Fall 2020

Don Mendelson, FIX Technical Architect
Jim Northey, GTC Co-chair,Americas and ISO TC68 Chair
Large scale investment in FIX infrastructure not likely

Few firms are providing resources to address operational inefficiencies

Therefore, any automation must be tactical and incremental

MUST address key FIX operational challenges
  - FIXatdl configuration
  - Certificate deployment issues

Any automation / process improvement must be applicable to non-FIX protocols
  - FIX now has multiple encodings and multiple session layers
  - Many venues have proprietary binary interfaces
However…

- Unmet needs
  - Operational inefficiencies
  - Operational risks
  - Takes to long to onboard trading partners
  - Translation / normalization effort
Where are the current pain points?

- Ambiguity and incompleteness of rules of engagement
- Must support multiple flavors of FIX
- Multiple internal messaging applications
- Multiple venue gateways
How to engage a FIX counterparty without Orchestra

“In short, the information we have is sparse and not directly actionable…”

Not to mention errror prone as well!
Having to configure device drivers?
Setting DIP Switches on interface cards?
Each interface needed an expert to get operational
Then came the magic of plug and play

The goal of Orchestra is to make financial messaging protocols more plug and play
What is Orchestra?
Two aspects of Orchestra

Define the service...

- Fields
- Messages
  - Scenarios
- Components and Groups
- Code sets and values
- Conditional Rules
- Workflows
- Actors and state transition

Deploy the service...

- Application Layer
  - Version
  - FIXatdl
- Encoding Layer
- Session Layer
  - counterparty identifiers
  - security
- Transport Layer
  - Connection details

Define the service...

- Deployment details
- Testing and validation
- Production integration

Deploy the service...

- Activation and monitoring
- Support and maintenance
- Continuous improvements
FIX Orchestra enables DevOps best practices
What is FIX Orchestra?

What it is
- A standard for exchanging machine-readable rules of engagement
- Describes your service offering
- How we deliver the standard
  - FIX Latest
  - FIX.4.2
  - FIX Session Layer
  - Recommended practices

What it is NOT
- A product
- A wire protocol
- A replacement for your existing infrastructure
- A FIX version
Two interchangeable representations of Orchestra

