Simple Binary Encoding

Release Candidate 3

Technical Proposal

December 2014

v0.2

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	

r0.2

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	May 14, 2014	Don Mendelson CME Group	Initial draft
v0.2	Dec. 4, 2014	Don Mendelson CME Group	Brought technical proposal up to date to be consistent with specification.

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
Fred Malabre	CME Group	Fred.malabre@cmegroup.com	HPWG co-chair SBE co-lead
Don Mendelson	CME Group	Don.mendelson@cmegroup.com	SBE co-lead

2 Requirements

2.1 Business Requirements

2.1.1 Semantic

Semantics of FIX messages and fields refers to the business meaning of encoded elements at the application layer. Initially, Simple Binary Encoding message schema provides a way to associate encodings the data types recognized by the tag=value encoding and registered in the FIX Repository. SBE RC2 makes some distinctions in encoding not made by traditional types. For example, tag=value encoding supports String data type, while SBE makes a distinction between fixed-length character arrays and variable-length strings.

RC3 should further clarify the relation between FIX tag=value data types and binary data types. While supporting all appropriate FIX data types, SBE should allow exchange of binary elements that have no suitable correspondence to a FIX data type. For example, a UUID can be declared at presentation layer as a 16 octet array, but it has no suitable FIX semantic type; the binary encoding is not a String. However, it does have business meaning as a unique identifier.

2.1.2 Versioning

One of the challenges raised with a binary encoding is the static aspect based on a given template as opposed to the dynamic aspect of the FIX tag=value encoding where tags can be added and removed on the fly.

SBE RC2 added the concept of versioning to templates that allow decoding of messages encoded with an older version template. Specific encoding and decoding rules need to be performed to support this optional feature. RC2 provided for appending new fields to a message or repeating group while maintaining back-compatibility with the previous version. However, the RC2 mechanism did not support adding new repeating groups or variable-length data to a message.

RC3 clarifies the extension mechanism to support addition of new repeating groups and variable-length data, but only within the restriction that the new group or field is at the end of a message.

2.2 Technical Requirements

2.2.1 Strings and raw data encoding

SBE RC2 enhanced the capability of specifying a character encoding, such as US-ASCII or UTF-8, for both fixed-length and variable-length String fields. In a message schema, fixed-length is specified as an ordinary <field> element while variable-length element that is either encoded text or raw data is specified as <data>.

Two clarifications are needed in RC3. First, there should be an explicit way to specify binary raw data which has no character encoding. Second, it should be possible to support binary data of fixed length.

2.2.2 Variable-length data in repeating groups

SBE RC2 only supported variable-length data at the end of a message. RC3 adds the capability to encode variable-length data within a repeating group entry as well. It is need to support existing FIX message definitions as well as new ones to be defined.

2.2.3 XML schema improvements

The XML schema (XSD) for RC3 is back compatible with the RC2 version but is somewhat less verbose. It supports variable-length data in repeating groups in an elegant way by treating repeating groups as a recursive application of the same XML type used for messages.

Also, composite encoding types are now defined in a more flexible way. Composites may now contain enumerated types or bitsets as well as simple data types. This feature was employed to encode a timestamp with precision on the wire. Its composite type contains the number of time ticks plus a time unit declared as an enumeration.

3 Issues and Discussion Points

3.1 Semantic types

RC3 does not provide a means to declare *new* semantic types. If such a mechanism is developed in the future, it should be supported across all FIX encodings.

3.2 Versioning

The working group considered a proposal for a more general purpose solution for versioning new repeating groups. However, in the judgment of the working group, if a message is changed radically, it is more likely to be published as a new template rather than as an updated version of an old one. A more productive solution would be to provide a protocol to convey message schemas in-band. That is outside the scope of SBE, and should be developed to work for all FIX encodings.

4 References

Reference	Version	Relevance	Normative
FIX Simple Binary Encoding RC2 Specifications	Final	Full specification as approved for RC1 in March 2014 by the FPL GTC.	
Simple Binary Encoding – Release Candidate 3	RC3 v0.x	Full specifications based on RC1 with the addition of the technical solutions from this document.	
SimpleBinary-ReleaseCandidate3	Draft	Full XSD supporting the specifications .	

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 3*). The standard defines wire format and message schema declaration.

8.2 Schema

An XML schema (XSD) is provided to standardize XML message schemas. The XSD file should published to users with the specification document. For this release, the name of the XSD file is SimpleBinary1-0.xsd. Internally, the XSD is identified as version="1.0RC3".

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).