ‘1’ (Ask)
146	NoRelatedSym	Y	Number of currency pair which market data is requested for. Supported value: ‘1’
=>55	Symbol FIXField::AggregatedBook	Y	Currency pair
<StandardMessageTrailer>		Y	

6.7 Market Data Reject

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'Y'
262	MdReqID	Y	Unique identifier of Market Data Request message
281	MDReqRejReason	N	Reject reason. Supported values: '0' (Unknown symbol) '1' (Duplicate MDReqID (262)) '4' (Unsupported SubscriptionRequestType (263)) '5' (Unsupported MarketDepth (264)) '6' (Unsupported MDUpdateType (265)) '8' (Unsupported MDEntryType (269))
58	Text	N	Description of reject.
<StandardMessageTrailer>		Y	

6.8 Market Data Snapshot/Full Refresh

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'W'
262	MdReqID	Y	Unique identifier of Market Data Request message

55	Symbol	Y	Currency pair
268	NoMDEntries	Y	Supported value: ‘2’
=> 269	MDEntryType	Y	Market Data entry type. Supported values: ‘0’ (Bid), ’1’ (Offer)
=> 270	MDEnrtyPx	Y	Bid/Ask price
=> 271	MDEntrySize	Y	Bid/Ask size
=> 15	Currency	N	Can be used to specify the currency of the quoted price.
<StandardMessageTrailer>		Y	

6.9 Market Data Incremental Refresh

Tag	Field name	Req'd	Comments
<StandardMessageHeader>		Y	MsgType = 'X'
262	MdReqID	Y	Unique identifier of Market Data Request message
268	NoMDEntries	Y	Supported value: ‘2’
=> 279	MDUpdateAction	Y	Type of Market Data update action.
=> 269	MDEntryType	C	Market Data entry type. Supported values: ‘0’ (Bid), ’1’ (Offer)
55	Symbol	Y	Currency pair
=> 270	MDEnrtyPx	Y	Bid/Ask price
=> 271	MDEntrySize	Y	Bid/Ask size
=> 15	Currency	N	Can be used to specify the currency of the quoted price.
<StandardMessageTrailer>		Y	

7. Security Master

SecurityMaster is an auxiliary mechanism available in the MTFIXBridge. It was designed to define complex instruments (like Futures, Options, etc) available on the ECN side and link them with corresponding MT4 symbols. MTFIXBridge uses SecurityMaster's data to prepare correct outgoing FIX messages (Orders, MDSubscriptions, etc) and translate incoming ECN instruments to the MT4 Symbol.
Instrument's parameters are configured in the XML file, that has following format:

<Instruments>
    <Templates>
        Contains templates of rules, used to update outgoing FIX message or process incoming FIX message
    </Templates>
    <Instrument Name="*instrument name*" Destination="*ECN name*" Enabled="true">
        <Params>
            Contains Named ECN parameters for this instrument
        </Params>
        <MT4Params>
            Contains MT4 parameters for this instrument
        </MT4Params>
        <FIXParams>
            Contains references to the templates and *rules, used to update outgoing FIX message or process incoming FIX message*
        </FIXParams>
    </Instrument>
    <Instrument Name="*instrument name*" Destination="*ECN name*" Enabled="true">
        …
    </Instrument>
    …
</Instruments>

7.1 Named ECN Parameters section

Named ECN Parameters section contains list of the parameters and their values. Named parameters could be referenced in the FIX rules or templates.

Following named parameters are predefined:

- ClosePositionAtExpireDate – parameter defines how available MT4 positions are closed, when Final Settlement price is received. Value 'OptionPL': positions are closed by settlement price and profit/loss is calculated for options. Value 'FinalSettlementPrice': positions are closed by settlement price and profit/loss is calculated as configured in MT4.
- HighLimitPrice – contains HighLimit value for the order's price for Instrument, parameter is used during HighLowPriceCheck procedure;
- LowLimitPrice – contains LowLimit value for the order's price for Instrument, parameter is used during HighLowPriceCheck procedure;
- MarketId – contains Market name for Instrument, parameter is used to bind instrument with Market, defined by the MarketDefinition configuration;
- MarketSegId – contains MarketSegment name for Instrument, parameter is used to bind instrument with Market, defined by the MarketDefinition configuration;

