FIX PROTOCOL
INDUSTRY-DRIVEN MESSAGING STANDARD℠

Simple Binary Encoding
Release Candidate 4
Technical Proposal

February 2016

v0.1

Proposal Status: Public Comment

For Global Technical Committee Governance Internal Use Only

Submission Date	Dec 18, 2013	Control Number	
Submission Status	Submitted	Ratified Date	
Primary Contact Person	Don Mendelson	Release Identifier	

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 FPL 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 FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”. THE FPL 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

Document History

Revision	Date	Author	Revision Comments
v0.1	Jan. 13, 2016	Don Mendelson Silver Flash LLC	Initial draft

1 Introduction

The High Performance Working Group was formed with the goal of improving the fit-for-purposefulness of FIX for high performance.

Recent improvements in the speed of hardware, software, and network connections (such as in co-location solutions) are putting pressure on the FIX protocol and highlighting some inefficiencies of the current version of the protocol (e.g., excessive echoing of input values, inefficient encoding). New financial applications such as high-frequency trading and market data feeds pose new performance requirements. In recent years, several financial organizations have avoided the performance limitations of FIX and introduced new proprietary protocols that are optimized for speed. These proprietary interfaces have been offered, sometimes along with a FIX interface, to support high-speed transactions and/or data feeds.

The current performance limitations of FIX can be removed by making changes and additions at multiple levels of the protocol. At the *application* level, there is a need to define less-verbose versions of some FIX messages and to streamline the message flow. At the *presentation* level, there is a need to provide new encodings that are faster and more compact than the traditional Tag=Value encoding of FIX. At the *session* level, there is a need to specify a new lightweight session protocol with basic recovery options. The High Performance Working Group is drafting a set of specifications and guideline documents to address all these aspects.

This proposal entails the use of an FPL designed *Simple Binary Encoding* to produce fast and compact encodings of FIX messages.

Simple Binary Encoding provides different characteristics than other binary encodings. It is optimized for low latency. This new FPL binary encoding complements the existing only binary encoding developed in 2005 (FAST) with a focus on reducing bandwidth utilization for market data. In addition, the encoding is also defined and controlled within FPL only in contrast to the binary encodings proposals to encode FIX with Google Protocol Buffers and ASN.1

1.1 Authors

Name	Affiliation	Contact	Role
Don Mendelson	CME Group	Donmendelson@gmail.com	SBE lead

2 Requirements

2.1 Business Requirements

2.1.1 Versioning

2.1.1.1 Schema extension clarifications

- **Issue 2: Schema extension is vague in terms of what compatibility means**
- **Issue 3: Extensibility of the Template ID**

Clarifications were made to section 5 of the specification regarding message schema extensions. Users wanted to know precisely when a change to a message template is back-compatible with a previous version or when a template ID is required due to a breaking change. A point was added to say that SBE message schema extension only handles the mechanics of versioning; it does not relieve developers of carefully planning migrations of message changes and documenting semantic differences.

Also, a sentence was added to the spec to say that valid values may be added to an enumeration in later versions of a schema. New values may be marked with a “sinceVersion” attribute, just like new fields.

To aid in understanding, a non-normative subsection was added to give a strategy for compatibility when a decoder’s version is different than the encoder’s schema version.

2.2 Technical Requirements

2.2.1 Message structure: repeating groups

- **Issue 6: Limiting maximum occurrences of repeating group**
- **Issue 8: blockLength for repeating groups of variable length**

A section was added to explicitly show how to limit the minimum or maximum number of entries in a repeating group.
A clarification was made to “blockLength” explanation to say that it only includes the length of fixed-length fields, not variable length fields or nested repeating groups.

2.2.2 Message structure: composite encodings

- **Issue 11: Offsets within composite types**

A change was made to the XML schema (XSD) to allow the “offset” attribute to be set on elements within a composite encoding of affect byte alignment. This attribute already existed for fields within a message.

2.2.3 Message schema: reusing encodings

- **Issue 12:** Composites reusing other types

A change was made to the XML schema (XSD) to allow a composite encoding to refer to an existing simple encoding as its element without having to redefine the contained element. (In RC3 and previous versions, all types within a composite encoding were defined in-line.) For example, an enumeration could be defined once, and that enumeration could be re-used in any number of composite types or fields.

3 Issues and Discussion Points

3.1 Reusable message blocks

A proposal was made to support definition of blocks of fields or entries of repeating groups that could be re-used in multiple message templates. FIX Repository has such a feature. For example, Instrument block is re-used in many FIX messages. However, there was no consensus on how this feature should work in SBE so it was not accepted for RC4. It may be re-introduced in a later release candidate.

3.2 Purpose of fields and message blocks vis-à-vis composite encodings

SBE has a concept of a concept of a composite encoding in which two or more primitive data types are combined to form a unit. For example, a decimal number is encoded in SBE as a pair of integers, representing mantissa and exponent. This encoding is used for FIX fields of data type Price and the like.

When the proposal was made for reusable message blocks, as described above, a counter argument was made that it could instead use the existing concept of composite encoding. From a strictly presentation layer view, there is no need for both composite encoding and reusable block concepts. They could be one and the same.

On the other hand, FIX historically does have concepts of fields and blocks, somewhat independently of their wire formats. FIX's greatest value as a protocol is its well-known semantics at the level of fields, blocks and messages. Although blocks and composite encodings may seem to be redundant concepts from a wire format perspective, blocks and fields map to application layer concepts while wire formats do not.

It would be helpful to hear other opinions on this matter to help resolve it. It is central to the effort to issue alternative encodings for FIX for different technical purposes while mapping to common semantics.

4 References

Reference	Version	Relevance	Normative
FIX Simple Binary Encoding RC3 Specifications	Final	Full specification as approved for RC3 in December 2014 by the FPL GTC.	Yes
GitHub project FIXTradingCommunity/fix-simple-binary-encoding		Final specifications as well as working drafts and issue tracking.	

5 Relevant and Related Standards

Related Standard	Version	Reference location	Relationship	Normative
None				

6 Intellectual Property Disclosure

Related Intellection Property	Type of IP (copyright, patent)	IP Owner	Relationship to proposed standard
None			

7 Definitions

Term	Definition

8 Simple Binary Encoding

8.1 Specifications

Full specifications for the Simple Binary Encoding are available in separate document (*FIX Simple Binary Encoding – Release Candidate 4*). The standard defines wire format and message schema declaration. The document is a snapshot of drafts now being developed in GitHub project FIXTradingCommunity/fix-simple-binary-encoding.

8.2 Schema

An XML schema (XSD) is provided to standardize XML message schemas. The XSD file is publically available in GitHub project in GitHub project FIXTradingCommunity/fix-simple-binary-encoding.

For this release, the name of the XSD file is SimpleBinary1-0.xsd. Internally, the XSD is identified as version="1.0RC4".

Appendix A - Usage Examples

Examples are provided in the specification document.

Appendix B – Compliance Strategy

Message schemas should be validated against the provided XML schema (XSD).