FeedOS

The Market Data Operating System

FeedOS Client API 2009-05-12

FIX Protocol Specification for MarketData

Table of content

Document History

Date	Author	Action
2005-07-12	D.Fenouil	Initial version
2007-04-01	D.Fenouil	Add sample sets of tags for common instruments
2009-05-10	D.Fenouil	Add support for FIX v4.4. DerivativeSecurityListRequest flagged as deprecated
2009-05-12	D.Fenouil	Update sample set of tags to reflect new feeds: Eurex “CEF Ultra+” and Xetra “CEF core”.

1 Overview

This document describes how to connect to a *FeedOS MarketData Server* using the FIX protocol.
The words *Market Data* relate to:

- Downloading referential ("static") data for financial instruments: Symbol, Maturity, etc.
- Receiving quotations (price-related data): trades, bid/ask, opening/closing prices, etc.

Nota Bene: Order Routing is **not** provided by *FeedOS MarketData*.

The areas covered in this document are:

1. How to *configure* a client FIX access towards a FeedOS system.
2. How to *reference instruments* and markets
3. The FIX *messages* supported (Requests & Responses).
4. For each message, the *subset of FIX values* expected.

1.1 FIX Version & Non-Standard extensions

Supported versions of FIX protocol are: v4.3, v4.4
Although the rule is to **enforce strict compliance** to the FIX standard, some deviations might be implemented on a case-by-case basis.
See http://www.fixprotocol.org/specifications/fix4.3fiximate/index.html

You should configure you FIX engine to accept these values for tag 167 (SecurityType):

- "FUT" (Future Contract)
- "OPT" (Option Contract)
- "INX" (Index)

It should be noted that the FIX interface can be configured to send a few non-standard tags.
Here is the list of tags and their meaning:

Tag number	Tag name	Syntax	Meaning
9506	PriceIncrement_static	double	Minimum price increment (aka "tick size")

1.2 Prerequisite

You need a FIX engine supporting FIX version 4.3 or 4.4.
We suggest using the open source QuickFix software.
See http://www.quickfixengine.org/

1.3 FIX Session Management

Normal FIX session mechanism allows recovering of missed messages in case of a disconnection. This is achieved by replaying those missed messages upon reconnection of both sides.
Although this behavior is essential in case of order routing, it may be a problem when it comes to sending large volumes of quotation data.
As a result, we decided to deactivate this feature by resetting the "sequence number" upon each new connection.

In this case the duration of a *FIX Session* (logical connection) equals the duration of the *FIX Connection* (network connection).

2 Configuring the connection

Please use these settings:

- Set FIX version to 4.3 or 4.4 (as specificed by QuantHouse staff)
- Set heartbeat interval to 30 seconds.
- Relax clock-synchronisation checkings, if any.

If using quickfix, please set:

- ResetOnLogout=Y
- ResetOnDisconnect=Y
- CheckLatency=N

A typical QuickFix configuration would be (for FIX v4.3):

```
[SESSION]
ConnectionType=initiator
ReconnectInterval=30
LogonTimeout=10
HeartBtInt=30
StartTime=07:15:00
EndTime=22:15:00
BeginString=FIX.4.3
SenderCompID=CLIENT_NAME
TargetCompID=FEEDOS_SERVER
DataDictionary=/my_fix_app/FIX43.xml
UseDataDictionary=Y
SocketConnectPort=6047
SocketConnectHost=66.66.66.66
ResetOnLogout=Y
ResetOnDisconnect=Y
CheckLatency=N
```

Nota Bene: parameters in **bold** will be provided by QuantHouse staff.

3 Referencing instruments

3.1 Markets

Financial markets are referenced by using the **M**arket **I**dentifier **C**ode (ISO 10383) in tag 207 (*Security Exchange*).

3.2 Instruments

Preferred solution: the Local Code String

Within a given market, individual instruments are referenced using a proprietary, market-specific code called the ***Local Code String***. The ***Local Code String*** is unique within a given market. It is implemented using tags 22 (*SecurityIDSource*) and 48 (*SecurityID*).

The combination of the MIC and the Local Code String allows to unambiguously reference any instrument within a FeedOS system.

This information is **provided by the FeedOS system** (when downloading referential data). Conversely, user should use the same information to reference instruments when building "request" messages.

Examples:

MIC	Local Code String	Comment
`XEUR`	`FDAX1209`	the Eurex DAX Future that expires on 2009-12
`XWCE`	`WABO1209P1150`	a PUT Option on Winnipeg Commodity Exchange
`XLIF`	`E+1xLS0306-1xLS0907`	a Calendar Spread on LIFFE

Thus referencing an instrument requires only 3 tags:

- Tag 207 (*Security Exchange*) = the **MIC**
- Tag 22 (*SecurityIDSource*) = **8** (*Exchange Symbol*)
- Tag 48 (*SecurityID*) = the **Local Code String**

These 3 tags are sufficient to unambiguously reference any instrument, including derivatives and multi-legs.

Nota Bene:

When appropriate, use tags 308, 305 and 309 (in Component Block UnderlyingInstrument).