- MaturityDate - contains MaturityDate for Instrument, parameter is used to execute Symbol expiration procedure (publish Final Settlement price, move related MT4 Symbol to the Expired group and disable trading);
- MDTypes – contains list of the market data types for the instrument, parameter is used when bridge prepares MarketDataSubscription(V) FIX message;
- MDTypeForCharts – defines marketData type (2 – Trade, 3 – Index, etx) that bridge pushes to the MT4 Charts;
- MDTypeForTicks – defines marketData type (supported values are ‘Trades’, ‘Index’, ‘BBO’) that bridge pushes to the MT4 tick chart;
- OptionType – parameter defines type of the option instrument. Value ‘1’ is used for CALL option; Value ‘0’ is used for PUT option. Parameter is required for ClosePositionAtExpireDate functionality;
- OrderTimeInForce – parameter is used for configuration of MT4 symbol and defines value of the ‘Order’ parameter. Values are ‘Daily’, ‘GTC’, ‘DailyNoStops’;
- TradingReferencePrice – parameter is used for configuration of MT4 symbol and contains reference price for instrument;
- StopTradeDateTime – parameter contains last trade date for the instrument.
- StrikePrice – parameter contains Strike price for the option instrument. Parameter is required for ClosePositionAtExpireDate functionality;
- Symbol – parameter contains ECN symbol name;
- UnderlyingSymbol – parameter contains ECN symbol name of the underlying instrument. Parameter is used for the derivatives.

Example of the Named ECN Parameters for “GCV6-GCZ7” Gold Futures at CME Globex:

```
<Params>
    <Currency Value="USD"/>
    <MarketId Value="CME"/>
    <MarketSegId Value="GLOBEX"/>
    <SecDescription Value="GCV6-GCZ7"/>
    <SecurityID Value="89522"/>
    <SecurityIDSource Value="8"/>
    <Symbol Value="GC"/>
    <CFICode Value="FMXXSX"/>
    <SecurityType Value="FUT"/>
    <MDTypes Value="0,1,2"/>
</Params>
```

7.2 MT4 Parameters for Security Master

MT4 Parameters section contains parameters of MT4 symbol. Bridge uses these parameters to bind MT4 Symbol with ECN Instrument.

Following parameters are supported:

- Symbol – contains name of the MT4 Symbol;
- SecGroup – contains name of the MT4 Security Group;

Example of the MT4 Parameters for “GCV6-GCZ7” Gold Futures at CME Globex:

```
<MT4Params>
    <Symbol Value=" GCV6GCZ7"/>
    <SecGroup Value="CMEFutures"/>
</MT4Params>
```

7.3 FIX rules section

This section contains rules applied to the FIX messages. For the incoming FIX messages, rules are used to compare tag's value from rule with tag's value from FIX message.
, which defines, how bridge should update outgoing FIX messages and process incoming FIX messages.
Format of the rule is following:
<Param <Message type filter> <FIX tag reference> <value> />

For example:
<Param MsgType="<list of FIX message types>" Tag="<tag reference>" Value="<value>"/>
<Param MsgType="<list of FIX message types>" Tag="<tag reference>" RefParamValue ="<Named parameter>"/>

"<list of FIX message types>" – is a simple filter that declares messages that should be processed by rule.

<FIX tag reference> contains reference to the FIX tag id located in the body of the message or group, for example:

1) Tag="55" – reference to the Symbol(55) tag that located in the message body
2) Tag="268[0]=>55" – reference to the Symbol(55) tag that located in the first entry (entry number starts from 0) of NoMDEntry(268) group

<Named parameter> and <value> defines value, that is expected in the incoming FIX message or has to be assigned when bridge prepared outgoing FIX message. Value could be the constant or reference to the Named ECN Parameter.

