

globalXLtrade

Pricing Tool Add-in for Microsoft Excel

Technical Specification

10 September 2004

1.1	Introduction.....	4
1.2	Intended Audience	4
1.3	Abbreviations	4
1.4	References.....	4
2.1	User Interfaces	5
2.1.1	Main menu	5
2.1.2	Properties Form – Configuring globalXLtrade.....	5
2.1.3	Control Form – Controlling and Monitoring globalXLtrade.....	6
2.1.4	Market Overview Spreadsheet – Data sent and received.....	7
3.1	Market Data definition.....	8
3.2	Storing Market Data.....	9
3.2.1	Market Overview Collection.....	9
3.3	Subscribing to Market Data	9
3.3.1	Overview of Market Data Subscription	9
3.3.2	Requesting Market Data	9
3.3.3	Receiving Market Data	9
3.3.4	Market Data Persistence and Recovery	9
4.1	Order definition.....	10
4.1.1	Security Identifier	10
4.1.2	Order Identifiers.....	10
4.1.3	OrderStatus	11
4.1.4	Order Priority	11
4.2	Storing Order Data.....	12
4.2.1	Orders Collection	12
4.2.2	Queue Collection	12
4.2.3	Request Collection	12
4.2.4	Pending Cancellation Collection.....	12
4.2.5	Pending Order Collection	12
4.2.6	Order Persistence and Recovery	12
4.3	Entering Orders.....	13
4.3.1	Overview of Order Entry	13
4.3.2	Preventing Crossing Trades	14
4.3.3	Preventing Duplicate or Similar Orders.....	14
4.3.4	Order Acknowledgments	14
4.3.5	Order Rejections	14
4.4	Canceling Orders	14
4.4.1	Overview of Canceling an Order.....	15
4.4.2	Canceling Queued Orders	16
4.4.3	Canceling Sent Orders (Orders Awaiting Acknowledgment)	16
4.4.4	Canceling Open Orders.....	16
4.4.5	Cancel Acknowledgments	16
4.4.6	Rejections.....	16
4.5	Queuing Requests	16
4.6	Sending Requests	16
4.6.1	Entry Request.....	16
4.6.2	Cancel Request.....	17
4.6.3	Request Event	17
4.7	Executions.....	17
4.7.1	Overview of an Execution	17

4.7.2	Partial Fills	17
4.7.3	Fills	17
5.1	FIX Messages.....	18
5.1.1	FIX messages from client to glox	18
5.1.2	FIX messages from glox to client	20
5.2	FIX Message Resends.....	21
5.3	Overview of FIX message flow	22

1 Preface

1.1 Introduction

globalXLtrade provides Microsoft Excel with a FIX interface to glox and allows a trader to receive market data, enter orders and cancel orders directly from Excel. As such, globalXLtrade has to handle a real-time market data feed and manage the flow of order requests. This draft document provides an overview of the mechanisms needed and provides a focus for further discussion.

1.2 Intended Audience

This document is intended for developers and business analysts. It covers the technical aspects of streaming prices to globalXLtrade via Microsoft Excel.

1.3 Abbreviations

Table 1 provides a description for each abbreviation used in the document:

Abbreviation	Description
COM	Component Object Model
FIX	Financial Information Exchange
ISIN	International Securities Identification Number
MOV	Market Overview
SWX	The Swiss Exchange
VTX	virt-x

Table 1

1.4 References

This document makes reference to the documents listed in Table 2.

Title	Author	Date
Connecting to glox using FIX	Joerg Thonnes	7 September 2004
Financial Information Exchange (FIX) Version 4.2 with Errata	FIX Protocol Organisation	1 May 2000

Table 2

2 Excel Interface

The interface between globalXLtrade and Excel is achieved using COM Interoperability. globalXLtrade is delivered as a COM Add-in for Excel.

2.1 User Interfaces

2.1.1 Main menu

The main menu can be found on the Excel menu bar (see Figure 1).

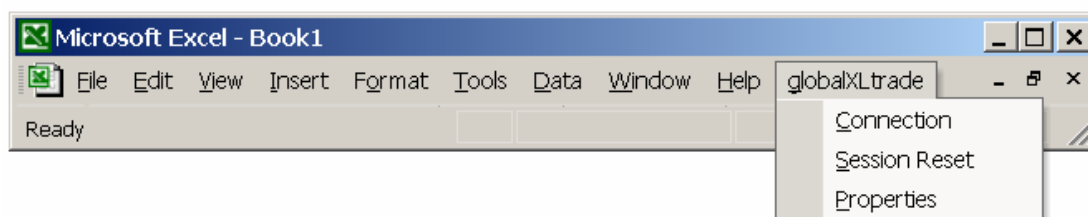


Figure 1

2.1.2 Properties Form – Configuring globalXLtrade

A form is provided for configuring globalXLtrade.

2.1.2.1 General Properties

The **Workbook** dropdown list contains a list of all the workbooks currently open. Use this list to select the workbook you wish to add the market data worksheet to. If the workbook you want is not listed then it needs to be opened in the normal way.

The **Security List** is used to limit and sort the securities subscribed to for market data. To set this, simply click the button right next to the Security List text box. This will open up an input box. Use the cursor to select the range containing the security data and then click the OK button. If a security list is not selected then no market data will be received.



Figure 2

2.1.2.2 FIX Link Properties

globalXLtrade connects to glox using FIX. There are a number of environments. The **FIX Link** properties (see Figure 3) allow you to specify how the FIX connection is established.

The **SenderCompID**, **TargetCompID**, **Connection Type**, **Acceptor Host**, **Acceptor Port**, **Data Dictionary** and **Begin String** details can be obtained from the glox technical help desk.

A **Data Dictionary** (FIX42.xml) is provided for data validation and this will be contained in the Program Files directory.

