
Appendix A, XML Interface version 1.0

This is the first and currently implemented XML interface for the communication over the TCP link between the client and the server. In essence, it is a simple request – response type protocol. A client sends in a *request* and the server responds with appropriate transaction data. The request can be either in the form of a *subscription* for future trading activity data or a *query* for historical data. Both requests and responses in the interface are sent over the interface in an XML format (a Document Type Definition (DTD) specifying this format is found below). The request is the only command defined by this interface, and the client is further restricted to sending only one request per connection. After the server has sent all of the data in response to the request, the connection is closed by the server.

A.1 Protocol

Here protocol is used in the sense that it defines the way the client and the server interacts, for example how communication is set up and ended. The exact specification of the messages used in the communication is described later. A client sends requests to the server of either subscription or query type. The protocol also requires the client to precede every request with the size of the message in bytes in a textual format. This is a temporary solution to let the server read the whole request from the socket before starting to parse it.

The server responds to the request by sending an XML document. Both query and subscription responses follow the same XML format. In the case of a query, however, all of the information is sent once it has been retrieved from the database. The signal to end the communication is the end tag of the document. If the database does not contain any of the data demanded, an empty XML document is sent.

If the request is a subscription, on the other hand, the beginning of the XML document is sent to the client when the subscription is added to the subscription list. This works as a message to the client that the subscription is accepted. Then, as long as the subscription remains valid, i.e. as long as specified in the request, subscribed information is sent as XML to the client immediately when it is received to the server from the CLICK system. Thus, the response forms a complete and valid XML document being sent successively at the same tempo as the information arriving from the exchange system. When the subscription expires, the server ends the communication by sending the XML document's end tag (the document is not a valid XML document until the end tag has been sent). The figure below illustrates the basic flow of messages between a client and the server.

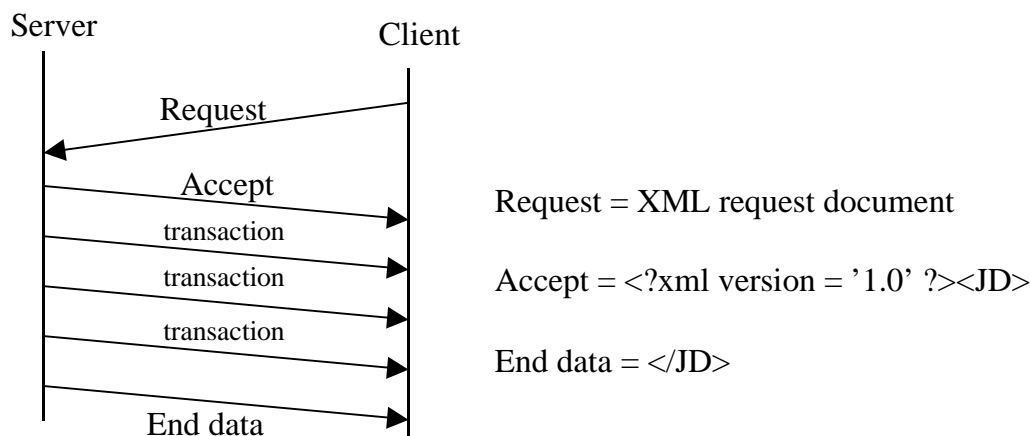


Figure 1, Communication flow client - server

Below follows a detailed description of the information in a request and the type of transaction messages the XML interface defines.

A.2 Request

The requests that the interface can handle are predefined. The format of the request has ten different parameters that define the query, where some of the parameters also have a predefined set of values that are valid.

A.2.1 Data in requests

The parameters in a request are:

Request type – one of the following predefined values:

1. Subscription
2. SubscriptionLight
3. Query
4. QueryLight

The light types return a response where a configurable amount of the information in the transaction messages is suppressed. The same transaction messages are used but with some fields suppressed. For example, it is possible to configure the messages so that they only contain information about what instrument and which participant it concerns, price and volume information. The configuration takes place in the server application.

Table 1, Request variables

Parameters	Predefined values	Explanation
Request type	Subscription	Future data.
	SubscriptionLight	Future data where some fields in the transaction messages are suppressed.
	Query	Stored data.
	QueryLight	Stored data where some fields in the transaction messages are suppressed.
Request name	All	This value is only valid for subscriptions in order to avoid queries for a too large amount of information.
	Surveillancedata	This is value returns the minimal subset of the available information necessary for the client to perform its surveillance functionality. For our implementation of the client the necessary information is quotes, orders, trades and order book changes.
	Transaction name	The name of a specific transaction message, one of the following: <ul style="list-style-type: none"> • Trade • Order • ExtBBO • IntBBO • Hedge • HedgeExecution • Quote • PriceImprovementQuote • OBChange
Instrument	All	Returns information for all instruments traded at the exchange. This only a valid value for subscriptions also to avoid large queries
	Instrument name	The name of an instrument should be in a readable form and not some numerical code. A name could for example be Volvo A, SSE. This is not implemented in the prototype due to the time it would take to develop. Instead the internal numerical representations are used.
StartTime	YYYY-MM-DD hh:mm:ss	The field is only considered for queries. Subscriptions are assumed to start when they are received.
StopTime	YYYY-MM-DD hh:mm:ss	The value is used in both queries and subscriptions
Customer	All	
	Empty string	
	Customer name	The name of a customer to the exchange, i.e. a broker firm.
User	All	
	Empty string	

	User name	The name of a user account at a brokerage firm, i.e. a specific broker.
Client	All	
	Empty string	
	Client name	The end client, this information is not available in the trading system today.
Order number	Empty string	
	Number	If this field is used the request is assumed to be a request, and the return is all the information about that order number found in the database.