Example of the Security Master Configuration:

```
<Instruments>
  <Templates>
        <FIXParamsTemplate Name="FIXTemplExample">
          <Param MsgType="V" Tag="146" Value="1"/>
          <Param MsgType="V" Tag="146[0]=>55" RefParamValue="Symbol"/>
          <Param MsgType="V" Tag="146[0]=>48" RefParamValue="SecurityID"/>
          <Param MsgType="V" Tag="146[0]=>22" RefParamValue="SecurityIDSource"/>
          <Param MsgType="V" Tag="146[0]=>167" RefParamValue="SecurityType"/>
          <Param MsgType="V" Tag="146[0]=>541" RefParamValue="MaturityDate"/>

          <Param MsgType="D,F,G,8,W,e,f" Tag="55" RefParamValue="Symbol"/>
          <Param MsgType="D,F,G,8,W,e,f" Tag="48" RefParamValue="SecurityID"/>
          <Param MsgType="D,F,G,8,W,e,f" Tag="22" RefParamValue="SecurityIDSource"/>
          <Param MsgType="D,F,G,e" Tag="167" RefParamValue="SecurityType"/>
          <Param MsgType="D,F,G,e" Tag="541" RefParamValue="MaturityDate"/>

          <Param MsgType="X" Tag="268[0]=>55" RefParamValue="Symbol"/>
          <Param MsgType="X" Tag="268[0]=>48" RefParamValue="SecurityID"/>
          <Param MsgType="X" Tag="268[0]=>22" RefParamValue="SecurityIDSource"/>
          <!-- Param MsgType="X" Tag="268[0]=>167" RefParamValue="SecurityType"/>
          <Param MsgType="X" Tag="268[0]=>541" RefParamValue="MaturityDate"/ -->
        </FIXParamsTemplate>
```

```
        <FIXParamsTemplate Name="FIXTemplUnderlying">
          <Param MsgType="V" Tag="146" Value="1"/>
          <Param MsgType="V" Tag="146[0]=>55" RefParamValue="Symbol"/>
          <Param MsgType="V" Tag="146[0]=>48" RefParamValue="SecurityID"/>
          <Param MsgType="V" Tag="146[0]=>22" RefParamValue="SecurityIDSource"/>
          <Param MsgType="V" Tag="146[0]=>167" RefParamValue="SecurityType"/>
          <Param MsgType="V" Tag="146[0]=>541" RefParamValue="MaturityDate"/>

          <Param MsgType="D,F,G,8,W,f" Tag="55" RefParamValue="Symbol"/>
          <Param MsgType="D,F,G,8,W,f" Tag="48" RefParamValue="SecurityID"/>
          <Param MsgType="D,F,G,8,W,f" Tag="22" RefParamValue="SecurityIDSource"/>
          <Param MsgType="D,F,G" Tag="167" RefParamValue="SecurityType"/>
          <Param MsgType="D,F,G" Tag="541" RefParamValue="MaturityDate"/>

          <Param MsgType="X" Tag="268[0]=>55" RefParamValue="Symbol"/>
          <Param MsgType="X" Tag="268[0]=>48" RefParamValue="SecurityID"/>
          <Param MsgType="X" Tag="268[0]=>22" RefParamValue="SecurityIDSource"/>
          <!-- Param MsgType="X" Tag="268[0]=>167" RefParamValue="SecurityType"/>
          <Param MsgType="X" Tag="268[0]=>541" RefParamValue="MaturityDate"/ -->
        </FIXParamsTemplate>
        <FIXParamsTemplate Name="FIXTemplEquityIndex">
          <Param MsgType="V" Tag="146" Value="1"/>
          <Param MsgType="V" Tag="146[0]=>55" RefParamValue="Symbol"/>
          <Param MsgType="V" Tag="146[0]=>48" RefParamValue="SecurityID"/>
          <Param MsgType="V" Tag="146[0]=>22" RefParamValue="SecurityIDSource"/>
          <Param MsgType="V" Tag="146[0]=>167" RefParamValue="SecurityType"/>
          <Param MsgType="V" Tag="146[0]=>541" RefParamValue="MaturityDate"/>