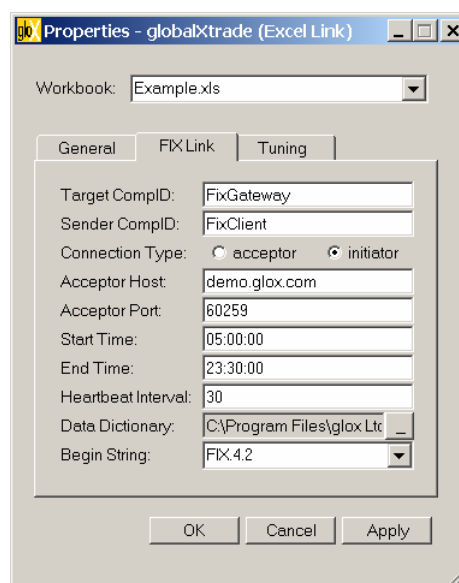


Figure 3

2.1.2.3 Tuning Properties

The **Tuning** properties allow you to control the way in which orders are submitted.

When you update your orders the requests will enter a prioritised queue. The requests are then sent to glox in order of priority. The mechanism is discussed in more detail in the Order Management section (see section 4). The **Max Requests** parameter allows you to specify how many outstanding requests (i.e. requests waiting an acknowledgement from the Exchange) are allowed at any one time (see section 4.6.3.1).

Under certain load conditions the Exchange may reject transaction requests due to a lack of resources. If this happens glox will return the rejection to Excel. The **Max Resubmits** parameter allows you to specify how many times a request will be automatically resubmitted if it is rejected by the Exchange (see section 4.6.3.2).

2.1.3 Control Form – Controlling and Monitoring globalXLtrade

The Control form provides a means to control and monitor the connection to glox. The **Start** button starts the FIX connection. The **Update** button updates all prices. The **Cancel** button cancels all orders. The **Auto** button switches on the automatic update of prices. The **Close** button in the top right hand corner of the Control window ends the FIX connection.

The Control window provides a summary of the last messages sent and received, and various statistics to monitor the messages. While the FIX connection is active the Target details are displayed in green. When the FIX connection ends the Target details turn red and then the Control window closes.

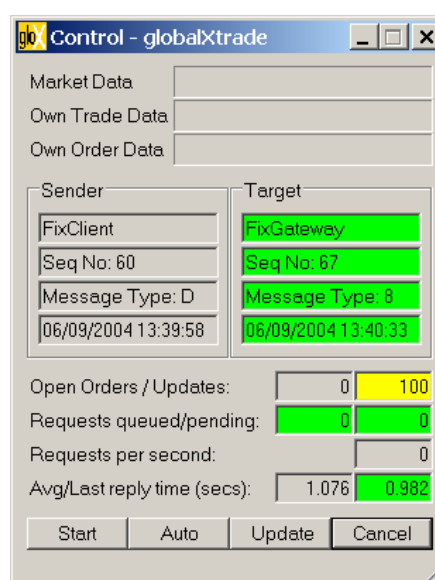


Figure 4

The **Open Orders/Updates** text boxes show the number of own open orders, and updates pending, respectively. The **Updates** text box will be yellow if there are updates pending. The **Requests queued** text box shows the number of transaction requests currently queued. If this number exceeds the Max Requests property then the box will be yellow. If the number queued exceeds 100 then the box will be red, otherwise it will be green. The **Requests pending** text box shows the number of transaction requests waiting for a response from the Exchange. If the number of open requests is greater than 1 then the box will be yellow, if the number equals the **Max Requests** property then the box will be red, otherwise it will be green.

2.1.4 Market Overview Spreadsheet – Data sent and received

When globalXLtrade is started a new sheet is created for the own order and market data:

	A	B	C	D	E	F	G	H	I	J	K
1	isin	ownBuySize	ownBuyPrice	bestBuySize	bestBuyPrice	bestSellPrice	bestSellSize	ownSellPrice	ownSellSize	closingPrice	closingPriceE
2	XS0141587310									99.50	20040903-13.2
3	XS0141608769									99.20	20040722-17.2
4	XS0141651264									107.11	20040903-10.2
5	XS0142019396									98.01	20040722-17.2
6	XS0142019479									100.19	20040722-17.2
7	XS0142391209									100.20	20040827-13.2
8	XS0142531739									99.85	20040722-17.2
9	XS0143875523									95.90	20040831-11.56

Figure 5

When the “GLOX_Market_Overview” worksheet is created it is protected to avoid accidental changes being made by the user. However, ten columns in this sheet are unprotected and provided for the user to enter order update data:

- updateBuySize* – the required size of the buy order
- updateBuyPrice* – the required price of the buy order
- updateSellPrice* – the required price of the sell order
- updateSellSize* – the required size of the sell order
- updateBuyText* – memo text shown on the buy order
- updateBuyFlag* – TRUE/FALSE this buy order should/shouldn't be updated
- updateBuyPriority* – the value used to prioritise the order while queued
- updateSellText* – memo text shown on the sell order
- updateSellFlag* – TRUE/FALSE this sell order should/shouldn't be updated
- updateSellPriority* – the value used to prioritise the order while queued

The cells in these columns can contain values or formulae.