A.2.2 Request DTD

```
<?xml version = '1.0' ?>
```

```
<!DOCTYPE REQUEST |
```

```
<!ELEMENT REQUEST (REQUESTTYPE , REQUESTNAME , INSTRUMENT,
STARTTIME, STOPTIME, CUSTOMER, USER, CLIENT, ORDERNUMBER)>
```

```
  <!ELEMENT REQUESTTYPE (#PCDATA)>
```

```
  <!ELEMENT REQUESTNAME (#PCDATA)>
```

```
  <!ELEMENT INSTRUMENT (#PCDATA)>
```

```
  <!ELEMENT STARTTIME (#PCDATA)>
```

```
  <!ELEMENT STOPTIME (#PCDATA)>
```

```
  <!ELEMENT CUSTOMER (#PCDATA)>
```

```
  <!ELEMENT USER (#PCDATA)>
```

```
  <!ELEMENT CLIENT (#PCDATA)>
```

```
  <!ELEMENT ORDERNUMBER (#PCDATA)>
```

```
|>
```

A.3 Trading information responses

The format of the trading information sent as XML to the client corresponds closely to the format of the information from the trading system itself. The messages are found in the table below. For an explanation of the meaning of the tags used for specific data fields, see the CLICK logb documentation.

Message	Explanation
Order	This is a transaction message with information about a new order in the system.
Quote	This message contains information from a market maker about the quotation of an instrument.
Trade	Information about a trade with references to the two matched orders.
Order book change	Information about any change in the order book. This record is

	generated after each trade, order entry and quote, among other events.
Internal Best Bid Offer	Information about the current best bid and best offer for a certain asset at the market place.
External Best Bid Offer	BBO at the away-market of the asset.
Hedge Order	This record indicates that an order is destined to be traded at the away market (hedged), because the BBO at the away market is better than at JIWAY.
Hedge Execution Report	Report of a successful hedge execution, where the original hedge order has been traded at the away market in part or fully.
Price improvement quote	Record with information that a market maker will trade a certain volume of an asset at the market place to the price specified by the external BBO.

A.3.1 Tags used

Since an XML representation of data is string-based, and can be quite “talkative”, the tags used in the interface are kept at a minimum length. The following tables present the tag names used, and their respective meaning.

XML transactions tags	Meaning
EB	External BBO
HO	Hedge Order
IB	Internal BBO
IQ	Improvement Quote
O	Order
OB	Orderbook Change
Q	Quote
S	Instrument Status Record
HE	Hedge Execution Report

XML data tags	Meaning
AEB	Ask External BBO
AIB	Ask Internal BBO
BA	BidAsk
BEB	Bid External BBO
BIB	Bid Internal BBO
C0	Customer Country

CC	Client Category
CL	Client
CN	Country Number
CON	Check OrderNumber
CU	Customer
DP	Deal Price
EN	Exchange/Country Number
ER	End Of Report
GSN	Global Sequence Number
HI	Hedge Order ID
HR	Hedge Execution Reference
HS	Hedge Status
IC	Instrument/Commodity Code
IG	Instrument Group
IN	Instrument/Series
IS	Instrument/Series Status
LR	Logb Reason
LT	Lock Type
MC	Market Code
MOD	Modifier
MOV	Market Order Volume
OC	Order category
ON	OrderNumber
OT	Order Type
PR	Premium
QY	Quantity
SN	Sequence Number
TA	Trade Ask side
TB	Trade Bid side
TON	Target Order Number
US	User
VM	Volume Multiplier
VO	Volume
VT	Validity Time

