Transitioning to Advanced Versions of Messaging Standards

Ryan Pierce

Townsend Analytics Ltd. / Archipelago LLC

February 26, 2001 – New York

Introduction

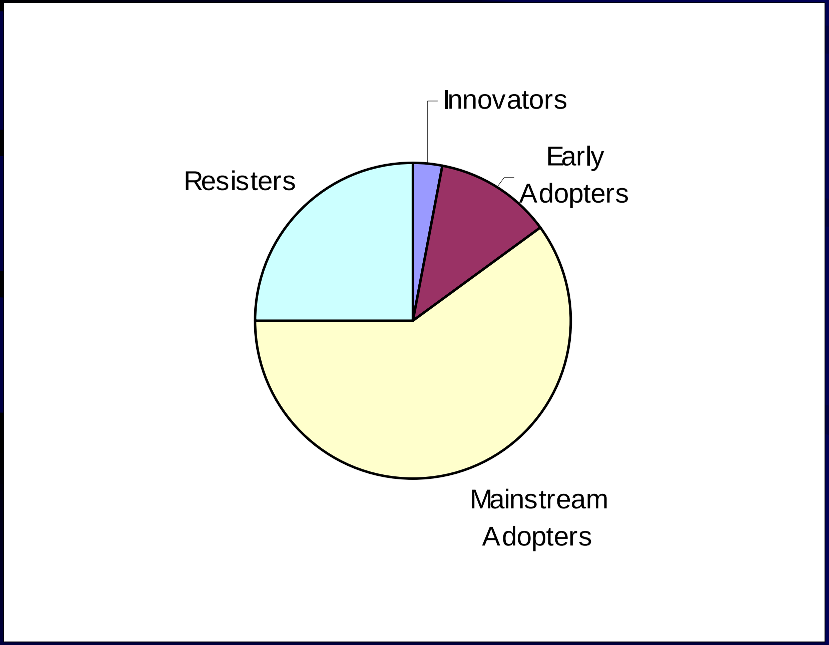
Outline

- Discussion of Technology, Protocols and Change
- Discussion of FIXML
- FIX Versions
    - 4.0 to 4.1 Migration, 4.1 New Features
    - 4.1 to 4.2 Migration, 4.2 New Features
- FIXML Implementation

Adoption of Technology

- New technology is good…
- …But someone has to pay for it.
- Management of risks and timing entry
  - Development / deployment / support costs
  - Will others adopt it?
  - What are the risks of NOT adopting it?

Adoption of Technology – Wheel of Change



- Innovators – creators of technology
- Early Adopters – often trading partners of innovators
- Mainstream Adopters – jumping on as technology reaches critical mass
- Resisters – do not want to support new technologies

Timing Adoption - Innovators

- Resulting protocol meets your business needs well; you’re not retrofitting a protocol ill-suited to your business
- Builds a positive, leading image for your firm
- High development costs
- Risk that nobody else will implement the protocol
- Risks can be lessened by working with a standards body rather than creating a proprietary standard.

Timing Adoption – Early Adopters

- Already is a (small) established client / vendor base
- Positive image for your firm
- Risks:
  - Protocol has not reached critical mass; no guarantee that it will take off
  - Development costs might not be justified by business revenues

Timing Adoption – Mainstream Adopters

- Wide client / vendor base to build upon
- Protocol has critical mass
- Risks:
  - Entering too late may mean you've lost business opportunities / market share
  - Protocols may soon be obsolete

Hub and Spoke Model

- Examples: SWIFT, GSTPA
- Hub can set standards, mandate version upgrades
- New entrants to the protocol know which protocol version to implement
- Upgrade schedule is often controlled by hub, not by users
- Upgrades often must happen less frequently
- Often less differences in protocol interpretation, as the hub is the authority on what is compliant

Peer to Peer Model

- Example: FIX
- Many protocol versions are simultaneously used in production
- New entrants to the protocol don’t have a clear-cut answer of what version to implement
- Often more differences in protocol interpretation, and more differences in feature support
- Upgrade schedule determined by users
- New versions of the standard can be released more frequently, as mandatory upgrades are not forced

What to Support?

- All versions! Maximum flexibility! …
- …But this comes at quite a lot of development and support costs.
- Just the latest version! Minimize costs! …
- …But this can seriously limit one’s trading partners.
- One needs to ascertain the versions supported by the industry as a whole, one’s current clients / vendors, one’s potential clients / vendors, and one’s competitors.

Understanding FIX vs. FIXML

- FIX and FIXML are not widely different protocols
- FIXML provides an XML syntax for writing FIX messages
  - FIXML still uses the same FIX data dictionary and business logic
  - FIXML and FIX difference is like a typewritten letter vs. a handwritten letter; the representation is different, however the content is identical
  - FIXML mapping rules to represent FIX in XML
- FIXML provides no session layer; one can encapsulate it in FIX or use another protocol

Understanding FIX vs. FIXML

	FIX	FIXML
Data Representation	Tag=Value[SOH] format	XML DTD (Schema version planned)
Session	Yes	No
Data Dictionary	Same	Same
Business Rules / State Changes	Same	Same

Why Not FIXML?

- Bandwidth. FIXML takes much more bandwidth to transmit the same amount of data. FIXML may take more than 5 times the bandwidth of FIX.
- Performance. XML parsers, especially validating parsers, might be slower than simple FIX engines. Note: I have no concrete numbers here; just keep the possibility in mind.
- Support. FIXML is not presently widely supported.

Why FIXML?

- Can use off-the-shelf parser software
  - Not a big selling point; writing a FIX Tag=Value parser is rather trivial.
- Automatic validation of messages
  - Also not a big selling point, as DTD validation is weak. XML Schema will be better.