	A	U	V	W	X	Y	Z	AA
1	isin	updateBuySize	updateBuyPrice	updateSellPrice	updateSellSize	updateBuyText	updateBuyFlag	updateBuyPriority
2	XS0141587310	20,000.00	103.20	107.20	20,000.00	Demo	TRUE	2064000
3	XS0141608769	20,000.00	103.20	107.20	20,000.00	Demo	TRUE	2064000
4	XS0141651264	20,000.00	106.11	110.11	20,000.00	Demo	TRUE	2122200
5	XS0142019396	20,000.00	100.02	104.02	20,000.00	Demo	TRUE	2000400
6	XS0142019479	20,000.00	104.19	108.19	20,000.00	Demo	TRUE	2083800
7	XS0142391209	20,000.00	102.20	106.20	20,000.00	Demo	TRUE	2044000
8	XS0142531739	20,000.00	103.85	107.85	20,000.00	Demo	TRUE	2077000
9	XS0143875523	20,000.00	101.90	105.90	20,000.00	Demo	TRUE	2038000

Figure 6

When the **Update** button on the Control form is clicked, buy and sell orders, with the updateSellFlag or updateBuyFlag set to TRUE, will be updated (see section 4).

3 Market Data Feed

3.1 Market Data definition

A Market data object consists of the following attributes:

Attribute	Description
<i>BestBuyPrice</i>	the best buy price in the order book
<i>BestBuySize</i>	the aggregated size of the best buy orders
<i>BestSellPrice</i>	the best sell price in the order book
<i>BestSellSize</i>	the aggregated size of the best sell orders
<i>ClosingPrice</i>	the closing price
<i>ClosingPriceDate</i>	the date and time of the closing price
<i>Currency</i>	the trading currency
<i>SecurityExchange</i>	the Exchange (e.g. swx = Swiss Exchange, vtx = virt-x)
<i>Isin</i>	the international security identification number
<i>LastTradedDate</i>	the data and time of the last trade
<i>LastTradedPrice</i>	the price of the last trade
<i>OpeningPrice</i>	the opening price
<i>OpeningPriceDate</i>	the date and time of the opening price
<i>OwnBuyPrice</i>	the best own buy price
<i>OwnBuySize</i>	the size of the best own sell order
<i>OwnBuyStatus</i>	the status of the best own order (see section 4.1.3)
<i>OwnSellPrice</i>	the best own sell price
<i>OwnSellSize</i>	the size of the best own sell order
<i>OwnSellStatus</i>	the status of the best own sell order (see section 4.1.3)
<i>MDReqID</i>	the unique identifier for the MarketDataRequest

Table 3

Some of these attributes map to FIX tags. The mapping between market data attributes and FIX tags is described in section 5.1.2.3.

3.2 Storing Market Data

3.2.1 Market Overview Collection

An object collection called Market Overview is maintained in memory for storing market data. It stores all market data using the FIX MDReqID as the primary key.

3.3 Subscribing to Market Data

3.3.1 Overview of Market Data Subscription

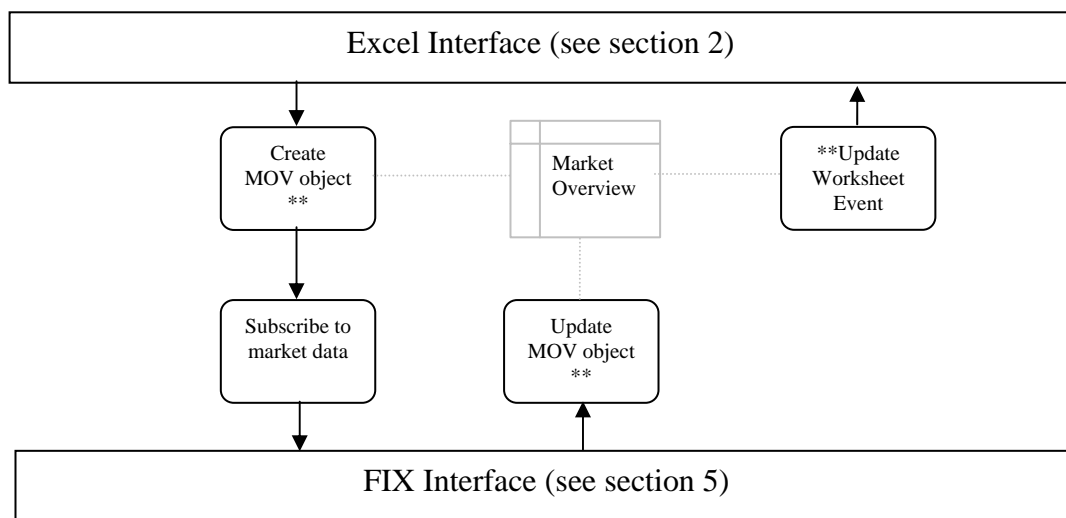


Figure 7

3.3.2 Requesting Market Data

When a market data subscription is made, a Market Overview object is added to the Market Overview collection for the security in question. This triggers an event to update the “GLOX_MarketOverview” worksheet.

3.3.3 Receiving Market Data

When new market data is received, the appropriate Market Overview object is updated and this triggers an event to update the “GLOX_MarketOverview” worksheet.

3.3.4 Market Data Persistence and Recovery

The Market Overview objects are recovered from glox each time globalXLtrade is started. They are only held in memory and not stored persistently.

4 Order Management

4.1 Order definition

An Order object consists of the following attributes:

Attribute	Description
<i>ClOrdID</i>	a unique ID assigned by globalXLtrade for the order
<i>Side</i>	indicates whether a buy or sell order
<i>Currency</i>	the trading currency
<i>Date</i>	the date the order was entered
<i>SecurityExchange</i>	the Exchange (e.g. swx = Swiss Exchange, vtx = virt-x)
<i>OrderID</i>	the unique ID required for canceling the order
<i>SecondaryOrderID</i>	the unique ID provided by the exchange
<i>Isin</i>	the international security identification number
<i>Price</i>	the limit price for the order
<i>Text</i>	free text
<i>OrderType</i>	this is always set to "Normal"
<i>OrderQty</i>	the original quantity of the order
<i>OrderStatus</i>	the order status (see section 4.1.3)
<i>LeavesQty</i>	the order quantity still open in the book
<i>Priority</i>	the priority value assigned to this order
<i>RequestType</i>	current request type (e.g. Enter or Cancel)
<i>SendAttempts</i>	the number of send attempts (see sections 4.3.5 & 4.4.5)
<i>RequestTime</i>	the time elapsed since the request was sent
<i>CancelCross</i>	flag used to prioritize a cancel request (see section 4.1.4.1)

Table 4

Some of these attributes map to FIX tags. The mapping between Order attributes and FIX tags is described in sections 5.1.1.1 and 5.1.2.1.