TO | Trade Order

A.3.2 XML response DTD

```
<?xml version = '1.0' ?>
```

```
<!DOCTYPE JD [
```

```
<!ELEMENT JD (O | Q | IQ | OB | T | EB | IB | H | HE | S)*>
```

```
  <!ELEMENT O (ON, TS, SN, IN, CO, CU, US, BA, QY, PR, VT, OT, CC, CL)>
```

```
  <!ELEMENT Q (ON, TS, SN, IN, CO, CU, US, BA, QY, PR)>
```

```
  <!ELEMENT IQ (ON, TS, SN, IN, CO, CU, US, BA, VO, VM)>
```

```
  <!ELEMENT EB (TS, SN, IN, BEB, AEB)>
```

```
    <!ELEMENT BEB (PR, VO)>
```

```
    <!ELEMENT AEB (PR, VO)>
```

```
  <!ELEMENT IB (TS, SN, IN, BIB, AIB)>
```

```
    <!ELEMENT BIB (PR, VO, MOV)>
```

```
    <!ELEMENT AIB (PR, VO, MOV)>
```

```
      <!ELEMENT MOV (VO)>
```

```
  <!ELEMENT OB (ON, TS, SN, IN, CO, CU, US, BA, QY, PR, LR, CON, OC, VT, OT, CC, CL, LT)>
```

```
    <!ELEMENT CON (ON)>
```

```
  <!ELEMENT H (ON, TS, SN, IN, CO, CU, US, BA, QY, PR, HI, OT, EN)>
```

```
  <!ELEMENT HE (ON, TS, SN, IN, CO, CU, US, BA, QY, PR, TON, HI, ER, HS, HR)>
```

```
    <!ELEMENT TON (ON)>
```

```
  <!ELEMENT T (TS, SN, IN, CO, CU, US, QY, PR, TB, TA)>
```

```
    <!ELEMENT TB (TO)>
```

```
    <!ELEMENT TA (TO)>
```

```
    <!ELEMENT TO (ON,CO,CU,US,QY,PR,VT,OC,OT,CC,CL)>
```

```
  <!ELEMENT S ( TS, SN, IN, CO, CU, US, IS)>
```

```
    <!ELEMENT ON (#PCDATA)>
```

```
    <!ELEMENT TS (#PCDATA)>
```

```
    <!ELEMENT SN (#PCDATA)>
```

```
    <!ELEMENT IN (#PCDATA)>
```

```
    <!ELEMENT CO (#PCDATA)>
```

```
    <!ELEMENT CU (#PCDATA)>
```

```
    <!ELEMENT US (#PCDATA)>
```

```
    <!ELEMENT BA (#PCDATA)>
```

<!ELEMENT QY (#PCDATA)>
<!ELEMENT PR (#PCDATA)>
<!ELEMENT VT (#PCDATA)>
<!ELEMENT OT (#PCDATA)>
<!ELEMENT CC (#PCDATA)>
<!ELEMENT CL (#PCDATA)>
<!ELEMENT VO (#PCDATA)>
<!ELEMENT VM (#PCDATA)>
<!ELEMENT LR (#PCDATA)>
<!ELEMENT OC (#PCDATA)>
<!ELEMENT LT (#PCDATA)>
<!ELEMENT HI (#PCDATA)>
<!ELEMENT EN (#PCDATA)>
<!ELEMENT ER (#PCDATA)>
<!ELEMENT HS (#PCDATA)>
<!ELEMENT HR (#PCDATA)>
<!ELEMENT IS (#PCDATA)>

|>

Appendix B, XML Interface version 2.0

B.1 Introduction

With the experience gained from the work done with the surveillance client and surveillance functionality, it was clear that the XML interface version 1.0 was quite CLICK- and JIWAY specific. One desirable feature of the XML interface is to be usable on different exchanges, as well as with different exchange systems, notably also the SAXESS system. With this in mind, the XML interface has undergone a thorough redesign. The aim of this has been to make it simpler, and at the same time more general and flexible. Also, the robustness of a strictly defined structure has been preserved in as much as possible of the new design.

B.2 Protocol

The protocol (i.e. the way the client and the server interact) remains the same in the new version. It is only the format and content of the messages containing trading information that change. The format for the requests also remains unchanged.

B.3 Trading information responses

The number of messages has been greatly reduced as compared to version 1.0, and their content is designed to reflect the information needed for market surveillance only. In this, the idea is to only use messages general enough to be valid for most electronic marketplaces. All flow of surveillance-relevant information on an exchange relates to *orders*. This is the most general entity – a market actor requesting to sell or buy a certain quantity of an asset to a certain price. An order may also change or be changed during its way through the exchange system. This leads to the following two main message types:

- Order entry
- Order change

Each of these messages contain data about that specific event. The table below illustrates the minimal set of such data identified as necessary for the surveillance process. The choice of what data is considered necessary is based on the experience gained from the pilot implementation of the surveillance client. Basically, the idea is that the surveillance process on any exchange is primarily concerned with the actions taken by its participants, and not internal exchange information. This follows from the fact that all market manipulation and insider trading deals with the information that is actually visible to the trading public. How this information (about orders, trades, prices, volumes, etc.) has been produced (on a technical level) is not interesting from this viewpoint. Thus, the surveillance client needs to know what was done at the exchange when, and who did it.

Order entry	Order change
Order identity	Event time
Instrument	Bid order identity
Event time	Ask order identity
Bid or Ask	Reason
Brokerage firm	

Broker	
Type	

As can be seen from the table, both of the basic messages include certain predefined fields, containing information that are generic to an “order”, or to an “order change”, respectively. Hence, an order will always be identified with an instrument and an event time, for example. In the case of the order change, either one or both of the bid- and ask order identity will be used, depending on the reason for the order change.

However, there are many types of orders, as well as reasons for an order change, so there should be a way to make the messages flexible and the interface extendable. Therefore, the messages were designed with a set of data fields common to all types, together with a part that is specific for each type of order or order change. This way, the interface is extendable; it is possible to add new order types and new reasons for order changes while the header part of the message remains the same.