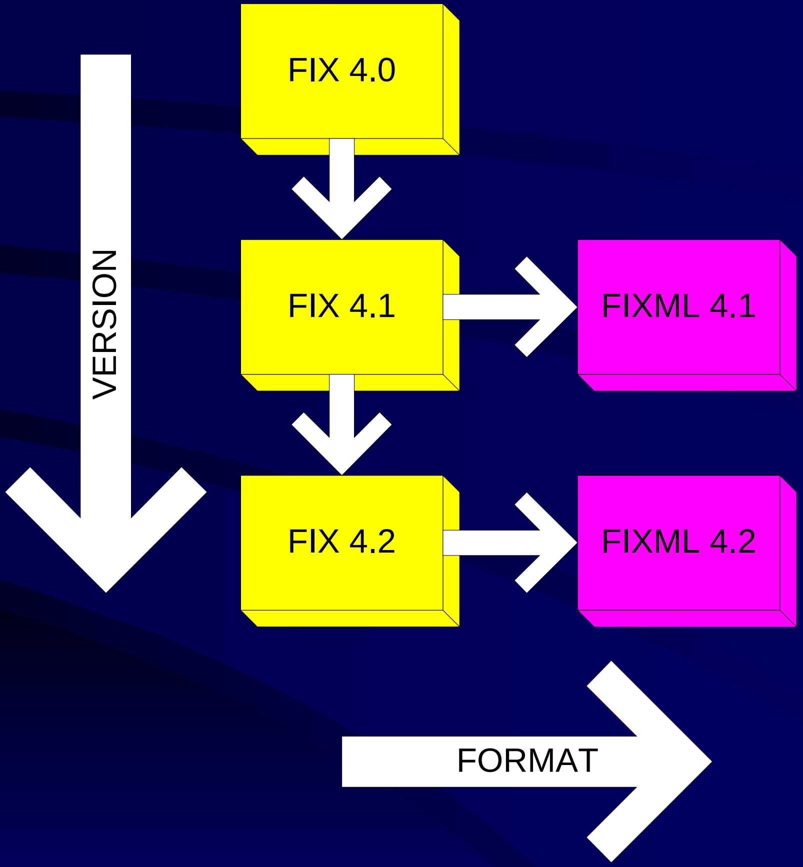
Why FIXML?

- XML is becoming the meta-language for standards. All kinds of industries are inventing vertical XML grammars. E-Commerce is redefining itself in XML, and FIX needs to support XML to stay with the times.
- Other in-house applications use XML. Transporting FIXML internally may make sense, as well as using it externally.

Why FIXML?

- You wish to use middleware or other protocols instead of the FIX session.
  - Microsoft's Message Queue (MSMQ), IBM's MQSeries, TIBCO Rendezvous
  - Horizontal XML initiatives like ebXML, SOAP, XML Protocol
- You want to use FIX with other XML protocols (FpML, OFX, etc.)
- If trading partner(s) want to use it.

FIX and FIXML Migration Path



Importance of FIX Version

- Protocol version has little impact on the session after FIX 4.0
- Version has quite a significant impact on application-level business processing
- FIX Version may be a bigger issue than FIX vs. FIXML
  - FIX to FIXML translating can be done externally
  - FIX version changes often must be carried back farther into the core of one’s system
- 4.0 to 4.1 likely more difficult than 4.1 to 4.2

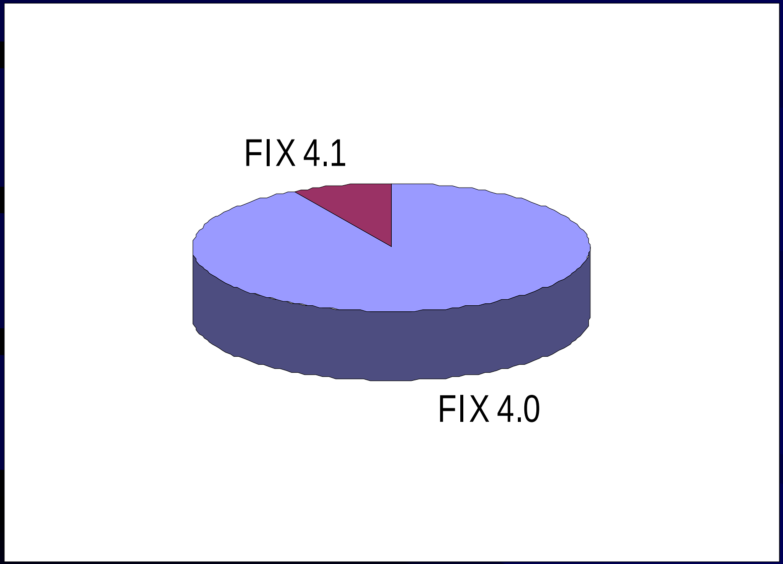
Easing the Transition Between Versions

- While not FIX compliant, trading partners can agree to implement some fields and messages from later FIX versions in earlier FIX versions.
  - Example: Using DiscretionInst, DiscretionOffset (introduced in FIX 4.2) in FIX 4.0 or 4.1
- Allows for additional functionality without requiring significant development effort to upgrade versions.
- Warning: May not be compatible with some FIX parser validation

FIX Version History

	FIX 3.0	**FIX 4.0**	**FIX 4.1**	**FIX 4.2**
Release Date	Sep 1995	Jan 1997	Apr 1998	Mar 2000
# Admin Msgs	7	7	7	7
# Business Msgs	17	20	21	39
# Fields	112	138	208	396
# Appendices	4	4	7	16
# pages in spec	57	69	106	265

Archipelago FIX Order Flow Sessions – 2/2001



- FIX 4.0: 92%
- FIX 4.1: 8%
- FIX 4.2: 1 interested client
- FIXML: 0 interested clients
- FIX 4.0 to 4.1 upgrades: 0

Why FIX 4.0?

- FIX 4.0 greatly improved upon the session reliability of 3.0. The 4.0 session remains almost unchanged in later versions.
- FIX 4.0 contains just about all one needs to handle US Equities
- FIX 4.0 was current when FIX gained critical mass
- Y2K: FIX 3.0 uses 2-digit years.

Why Not FIX 4.0?

- Later protocols have more functionality
  - More business uses
  - More instrument types (derivatives)
  - Better international support
- Later protocols have much less ambiguity (although trading partners can agree on how to handle ambiguous cases)
- Currently no FIXML DTD for FIX 4.0

FIX 4.0 Ambiguities

- Partial fills while a cancel is pending
  - FIX 4.0 uses OrdStatus without really defining it; FIX 4.1 uses OrdStatus (current state of order) and ExecType (type of action done to order)
  - Is OrdStatus = Partially Filled, or Pending Cancel / Replace?