Other naming schemes

You can still use the whole *Instrument Component Block* (tags `Symbol, CFICode, SecurityType, StrikePrice, MaturityDate, MaturityMonthYear, OptAttribute, NoLegs, InstrumentLeg`, etc) to reference instruments, although this is less reliable and less efficient.

FeedOS to FIX mapping

Most of FeedOS internal data follows the FIX protocol, hence the mapping is usually straightforward (especially for quotation data).
Regarding referential data, here is a list of sample values for different kinds of instruments.
You can see that most of internal “attributes” are mapped directly into FIX tags.

3.3 Stock

FeedOS Attributes	FIX tags
FOSMarketId=89 (*XETR*)	**SecurityExchange**=XETR
Symbol=BMW	**Symbol=**BMW
PriceCurrency=EUR	**Currency=**EUR
LocalCodeStr=DE0005190003	**SecurityIdSource**=8 (*Exchange Symbol*)
	SecurityId=DE0005190003
ISIN=DE0005190003	**SecurityAltIdSource**=4 (*ISIN*)
	SecurityAltId=DE0005190003
Description= BAY.MOTOREN WERKE AG ST	**SecurityDesc**= BAY.MOTOREN WERKE AG ST
SecurityType=NONE	**SecurityType**=NONE
CFICode=EXXXXX	**CFICode**=EXXXXX

3.4 Index

FeedOS Attributes	FIX tags
FOSMarketId=89 (*XETR*)	**SecurityExchange**=XETR
Symbol=DAX	**Symbol=**DAX
LocalCodeStr=DE0008469008	**SecurityIdSource**=8 (*Exchange Symbol*)
	SecurityId=DE0008469008
ISIN=DE0008469008	**SecurityAltIdSource**=4 (*ISIN*)
	SecurityAltId=DE0008469008
Description= DAX (PERFORMANCE-INDEX)	**SecurityDesc**= DAX (PERFORMANCE-INDEX)
SecurityType=NONE	**SecurityType**=NONE
CFICode=MRIXXX	**CFICode**=MRIXXX

3.5 Currency Exchange Rate

FeedOS Attributes	FIX tags
FOSMarketId=502 (*Xfor*)	**SecurityExchange**=Xfor
LocalCodeStr=EUR/AUD	**SecurityIdSource**=8 (*Exchange Symbol*)
	SecurityId=EUR/AUD
Description=AUD for 1 EUR	**SecurityDesc**=AUD for 1 EUR
SecurityType=NONE	**SecurityType**=NONE
CFICode=MRCXXX	**CFICode**=MRCXXX

3.6 Future

FeedOS Attributes	FIX tags
FOSMarketId=12 (*XEUR*)	**SecurityExchange**=XEUR
Symbol=FDAX	**Symbol=**FDAX
PriceCurrency=EUR	**Currency=**EUR
LocalCodeStr=FDAX1209	**SecurityIdSource**=8 (*Exchange Symbol*)
	SecurityId=FDAX1209
Description= DAX(R) Futures	**SecurityDesc**= DAX(R) Futures
SecurityType=FUT	**SecurityType**=FUT
CFICode=FFIXXX	**CFICode**=FFIXXX
MaturityYear=2009 **MaturityMonth**=12 **MaturityDay**=08	**MaturityMonthYear**=200912
	MaturityDate=20091218
ContractMultiplier=25	**ContractMultiplier**=25
PriceIncrement_static=0.5	**9506**=0.5

3.7 Option

FeedOS Attributes	FIX tags
FOSMarketId=12 (*XEUR*)	**SecurityExchange**=XEUR
Symbol=ODAX	**Symbol=**ODAX
PriceCurrency=EUR	**Currency=**EUR
LocalCodeStr=ODAX1209C6000	**SecurityIdSource**=8 (*Exchange Symbol*)
	SecurityId=ODAX1209C6000
Description=DAX(R) Options	**SecurityDesc**=DAX(R) Options
SecurityType=OPT	**SecurityType**=OPT
CFICode=OCXIX	**CFICode**=OCXIXX
StrikePrice=6000	**StrikePrice**=6000
OptAttributeVersion=***n*** (*not present when 0*)	**OptAttribute**=***n*** (*not present when 0*)
MaturityYear=2009 **MaturityMonth**=12 **MaturityDay**=08	**MaturityMonthYear**=200912
	MaturityDate=20091218
ContractMultiplier=5	**ContractMultiplier**=5
PriceIncrement_static=0.1	**9506**=0.1

NB: tag PutOrCall(201) is deprecated. Use 2nd letter of CFICode instead.