4.1.1 Security Identifier

Each security is uniquely identified by ISIN, Currency and SecurityExchange.

4.1.2 Order Identifiers

There are three separate order identifiers which serve different purposes.

4.1.2.1 ClOrdID

The *ClOrdID* is assigned by globalXLtrade and is used as the unique identifier for controlling Order requests.

4.1.2.2 OrderID

The *OrderID* is assigned by the Exchange's local trading system (located at glox) when it sends the Order request. This identifier is returned with the Order acknowledgment and it is required for canceling the Order. It must be submitted with the OrderCancelRequest (see section 5.1.1.2).

4.1.2.3 SecondaryOrderID

The *SecondaryOrderID* is assigned by the Exchange system. This identifier is returned with the Order acknowledgment and is the unique Exchange reference for the Order.

4.1.3 OrderStatus

The Order life cycle consists of the following order states.

4.1.3.1 Created

Initially an Order is created and added to the Orders collection with an *OrderStatus* of *Created*.

4.1.3.2 Queued

When an Order is queued it is added to the Queue collection and the *OrderStatus* is changed to *Queued*.

4.1.3.3 Sent

When an entry request is ready to send, it is removed from the Queue collection and then added to the Request collection. The *OrderStatus* is then changed to *Sent*.

4.1.3.4 Open

When an entry request receives a successful acknowledgment or a cancel request is rejected then the *OrderStatus* is changed to *Open*.

4.1.3.5 CancelQueued

When a cancel request is queued it is added to the Queue collection and the *OrderStatus* is changed to *CancelQueued*.

4.1.3.6 CancelSent

When the cancel request is ready to send, it is removed from the Queue collection and then added to the Request collection. The *OrderStatus* is then changed to *CancelSent*.

4.1.3.7 Deleted

When a cancel request receives a successful acknowledgment or an entry request is rejected then the *OrderStatus* is changed to *Deleted*.

4.1.4 Order Priority

The priority of a request is determined by the order of the Queue Collection (see section 4.2.2). The Queue Collection is a sorted list and each new request is inserted into the Queue collection in order of priority. The priority is based on three properties: the *CancelCross* flag, the *RequestType* and the *Priority* value. The priority request is therefore always the first request in the Queue collection.

4.1.4.1 CancelCross Flag

When a cancel request is added to the Queue collection it can be flagged as being related to a dependent Order entry that would otherwise cross with the Order being canceled. Cancel requests that have the *CancelCross* flag set, have higher priority than other requests.

4.1.4.2 RequestType

There are two *RequestTypes*: Enter and Cancel. Enter requests have a higher priority than Cancel requests (unless the *CancelCross* flag is set, see section 4.1.4.1)

4.1.4.3 Priority Value

Each Order has a *Priority* property. A value can be assigned to this property when the Order is created. Within each *RequestType* (see section 4.1.4.2) requests are sorted according to priority. The highest value has the highest priority.

4.2 Storing Order Data

globalXLtrade maintains five distinct Order collections for storing Orders and managing the lifecycle of each Order.

4.2.1 Orders Collection

The Orders collection stores all Orders using the FIX ClOrdID as the primary key. It also maintains a sorted list, sorted on the basis of ISIN, Currency, Exchange, Side, and ClOrdID. Each time the Orders collection is updated, an event is triggered to update the “GLOX_MarketOverview” worksheet. The Orders collection maintains statistics on the following:

LastRequestTime – the roundtrip time for the last request

AverageRequestTime – the average roundtrip time for all requests

TotalRequestCount – the total number of requests sent

4.2.2 Queue Collection

The Queue collection is a sorted list and stores all requests (i.e. both entry and cancel) currently waiting to be sent. When an Order is created, a priority can be specified. The Queue is sorted based on the *Priority* property and *RequestType* (see section 4.1.4). Each time a new request is added to the collection it is inserted in order of priority. The priority Order is therefore always the first request in the collection.

4.2.3 Request Collection

The Request collection holds all requests (i.e. both entry and cancel) that have been sent and are awaiting acknowledgment. It stores the requests using the FIX ClOrdID as the primary key.

4.2.4 Pending Cancellation Collection

When a cancel request is made for an Order that has an *OrderStatus* of *Sent* (see section 4.1.3.3), the request is stored in the Pending Cancellation collection, using the FIX ClOrdID as the primary key. If there is a pending cancellation for an Order when it receives an acknowledgment then a cancel request for this Order will be added to the Orders collection and then queued (see section 4.5).

4.2.5 Pending Order Collection

Certain entry requests are dependent on the completion of related cancel requests (e.g. if the Order being entered will cross with an existing own Order then the existing Order must be canceled first). Orders pending the completion of a related cancel request are added to the Pending Order collection, using the FIX ClOrdID of the related cancel request as the primary key. This collection is checked each time a cancel request succeeds and if a pending Order exists then it is added to the Orders collection and then queued (see section 4.5).

4.2.6 Order Persistence and Recovery

While globalXLtrade is running it maintains all the Order objects in memory. To ensure that the order information will not be lost when the process stops and restarts, orders are serialized to a persistent store. When globalXLtrade starts, it de-serializes the orders from the persistent store and sends an *OrderStatusRequest* to glox for each order to recover the order information.