FIX 4.0 Ambiguities

- Response to Cancels – what the protocol says:
  - Order has ClOrdID = X, Cancel has ClOrdID=Y
  - First, send Execution Report with ClOrdID = Y, OrdStatus = New
  - Then, send Execution Report with ClOrdID = X, OrdStatus = Canceled

FIX 4.0 Ambiguities

- Response to Cancels – what many people do:
  - Order has ClOrdID = X, Cancel has ClOrdID=Y
  - First, send Execution Report with ClOrdID = X, OrdStatus = Pending Cancel/Replace
  - Then, send Execution Report with ClOrdID = X, OrdStatus = Canceled

FIX 4.0 Ambiguities

- Response to Cancel/Replace – what the protocol says:
  - Order has ClOrdID = X, Cancel/Replace has ClOrdID = Y
  - First, send Execution Report with ClOrdID = X, OrdStatus = Pending Cancel/Replace
  - Then, send Execution Report with ClOrdID = Y, OrdStatus = New

FIX 4.0 Ambiguities

- Response to Cancel/Replace – what some people do:
  - Order has ClOrdID = X, Cancel/Replace has ClOrdID = Y, for second Execution Report:
  - Send Execution Report with ClOrdID = Y, OrdStatus = Replaced
  - Or send Execution Report with ClOrdID = X, OrdStatus = Replaced

Recommendations to Ease Version Upgrades

- Treat FIX Version as an integer.
- Assume anything done in one FIX version will carry over into later versions. You'll usually be right.
- C++ Example:

```
if ( m_iFixVersion >= FIX_VERSION_4_1 )
   pMsg->AddField( FIX_LEAVES_QTY, iLeavesQty );
```

Recommendations to Ease Version Upgrades

- Always look at the latest Errata for each spec. At present, 4.1 has Errata.
- For FIX 4.1, consult FIX 4.2 Appendix D (State Change Matrices)
  - 4.2 Appendix D covers many more scenarios
  - In general, the rules haven’t changed. Just look out for FIX 4.2-specific features (separate Pending Cancel and Pending Replace states, ExecType=Restated, Good Till orders, etc.)

Recommendations to Ease Version Upgrades

- Do not skip versions (I.e. FIX 4.0 to 4.2)
  - Changes are usually cumulative, hence little to no extra work is needed to develop the intermediate version
  - Gives you more flexibility in communicating with clients / vendors
  - Gives you an intermediate milestone for your Q/A group to test, or for clients to use in production

FIX 4.0 to FIX 4.1 Migration

- When responding to a Resend Request, SeqReset – Gap Fill and SeqReset – Reset must be sent Poss Dupe.
  - This is implied in FIX 4.0, but was made explicit in FIX 4.1. This is a good practice to avoid certain session problems.
- All ID fields are now type char.
  - Includes AdvID, AdvRefID, ExecID, ExecRefID, IOIid, IOIRefID, AllocID, RefAllocID.
  - One can generally treat these as strings when receiving them regardless of version, and when generating an ID, use an integer in FIX 4.0.

FIX 4.0 to FIX 4.1 Migration

- CxlType is removed.
  - One can no longer use the Cancel Request message to do a partial cancel. This must be done via a Cancel / Replace (Modification) Request.
- Subject now a required field in Email.
- Cancel Reject must specify both ClOrdID and OrigClOrdID to identify the Cancel/Change and the Order.
- LeavesQty added to ExecutionRpt and ListStatus.
  - LeavesQty=OrderQty – CumQty if the order is open; otherwise it is 0.

FIX 4.0 to FIX 4.1 Migration

- Significant changes to order state transitions by addition of ExecType,
  - ExecType denotes the type of message, OrdStatus the current state of an order. A precedence value for these is defined.
  - A trade happening while a cancel is pending will have OrdStatus=Pending Cancel/Replace, and ExecType=Partial Fill
- Each transition requires a separate FIX message
  - I.e. you cannot report a partial fill and cancel the order in one message.

FIX 4.0 to FIX 4.1 Migration

- Chaining of ClOrdID and OrigClOrdID
  - When acknowledging accepting, or rejecting a change or cancel, you must specify both OrigClOrdID and ClOrdID.
  - I.e. accepting a cancel or reporting an order as canceled, ClOrdID = ID of cancel, OrigClOrdID = ID of order.
- OrdStatus added to Cancel Reject.
- Removed NoMiscFees, MiscFees; to be handled with Allocation message
- Significant changes made to Allocations

FIX 4.1 New Features

- 24-hour sessions through ResetSeqNumFlag
- LocationID fields added to header
- Common symbology block used in all messages with Symbol
  - Fields include SymbolSfx, IDSource, SecurityID, Issuer, SecurityDesc
- Additional symbology for derivatives:
  - SecurityType, MaturityMonthDate, MaturityDay, PutOrCall, StrikePrice, OptAttribute, SecurityExchange

FIX 4.1 New Features

- SecondaryOrderID in ExecutionRpt and Order Cancel Reject. Used when routing orders; one can inform the client of the order's ID at the execution point.
- Added two new Side values
  - Undisclosed (for IOIs)
  - Cross (for exchanges)

FIX 4.1 New Features

- Additions to Quote and/or QuoteRequest
  - TransactTime (Quote), OrderQty, OrderType (Request), Fields for Forex (FutSettDate, OrderQty2, FutSettDate2, BidSpotRate, BidForwardPoints, OfferSpotRate, OfferForwardPoints)
- NoRelatedSym added to News and Email so that securities listed are part of a repeating group.
- EmailThreadID added to Email.
- Headline added to News.
- URLLink added to IOI, Advertisement, News.

FIX 4.1 New Features

- “ISO Currency Code” and “ISO Country Code” added to IDSource for use with News and Email.
- TradeDate, LastMkt added to Advertisement.
- TransactTime added to IOI.
- Added NoIOIQualifiers to make a repeating group. Added “At the Midpoint” and “Pre-open” as IOIQualifier enumerations.
- New ExecInst “U” to denote customer display instruction (SEC Rule 11Ac1-1/4)
- CashOrderQty added to Orders

FIX 4.1 New Features

- New OrdRejReason values: Unknown Order and Duplicate Order
- Numerous F/X changes made to order-related messages.
- For options trading, added OpenClose, CoveredOrUncovered, CustomerOrFirm to New Order Single and List, Cancel/Replace.
- Added missing Rule80A enumerations

FIX 4.1 New Features

- Added MaxShow to New Order Single and List, Cancel/Replace
  - Defines what is shown to customers of the ECN or trading network.
  - Used with MaxFloor to specify two levels of reserve quantities; MaxFloor indicates what is shown to the public
- LocateReqd in Order Cancel/Replace. Modified list of fields which may change.