Practically, one defines a new DTD for each exchange system that is to be used with the surveillance client application. This DTD must be compliant with the XML interface version 2.0. Therefore, the DTD needs to have the order entry and the order change entities at the central structural unit. It is then possible to define several different values for the Type entities of the Order entry, and the Reason entities of the Order change, respectively, in order to customise the DTD for use with the particular system considered. The Type- and Reason entity definitions are simply XML definitions of the allowable sub-entities of compound elements, reflecting the specific order entry or change that needs to be described. Examples include a simple order entry (e.g., with a price, volume and ordering market participant) and a trade (an order change possibly including a price and a volume).

Below a more detailed explanation is given to the XML interface version 2.0, described in the light of two examples. The first such example is an example mapping of the XML interface version 2.0 to the CLICK system run at JIWAY. After this, another example is presented in the mapping of the XML interface to a SAXESS-run exchange.

B.4 Mapping the interface to JIWAY

For JIWAY (running the CLICK exchange system), there are a couple of special events that can occur, and which are interesting for the surveillance process. The different possible order types are:

1. Simple order
2. Quote
3. Price Improvement Quote

The reason for an order change field can be of any one of the above types or, in addition, any one of the following:

4. Trade
5. Hedge
6. Delete

The respective Type- and Reason entity definitions include the following fields:

Type/ Reason	Simple order	Quote	Price I. Q.	Trade	Hedge	Delete
Data	Price	Price	Volume multiplier	Price	Price	
	Volume	Volume	Volume limit	Volume	Volume	
	End party					

Due to the fact that this particular DTD definition deals with the JIWAY exchange only, certain CLICK-supported transactions that are not used on JIWAY, such as combination orders, are not defined here. Combination orders are different in the way that they contain several orders that all need to be traded together. An extension including combination orders would be quite straightforward. An example implementation could include an additional Type definition, specifying an order message as a part of a combination order with a combination order identity (each combination order would be sent over a number of individual Order entities). There are other things to take into consideration as well, for example different price and volume types used by different exchange systems. The general guideline is that the interface needs to transfer the information that is public on the market to the surveillance process.

B.4.1 Tags used

As in the XML interface version 1.0, the tags used in the JIWAY version of the interface are abbreviated in order to reduce the XML overhead. The two tables below present the tags used.

Table 2 – Tags defining compound entities

Compound XML tags	Meaning
OE	Order entry
OC	Order change
IQ	Price Improvement Quote
SO	Simple Order
Q	Quote
TR	Trade
H	Hedge
D	Delete
T	Type
R	Reason

Table 3 – Tags defining simple data types

XML data tags	Meaning
ID	Order identity / Order

	change identity
P	Price
V	Volume
BA	Bid or Ask
IN	Instrument
TS	Time stamp
BR	Brokerage firm
DE	Dealer
EP	End party
VM	Volume Multiplier
VL	Volume Limit

B.4.2 XML response DTD

```

<?xml version = '1.0' ?>
<!DOCTYPE JD [
<!ELEMENT JD (OE | OC)*>
  <!ELEMENT OE (ID,IN,TS,BA,BR,DE,T)>
    <!ELEMENT T( SO | Q | IQ)>
      <!ELEMENT SO (P, V, EP)>
      <!ELEMENT Q (P, V)>
      <!ELEMENT IQ (VM, VL)>

  <!ELEMENT OC (TS,BO,AO,R)>
    <!ELEMENT R( SO | Q | IQ | TR | H | D)>
      <!ELEMENT TR (P, V)>
      <!ELEMENT H (P, V)>
      <!ELEMENT D (P, V)>

<!ELEMENT ID (#PCDATA)>
<!ELEMENT IN (#PCDATA)>
<!ELEMENT TS (#PCDATA)>
<!ELEMENT BA (#PCDATA)>
<!ELEMENT BR (#PCDATA)>
<!ELEMENT DE (#PCDATA)>
<!ELEMENT EP (#PCDATA)>

```

<!ELEMENT V (#PCDATA)>

<!ELEMENT P (#PCDATA)>

<!ELEMENT D (#PCDATA)>

B.4.3 Transaction message mapping

For each exchange implementing the XML interface version 2.0, a server has to be set up in order to map the informational content to the XML interface. Mapping the interface to the JIWAY exchange is quite straightforward, since the order types defined above match the transaction messages from the exchange closely. In the XML interface version 2.0, external and internal BBO records should be disregarded (since they are not used in the surveillance process), which reduces the message load from the CLICK exchange system. The mapping would be as follows:

Jiway message	Interface message
Order entry	Order – simple order
Quote	Order – quote
Price improvement quote	Order – price improvement quote

For the messages mapping to order changes, some extra consideration needs to be taken. The table below shows a draft for how this could be done.

Jiway message	Interface message
Trade	Order change – trade
Order/Quote in order book, trade	Disregarded, redundant
Hedge Execution order	Order change – hedge
Order delete	Disregarded, redundant
External BBO	Disregarded, redundant
Internal BBO	Disregarded, redundant
Order/Quote in order book, deleted	Order change – deleted

As can be seen from the table, all order/quote in order book messages are ignored except the ones concerning a deletion of an order.