4.3.2 Preventing Crossing Trades

When a new Order is entered, a check is made to ensure that it will not match against any existing own Orders resulting in a cross trade. If a matching own Order exists on the opposite side of the book then the new Order is added to the Pending Orders collection (see section 4.2.5) and the existing Order is canceled. As soon as the cancellation of the existing Order is confirmed, the new Order is added to the Orders collection (see section 4.2.1) and the request is then queued (see section 4.5).

4.3.3 Preventing Duplicate or Similar Orders

The order management allows for a single buy and sell Order in each security. If a new Order is entered with the same ISIN, Currency, Exchange, Side, Price and Size as an existing Order then the request will be deemed as a duplicate and ignored. Similarly, if a new Order is entered with the same ISIN, Currency, Exchange and Side as an existing Order but a different price or size, then the existing order will be canceled.

4.3.4 Order Acknowledgments

When the Order acknowledgment is received the Order is removed from the Request Collection and the Order is updated (e.g. the *OrderStatus* is changed to *Open*, the *OrderID* is added and the *RequestTime* is calculated). The Orders collection statistics (see section 4.2.1) are also updated.

4.3.5 Order Rejections

When a rejection is received the Order is removed from the Request collection (see section 4.2.3)

4.3.5.1 Invalid Order Data

If the price does not match the price step for the security or the size does not match the minimum tradable unit for the security then the Order will be rejected with an appropriate response. The Order will not be automatically resubmitted.

4.3.5.2 Insufficient Exchange Resources

If the Exchange has insufficient resources to handle the request then the Order may be rejected with an appropriate response. Such rejections can be automatically resubmitted. The Order's *SendAttempts* attribute maintains a record of the number of times an order has been submitted unsuccessfully. It is therefore possible to limit the number of resubmissions to a specified number (see section 4.6.3.2). The order can be resubmitted by queuing the request again (see section 4.5)

4.4 Canceling Orders

The method for canceling an Order will vary depending on the state of the Order. If the Order is in a state of *CancelQueued* or *CancelSent* then there is no need to cancel the order again.

4.4.1 Overview of Canceling an Order

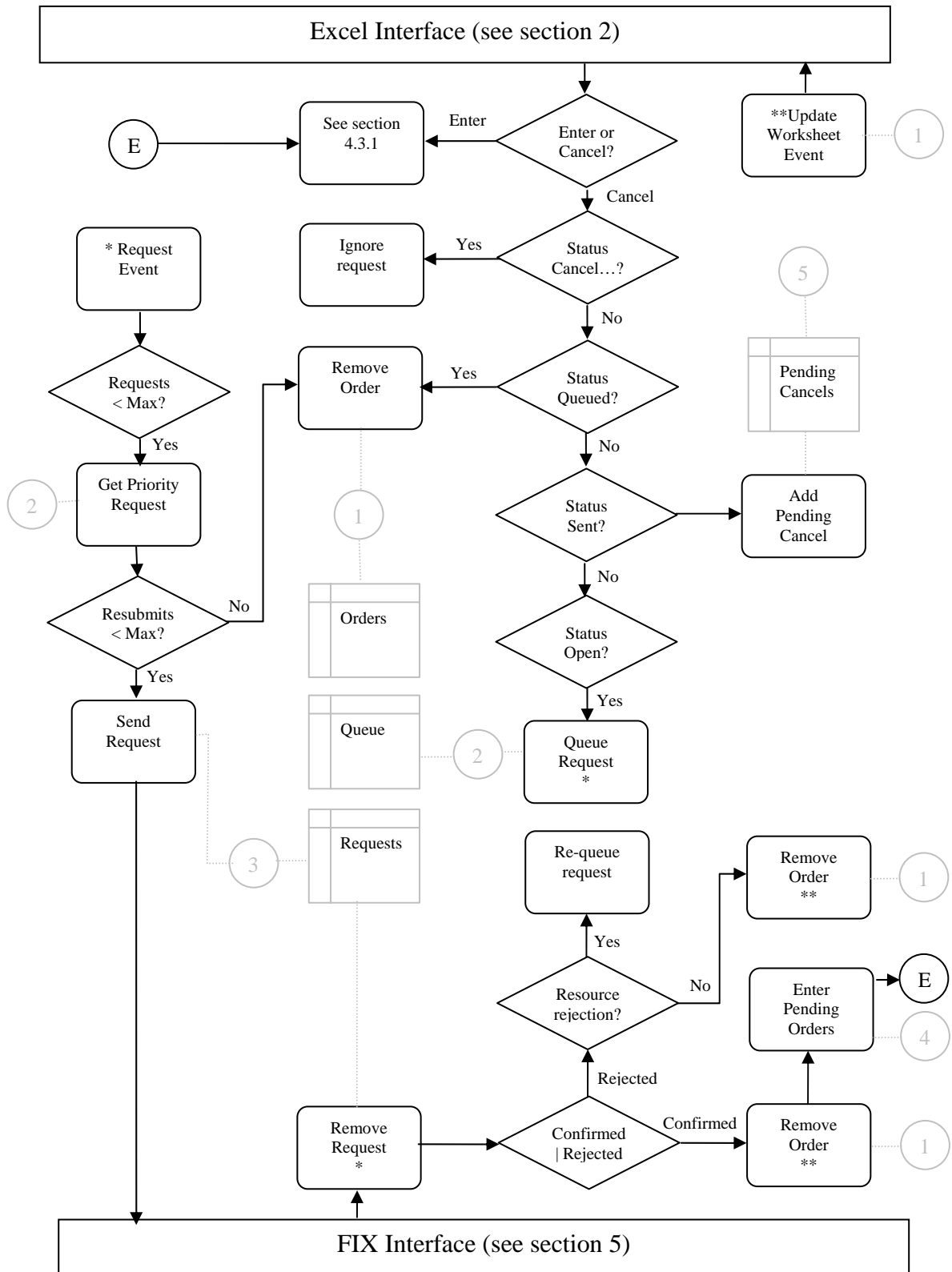


Figure 9

4.4.2 Canceling Queued Orders

If the order has only been *Created*, or *Queued* then it can be removed instantly.