          <Param MsgType="D,F,G,8,W,f" Tag="55" RefParamValue="Symbol"/>
          <Param MsgType="D,F,G,8,W,f" Tag="48" RefParamValue="SecurityID"/>
          <Param MsgType="D,F,G,8,W,f" Tag="22" RefParamValue="SecurityIDSource"/>
          <Param MsgType="D,F,G" Tag="167" RefParamValue="SecurityType"/>
          <Param MsgType="D,F,G" Tag="541" RefParamValue="MaturityDate"/>

          <Param MsgType="X" Tag="268[0]=>55" RefParamValue="Symbol"/>
          <Param MsgType="X" Tag="268[0]=>48" RefParamValue="SecurityID"/>
          <Param MsgType="X" Tag="268[0]=>22" RefParamValue="SecurityIDSource"/>
          <!--Param MsgType="X" Tag="268[0]=>167" RefParamValue="SecurityType"/>
          <Param MsgType="X" Tag="268[0]=>541" RefParamValue="MaturityDate"/-->
        </FIXParamsTemplate>
   </Templates>

<Instrument Name="ADVANC" Destination="SET" Enabled="true">
        <Params>
                <Currency Value="THB"/>
                <HighLimitPrice Value="284"/>
                <InstrumentKey
Value="MarketId:SET#MarketSegId:COMMON_STOCK#SecurityID:1069#SecurityIDSource:8#Sym
bol:ADVANC#"/>
                <LowLimitPrice Value="153.5"/>
                <MDTypes Value="2"/>
```

```
			<MarketId Value="SET"/>
			<MarketSegId Value="COMMON_STOCK"/>
			<OrderTimeInForce Value="Daily"/>
			<SecDescription Value="ADVANCED INFO SERVICE"/>
			<SecurityID Value="1069"/>
			<SecurityIDSource Value="8"/>
			<SecurityType Value="CS"/>
			<Symbol Value="ADVANC"/>
			<MDTypes Value="2"/>
			<OrderTimeInForce Value="Daily"/>
		</Params>
		<MT4Params>
			<Symbol Value="ADVANC"/>
			<Description Value="ADVANCED INFO SERVICE"/>
			<SecGroup Value="Underlying"/>
			<ContractSize Value=""/>
			<DigitsNumber Value="4"/>
		</MT4Params>
		<FIXParams>
			<RefTemplate Name="FIXTemplUnderlying"/>
		</FIXParams>
</Instrument>

<Instrument Name="USDZ14F15" Destination="SET" Enabled="true">
		<Params>
			<Currency Value="THB"/>
			<InstrumentKey
Value="MarketId:TXC#MarketSegId:USDFC#SecurityID:11453#SecurityIDSource:8#Symbol:USDZ
14F15#"/>
			<MDTypes Value="0,1,2,4,7,8,9,u"/>
			<MarketId Value="TXC"/>
			<MarketSegId Value="USDFC"/>
			<MaturityDate Value="20141229"/>
			<SecDescription Value="USD Futures Combo"/>
			<SecurityID Value="11453"/>
			<SecurityIDSource Value="8"/>
			<SecurityType Value="COMB"/>
			<StopTradeDateTime Value="20141229-04:00:00.000"/>
			<Symbol Value="USDZ14F15"/>
			<TotalUnderlyingSec Value="2"/>
			<TradingReferencePrice Value="0.05"/>
			<UnderlyingSecIdSource_0_0 Value="8"/>
			<UnderlyingSecIdSource_0_1 Value="8"/>
			<UnderlyingSecId_0_0 Value="9344"/>
			<UnderlyingSecId_0_1 Value="11376"/>
			<UnderlyingSymbol_0_0 Value="USDZ14"/>
			<UnderlyingSymbol_0_1 Value="USDF15"/>
			<MDTypes Value="0,1,2,4,7,8,9,u"/>
		</Params>
		<MT4Params>
			<Symbol Value="USDZ14F15"/>
```

```
            <Description Value="USD Futures Combo"/>
            <SecGroup Value="TXC"/>
            <ContractSize Value=""/>
            <DigitsNumber Value=""/>
        </MT4Params>
        <FIXParams>
            <RefTemplate Name="FIXTemplExample"/>
        </FIXParams>
</Instrument>

</Instruments>
```