Orchestra XML
- Used widely in operations
- Standard in the industry

Tablature
- Simple
- Human Readable
- Human Writeable!
- GitHub Markdown

Both are machine readable!
Here is an example in both XML and Markdown:

```xml
<fix:message msgType="W" id="10116" name="MarketDataSnapshotFullRefRefresh">
  ...
</fix:message>

<fix:message msgType="D" id="10206" name="NewOrderSingle">
  <fix:structure>
    <fix:componentRef presence="required"/>
    <fix:fieldRef id="11" presence="required"/>
    <fix:fieldRef id="18"/>
    <fix:fieldRef id="111"/>
    <fix:fieldRef id="1300"/>
    <fix:fieldRef id="54" presence="required"/>
    <fix:fieldRef id="60" presence="required"/>
    <fix:fieldRef id="38" presence="required"/>
    <fix:fieldRef id="43" presence="required"/>
    <fix:fieldRef id="99" presence="required"/>
    <fix:when OrderType='4 or OrderType='5'>
      <fix:rule presence="required"/>
    </fix:when>
  </fix:structure>
</fix:message>

<fix:message msgType="E" id="10206" name="OrderCancelRequest">
  ...
</fix:message>

<fix:message msgType="G" id="10210" name="OrderCancelReplaceRequest">
  <fix:structure>
    <fix:componentRef presence="required"/>
    <fix:fieldRef id="41" presence="required"/>
  </fix:structure>
</fix:message>

---

**Message NewOrderSingle type D**

The New Order — Single message type is used by institutions wishing to electronically submit orders to a broker for execution.

<table>
<thead>
<tr>
<th>Name</th>
<th>Tag</th>
<th>Presence</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>StandardHeader</td>
<td>c</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>COrdID</td>
<td>11</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>Account</td>
<td>1</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>ExecInst</td>
<td>18</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>MaxFloor</td>
<td>111</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>55</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>MarketSegmentId</td>
<td>1300</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>Side</td>
<td>54</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>TransactTime</td>
<td>60</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>OrderQty</td>
<td>38</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>OrdType</td>
<td>40</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>44</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>StopPx</td>
<td>99</td>
<td>required if OrderType='4 or OrderType='5</td>
<td></td>
</tr>
<tr>
<td>TimeInforce</td>
<td>59</td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>ExpireDate</td>
<td>432</td>
<td>required if TimeInforce = 6</td>
<td></td>
</tr>
<tr>
<td>OrderCapacity</td>
<td>528</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>StandardTrailer</td>
<td>c</td>
<td>required</td>
<td></td>
</tr>
</tbody>
</table>

---

**Message OrderCancelRequest type F**
How to engage a FIX counterparty with Orchestra
How can you use Orchestra?
How can you benefit from Orchestra right now?

- Looking for a simpler way to create a richer and less ambiguous rules of engagement document?
- Need to document and understand your internal electronic trading infrastructure better?
- Just want to share message definitions and conditional fields?
- Want to extend to describe message responses, scenarios, and basic states?
- Want to fully model in detail the FIX service?

*We can do it! let’s start today!*
Realizing your own orchestra machine readable rules of engagement

- Log files
- A QuickFIX data dictionary
- A Unified Repository
- FIXLatest Orchestra Repository
- An existing RoE (spec) in PDF and MSWord

- log2orchestra
- quickfix2orchestra
- unified2orchestra
- orchid
- pandoc

Your rules of engagement in Tablature Orchestra format
Your Orchestra rules of engagement in Orchestra format

Any Markdown editor

Start Here
log2orchestra

FIX Log to Orchestra

Creates an Orchestra file from one or more FIX message logs (tag-value encoding)

Input

Reference Orchestra file

FIX message log files

Configuration file for scenarios (optional)

Output

Orchestra file to create (*.xml)

myorchestra.xml

Append only (removes no scenarios)

Create Orchestra file

Help

Version 1.0.0
© Copyright 2020, FIX Protocol Ltd.
A group of tooling to support Orchestra

- **log2orchestra**: Ability to derive an orchestra file from Atom/VS Code/Zettlr/GitHub Editors for tablature
- **md2orchestra**
  - **orchestra2md**
  - **md2interface**
  - **Interface2md**: Converts between tablature and orchestra
- **docgen**: Generates an HTML version of specification with workflow diagrams and state diagrams
- **FIXimate**: Ability to generate custom FIXimate
FIX Orchestra supports innovation

- Possible uses and tools
  - Generate and run conformance tests
  - Capture best practices as an Orchestra file instead of text
  - Regulate internal flows within a large organization as well as between counterparties
  - Orchestra is a contract for behavior – use it to generate an emulator for testing
  - Analyze FIX logs for conformance to specified behavior
  - Let’s go further…
    - Generate Execution Management, Order Management, Smart Order Routing, Order Matching behavior based upon exchange of state machine descriptions contained within FIX Orchestra files
FIX A New Level of Openness

- FIX develops technical standards in an open manner

- Technical Standards Process
  - Anyone can propose a technical standard
  - Anyone can work on a technical standard
  - Work is done by a technical working group
  - Available to everyone on GitHub
In conclusion…

- Orchestra is a standard to define an interface for service offerings
- You don’t need to modify the internals of your applications
- Orchestra can be used to define your service in one place within version control
- Simple scripts can be created to read Orchestra files and update configuration files for various services
- You can start to benefit from Orchestra immediately with minimal investment
How can you get involved?
Want to get more involved?

- FIX Orchestra Working Group
- MOST Monitoring, Onboarding, Software Testing Working Group
- Create your specification
- Test open source and member based tools and services