3.8 Double Leg Strategy

FeedOS Attributes	FIX tags
FOSMarketId=12 (*XEUR*)	**SecurityExchange**=XEUR
Symbol=FDAX	**Symbol=**FDAX
PriceCurrency=EUR	**Currency=**EUR
LocalCodeStr=+1xFDAX0609-1xFDAX1209	**SecurityIdSource**=8 (*Exchange Symbol*)
	SecurityId=+1xFDAX0609-1xFDAX1209
SecurityType=MLEG	**SecurityType**=MLEG
CFICode=MRXXXX	**CFICode**=MRXXXX
NbLegs=2	**NoLegs**=2
LegFOSInstrumentCode(0)=12/709448	**InstrumentLeg(0)/SecurityIdSource**=100 (*FeedOS internal code*)
	InstrumentLeg(0)/SecurityId=12/709448
LegRatioQty(0)=1	**InstrumentLeg(0)/LegRatioQty=**1
LegSide(0)=1	**InstrumentLeg(0)/LegSide=**1
LegFOSInstrumentCode(1)=12/925827	**InstrumentLeg(1)/SecurityIdSource**=100 (*FeedOS internal code*)
	InstrumentLeg(1)/SecurityId=12/925827
LegRatioQty(1)=1	**InstrumentLeg(1)/LegRatioQty=**1
LegSide(1)=2	**InstrumentLeg(1)/LegSide=**2

4 Overview of supported messages

4.1 Administrative Messages

All administrative messages supported by FIX 4.3 are used. These messages are:

- Logon
- Heartbeat
- TestRequest
- ResendRequest
- Reject
- SequenceReset
- Logout

4.2 Application Messages

The following client **Request** messages are supported (accepted by the server):

- SecurityListRequest
- DerivativeSecurityListRequest
- MarketDataRequest

In response to these requests, the following **Response** messages are supported (sent by the server):

- SecurityList
- DerivativeSecurityList
- MarketDataSnapshotFullRefresh
- MarketDataRequestReject
- BusinessMessageReject

5 Referential ("static") Data

5.1 Downloading the referential data

The list of financial instruments available on the server may be downloaded by using the following request message.
We suggest that user application perform this "full download" (once a day for example) and use data locally afterwards.

Request Message: SecurityListRequest

Expected tag values:

- SecurityListRequestType=0,1 or 4 (*all securities*)
- SubscriptionRequestType=0 (*snapshot*)

Optional tags in Component Block "Instrument" may be used as a filter, for example by specifying values for:

- CFICode
- SecurityType
- SecurityExchange

Optionally, continuous real-time updates may be received when using SubscriptionRequestType=1 (*snapshot+updates*).

Response Message: SecurityList

As a result of a successful SecurityListRequest, a bunch of SecurityList messages will be sent to the client.
Various fields will be filled with instruments' characteristics. Although FeedOS attempts to fill as many tags as possible, the exact content is market-dependent.

5.2 Requesting the list of derivatives

Use this to retrieve the list of derivatives pertaining to an "underlying" instrument.
Nota Bene: this is not effective on every market. DEPRECATED

Request Message: DerivativeSecurityListRequest

Expected tag values:

- Component Block "Underlying Instrument"
- SecurityListRequestType=4 (*all securities*)
- SubscriptionRequestType=0 (*snapshot*)
- Component Block "UnderlyingInstrument" should be used to select the underlying.

Response Message: DerivativeSecurityList

As a result of a successful DerivativeSecurityListRequest, this message returns the list of derivatives.

6 Quotation Data

The MarketDataRequest is used to snapshot or subscribe to price information.

6.1 Performing a snapshot

Use this to perform a snapshot of quotation data for one or more instruments.

Request Message: MarketDataRequest

Expected tag values:

- MDReqID
- Component Block “Instrument”, one occurrence per target instrument
- SubscriptionRequestType=0 (*snapshot*)
- AggregatedBook=Y (*yes*)
- MarketDepth=***n***

Response Message: MarketDataSnapshotFullRefresh

This message is sent after a successful MarketDataRequest, one per target instrument.

6.2 Starting a subscription

Use this to start a subscription (continuous real time updates) of quotation data for one or more instruments.

Request Message: MarketDataRequest

Expected tag values:

- MDReqID
- Component Block “Instrument”, one occurrence per target instrument
- SubscriptionRequestType=1 (*snapshot* + *updates*)
- MDUpdateType=0 (Full Refresh)
- AggregatedBook=Y (yes)
- MarketDepth=***n***
    - ***n*** =0 to disable the sending of order book data
    - ***n*** =-1 to get “all” levels (“all” depends on what is available from the market)
    - ***n*** >=2 to receive only the best ***n*** levels

  Maximum depth available depends on the market, the instrument type, and the user’s permissions.

Response Message: MarketDataSnapshotFullRefresh

In case of a successful subscription, a first “full refresh” message (one per target instrument) is sent immediately. After that, “update” messages are sent for each event occurring on a target instrument (transaction, change in Ask/Bid limits, etc).

Response Message: MarketDataRequestReject

This is sent in case of a missing or invalid argument, or if the requested MarketDepth is not available. The tags MDReqRejReason (and eventually Text) tell the reason of the failure.

6.3 Stopping an ongoing subscription

After a successful MarketDataRequest, you may want to stop receiving continuous updates.

Request Message: MarketDataRequest

Expected tag values:

- MDReqID of previous MarketDataRequest that started the subscription.
- SubscriptionRequestType=2 (*disable updates*)

Response Message

No response is sent. The “unsubscription” always succeeds.