

# Financial Information eXchange

Chris Prouty

Winter Seminar 2012

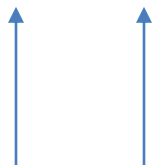
- Financial Information eXchange (FIX) is a protocol used by banks, investment funds, exchanges, and other market participants to communicate order data
- FIX can be used to communicate between OTC counterparties or with exchanges
  - Buy/sell stocks
  - Buy/sell futures
  - Buy/sell options
- FIX is used in some automated trading shops to execute orders generated by algorithm

- FIX is an open protocol, with detailed specifications at [www.fixprotocol.org](http://www.fixprotocol.org)
  - The most recent version of FIX is 5.0, but 4.4 is considered an industry standard
- Over the years, the open source community has contributed substantially to FIX implementations
- There exist several open source FIX engines, the best (in my opinion) being QuickFIX ([www.quickfixengine.org](http://www.quickfixengine.org))
- A FIX engine handles the nuts and bolts of transmitting and receiving messages

- The FIX message structure is simple
  - A FIX message is a string and consists of a collection of “tags”

- Example:

8=FIX.4.19=15435=649=BRKR56=INVMGR34=23652=19980604-  
07:58:4823=11568528=N55=SPMI.MI54=227=20000044=10100.00000025=H10=159



Each tag is labeled by a number.  
Each number has a specific definition, given in the protocol specifications.

Each numbered tag has data associated. This data describes the order. For instance, tag 55 is defined as the “Symbol” tag of the order. Here we see an example equity symbol associated with tag 55.

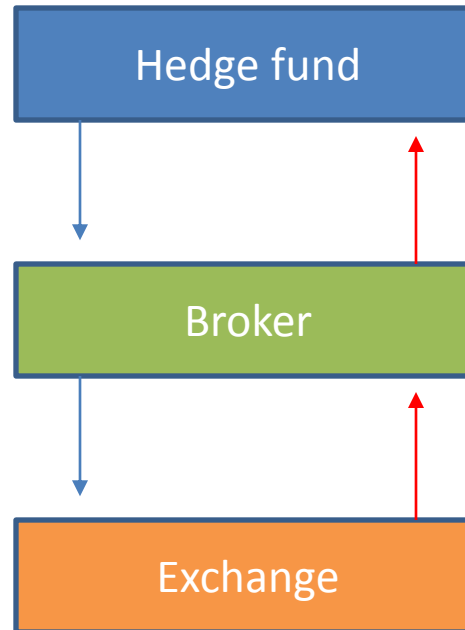
- The tags are separated with ASCII character code “1”

- Below is an illustration of the life of a FIX message

A hedge fund decides to place an electronic order to buy 1000 MSFT shares at the market.

The broker receives the order and after some basic validation, sends it to the exchange.

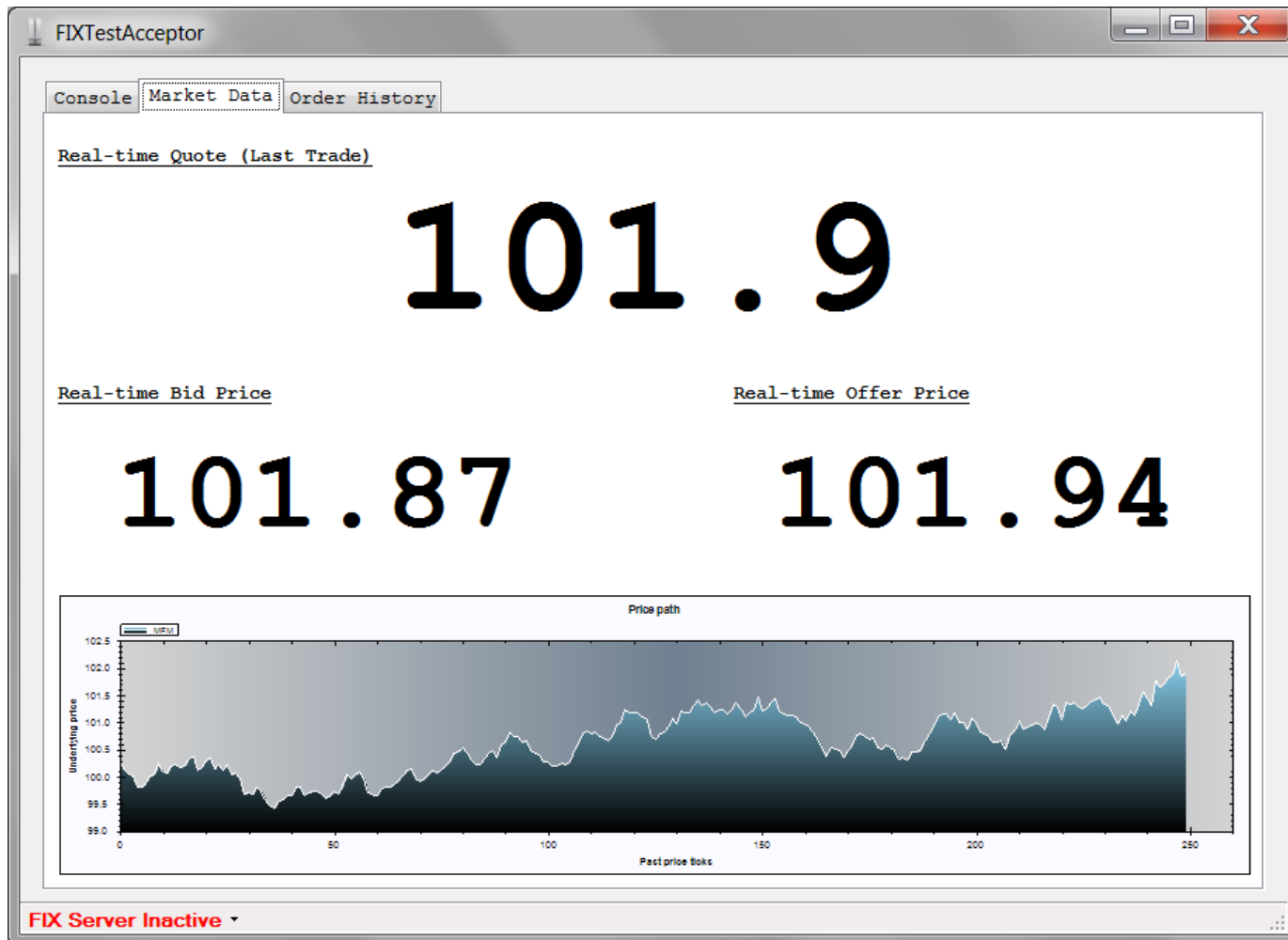
The trade is matched by the exchange's order matching algorithm. The order is now filled.



