



FIXML

FIX Technical Background



FIX Session and Application Layers

Trading Application

Session Layer

- Establishing and terminating the connection
- Message Delivery
data integrity, sequencing, addressing

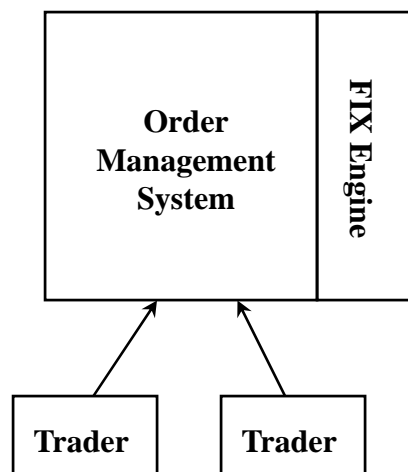
Application Layer

- Business related content
Order, Execution, IOI, etc

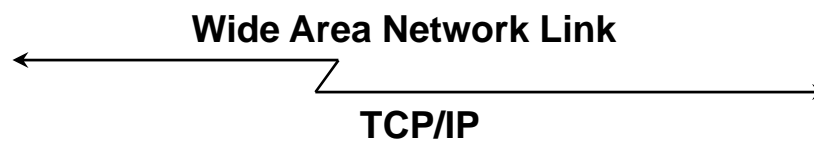
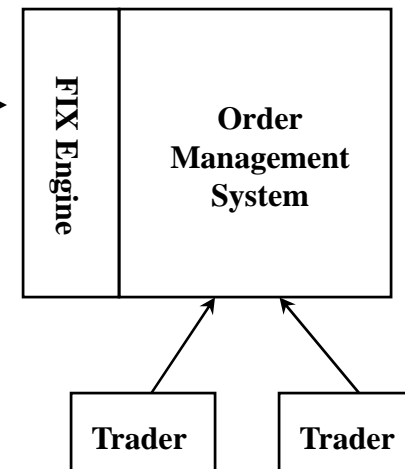
FIX Protocol

FIX Connectivity

Investment Manager



Broker/Dealer





FIX Message Structure



Message Header

Identifies message type, message length, sender/destination ,sequence number, sending time, etc

Message Body

Contains specific session and application message content

Message Trailer

Contains optional digital signature and the required checksum value



FIX 4.1 Message Types

Session Messages

Heartbeat
Logon
Test Request
Resend Request
Reject
Sequence Reset
Logout

Application Messages

Advertisement
Indications of Interest
News
New Order - Single
Execution Report
Don't Know
Order Cancel/Replace
Order Cancel Request
Order Cancel Reject
Order Status Request
Allocation
Allocation ACK

New Order - List
List Status
List Execute
List Cancel Request
List Status Request
Quote
Quote Request
Email
Don't Know Trade
Settlement Instr.



Message Types Added for 4.2

Application Messages

Mass Quote

Quote Ack

Market Data Request

Market Data Request Reject

Market Data - Full/Snapshot/Incremental

Security Request/Definition/Status

Trading Session - Request/Status

Bid Request/Response

Business Message Reject



FIX Fields

<TAG>=<VALUE>**<DELIMITER>**

Composed of four parts

----- <TAG>

----- “=”

-----<VALUE>

-----<DELIMITER>

non-printing ASCII character called the SOH
character

Example:

“8=FIX.4.2^”



FIX Single Order Example

8=FIX.4.1^9=0235^35=D^34=10^43=N^49=VENDOR^50=CUSTOMER^56=BROKER^52=19980930-09:25:58
^1=XQCCFUND^11=10^21=1^55=EK^48=277461109^22=1^54=1^38=10000^40=2^44=76.750000^59=0^10=165

Header

8=FIX.4.1
9=235
35=D
34=10
43=N
49=VENDOR
115=CUSTOMER
56=BROKER
52=19980930-
09:25:58

Begin String
Body Length
MsgType
MsgSeqNum
PossDupFlag
SenderCompID
OnBehalfOfCompID
TargetCompID
Sending Time

Body

1=XQCCFUND
11=12345
21=1
55=EK
48=277461109
22=1
54=1
38=10000
40=2
44=76.750000
59=0

Account (optional)
ClOrdID
HandInst
Symbol
SecurityID (optional)
IDSource (optional)
Side
OrderQty
OrdType
Price (optional)
TimeInForce (optional)

Trailer

10=165

Checksum



FIXML

FIX and XML



FIXML: Example Syntax

```
8=FIX.4.1^9=0235^35=D^34=10^43=N^49=VENDO
R^50=CUSTOMER^56=BROKER^52=19980930-
09:25:58
^1=XQCCFUND^11=10^21=1^55=EK^48=27746110
9^22=1^54=1^38=10000^40=2^44=76.750000^59=0
^10=165
```

Becomes...

```
<?xml version='1.0'?><!DOCTYPE FIXML SYSTEM 'fixmlmain.dtd'>
<FIXML>
  <FIXMLMessage>
    <Header>
      ...
    </Header>
    <ApplicationMessage>
      <Order>
        <ClOrdID>12345</ClOrdID>
        <HandlInst Value="1"/>
        <Instrument>
          <Security>
            <Symbol>EK</Symbol>
          </Security>
        </Instrument>
        <Side Value="1"/>
        <OrderQuantity>
          <OrderQty>10000</OrderQty>
        </OrderQuantity>
        <OrderType>
          <MarketOrder OrdType="1"/>
        </OrderType>
        <Currency Value="USD"/>
      </Order>
    </ApplicationMessage>
  </FIXMLMessage>
</FIXML>
```

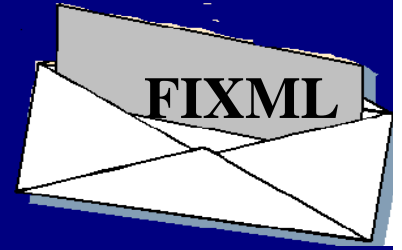


FIXML: Implementation Issues

- Easy migration for existing FIX engines
 - “Embedded FIXML”
- Backward-compatibility
 - optional field can co-exist with “standard” tag=value data
 - XML attributes represent existing FIX tags
- Session Layer remains intact
 - core engine is not affected



FIXML: Embedded messages



FIXML can be “wrapped” as a value of a standard FIX tag

49=BROKER

56=HUB

128=INST

212=245

213=<FIXML>

<Header>

...

</Header>

<Indication>

...

</Indication>

</FIXML>

SenderCompID

TargetCompID

DeliverToCompID

FIXML Data Length

FIXMLData



FIXML: Major Business Goals

- Utilize existing systems and processes
- Protect investment in traditional FIX
- Provide migration path to next generation FIX systems
- Impose little or no impact on existing business applications
- Position FIX for greater interoperability with other industry standards