4.4.3 Canceling Sent Orders (Orders Awaiting Acknowledgment)

It is only possible to send a cancel request if the *OrderID* (see section 4.1.2.2) is available. If the Order is in the *Sent* state (see section 4.1.3.3) then the *OrderID* will not be available. The request is therefore added to the Pending Cancellation collection (see section 4.2.4). When the Order receives an acknowledgment the cancel request will be queued immediately (see section 4.5).

4.4.4 Canceling Open Orders

If the Order is *Open* then a cancel request is queued (see section 4.5).

4.4.5 Cancel Acknowledgments

When the Order acknowledgment is received the Order is updated (e.g. the *OrderStatus* is changed to *Deleted* and the *RequestTime* is calculated). The Orders collection statistics (see section 4.2.1) are also updated. The Order is then removed from all collections.

4.4.6 Rejections

4.4.6.1 Unknown OrderID

Occasionally a cancel request will be rejected because the Order is no longer in the order book. This can happen when the Order is deleted (e.g. matched against another order) shortly before the cancel request arrives at the Exchange. If this happens, the cancel request will not be automatically resubmitted.

4.4.6.2 Insufficient Exchange Resources

If the Exchange has insufficient resources to handle the request then the cancel request may be rejected with an appropriate response. Such rejections can be automatically resubmitted. The Order's *SendAttempts* attribute maintains a record of the number of times an Order has been submitted unsuccessfully. It is therefore possible to limit the number of resubmission attempts to a specified number (see section 4.6.3.2). The cancel request can be resubmitted by queuing the request again (see section 4.5)

4.5 Queuing Requests

When a request is queued, it is inserted into the Queue collection, in order of priority. The *OrderStatus* is changed to *Queued* for entry requests, or changed to *CancelQueued* for cancel requests. Each time a new request is added to the queue, a request event is raised (see section 4.6.3).

4.6 Sending Requests

4.6.1 Entry Request

When an entry request is made the Order's *OrderState* is changed to *Sent* and the request is added to the Request Collection. When the request receives an acknowledgement (see section 4.3.4) a request event is raised (see section 4.6.3).

4.6.2 Cancel Request

When a cancel request is made the Order's *OrderState* is changed to *CancelSent* and the request is added to the Request Collection. When the request receives an acknowledgement (see section 4.3.4) a request event is raised (see section 4.6.3).

4.6.3 Request Event

Each time the request event is raised, the next Priority Order is selected from the Queue collection for sending. globalXLtrade maintains configurable parameters for controlling the maximum number of concurrent requests and the maximum number of resubmits.

4.6.3.1 Maximum Number of Concurrent Requests

A request (entry or cancel) will only be added to the Request collection and sent if the number of requests in the Request collection is less than the maximum number of concurrent requests allowed. By setting a maximum number of concurrent requests (see section 2.1.2.3) the trader can control the maximum number of requests that will be allowed to sit in the Exchange queue.

4.6.3.2 Maximum Number of Resubmits

A request (entry or cancel) will only be added to the Request collection and sent if the Order's *SendAttempts* attribute is less than the maximum number of resubmits (see section 2.1.2.3). This allows the trader to control the automatic resubmission of requests rejected because of a lack of Exchange resources.

4.7 Executions

4.7.1 Overview of an Execution

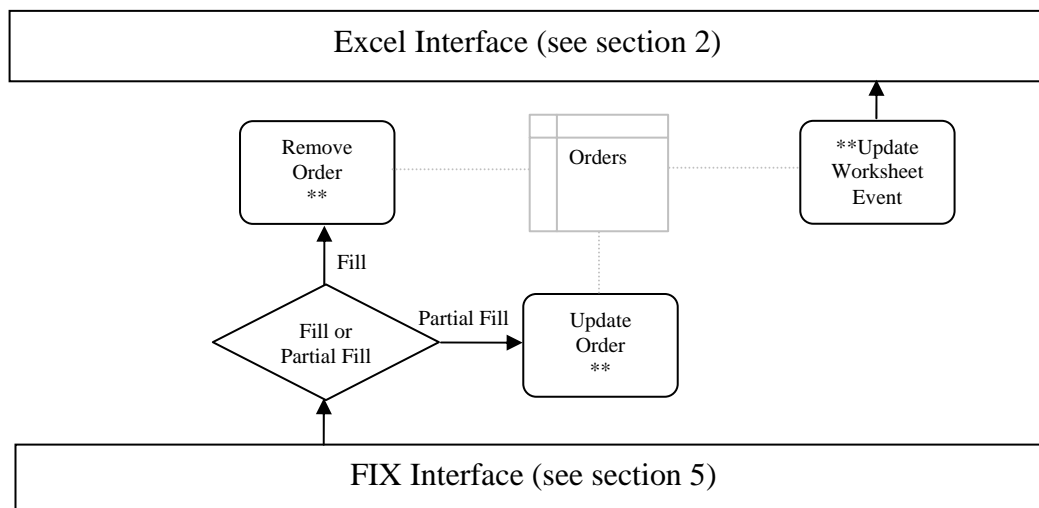


Figure 10

4.7.2 Partial Fills

Each time an execution occurs, the associated Order is updated with the remaining quantity.

4.7.3 Fills

When an Order fully matches, the *OrderState* is change to *Deleted* and the Order is removed from all collections.