FIX 4.1 New Features

- PegDifference added to order-related messages.
  - Example: allows one to peg to bid plus 1/8.
  - The sign of the field is not defined in 4.1; it becomes explicit in 4.2
- Significant overhaul of Allocation messages, and Settlement Instructions message added

FIX 4.1 to FIX 4.2 Migration

- Quantity fields change from int to float to support instruments not trading in integer units
  - Not necessarily needed to implement if you only trade instruments in integer shares.
- Optional millisecond accuracy in time fields.
- No more maximum sequence number of 999999. EndSeqNo=0 in a Resend Request represents infinity.

FIX 4.1 to FIX 4.2 Migration

- Removed limit of 9999 on BodyLength, and added MaxMessageSize to Logon to negotiate maximums
- Sequence Reset – Reset must always be processed, even if the Sequence Reset message’s sequence number is less than expected.
  - This should apply to previous FIX versions as well.
- Added Business Message Reject message
- IOIOthSvc field removed as it is no longer used

FIX 4.1 to FIX 4.2 Migration

- Don’t Know Trade (DK) message – OrderID and ExecID are now required
- TransactTime is required on New Order Single & List, Order Cancel Request & Cancel / Replace Request, List Execute, List Cancel Request
  - Used to flag stale orders
- ExecRestatementReason required for “unsolicited” cancels of orders
- Meaning of the sign of PegDifference is now defined

FIX 4.1 to FIX 4.2 Migration

- CxlRejResponseTo is required for Order Cancel Reject. Specifies whether a Cancel or Cancel/Replace is being rejected
- OrdStatus / ExecType Pending Cancel/Replace changed into two types: Pending Cancel and Pending Replace.

FIX 4.2 New Features

- XmlDataLen and XmlData tags for including FIXML and other XML protocols (FpML)
- LastMsgSeqNumProcessed added to standard header to determine if your trading partner is getting backlogged
- Added OnBehalfOfSendingTime to standard header for use when a hub relays messages.
- RefTagID, RefMsgType, SessionRejectReason added to Reject.

FIX 4.2 New Features

- Added NoMsgTypes, RefMsgType, MsgDirection to Logon to help auto-negotiate session use.
  - WARNING! The spec is WRONG in stating the tag to use for this is MsgType, and will likely be corrected in the Errata.
- New Security Type: Convertible Bond
- New IDSource values: Exchange Symbol, Consolidated Tape Association (CTA) Symbol
- New Side enumeration: CrossShort

FIX 4.2 New Features

- “Encoded” text fields (Raw Data) for non-ASCII character sets (I.e. Japanese), and MessageEncoding in the standard header.
- Better support added for Forex trading
- CouponRate and ContractMultiplier added to standard symbol block for Fixed Income and Derivatives.
- Mass Quote messages added, primarily for making markets in options.

FIX 4.2 New Features

- IOI routing through NoRoutingIDs, RoutingType, RoutingID. Supports targeting and blocking IOIs.
- “Ready to Trade” added as an IOIQualifier
- SpreadToBenchmark and Benchmark added to IOI to support Fixed Income (High Yield / High Grade Corporate Bonds)
- Security Definition Request, Security Definition
  - Allows one to define complex multi-instrument trading strategies, i.e. options spreads, FLEX options, etc.

FIX 4.2 New Features

- Security Status Request, Security Status messages
  - Allows Exchanges to convey trading halts, etc.
- Trading Session Status Request, Trading Session Status messages
  - Allows Exchanges to convey information about global trading activity, like market halts, total volume traded in a session

FIX 4.2 New Features

- Market Data Request, Market Data – Snapshot / Full Refresh, Market Data – Incremental Refresh messages
  - Allows FIX to be used by ECNs, Exchanges, data vendors for their data feeds
  - Can be used for full depth of book or just top of book
  - Can be used on a query basis, a subscription basis, or as a data feed
  - Designed for bandwidth efficiency, to be used in high volume real-time applications

FIX 4.2 New Features

- New OrdType: Funari (Japanese)
- Account field added to Order Cancel, Order Cancel Reject, Order Status
- CashOrderQty added to Execution Report

FIX 4.2 New Features

- TradingSessionID added to messages containing LastMkt, ExDestination.
  - Trading Sessions defined by Exchanges / ECNs
  - A list of them can be specified on Orders, Cancel/Replaces (NoTradingSessionID, TradingSessionID repeating Group)
  - Can be used to facilitate extended hours trading, or multiple trading sessions on a given exchange (i.e. open outcry vs. electronic, round vs. odd-lot)

FIX 4.2 New Features

- NoContraBrokers, ContraBroker, ContraTrader, ContraTradeQty, ContraTradeTime repeating group added to Execution Report
- ComplianceID field added to Order, Cancel, Cancel / Replace, Execution Rpt
  - Used for audit IDs, i.e. OATS
- SolicitedFlag added to Order, Cancel / Replace, Execution Rpt messages
- Pre-trade allocation (AllocAccount, AllocShares)

FIX 4.2 New Features

- ExecType=Restated
  - Allows sell-side to modify order for several reasons
  - ExecRestatementReason:
    - GT Corporate Action
    - GT renewal / restatement (no corporate action)
    - Verbal change
    - Repricing of order
    - Broker option
    - Partial Decline of order
- EffectiveTime in New Order Single & List, Cancel/Replace messages allows for staging orders to go live at a later time

FIX 4.2 New Features

- More fields from order carried into Execution Rpt
  - HandlInst, MinQty, MaxFloor, MaxShow, OpenClose
- CxlRejReason of “Broker option” added
- GrossTradeAmt added to Execution Rpt for Japanese requirements
- New ExecInst: “Fixed Peg to Local Best Bid / Offer at time of order”
- New ExecInst: “Peg to VWAP”

FIX 4.2 New Features

- DiscretionInst, DiscretionOffset added to New Order Single & List, Cancel / Replace Request
  - Order has a displayed price (represented by Price) and a better discretionary price at which the order will trade
  - DiscretionInst defines benchmark for the discretionary price (i.e. displayed price, one of the pegging instructions) while DiscretionOffset determines the difference from the benchmark
  - DiscretionOffset is always added to the benchmark, i.e. when selling, if discretion is relative to the displayed price, then DiscretionOffset must be negative

FIX 4.2 New Features

- Added Program/Basket/List Trading functionality
  - Bid Request, Bid Response, List Strike Price messages added
  - WaveNo field removed
  - Accepted for Bidding added to OrdStatus
- Good Till (GT) functionality
  - Added DayOrderQty, DayCumQty, DayAvgPx, GTBookingInst to Execution Report
  - ExpireDate added as alternative to ExpireTime

FIX 4.2 New Features

- ExecType = Stopped defined
  - LastShares, LastPx defines guaranteed price / volume
- CxlRejReason of “Order already in cancel/replace status” added
- OrdRejReason of “Stale order” and “Duplicate of verbally communicated order” added
- ClearingFirm, ClearingAccount added to Order, Execution Rpt, Cancel/Replace messages.
  - ClearingFirm can be used in US options trade for CMTA number.

