FIX Orchestra Technical Standard Proposal

Release Candidate 4

February 21, 2019

V0.3

Proposal Status: Public Review

For Global Technical Committee Governance Internal Use Only

Submission Date	February 21, 2019	Control Number	
Submission Status	Public Review	Ratified Date	
Primary Contact Person	Don Mendelson	Release Identifier	

r0.3

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE "FIX PROTOCOL") ARE PROVIDED "AS IS" AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

DRAFT OR NOT RATIFIED PROPOSALS (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FIX GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FIX WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”. THE FIX GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Table of Contents

Table of Figures

Document History

Revision	Date	Author	Revision Comments
RC4 v0.1	Jan. 14, 2019	Don Mendelson Silver Flash LLC	Initial draft
v0.2	Jan. 23, 2019	Don Mendelson Silver Flash LLC	Minor edits
v0.3	Feb. 21, 2019	Don Mendelson Silver Flash LLC	Minor edits

1 Introduction

FIX Orchestra was conceived as **machine readable rules of engagement** between counterparties. As such, it is a standard for exchange of metadata about the behavior of FIX applications. Orchestra is intended to cut time to onboard counterparties and improve accuracy of implementations.

Orchestra does not change FIX protocol itself in any way, nor does it obsolete existing FIX engines or tools.

1.1 Authors

Name	Affiliation	Contact	Role
Don Mendelson	Silver Flash LLC	donmendelson@silverflash.net	Working group co-chair
Jim Northey	GTC Co-chair, Americas and ISO TC68 Chair	jim.northey@fintechstandards.us	Working group co-chair

2 Requirements

2.1 Business Requirements

2.1.1 Rules of engagement

FIX Orchestra was designed to overcome often vague, humanly readable specifications and thus cut time for onboarding and ongoing maintenance.

The contents of Orchestra files are machine readable (that is, processed as data) may include:

- Message structure by each scenario, implemented as an extension of FIX Repository.
- Accepted values of enumerations by message scenario
- Workflow: when I send this message type under this condition, what can I expect back?
- How external states affect messages, e.g. market phases
- Express a condition such as for a conditionally required field using a Domain Specific Language (DSL)
- Document and exchange the Algorithmic Trading Definition Language (FIXatdl) files associated with a FIX service offering
- FIX session identification and transport configuration

Given a standard for information interchange, firms and vendors will be enabled to develop tools to automate configuration of FIX engines and applications, and generation of code, test cases, and documentation. The various aspects are not an all-or-nothing proposition, however. Users may implement only the features that they find most beneficial, and add features as needed.

2.1.2 Optimize for tools

The main theme of Release Candidate 4 is to refine the Orchestra XML schema to optimize tools rather than adding new business features.

2.2 Technical Requirements

This section discusses enhancements to the Orchestra standard since Release Candidate 3.

2.2.1 Orchestra XML schema

2.2.1.1 Field and code set scenarios

Like messages and components, fields and code sets now support scenarios in RC4. The main benefit is that a code set can be constrained to different sets of codes in different use cases. For example, PartyRole can have enriched values in outbound messages compared to the requirements for inbound messages. Additionally, field and code sets scenarios can have different documentation than their base implementations.

2.2.1.2 Separation of components and groups

Previously, component blocks and repeating groups were stored together under one parent element. To optimize operations, they were separated.

2.2.1.3 Object ID dropped

Previously, there was a proposal to generate a universally unique ID for every element in Orchestra. A satisfactory solution was not found, and the idea was deemed unnecessary.

2.2.1.4 Internal IDs normalized

In FIX Repository, every message element reference, such as fieldRef, carried both a humanly readable name and a numeric ID. There was always the chance that the two items would be inconsistent with each other. In Orchestra RC4, only the numeric ID is carried in references. The object that is referenced, such as a field, still carries its name.

2.2.2 Demonstration projects

This project has been added since RC3:
Orchestra2sbe--generates a Simple Binary Encoding (SBE) message schema from an Orchestra file.

All other demonstration projects have been updated to conform to the RC4 schema.