B.5 Mapping the interface to SAXESS

An exchange running SAXESS would have to adapt the above-presented XML format for translation from the messages used in the SAXESS Interprocess Protocol. This section constitutes an example outline of how this could be done. A certain care should be put into the interpretations of this outline, however – the mapping may contain errors due to a misunderstanding of the SAXESS system. The reason for this is a shortage of time to study the SAXESS system.

The order types identified are:

1. Simple order
2. Quote

3. Combination order
4. Linked Order
5. Stop loss order

The reason for an order change field can be of any one of the above types or, in addition, any one of the following:

6. Trade
7. Delete

The respective Type- and Reason entity definitions include the following fields:

Type/ Reason	Simple order	Linked Order	Combination order	Stop loss order
Data	Price	Price	Price	Price
	Volume	Volume	Volume	Volume
	Open volume			Stop loss trigger price
	Price condition			Stop loss new price
				Price type

Type/ Reason	Quote	Trade	Delete
Data	Price	Price	
	Volume	Volume	

In the SAXESS system it is possible to specify market participants acting for other participants. A decision of how to handle this needs to be taken. One simple approach would be to disregard the participant acting for another participant and only use the participant from where the transaction originates. Otherwise it would be necessary to keep lists with the participants that act in the name of other participants, so that the XML information correctly reflects the real intentions of the active exchange members.

Another thing to consider is how the price and volume information is to be treated, since it is possible to specify price and volume conditions for incoming orders. In the current implementation, the price condition of the order is specified in the XML message. For volume conditions, the volume field contains the total volume left to be traded and the open volume field contains the remaining volume shown on the exchange.

The SAXESS system also supports so-called combination- and linked orders. Combination orders are merely a combination of 2-8 individual legs, associated with an AND condition. Linked orders are like combination orders, except that the condition is a XOR logic. Combination- and linked order messages in the SAXESS system are broken up into one XML order message for each leg in the SAXESS combination order. The XML messages are linked together simply by having the same order identity in the header.

Finally, there is an order type called stop-loss order, indicating an order with a price- or volume condition attached to it. At the time when this condition is fulfilled, the order changes price and/or volume following a predefined pattern. In the present adaptation of the XML interface, the stop-loss order has its own message specification with values for the trigger price

and the new price when the trigger price is reached. A definition of the trigger price type is also included.

B.5.1 Tags used

The tags used in the interface are the same except that the Jiway specific message types price improvement quote and hedge are not used. Instead, a tag for combination orders is added along with some other tags, such as one for the open volume field. Please see the table below for the complete listing of the new tags used.

Table 4 – New tags for SAXESS

New XML tags	Meaning
CO	Combination order
LO	Linked order
STO	Stop-loss order
OV	Open volume
TP	Stop loss trigger price
NP	Stop loss new price
PT	Price type
PC	Price condition

B.5.2 XML response DTD

```

<?xml version = '1.0' ?>
<!DOCTYPE SX [
<!ELEMENT SX (OE | OC)*>
  <!ELEMENT OE (ID,IN,TS,BA,BR,DE,T)>
    <!ELEMENT T( SO | Q | CO | LO | STO)>
      <!ELEMENT SO (P, V, OV, PC)>
      <!ELEMENT Q (P, V)>
      <!ELEMENT CO (P, V)>
      <!ELEMENT LO (P, V)>
      <!ELEMENT STO (P, V, TP, NP, PT)>

  <!ELEMENT OC (TS,BO,AO,R)>
    <!ELEMENT R( SO | STO | Q | TR | D )>
    <!ELEMENT TR (P, V)>

```

<!ELEMENT ID (#PCDATA)>
<!ELEMENT IN (#PCDATA)>
<!ELEMENT TS (#PCDATA)>
<!ELEMENT BA (#PCDATA)>
<!ELEMENT BR (#PCDATA)>
<!ELEMENT DE (#PCDATA)>
<!ELEMENT EP (#PCDATA)>
<!ELEMENT V (#PCDATA)>
<!ELEMENT P (#PCDATA)>
<!ELEMENT D (#PCDATA)>
<!ELEMENT OV (#PCDATA)>
<!ELEMENT NP (#PCDATA)>
<!ELEMENT TP (#PCDATA)>
<!ELEMENT PT (#PCDATA)>
<!ELEMENT PC (#PCDATA)>

B.5.3 SX Session message mapping

As an illustrative example, the SX messages from SAXESS can be mapped into this XML format in the following way:

XML order messages

- The XML message of type simple order (SO) maps from the SX order insert message when the order is not a stop-loss order. If the SX order message is indeed a stop-loss order, it maps to the XML stop-loss order.
- Several XML messages of combination order type map from each SX combination order insert message, and several XML messages of linked order type map from each SX linked order insert.
- XML quote messages map from either a leg in a bundled quote message or an order insert message where the isQuote flag is set to yes.

XML order change messages

- XML trade messages are mapped from SX trade insert messages.
- XML order, quote and stop loss order messages for order changes map from the SX update and adjust messages for single, linked and combination orders.
- Delete messages in XML are mapped from the cancel messages for the different types of orders.

Appendix C, Database specification

The final version of the physical database design for a relational database is presented below.