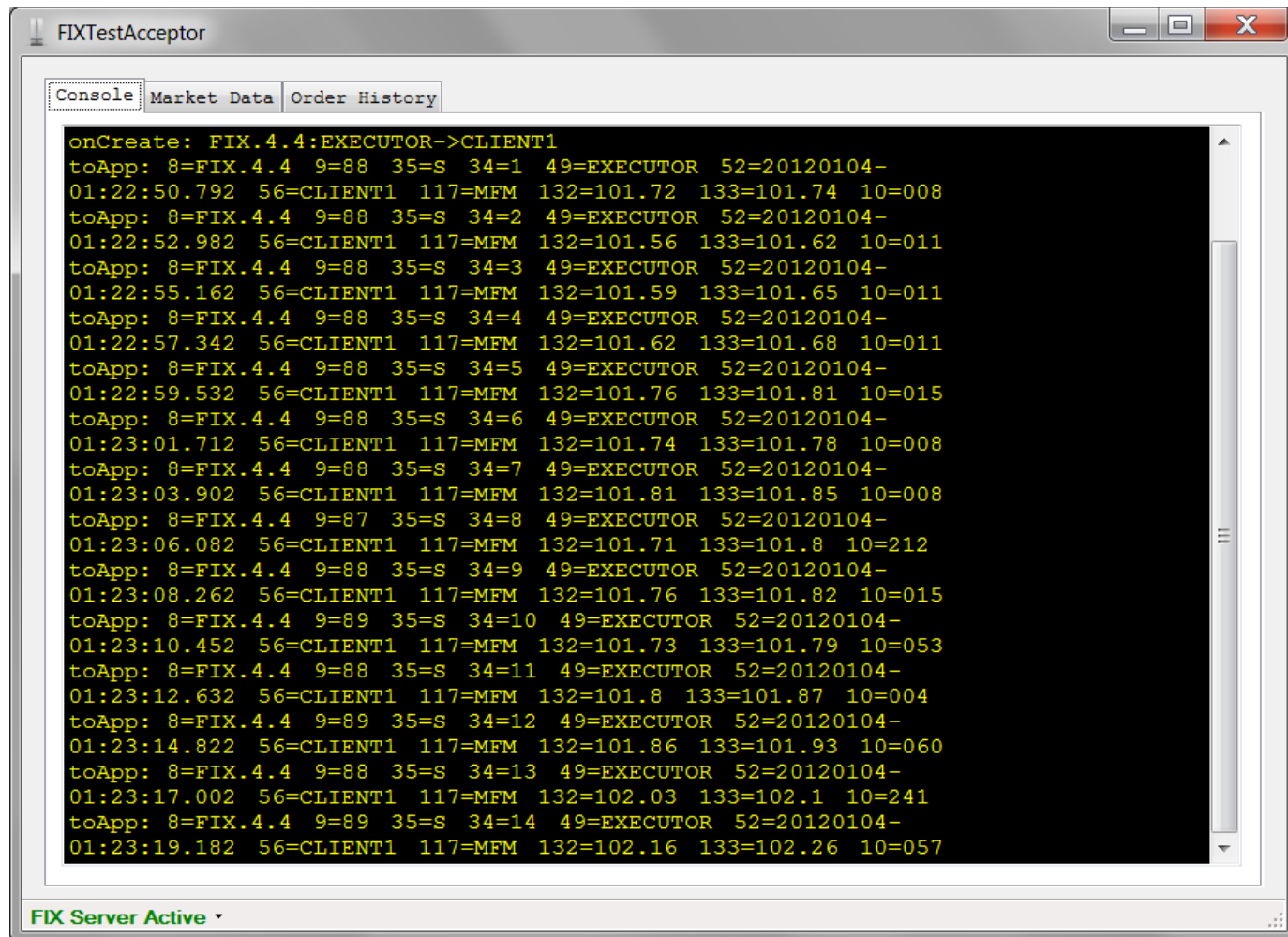
An order execution report is generated and sent to the broker, who in turn sends it to their customer (the hedge fund). Fill information is transmitted in this manner.