7.4 Security Master Generator

Bridge allows to generate SecurityMaster XML configuration file automatically according to the XML templates and data from the SecurityList and DerivativeSecurityList FIX messages.

Example of the Security Master Generator Configuration:

securitydefgenerator.xml

8. Market Definition

Market Definition allows configuring ECN trading sessions and defines auxiliary ECN parameters. Market Definition XML file may contains multiple market definitions with different names and destinations. Bridge automatically generates MarketDefinition XML file using data from incoming MarketDefinition(BU) and TradingSessionList(BJ) FIX messages.

Format of the ECN market definition is following:

```
<MarketDefinitions>
        <Market Name="<name of the market>" Destination="<destination name >">
                <MT4Params>
                        List of MT4 parameters specific for this market
                </MT4Params>
                <Sessions>
                        List of the market sessions
                </Sessions>
        </Market>
        ….
</MarketDefinitions>
```

8.1 MT4 Parameters section

MT4 Parameters section contains parameters of MT4 symbols related to this market.

Following parameters are supported:

- TickSize – defines MT4 symbol's TickSize value. TickSize value could be different depend on the price of the instrument. Example of the TickSize configuration:

```
<TickSize StartPrice="0" Value="0.01"/>
<TickSize StartPrice="10" Value="0.1"/>
<TickSize StartPrice="25" Value="0.25"/>
<TickSize StartPrice="100" Value="0.5"/>
<TickSize StartPrice="200" Value="1"/>
```

- PositionVolumeLimit – defines limit for the order's volume that could be submitted for the instruments that are traded on this market. Example of the configuration:

```
<PositionVolumeLimit Value="1000"/>
```

8.2 Market Sessions section

Market Sessions section contains list of the sessions defined for the market and session's parameters - start and stop time, MarketData and Order flow.

Format of the market session is following:

```
<Sessions>
```

<Session Name="name of the session" StartTime="start datetime" EndTime="finish datetime">
Market Data Flow parameters
Order Flow parameters
</Session>
….
</Sessions>

MarketData Flow parameters

Following parameters are supported:

- Enable/Disable MarketData flow – bridge will establish MarketData FIX sessions, if enabled
- Enable/Disable MarketData subscription – bridge will send MarketData subscription FIX messages, if enabled
- Enable/Disable Charts update – bridge will update MT4 Charts according to incoming marketData FIX messages, if enabled

Example of the MarketData Flow parameters:

<MarketDataFlow Enabled="false">
<MarketData Enabled="false"/>
<Charts Enabled="false" PushLastPriceIfNoData="false"/>
</MarketDataFlow>

Order Flow parameters

Following parameters are supported:

- Enable/Disable Order flow – bridge will establish Order FIX session, if enabled
- Enable/Disable Market, Limit and Stop order types – bridge allows to send Market, Limit and Stop orders to the Order FIX session, if enabled

Example of the Order Flow parameters:

<OrderDataFlow Enabled="false">
<MarketOrder Enabled="false"/>
<LimitOrder Enabled="false"/>
<StopOrder Enabled="false"/>
</OrderDataFlow>

Example of the Market Definition:

<MarketDefinitions>
<Market Name="SET:EQUITY_INDEX" Destination="SET">
<MT4Params>
<SecGroup Name="SET"/>
<TickSize StartPrice="0.01" Value="0.01"/>
<PositionVolumeLimit Value="0"/>
</MT4Params>
<Sessions>

```
			<Session Name="Startup" StartTime="20150310-04:00:00"
EndTime="20150310-08:30:00">
				<MarketDataFlow Enabled="false">
					<MarketData Enabled="false"/>
					<Charts Enabled="false" PushLastPriceIfNoData="false"/>
				</MarketDataFlow>
				<OrderDataFlow Enabled="false">
					<MarketOrder Enabled="false"/>
					<LimitOrder Enabled="false"/>
					<StopOrder Enabled="false"/>
				</OrderDataFlow>
				<DropCopyFlow Enabled="false"/>
			</Session>
			<Session Name="Open" StartTime="20150310-08:30:00"
EndTime="20150310-09:00:00">
				<MarketDataFlow Enabled="true">
					<MarketData Enabled="true"/>
					<Charts Enabled="true" PushLastPriceIfNoData="true"/>
				</MarketDataFlow>
				<OrderDataFlow Enabled="true">
					<MarketOrder Enabled="false"/>
					<LimitOrder Enabled="true"/>
					<StopOrder Enabled="false"/>
				</OrderDataFlow>
				<DropCopyFlow Enabled="true"/>
			</Session>
			<Session Name=" Intermission " StartTime="20150310-09:00:00"
EndTime="20150310-14:00:00">
				<MarketDataFlow Enabled="true">
					<MarketData Enabled="true"/>
					<Charts Enabled="true" PushLastPriceIfNoData="true"/>
				</MarketDataFlow>
				<OrderDataFlow Enabled="true">
					<MarketOrder Enabled="true"/>
					<LimitOrder Enabled="true"/>
					<StopOrder Enabled="true"/>
				</OrderDataFlow>
				<DropCopyFlow Enabled="true"/>
			</Session>
			<Session Name="Closed" StartTime="20150310-14:00:00"
EndTime="20150310-19:00:00">
				<MarketDataFlow Enabled="false">
					<MarketData Enabled="false"/>
					<Charts Enabled="false" PushLastPriceIfNoData="false"/>
				</MarketDataFlow>
				<OrderDataFlow Enabled="false">
					<MarketOrder Enabled="false"/>
```

```
				<LimitOrder Enabled="false"/>
				<StopOrder Enabled="false"/>
			</OrderDataFlow>
			<DropCopyFlow Enabled="false"/>
		</Session>
	</Sessions>
</Market>
</MarketDefinitions>
```

9. Custom FIX Rules

Custom FIX rules allows updating outgoing and incoming FIX messages. These rules are applied to the FIX message before it will be send to the FIX session or processed by the bridge. Bridge processes each rule defined in the XML file. Bridge applies rule to the FIX message, if all rule's filter conditions are satisfied.

Format of the Custom FIX rules is following:

```
<CustomFields>
	<Rule Description="some description">
		List of condition filters
		List of the actions
	</Rule>
	….
</CustomFields>
```

List of condition filters

User able to configure following conditions:

- Destination. Rule could be applied to the message with specific destination or any. Example:
  `<Destination Name = "*"/>`
- ExistField - condition is satisfied if FIX tag exists in the FIX message
- NotExistField - condition is satisfied if FIX tag doesn't exist in the FIX message
- EqualField – condition is satisfied if FIX tag contains specified value
- NotEqualField – condition is satisfied if FIX tag doesn't contain specified value
- MatchField – condition is satisfied if value from FIX tag matches specified pattern
- NotMatchField – condition is satisfied if value from FIX tag doesn't matche specified pattern

List of the actions

User able to configure following actions

- SetField – action assigns specified value to the FIX tag
- CopyField – action copies value from one to another FIX tag
- MoveField – action moves value from one to another FIX tag
- RemoveField – action removes FIX tag from message

Example of the Custom FIX rules:

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomFields>
	<Rule Description="testRule">
		<Destination Name = "*"/>
		<Condition>
			<EqualField Field="35" Value="D"/>
		</Condition>
		<Action>
			<RemoveField Field="100"/>
```

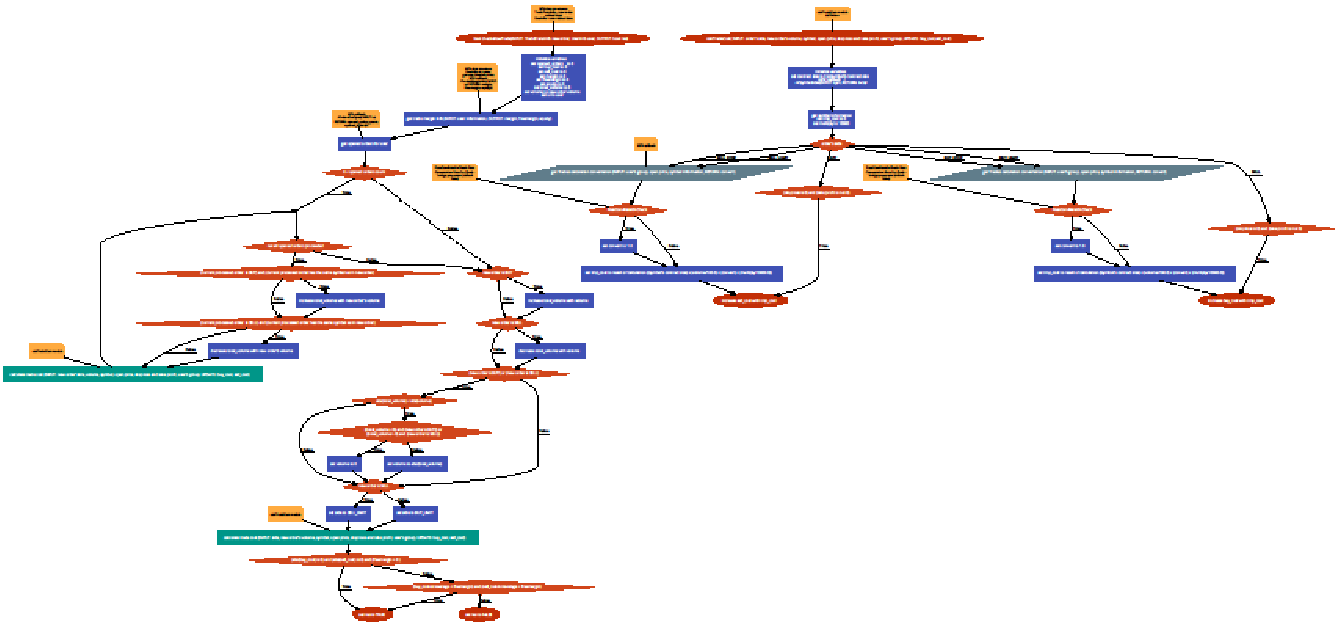
```
            </Action>
        </Rule>

        <Rule Description="testRule">
            <Destination Name = "*"/>
            <Condition>
                <EqualField Field="35" Value="D"/>
                <EqualField Field="40" Value="1"/>
            </Condition>
            <Action>
                <SetField Field="44" Value="1.3454"/>
            </Action>
        </Rule>

        <Rule Description="testRule">
            <Destination Name = "*"/>
            <Condition>
                <EqualField Field="35" Value="8"/>
            </Condition>
            <Action>
                <CopyField SourceField="583" TargetField="37" IsRequiredField="N"/>
            </Action>
        </Rule>

        <Rule Description="testRule">
            <Destination Name = "*"/>
            <Condition>
                <EqualField Field="35" Value="8"/>
                <EqualField Field="150" Value="A"/>
            </Condition>
            <Action>
                <SetField Field="150" Value="0"/>
            </Action>
        </Rule>

        <Rule Description="testRule">
            <Destination Name = "*"/>
            <Condition>
                <EqualField Field="35" Value="8"/>
                <EqualField Field="150" Value="4"/>
            </Condition>
            <Action>
                <CopyField SourceField="11" TargetField="41" IsRequiredField="N"/>
            </Action>
        </Rule>

    </CustomFields>
```

10. Buying Power Check Algorithm

Bridge performs buying power check before sending orders to ECN using the following algorithm.

First it gets trade margin information and checks total volume of all open orders including pending orders for the user by adding there total size of open buy orders and subtracting sell orders. Each time when user tries creating new order, bridge calculates trade cost value (either buy or sell) based on total trade volume, new order size (adding it to total trade volume for buys and subtracting for sells) and contract size for the instrument. If quotient of buy/sell trade cost to leverage is less than free margin then the result of check is 'true' and bridge allows new order to trade, otherwise it generates warning "Not enough money".

More detailed algorithm flow is shown on the picture below:



Contact Us

sales@btobits.com
Phone: +1-888-378-0666

Global Headquarters
US Client Support and Delivery Center

EPAM Systems, Inc
41 University Drive
Suite 202
Newtown, PA 18940
Phone: +1-267-759-9000
Fax: +1-267-759-8989

<epam>