Table 5, Quote

Column name	MySQL data type	Comment
OrderNumber	BIGINT UNSIGNED	
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
BidAsk	TINYINT UNSIGNED	
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	Primary key
Quantity	INT UNSIGNED	
Premium	INT UNSIGNED	
Customer	CHAR(5)	
User	CHAR(5)	
Country	CHAR(2)	

Table 6, OrderEntry

Column name	MySQL data type	Comment
OrderNumber	BIGINT UNSIGNED	Primary key
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
BidAsk	TINYINT UNSIGNED	
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	
Quantity	INT UNSIGNED	
Premium	INT UNSIGNED	
OrderType	TINYINT UNSIGNED	
ClientCategory	TINYINT UNSIGNED	
Customer	CHAR(5)	
User	CHAR(5)	

Country	CHAR(2)	
ValidityTime	SMALLINT UNSIGNED	
ClientID	CHAR(10)	

Table 7, Hedge

Column name	MySQL data type	Comment
OrderNumber	BIGINT UNSIGNED	Primary key
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	Primary key
HedgeID	SMALLINT UNSIGNED	
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
Premium	INT UNSIGNED	
Quantity	INT UNSIGNED	
BidAsk	TINYINT UNSIGNED	
ExchangeID	TINYINT UNSIGNED	
OrderType	TINYINT UNSIGNED	

Table 8, HedgeExecution

Column name	MySQL data type	Comment
OrderNumber	BIGINT UNSIGNED	
EventTime	DATETIME	Index
SequenceNumber	INT UNSIGNED	Primary key
Customer	CHAR(5)	
User	CHAR(5)	
Country	CHAR(2)	
Premium	INT UNSIGNED	
Quantity	INT UNSIGNED	
SeqExecution	SMALLINT UNSIGNED	
EndReport	TINYINT UNSIGNED	
HedgeStatus	TINYINT UNSIGNED	
ExecutionRef	CHAR(30)	
OrderTarget	BIGINT UNSIGNED	

HedgeID	SMALLINT	
---------	----------	--

Table 9, ImprovementQuote

Column name	MySQL data type	Comment
OrderNumber	BIGINT UNSIGNED	
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
BidAsk	TINYINT UNSIGNED	
SequenceNumber	INT UNSIGNED	Primary key
VolumeLimit	INT UNSIGNED	
VolumeMultiplier	INT UNSIGNED	
Customer	CHAR(5)	
User	CHAR(5)	
Country	CHAR(2)	

Table 10, InstrumentStatus

Column name	MySQL data type	Comment
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	Primary key
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
Status	TINYINT UNSIGNED	

Table 11, ExternalBBO

Column name	MySQL data type	Comment
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	Primary key
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
PremiumBid	INT UNSIGNED	

PremiumAsk	INT UNSIGNED	
VolumeBid	INT UNSIGNED	
VolumeAsk	INT UNSIGNED	

Table 12, InternalBBO

Column name	MySQL data type	Comment
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	Primary key
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
PremiumBid	INT UNSIGNED	
PremiumAsk	INT UNSIGNED	
VolumeBid	INT UNSIGNED	
VolumeAsk	INT UNSIGNED	
VolumeMOBid	INT UNSIGNED	
VolumeMOAsk	INT UNSIGNED	

Table 13, OBChange

Column name	MySQL data type	Comment
OrderNumber	BIGINT UNSIGNED	
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
BidAsk	TINYINT UNSIGNED	
EventTime	DATETIME	
SequenceNumber	INT UNSIGNED	Primary key
Quantity	INT UNSIGNED	
CheckOrderNumber	BIGINT UNSIGNED	
LogbReason	TINYINT UNSIGNED	
OrderCategory	TINYINT UNSIGNED	
LockType	TINYINT UNSIGNED	

Table 14, Trade

Column name	MySQL data type	Comment
EventTime	DATETIME	Primary key
SequenceNumber	INT UNSIGNED	Primary key
InstrumentCode	SMALLINT UNSIGNED	Index
InstrumentGroup	TINYINT UNSIGNED	
InstrumentMarket	TINYINT UNSIGNED	
DealPrice	INT UNSIGNED	
DealQuantity	INT UNSIGNED	
BidOrderNumber	BIGINT UNSIGNED	
BidCustomer	CHAR(5)	
BidUser	CHAR(5)	
BidCountry	CHAR(2)	
BidClient	CHAR(10)	
AskOrderNumber	BIGINT UNSIGNED	
AskCustomer	CHAR(5)	
AskUser	CHAR(5)	
AskCountry	CHAR(2)	
AskClient	CHAR(10)	

C.1 Index versions tested

Several different versions of index creation. The one used in the tables above is version 5. The other index versions are illustrated in the tables below except for version one which was lost.