- Depending on the customer's speed requirement, the broker may not be present

- This module is centered around a FIX server application that I built
- The server generates a price time series



- When the server is activated, it generates a stream of quotes to any connected clients
- It also accepts market orders to buy and sell

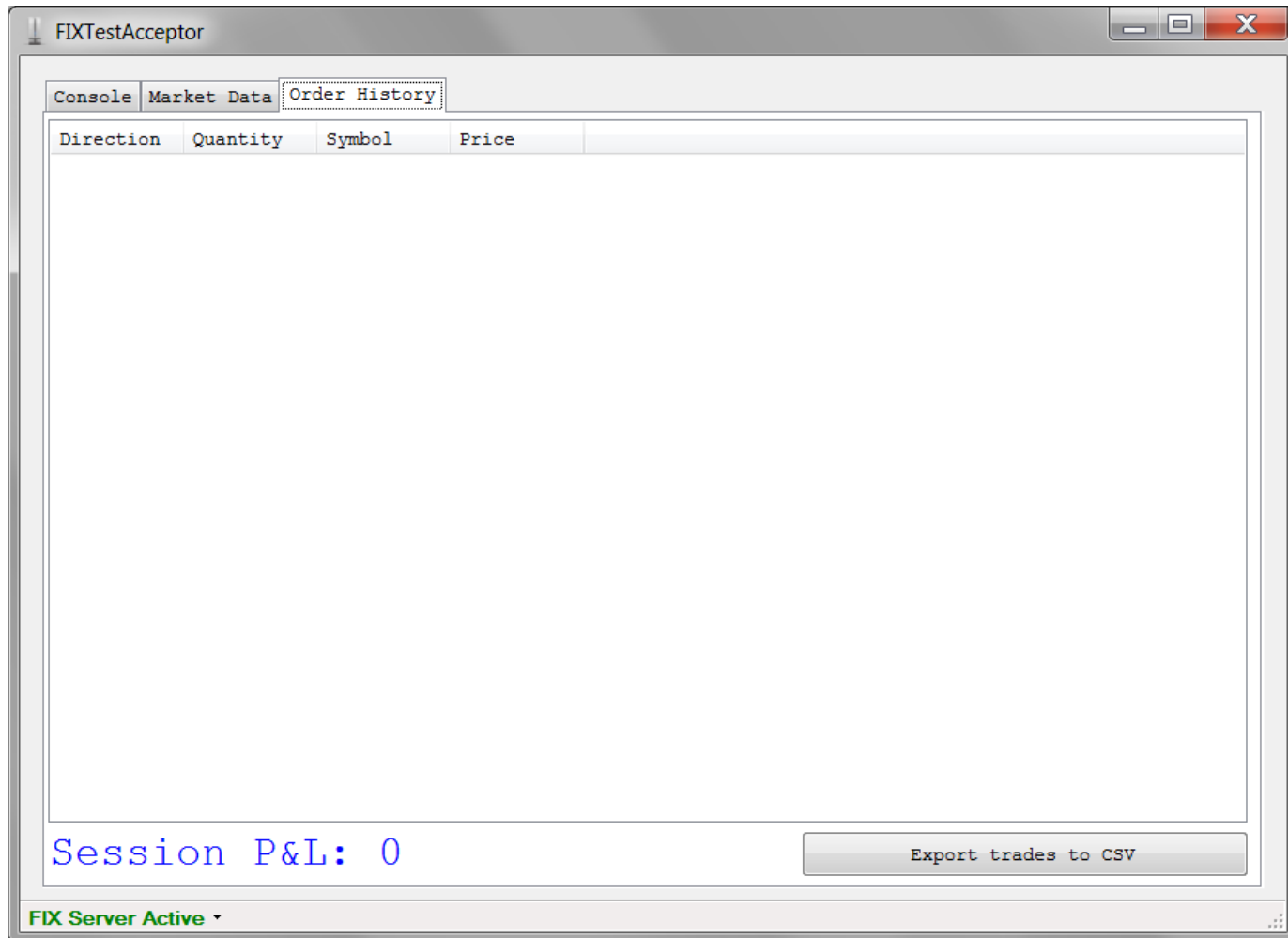


The screenshot shows a window titled "FIXTestAcceptor" with three tabs: "Console", "Market Data", and "Order History". The "Console" tab is active, displaying a stream of FIX messages. The messages are formatted as follows:

```
onCreate: FIX.4.4:EXECUTOR->CLIENT1
toApp: 8=FIX.4.4 9=88 35=S 34=1 49=EXECUTOR 52=20120104-
01:22:50.792 56=CLIENT1 117=MFM 132=101.72 133=101.74 10=008
toApp: 8=FIX.4.4 9=88 35=S 34=2 49=EXECUTOR 52=20120104-
01:22:52.982 56=CLIENT1 117=MFM 132=101.56 133=101.62 10=011
toApp: 8=FIX.4.4 9=88 35=S 34=3 49=EXECUTOR 52=20120104-
01:22:55.162 56=CLIENT1 117=MFM 132=101.59 133=101.65 10=011
toApp: 8=FIX.4.4 9=88 35=S 34=4 49=EXECUTOR 52=20120104-
01:22:57.342 56=CLIENT1 117=MFM 132=101.62 133=101.68 10=011
toApp: 8=FIX.4.4 9=88 35=S 34=5 49=EXECUTOR 52=20120104-
01:22:59.532 56=CLIENT1 117=MFM 132=101.76 133=101.81 10=015
toApp: 8=FIX.4.4 9=88 35=S 34=6 49=EXECUTOR 52=20120104-
01:23:01.712 56=CLIENT1 117=MFM 132=101.74 133=101.78 10=008
toApp: 8=FIX.4.4 9=88 35=S 34=7 49=EXECUTOR 52=20120104-
01:23:03.902 56=CLIENT1 117=MFM 132=101.81 133=101.85 10=008
toApp: 8=FIX.4.4 9=87 35=S 34=8 49=EXECUTOR 52=20120104-
01:23:06.082 56=CLIENT1 117=MFM 132=101.71 133=101.8 10=212
toApp: 8=FIX.4.4 9=88 35=S 34=9 49=EXECUTOR 52=20120104-
01:23:08.262 56=CLIENT1 117=MFM 132=101.76 133=101.82 10=015