5 The FIX Interface

globalXLtrade incorporates the QuickFIX engine (<http://www.quickfixengine.org/>) and uses FIX to interface to glox.

This document only provides an overview of the FIX messages used by globalXLtrade. More detailed information is contained in the glox FIX specification. Currently glox uses version FIX 4.2.

5.1 FIX Messages

A subset of FIX messages are used by the glox FIX interface. The following messages are used by globalXLtrade.

5.1.1 FIX messages from client to glox

5.1.1.1 OrderSingle (MsgType = D)

The OrderSingle message is used for submitting a new order to the Exchange.

Tag	Field Name	Assoc. Order Attribute(s)	Comment
11	ClOrdID	<i>ClOrdID</i>	Automatically assigned
63	SettlmentTyp		'0'=Regular
21	HandlInst		'1'=Automated, Private, No Broker
55	Symbol	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
48	SecurityID	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
22	IDSource		'8'=Exchange Symbol
207	SecurityExchange	<i>SecurityExchange</i>	"S" = swx, "VX" = vtx
54	Side	<i>Side</i>	'1'=Buy, '2' = Sell
60	TransactTime		Time at which order sent
38	OrderQty	<i>OrderQty</i>	This is populated by the <i>updateBuySize</i> or <i>updateSellSize</i> cells from "GLOX_MarketOverview"
40	OrdType		'2'=Limit
44	Price	<i>Price</i>	This is populated by the <i>updateBuyPrice</i> or <i>updateSellPrice</i> cells from "GLOX_MarketOverview"
59	TimeInForce		'0'=Day
58	Text	<i>Text</i>	This is populated by the <i>updateBuyText</i> or <i>updateSellText</i> cells from "GLOX_MarketOverview". Length < 80 characters.

5.1.1.2 OrderCancelRequest (MsgType = F)

The OrderCancelRequest message is used to cancel an existing order.

Tag	Field Name	Assoc. Order Attribute(s)	Comment
41	OrigClOrdID	<i>ClOrdID</i>	ClOrdID of original OrderSingle
37	OrderID	<i>OrderID</i>	Contained in the response to the original OrderSingle (see section 5.1.2.1)
11	ClOrdID	<i>ClOrdID</i>	Same as OrigOrdID
55	Symbol	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
48	SecurityID	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
22	IDSource		'8'=Exchange Symbol
207	SecurityExchange	<i>SecurityExchange</i>	"S" = swx, "VX" = vtx
54	Side	<i>Side</i>	'1'=Buy, '2' = Sell
60	TransactTime		Time at which order sent

5.1.1.3 OrderStatusRequest (MsgType = H)

The OrderStatusRequest is used to recover the current order information from glox.

Tag	Field Name	Assoc. Order Attribute(s)	Comment
11	ClOrdID	<i>ClOrdID</i>	Same as ClOrdID on original OrderSingle
55	Symbol	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
48	SecurityID	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
54	Side	<i>Side</i>	'1'=Buy, '2' = Sell

5.1.1.4 MarketDataRequest (MsgType = V)

The MarketDataRequest is used to subscribe to market data from glox.

Tag	Field Name	Assoc. Market Data Attribute(s)	Comment
262	MDReqID	<i>MDReqID</i>	
263	SubscriptionRequestType		'1' =Subscribe
264	MarketDepth		'1'=Top of Book
265	MDUpdateType		'0'=Full Refresh
267	NoMDEntryTypes		'5'
269	MDEntryType		'0'=Bid
269	MDEntryType		'1'=Offer
269	MDEntryType		'2'=Trade
269	MDEntryType		'4'=Opening Price
269	MDEntryType		'5'=Closing Price
146	NoRelatedSym		'1'
55	Symbol	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
48	SecurityID	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
22	IDSource		'8'=Exchange Symbol
207	SecurityExchange	<i>SecurityExchange</i>	"S" = swx, "VX" = vtx

5.1.2 FIX messages from glox to client

5.1.2.1 Execution Report (MsgType = 8)

ExecutionReports are use by glox to send order and execution data to globalXLtrade.

Tag	Field Name	Assoc. Order Attribute(s)	Comment
37	OrderID	<i>OrderID</i>	Required for OrderCancelRequests
198	SecondaryOrderID	<i>SecondaryOrderID</i>	
11	ClOrdID	<i>ClOrdID</i>	Unique order identifier
17	ExecID		Unique report identifier
20	ExecTransType		
150	ExecType		See note
39	OrdStatus		See note
103	OrdRejReason		See note
63	SettlmntTyp		Always '0'=Automatic
55	Symbol	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
48	SecurityID	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
22	IDSource		'8'=Exchange Symbol
207	SecurityExchange	<i>SecurityExchange</i>	"S" = swx, "VX" = vtx
54	Side	<i>Side</i>	'1'=Buy, '2' = Sell
60	TransactTime		Time at which ack sent
38	OrderQty	<i>OrderQty</i>	
40	OrdType		Always '2'=Limit
44	Price	<i>Price</i>	
151	LeavesQty	<i>LeavesQty</i>	Remaining open quantity
14	CumQty		
6	AvgPx		
58	Text		Free text

5.1.2.2 OrderCancelReject (MsgType = 9)

OrderCancelRejects are used by glox to reject order cancel requests.

Tag	Field Name	Assoc. Order Attribute(s)	Comment
37	OrderID	<i>OrderID</i>	Contained in the response to the original OrderSingle (see section 5.1.2.1)
11	ClOrdID	<i>ClOrdID</i>	Same as OrigOrdID
41	OrigClOrdID	<i>ClOrdID</i>	ClOrdID of original OrderSingle
60	TransactTime		Time at which the cancel reject was sent
434	CxlRejResponseTo		Always '1'=OrderCancelRequest
102	CxlRejReason		Reject reason
58	Text		Error code

5.1.2.3 MarketDataSnapshotFullRefresh (MsgType = W)

The MarketDataSnapshotFullRefresh message is used by glox to send market data to globalXLtrade.