Table 15, Index version 2

Table	Primary key	Index
Quote	InstrumentCode, SequenceNumber	
OrderEntry	InstrumentCode, OrderNumber	
Hedge	InstrumentCode, OrderNumber, SequenceNumber	
HedgeExecution	OrderNumber, SequenceNumber	
ImprovementQuote	InstrumentCode, SequenceNumber	
InstrumentStatus	InstrumentCode, SequenceNumber	

ExternalBBO	SequenceNumber	
InternalBBO	SequenceNumber	
OBChange	InstrumentCode, OrderNumber, SequenceNumber	
Trade	EventTime, SequenceNumber	

Table 16, Index version 3

Table	Primary key	Index
Quote	InstrumentCode, SequenceNumber	
OrderEntry	InstrumentCode, OrderNumber	
Hedge	InstrumentCode, OrderNumber, SequenceNumber	
HedgeExecution	SequenceNumber	
ImprovementQuote	InstrumentCode, SequenceNumber	
InstrumentStatus	InstrumentCode, SequenceNumber	
ExternalBBO	SequenceNumber	
InternalBBO	SequenceNumber	
OBChange	InstrumentCode, SequenceNumber	
Trade	EventTime, SequenceNumber	

Table 17, Index version 4

Table	Primary key	Index
Quote	SequenceNumber	InstrumentCode, EventTime
OrderEntry	OrderNumber	InstrumentCode, EventTime
Hedge	OrderNumber, SequenceNumber	InstrumentCode, EventTime
HedgeExecution	SequenceNumber	EventTime
ImprovementQuote	SequenceNumber	InstrumentCode
InstrumentStatus	SequenceNumber	InstrumentCode, EventTime
ExternalBBO	SequenceNumber	InstrumentCode, EventTime
InternalBBO	SequenceNumber	InstrumentCode,

		EventTime
OBChange	InstrumentCode, SequenceNumber	InstrumentCode, EventTime
Trade	EventTime, SequenceNumber	InstrumentCode

Appendix D, Internal information objects

The appendix describes the different classes containing information from the Jiway exchange system that are used by the server internally. The information has been categorised as concerning orders, trade or best bid offer. The class diagrams below show the attributes of the classes but not the operations defined. The operations are accessing methods to retrieve the data in the classes in different formats, such as numerical, string or XML format. The class diagrams only shown top level classes, in the implementation each attribute is created in two levels of subclasses. For example the customer attribute is a customer class and the customer class is a subclass of the stringfield class. The stringfield class is the class that actually holds the character data. As discussed in the report this probably makes the creation of objects unnecessarily demanding.

Order related information

The diagram below show classes related to orders. The classes at the bottom of the diagram are the ones that actually represent transaction messages from the exchange system.

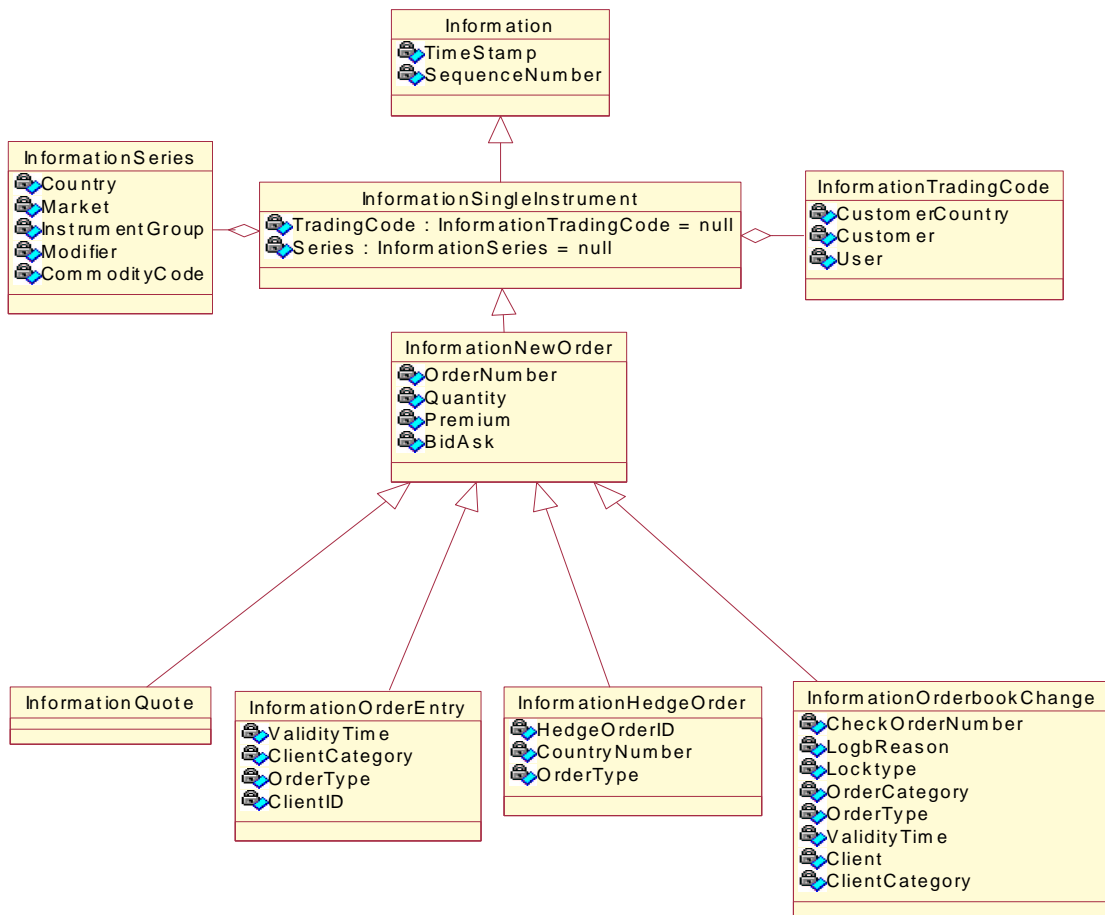


Figure 2, Order related classes

D.1 Trade related information

The diagram below shows the classes concerning trades and hedge executions. A hedge execution report is sent when a hedge order has been traded at an away market.