FIX 4.2 New Features

- Added support for Japanese “executed price” allocations as opposed to “average price”
- ConsumptionTax added to Allocation to support Japanese requirement
- SettlInstRefID added to Settlement Instructions message to support “Cancel” and “Replace” SettlInstTransType values

Recommendations on Which Versions to Support

- Check with your trading partners, competitors
- If you need certain features (derivatives, GT order handling, FIXML, etc.) you may not have much choice.
- Assess the leverage that you and your trading partners have in dictating version
- Ryan’s thoughts: 3.0 is mostly dead, for equities 4.0 and 4.1 are good candidates to implement due to their wide support, be wary of implementing only 4.2 as support isn’t quite there yet

XML Basics

- XML is very much like HTML in syntax, except:
  - XML is case-sensitive. `<Order>` != `<ORDER>`
  - Attributes must be surrounded by quotes. `<img width=50`...`>` won't work anymore; you must say `<img width="50"`...`/>`
  - All elements must be closed. I.e. no `<HR>` is allowed in the middle of a document. Empty elements can be denoted like this: `<Rule80A Value="A"></Rule80A>` or like this: `<Rule80A Value="A" />`

XML Validation

- One can optionally have a DTD (Document Type Definition) to define a document’s structure
- The DTD allows a validating parser to guarantee that all required elements and attributes are present
- XML validation is order sensitive. Elements MUST appear in the same order as they are specified in the DTD, otherwise the document cannot be validated against the DTD
- Additional fields MUST NOT BE ADDED without first defining them in the DTD or XML message

APIs for Parsing XML

- DOM – Document Object Model
  - Reads a whole document into a tree structure
  - Your application traverses the nodes of the tree
- SAX – Simple API for XML
  - Event-driven parser. The parser makes calls into your application while the XML file is being parsed

Beginning a FIXML Order

- First, start with the XML headers
- XML version is defined, as well as document encoding
- DOCTYPE defines which DTD to use (fixmlmain.dtd)
- Note that this example uses the latest FIX 4.2 DTD (version 1.0.0, 9 Feb 2001)

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE FIXML SYSTEM "fixmlmain.dtd">
```

Adding the root FIXML element

- Root element name is <FIXML>
- DTD says:

```
<!ELEMENT FIXML (FIXMLMessage+)>

<!ATTLIST FIXML  DTDVersion NMTOKEN  #FIXED '1.0.0'
                 FIXVersion NMTOKEN  #FIXED '4.2' >
```

- The FIXML element contains one or more FIXMLMessage elements (note the +)
- #FIXED attributes get added into the document by the parser

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE FIXML SYSTEM "fixmlmain.dtd">

<FIXML>
</FIXML>
```

Adding the FIXMLMessage element

- DTD says:

```
<!ELEMENT FIXMLMessage (Header ,
    ApplicationMessage)>
```

- The FIXMLMessage element contains exactly one Header element and one ApplicationMessage element

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE FIXML SYSTEM "fixmlmain.dtd">

<FIXML>
  <FIXMLMessage>
    <Header>
    </Header>
    <ApplicationMessage>
    </ApplicationMessage>
  </FIXMLMessage>
</FIXML>
```

Adding the Header element

- DTD says:

```
<!ENTITY % HeaderCustom "">

<!ENTITY % HeaderContent "Sender , OnBehalfOf?
    , Target , DeliverTo? , SendingTime? ,
    PossDupFlag? , PossResend?">

<!ELEMENT Header (%HeaderContent;)>
```

- A ? after an attribute indicates that it is optional. I.e. Sender is required, OnBehalfOf is optional
- ENTITY allows for substitution

```
<Header>
  <Sender>
  </Sender>
  <Target>
  </Target>
  <SendingTime />
</Header>
```

Adding the Sender and Target elements

- DTD says:

```
<!ELEMENT Sender (CompID , SubID? ,
    LocationID?)>

<!ELEMENT Target (CompID , SubID? ,
    LocationID?)>
```

```
<Header>
  <Sender>
    <CompID></CompID>
    <SubID></SubID>
  </Sender>
  <Target>
    <CompID></CompID>
    <SubID></SubID>
  </Target>
  <SendingTime />
</Header>
```

Completing the Header

- DTD says:

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

<!ATTLIST CompID  FIXTag           CDATA
    #FIXED '49-56-115-128'
                    SenderFIXTag     CDATA
    #FIXED '49'
                    TargetFIXTag     CDATA
    #FIXED '56'
                    OnBehalfOfFIXTag CDATA
    #FIXED '115'
                    DeliverToFIXTag  CDATA
    #FIXED '128'
                    DataType         CDATA
    #FIXED 'String' >
```

- #PCDATA = Parsed Character Data, i.e. actual text
- DTD cannot validate format of #PCDATA

```
<Header>
  <Sender>
    <CompID>MEEF</CompID>
    <SubID>IAN</SubID>
  </Sender>
  <Target>
    <CompID>TAL</CompID>
    <SubID>ARCA</SubID>
  </Target>
  <SendingTime>20010226-15:00:00</SendingTime>
</Header>
```

Starting the ApplicationMessage

- DTD says:

```
<!ELEMENT ApplicationMessage (Advertisement |
    Indication | News | Email | QuoteReq |
    Quote | Order | NewOrderList |
    ExecutionReport | DK_Trade |
    OrderModificationRequest |
    OrderCancelRequest | OrderCancelReject |
    OrderStatusRequest | Allocation |
    AllocationACK | SettlementInstructions |
    ListStatus | ListExecute |
    ListCancelRequest | ListStatusRequest |
    MarketData | MarketDataInc | MarketDataReq
    | MarketDataReqRej | MassQuote | QuoteAck |
    QuoteCancel | QuoteStatusReq | SecurityDef
    | SecurityDefReq | SecurityStatus |
    SecurityStatusReq | TrdSesStatus |
    TrdSesStatusReq | BusinessReject | Custom)>
```