3 Issues and Discussion Points

3.1 Questions to be decided beyond RC4

3.1.1 Provenance and pedigree format

The Orchestra file format has a provision for metadata about the file that tells who issued it, when, and in what format. Categorically, this is known as provenance. This metadata is based on the standardized Dublin Core model.

Also, FIX Repository contains the history of each message element—when it was added, updated, and possibly deprecated. Element history is called pedigree. Orchestra has so far kept the pedigree format of Repository 2010 Edition. However, to make it easier to interact with other protocols, it would be desirable to use a standardized model. The W3C PROV standard has been proposed for this. It is based on a comprehensive model of provenance and pedigree. PROV has multiple representations that follow the same semantics, including an XML encoding.

3.1.2 Reference implementations

FIX Orchestra is intended to be a standard for information exchange, not a software product. However, the working group has sponsored reference implementations of some aspects of the standard. This will help firms and vendors adopt the standard while adding their own special value. Up until now, all demonstration projects were made available to the public in GitHub so any interested software developer may take advantage, whether they were a member of FIX Trading Community or not.

However, it is now proposed that certain advanced applications should be considered premium and only be made available to members. Complex applications require more development time and maintenance, and therefore should be made available to firms that contribute. At the same time, those applications will serve as a value-added to membership, providing an incentive for firms to join.

Premium applications currently under development are:

Orchestra Editor

Log2orchestra—creates an Orchestra file from FIX message logs

4 References

Reference	Version	Relevance	Normative
None			

5 Relevant and Related Standards

Related Standard	Version	Reference location	Relationship	Normative
Dublin Core XML Schemas	2008-02-11	http://dublincore.org/schemas/xmls/	Dependency	Yes
XML Schema for FIX	2016		Technical guide	Yes
XML Schema	2012	https://www.w3.org/TR/xmlschema11-1/	Dependency	Yes
Namespaces	2006	https://www.w3.org/TR/xml-names11/	Dependency	Yes

6 Intellectual Property Disclosure

Related Intellection Property	Type of IP (copyright, patent)	IP Owner	Relationship to proposed standard

7 Definitions

Term	Definition
Pedigree	The recorded history of an artifact
Provenance	A record of ownership of an artifact

8 FIX Orchestra

8.1 Project milestones

Since Orchestra has many facets, features will be delivered in several release candidates rather than attempting a big-bang approach.

8.1.1 Release candidate 4 deliverables

These artifacts will be delivered as Release Candidate 4:

- The technical specification is a separate document "FIX Orchestra Technical Specification". The document will be displayed in the Tech/Specs section of FIX Trading Community website as well as in GitHub project FIXTradingCommunity/fix-orchestra-spec.

These resources have been published in GitHub project FIXTradingCommunity/fix-orchestra:

- XML schema (XSD) for Repository 2016 Edition and Orchestra plus documentation of the schema
- XML schema (XSD) for interfaces and session configuration plus documentation of the schema
- Grammar and implementation of the Score DSL
- A script to populate Repository 2016 from 2010 Edition (message structure only)
- Example Orchestra files
- Demonstration projects:
    - Documentation generator
    - QuickFIX data dictionary and code generators
    - XML diff/merge utility to manage Orchestra file changes
    - QuickFIX session configuration
    - Test generator

8.1.2 Roadmap

The next task of the Orchestra working group is develop a roadmap for release candidate 4 and beyond. In addition to standard development, a plan will be created to migrate the building of Extension Packs from Repository 2010 Edition to Orchestra. This plan will need to account for the tools that consume Repository, including FIXimate, FIXML schema generation, and so forth.

Appendix A - Usage Examples

These example Orchestra files are posted in GitHub.

Example order entry file developed by MilleniumIT

fix-orchestra/repository2016/src/test/resources/examples/

Sample interface file

https://github.com/FIXTradingCommunity/fix-orchestra/blob/master/interfaces2016/src/test/resources/SampleInterfaces.xml

A non-FIX exchange API interpreted in Orchestra

https://github.com/FIXTradingCommunity/orchestrations/tree/master/NYSE%20Pillar

Appendix B – Compliance Strategy

The first level of compliance will be provided by existing XML tools that verify conformity of a file to its schema. A test is provided to validate that isolated DSL expressions conform to the grammar. However, a comprehensive compliance test has not yet been developed.