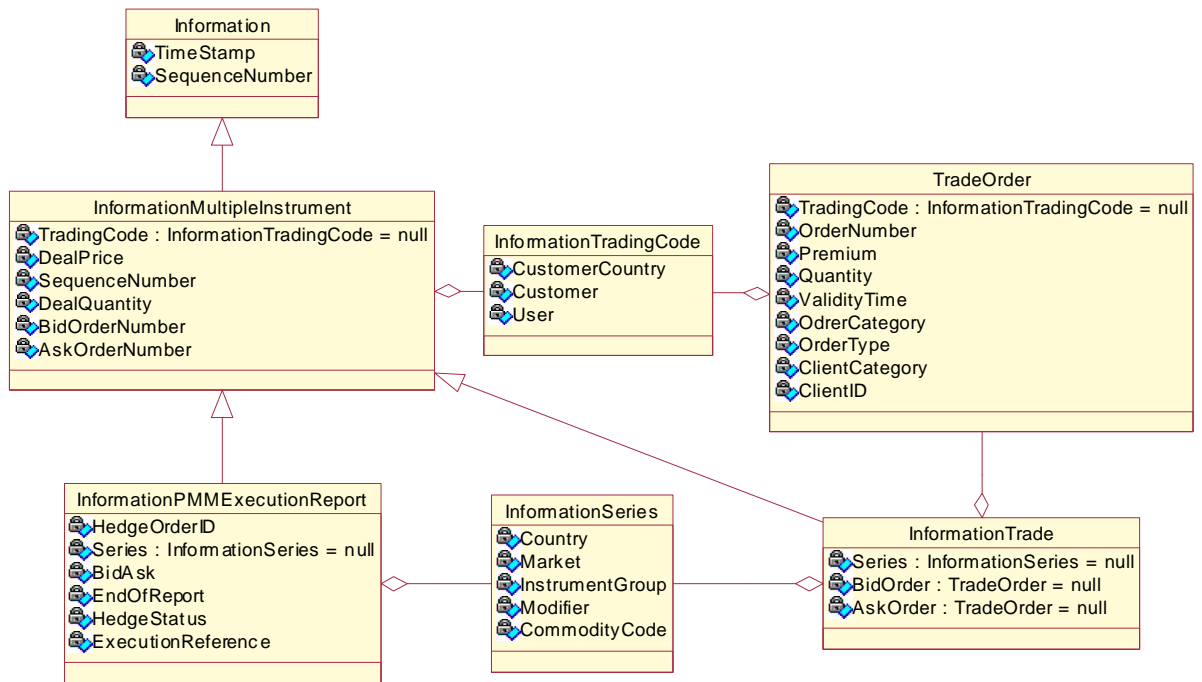


Figure 3, Trade related classes

D.2 Best Bid Offer related information

The classes below contain information about the current best bid offer at Jiway and at away markets.

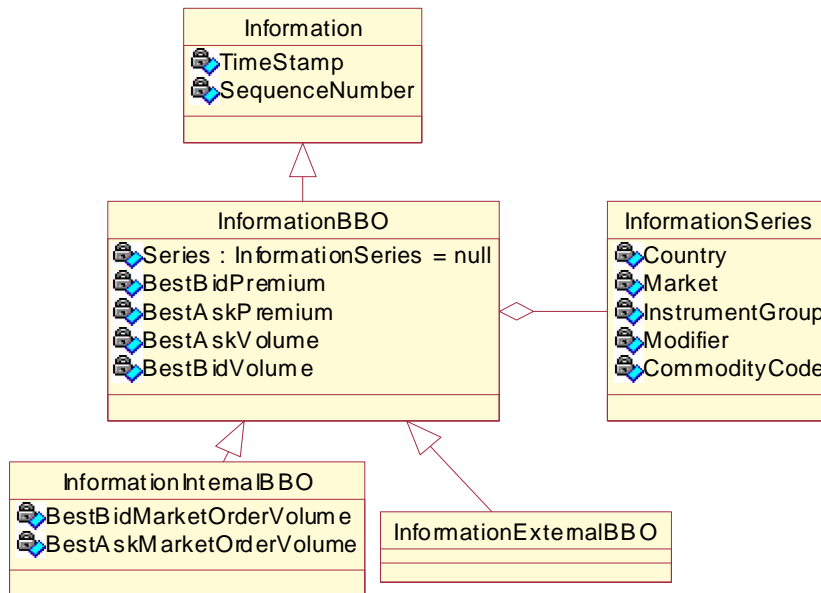


Figure 4, BBO related classes