- The | indicates that this element can contain one of the following elements

```
<ApplicationMessage>
  <Order>
  </Order>
</ApplicationMessage>
```

Starting the Order

- DTD says:

```
<!ENTITY % OrderCustom "">

<!ENTITY % OrderContent "ClOrdID , ClientID? ,
    ExecBroker? , Account? , OrdAllocGroupList?
    , Settlement? , HandInst , ExecInstList? ,
    MinQty? , MaxFloor? , ExecDestination? ,
    TrdSesList? , ProcessCode? , Instrument ,
    PrevClosePx? , Side , LocateReq? ,
    TransactTime , OrderQuantity , OrderType ,
    Currency? , ComplianceID? , SolicitedFlag?
    , IOI_ID? , QuoteID? , OrderDuration? ,
    EffectiveTime? , Commission? , Rule80A? ,
    ForexReqOrder? , Text? , EncodedTextGroup?
    , MaxShow? , DiscretionInst? ,
    DiscretionOffset? , ClearingFirm? ,
    ClearingAcct?">

<!ELEMENT Order (%OrderContent;)>

<!ATTLIST Order  FIXTag   CDATA  #FIXED '35'
                   DataType CDATA  #FIXED
    'String'
                   Value    CDATA  #FIXED 'D' >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID />
    <HandInst />
    <Instrument />
    <Side />
    <TransactTime />
    <OrderQuantity />
    <OrderType />
    <Currency />
    <OrderDuration />
    <Rule80A />
  </Order>
</ApplicationMessage>
```

Adding #PCDATA Elements to the Order

- DTD says:

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

<!ATTLIST ClOrdID  FIXTag   CDATA
    #FIXED '11'
                     DataType CDATA
    #FIXED 'String' >

<!ELEMENT TransactTime (#PCDATA)>

<!ATTLIST TransactTime  FIXTag   CDATA
    #FIXED '60'
                          DataType
    CDATA  #FIXED 'UTCTimeStamp' >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst />
    <Instrument />
    <Side />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency />
    <OrderDuration />
    <Rule80A />
  </Order>
</ApplicationMessage>
```

Adding Elements with Attributes to the Order

- DTD says:

```
<!ELEMENT HandInst EMPTY>

<!ATTLIST HandInst  FIXTag   CDATA  #FIXED '21'
                      DataType CDATA  #FIXED
    'char'
                      Value     (1 | 2 | 3 )
    #REQUIRED
                      SDValue   (AutoExecPriv |
    AutoExecPub | Manual )  #IMPLIED >
```

- “Value” is a #REQUIRED attribute, and must be specified.
- “SDValue” is an #IMPLIED attribute, meaning it is optional

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument />
    <Side />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency />
    <OrderDuration />
    <Rule80A />
  </Order>
</ApplicationMessage>
```

Adding Elements with Attributes to the Order

- DTD says:

```
<!ELEMENT Side EMPTY>

<!ATTLIST Side  FIXTag   CDATA  #FIXED
    '54'
                  DataType CDATA
    #FIXED 'char'
                  Value     (1 | 2 | 3
    | 4 | 5 | 6 | 7 | 8 | 9 )
    #REQUIRED
                  SDValue   (Buy |
                             Sell |
                             BuyMin |
                             SellPlus |
                             SellSht |
                             SellShtEx
    |
                             Undisc |
                             Cross )
    #IMPLIED >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument />
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency />
    <OrderDuration />
    <Rule80A />
  </Order>
</ApplicationMessage>
```

Adding Elements with Attributes to the Order

- DTD says:

```
<!ENTITY % currCodes
    "AED|AFA|ALL|ANG|AOK|ARP|ATS|AUD|BBD|BDT|BE
    C|BEF|BEL|BGL|BHD|BIF|BMD|BOP|BRC|BSD|BUK|B
    WP|BZD|CAD|CHF|CLP|CNY|COP|CRC|CSK|CUP|CVE|
    CYP|DDM|DEM|DJF|DKK|DOP|DZD|ECS|EGP|ESA|ESB
    |ESP|ETB|FIM|FJD|FKP|FRF|GBP|GHC|GIP|GMD|GN
    S|GQE|GRD|GTQ|GWP|GYD|HKD|HNL|HTG|HUF|IDR|I
    EP|ILS|INR|IQD|IRR|ISK|ITL|JMD|JOD|JPY|KES|
    KHR|KMF|KPQ|KRW|KWD|KYD|LAK|LBP|LKR|LRD|LSM
    |LUF|LYD|MAD|MGF|MLF|MNT|MOP|MRO|MTP|MUR|MV
    R|MWK|MXP|MYR|MZM|NGN|NIC|NLG|NOK|NPR|NZD|O
    MR|PAB|PES|PGK|PHP|PKR|PLZ|PTE|PYG|QAR|ROL|
    RWF|SAR|SBD|SCR|SDP|SEK|SGD|SHP|SLL|SOS|SRG
    |STD|SUR|SVC|SYP|SZL|THB|TND|TOP|TPE|TRL|TT
    D|TWD|TZS|UGS|USD|USN|USS|UYP|VEB|VND|VUV|W
    ST|XAF|XAU|YDD|YER|YUD|ZAL|ZAR|ZMK|ZRZ|ZWD"
    >

<!ELEMENT Currency EMPTY>

<!ATTLIST Currency  FIXTag   CDATA  #FIXED '15'
                     DataType CDATA  #FIXED
    'Currency'
                     Value     (%currCodes; )
    #REQUIRED >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument />
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency Value="USD“ />
    <OrderDuration />
    <Rule80A />
  </Order>
</ApplicationMessage>
```

Adding Elements with Attributes to the Order

- DTD says:

```
<!ELEMENT Rule80A EMPTY>

<!ATTLIST Rule80A  FIXTag   CDATA  #FIXED '47'
                     DataType CDATA  #FIXED
    'char'
                     Value     (A |
                                B |
                                C |
[SNIP]
                                X |
                                Y |
                                Z )  #REQUIRED
                     SDValue   (AgencySingle |
                                ShtExTranA |
[SNIP]
ShtExTranNonMem )  #IMPLIED >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument />
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency Value="USD“ />
    <OrderDuration />
    <Rule80A Value="A" />
  </Order>
</ApplicationMessage>
```