toApp: 8=FIX.4.4 9=89 35=S 34=10 49=EXECUTOR 52=20120104-
01:23:10.452 56=CLIENT1 117=MFM 132=101.73 133=101.79 10=053
toApp: 8=FIX.4.4 9=88 35=S 34=11 49=EXECUTOR 52=20120104-
01:23:12.632 56=CLIENT1 117=MFM 132=101.8 133=101.87 10=004
toApp: 8=FIX.4.4 9=89 35=S 34=12 49=EXECUTOR 52=20120104-
01:23:14.822 56=CLIENT1 117=MFM 132=101.86 133=101.93 10=060
toApp: 8=FIX.4.4 9=88 35=S 34=13 49=EXECUTOR 52=20120104-
01:23:17.002 56=CLIENT1 117=MFM 132=102.03 133=102.1 10=241
toApp: 8=FIX.4.4 9=89 35=S 34=14 49=EXECUTOR 52=20120104-
01:23:19.182 56=CLIENT1 117=MFM 132=102.16 133=102.26 10=057
```

At the bottom of the window, a status bar indicates "FIX Server Active".

- Orders are recorded and the profit & loss of the trades is calculated with every market tick
- Demonstration of server and client



- The time series being generated by the server is geometric Brownian motion with a signal embedded
- Participants in my module have the following tasks:
  - Familiarize themselves with the FIX protocol and open source FIX messaging engine(s)
  - Build a FIX client that has the capacity to connect to my server app
  - Read and save a lengthy time series of FIX quotes generated by the server
  - Analyze the time series to determine the signal embedded in the Brownian motion
  - Expand the client application to submit profitable trades without human interaction

- Participants can expect to learn the following things from the module:
  - A good understanding of the FIX protocol
  - Implementation and configuration of an open source FIX messaging engine
  - Some time series analysis
  - Trading strategy design
- Note that this module will require intensive programming and intermediate/advanced understanding of C#