Tag	Field Name	Assoc. Market Data Attribute(s)	Comment
262	MDReqID	<i>MDReqID</i>	
55	Symbol	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
48	SecurityID	<i>Isin and Currency</i>	<i>Isin & "." & Currency</i>
22	IDSource		'8'=Exchange Symbol
207	SecurityExchange	<i>SecurityExchange</i>	"S" = swx, "VX" = vtx
263	SubscriptionRequestType		'1' =Subscribe
264	MarketDepth		'1'=Top of Book
265	MDUpdateType		'0'=Full Refresh
268	NoMDEntries		'5'
269	MDEntryType		'0'=Bid
270	MDEntryPx	<i>BestBuyPrice</i>	
271	MDEntrySize	<i>BestBuySize</i>	
290	MDEntryPositionNo		'1'
269	MDEntryType		'1'=Offer
270	MDEntryPx	<i>BestSellPrice</i>	
271	MDEntrySize	<i>BestSellSize</i>	
290	MDEntryPositionNo		'1'
269	MDEntryType		'2'=Trade
270	MDEntryPx	<i>LastTradedPrice</i>	
272	MDEntryDate	<i>LastTradedDate</i>	
273	MDEntryTime		
269	MDEntryType		'4'=Opening Price
270	MDEntryPx	<i>OpeningPrice</i>	
272	MDEntryDate	<i>OpeningPriceDate</i>	
273	MDEntryTime		
269	MDEntryType		'5'=Closing Price
270	MDEntryPx	<i>ClosingPrice</i>	
272	MDEntryDate	<i>ClosingPriceDate</i>	
273	MDEntryTime		

5.2 FIX Message Resends

The FIX session layer implements a method of message resends for resynchronisation on reconnection. If the FIX connection between globalXLtrade and glox should fail, on reconnection glox may send ResendRequests to pick up missed messages. If these messages were resent then this could lead to stale Orders being sent to glox. globalXLtrade therefore prevents the resend of OrderSingle messages on reconnection and sends gap fills instead.

5.3 Overview of FIX message flow

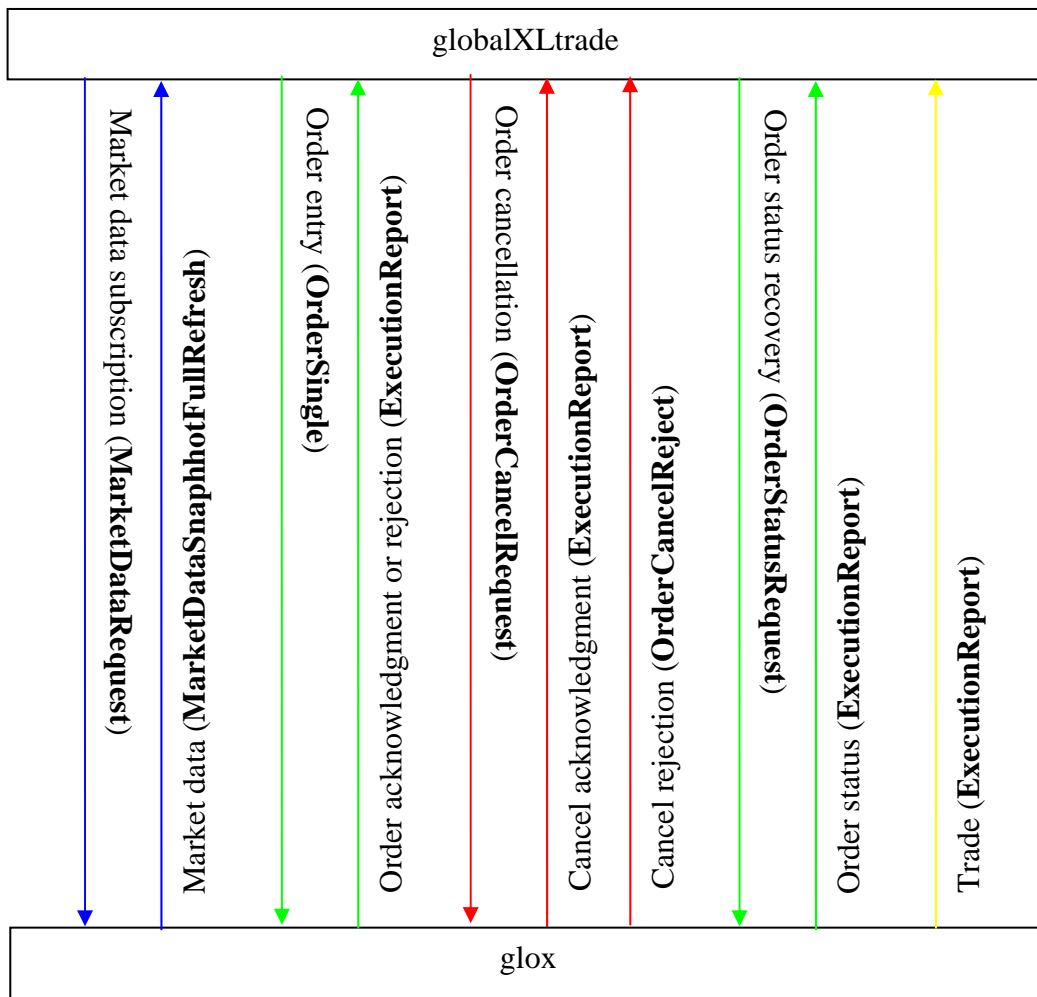


Figure 11