Adding the Instrument to the Order

- DTD says:

```
<!ENTITY % instrdetails "SymbolSfx? ,
    SecurityID? , IDSource? ,
    SecurityType? , SecurityExch? ,
    Issuer? , EncodedIssuerGroup? ,
    SecurityDesc? ,
    EncodedSecDescGroup?">

<!ELEMENT Instrument (Symbol ,
    %instrdetails;)>

<!ELEMENT Symbol (#PCDATA)>

<!ATTLIST Symbol  FIXTag   CDATA
    #FIXED '55'
                    DataType CDATA
    #FIXED 'String' >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument>
      <Symbol>CSCO</Symbol>
      <SecurityType />
    </Instrument>
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency Value="USD“ />
    <OrderDuration />
    <Rule80A Value="A" />
  </Order>
</ApplicationMessage>
```

Adding the Instrument to the Order

- DTD says:

```
<!ELEMENT SecurityType (Equity |
    MutualFund | ForeignExchange |
    FixedIncome | Warrant | Option |
    Future)>

<!ELEMENT Equity EMPTY>

<!ATTLIST Equity  FIXTag   CDATA
    #FIXED '167'
                    DataType CDATA
    #FIXED 'String'
                    Value     (CS | PS
    )  #REQUIRED >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument>
      <Symbol>CSCO</Symbol>
      <SecurityType>
        <Equity Value="CS" />
      </SecurityType>
    </Instrument>
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity />
    <OrderType />
    <Currency Value="USD“ />
    <OrderDuration />
    <Rule80A Value="A" />
  </Order>
</ApplicationMessage>
```

Alternate Instrument Example - Options

- DTD says:

```
<!ELEMENT Option (PutCall , Cover? ,
     CustomerFirm? , OpenClose? , Maturity ,
     StrikePx , OptAttribute?)>

<!ATTLIST Option  FIXTag   CDATA  #FIXED '167'
                    DataType CDATA  #FIXED
     'String'
                    Value    CDATA  #FIXED
     'OPT' >

<!ELEMENT PutCall EMPTY>

<!ATTLIST PutCall  FIXTag   CDATA  #FIXED '201'
                     DataType CDATA  #FIXED
     'String'
                     Value    (0 | 1 )
     #REQUIRED
                     SDValue  (Put | Call )
     #IMPLIED >
```

```
<Instrument>
  <Symbol>CYQ</Symbol>
  <SecurityType>
    <Option>
      <PutCall Value=“1”>
      <Maturity />
      <StrikePx />
    </Option>
  </SecurityType>
</Instrument>
```

Alternate Instrument Example - Options

- DTD says:

```
<!ELEMENT Maturity (MonthYear , Day?)>

<!ELEMENT MonthYear (#PCDATA)>

<!ATTLIST MonthYear  FIXTag   CDATA
    #FIXED '200'
                       DataType CDATA
    #FIXED 'month-year' >

<!ELEMENT StrikePx (#PCDATA)>

<!ATTLIST StrikePx  FIXTag   CDATA
    #FIXED '202'
                      DataType CDATA
    #FIXED 'Price' >
```

```
<Instrument>
  <Symbol>CYQ</Symbol>
  <SecurityType>
    <Option>
      <PutCall Value="1">
      <Maturity>
        <MonthYear>200103</MonthYear>
      </Maturity>
      <StrikePx>25.00</StrikePx>
    </Option>
  </SecurityType>
</Instrument>
```

Adding the OrderQuantity to the Order

- DTD says:

```
<!ELEMENT OrderQuantity (OrderQty |
    CashOrderQty)>

<!ELEMENT OrderQty (#PCDATA)>

<!ATTLIST OrderQty  FIXTag   CDATA
    #FIXED '38'
                      DataType CDATA
    #FIXED 'Qty' >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument>
      <Symbol>CSCO</Symbol>
      <SecurityType>
        <Equity Value="CS" />
      </SecurityType>
    </Instrument>
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity>
      <OrderQty>5000</OrderQty>
    </OrderQuantity>
    <OrderType />
    <Currency Value="USD“ />
    <OrderDuration />
    <Rule80A Value="A" />
  </Order>
</ApplicationMessage>
```

Adding the OrderType to the Order

- DTD says:

```
<!ELEMENT OrderType (MarketOrder |
    LimitOrder | StopOrder |
    StopLimitOrder | PrevQuotedOrder |
    PrevIndicatedOrder | ForexSwapOrder
    | PeggedOrder | ForexOrder |
    FunariOrder)>

<!ELEMENT LimitOrder (Price)>

<!ELEMENT Price (#PCDATA)>

<!ATTLIST Price  FIXTag   CDATA  #FIXED
    '44'
                   DataType CDATA
    #FIXED 'Price' >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument>
      <Symbol>CSCO</Symbol>
      <SecurityType>
        <Equity Value="CS" />
      </SecurityType>
    </Instrument>
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity>
      <OrderQty>5000</OrderQty>
    </OrderQuantity>
    <OrderType>
      <LimitOrder>
        <Price>32.25</Price>
      </LimitOrder>
    </OrderType>
    <Currency Value="USD“ />
    <OrderDuration />
    <Rule80A Value="A" />
  </Order>
</ApplicationMessage>
```

Adding the OrderDuration to the Order

- DTD says:

```
<!ELEMENT OrderDuration (TimeInForce |
    GTD_TimeInForce)>

<!ELEMENT TimeInForce EMPTY>

<!ATTLIST TimeInForce  FIXTag   CDATA  #FIXED
    '59'
                         DataType CDATA  #FIXED
    'char'
                         Value     (0 | 1 | 2 |
    3 | 4 | 5 )  #REQUIRED
                         SDValue   (Day |

    GoodTillCancel |

    AtTheOpening |

    ImmediateOrCancel |
                                    FillOrKill
    |

    GoodTillCrossing )  #IMPLIED >
```

```
<ApplicationMessage>
  <Order>
    <ClOrdID>FOO 1234</ClOrdID>
    <HandInst Value=“2”/>
    <Instrument>
      <Symbol>CSCO</Symbol>
      <SecurityType>
        <Equity Value="CS" />
      </SecurityType>
    </Instrument>
    <Side Value="1“ />
    <TransactTime>20010226-15:00:00</TransactTime>
    <OrderQuantity>
      <OrderQty>5000</OrderQty>
    </OrderQuantity>
    <OrderType>
      <LimitOrder>
        <Price>32.25</Price>
      </LimitOrder>
    </OrderType>
    <Currency Value="USD“ />
    <OrderDuration>
      <TimeInForce Value="0" />
    </OrderDuration>
    <Rule80A Value="A" />
  </Order>
</ApplicationMessage>
```

The Final FIXML Message

- Demonstration: Display message in Internet Explorer 5.
- Make sure file (called order.xml) is in the same directory as DTD.
- Note that IE5 likely is not doing all validation checks a validating XML parser should do.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE FIXML SYSTEM "fixmlmain.dtd">

<FIXML>
  <FIXMLMessage>
    <Header>
      <Sender>
        <CompID>MEEF</CompID>
        <SubID>IAN</SubID>
      </Sender>
      <Target>
        <CompID>TAL</CompID>
        <SubID>ARCA</SubID>
      </Target>
      <SendingTime>20010226-15:00:00</SendingTime>
    </Header>
    <ApplicationMessage>
      <Order>
        <ClOrdID>FOO 1234</ClOrdID>
        <HandInst Value="2" />
        <Instrument>
          <Symbol>CSCO</Symbol>
          <SecurityType>
            <Equity Value="CS" />
          </SecurityType>
        </Instrument>
        <Side Value="1" />
        <TransactTime>20010226-15:00:00</TransactTime>
        <OrderQuantity>
          <OrderQty>5000</OrderQty>
        </OrderQuantity>
        <OrderType>
          <LimitOrder>
            <Price>32.25</Price>
          </LimitOrder>
        </OrderType>
        <Currency Value="USD" />
        <OrderDuration>
          <TimeInForce Value="0" />
        </OrderDuration>
        <Rule80A Value="A" />
      </Order>
    </ApplicationMessage>
  </FIXMLMessage>
</FIXML>
```

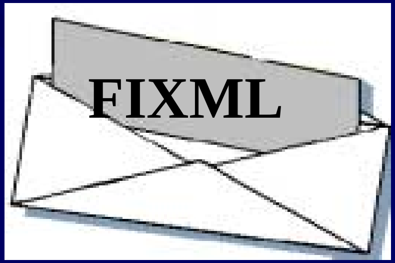
Embedding FIXML in FIX

- FIXML message placed in XmlData field of standard FIX message
- FIX Engine provides session layer, assigns sequence number, etc.
- Little to no change needed for either Business Application or FIX Engine (assuming FIX 4.1 or 4.2)

Embedding FIXML in FIX

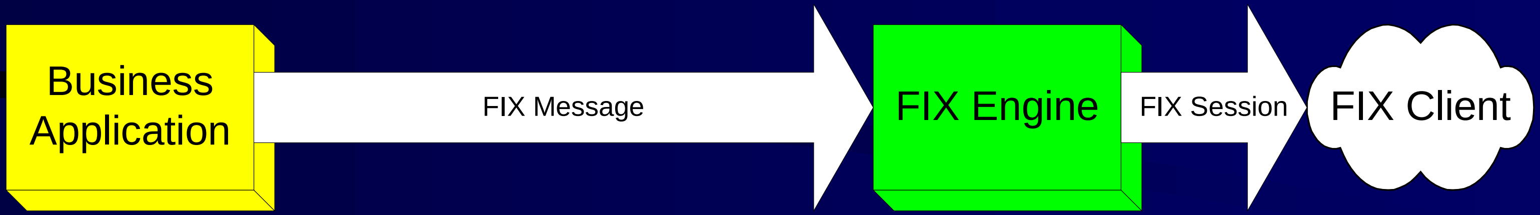
```
49=MEEF                                        SenderCompID
56=TAL                                         TargetCompID
212=1173                                       XmlDataLength
213=                                           XmlData
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE FIXML SYSTEM "fixmlmain.dtd">

<FIXML>
  <FIXMLMessage>
    <Header>
…
```



Architecting FIXML Systems

Normal FIX Message Flow



Business Application

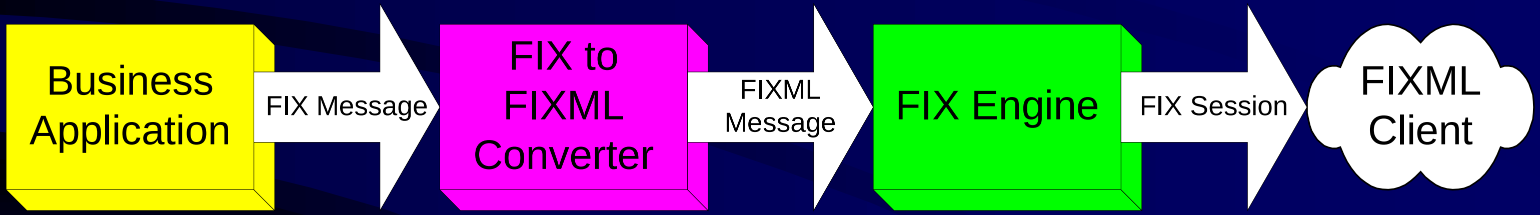
FIX Message

FIX Engine

FIX Session

FIX Client

Embedded FIXML Message Flow



Business Application

FIX Message

FIX to FIXML Converter

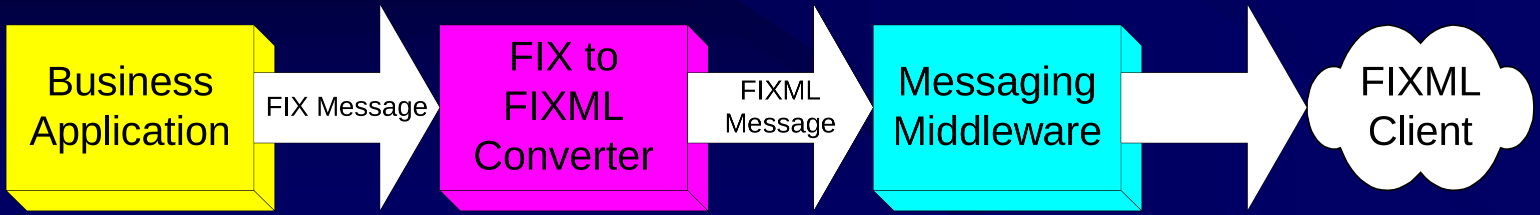
FIXML Message

FIX Engine

FIX Session

FIXML Client

FIXML Message Flow through Generic Messaging Middleware



Business Application

FIX Message

FIX to FIXML Converter

FIXML Message

Messaging Middleware

FIXML Client

Questions