High Performance Working Group

Encoding FIX Using Google Protocol Buffers

April 17, 2013

Revision v0.4

Proposal Status: Draft

For Global Technical Committee Governance Internal Use Only

Submission Date		Control Number	
Submission Status		Ratified Date	
Primary Contact Person		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	2013-04-09	Greg Malatestinic, Jordan & Jordan Sara Rosen, EBS Alessandro Triglia, OSS Nokalva	Initial draft.
v0.2	2013-04-11	Greg Malatestinic, Jordan & Jordan Sara Rosen, EBS	Change to Business workflow to include Users Guide in the set of documents to be admitted for approval.
v0.3	2013-04-16	Greg Malatestinic, Jordan & Jordan Sara Rosen, EBS Alessandro Triglia, OSS Nokalva	Changes based on feedback from HPWG review.
v0.4	2013-04-17	Greg Malatestinic, Jordan & Jordan Sara Rosen, EBS	Draft for GTC review.

1 Introduction

1.1 Background

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.

1.2 Google Protocol Buffers Encodings

This proposal entails the use of *Google Protocol Buffers* (GPB) technology to produce fast and compact encodings of FIX messages.

The use of GPB is one of the three major approaches to the efficient encoding of FIX that have been developed and are being proposed by the High Performance Working Group—the others being the use of *Abstract Syntax Notation 1* (ASN.1) and the use of a new, FIX-specific binary encoding called *Simple Binary Encoding* (SBE). Each of these three encoding mechanisms supports a full mapping of the existing FIX specification to efficient binary messages. However, each encoding mechanism emphasizes different quality characteristics in terms of degree of message compression, performance overhead, flexibility, ease of adoption, and standards conformance. These new encodings are intended to join the ranks of existing FIX encodings—ASCII Tag=Value, FIXML, and FAST, to provide an array of encoding options to support the varying needs of different organizations, while preserving the semantic richness of the FIX interface.

GPB is a data interchange format developed by Google which provides a language-neutral, platform-neutral, extensible mechanism for serializing structured data. Originally developed by Google for their own internal use, it was made available to the Open Source community in 2008. The technical specification draft, “Encoding FIX using Google Protocol Buffers”, attached to this proposal, contains provisions for the mapping of the content of the FIX Unified Repository to GPB.

The mapping to GPB defined in the proposed technical specification can be used for any FIX message (as defined in the FIX Unified Repository), and generates a set of GPB data structures. GPB data structures are defined as “messages” in a “.proto” template file. The template provides a machine-readable interface definition which is input to a language-specific code generator (protoc) to produce message encoders and decoders. These expose getters and setters for each field as well as methods to serialize/parse the whole structure to/from raw bytes.

The source of the mapping to GPB can be either the original FIX Unified Repository or any XML document that resembles the FIX Unified Repository. For example, it can be a subset of the Unified Repository, a custom variant of the Unified Repository, or the result of a formal transformation of the Unified Repository, such as the application of one or more scenarios to some messages in the Unified Repository. The FIX Basic Repository cannot be used as the source of the mapping for the following reason. Although the two forms of the FIX Repository have equivalent

content, they have a different structure (described by different XML schemas), and the proposed mapping specification relies on many aspects of the structure of the Unified Repository.

An overview of Protocol Buffers can be found on the Google Developers website at https://developers.google.com/protocol-buffers/docs/overview.

An overview of the Protocol Buffers API Reference can be found on the Google Developers website at https://developers.google.com/protocol-buffers/docs/reference/overview.

2 Business Workflow

It is proposed that the attached technical specification draft ("Encoding FIX Using Google Protocol Buffers v0.9.docx") and its associated User Guide ("Encoding FIX Using Google Protocol Buffers - User Guide") be admitted into the FPL standardization process and eventually be made available to FIX implementers and users.

The encoding of FIX over GPB follows a standard developed and maintained by Google which has been made available as an Open-Source project. There is therefore no need to create a new encoding specification for GPB encoding of FIX. However, the richness of GPB supports multiple solutions to mapping the existent FIX data types and messages to GPB. In the interest of standardization, this proposal specifies a normative encoding of FIX to protocol buffers. The proposed technical specification contains provisions for a mapping of the content of the FIX Unified Repository (or any XML document that resembles the FIX Unified Repository) to a GPB ".proto" template file. Additionally, the User Guide, "Encoding FIX using Google Protocol Buffers – User Guide", provides a set of best practice guidelines for maximizing the efficiency of encoding FIX via protocol buffers.

3 Issues and Discussion Points

3.1 Decimal Prices

The decision jointly reached by the SBE, GPB, and ASN.1 encoding subgroups, was to represent decimal prices by a pair of integers – a mantissa and an exponent, where the exponent determines placement of the decimal point. The motivation behind this was to eliminate the need to transmit the exponent in the binary payload in situations where both sending and receiving parties have negotiated the value through rules of engagement.

The GPB mapping provides mechanisms to support three encoding styles for the FIX Price datatype: one which includes only the mantissa; a second which includes an optional exponent; and a third which encodes fractional prices as IEEE floating point numbers.

3.2 Optional vs. Required Fields

All FIX fields in the GPB message definitions are to be marked as optional in accordance with Google's recommendation. (See https://developers.google.com/protocol-buffers/docs/proto.) It is expected that the enforcement of required fields be done in the application layer.

3.3 Message Type Framing

In order to decode a GPB message it is necessary to invoke a parsing method of the class to which the message will be deserialized. The GPB message type must therefore be known before invoking the decoder.

For FIX messages, two alternative solutions were considered. The first was to define all possible FIX messages as optional fields of a single FIX "super-message" in a single proto file. This approach was rejected by the Google

Protocol Buffers Encoding Subgroup because of the deployment challenges of a single FIX message. Separate proto files allow multiple teams to work independently, and better support message evolution. The second, and recommended approach, is that the FIX message type is to be specified outside the binary payload via the session layer.

3.4 Direct Access

Direct access to selective fields for high performance access is supported under the conditions listed in the User Guide. Additionally, fields must be encoded sequentially by field number. The C++, Python and Java implementations provided by Google support this behavior. We recommend that users of non-Google provided implementations, who depend on sequential field ordering, verify that their implementations honor this ordering policy.

3.5 Restricting Enumerated Values

The FIX to GPB mapping specification follows the convention of the FIX Repository in listing all possible enumerated values in the proto specification. Restricting enumerated values per message can be supported in Google Protocol Buffers by nesting the enumeration definition with its restricted set of values inside each message definition. GPB supports declarations of nested enumeration types of the same name provided they have different scopes. These are treated as distinct enumeration definitions. This allows enumerated values to be restricted based on the message in which they appear. E.g. MsgType can only have a single value for NewOrderSingle.

The FIX Repository does not support automated creation of proto files in this manner today.

3.6 Production of the Tag=Value encoding from a GPB template

Although support for additional encodings. such as the Tag=Value encoding used by FIX, is not provided by Google, mappings to various human-readable formats, such as XML, HTML, and JSON, have been developed by the open source community. (See http://code.google.com/p/protobuf/wiki/ThirdPartyAddOns for a list.)

Similarly, procedures can be developed that encode instances of the GPB classes generated by the protocol buffer compiler into FIXML or standard FIX wire-formatted messages and, conversely, decode FIXML or FIX wire-formatted data and populate the fields of an instantiated GPB class.

4 Proposed Message Flow

This proposal does not include any new message flow. A FIX message encoded in GPBcan carry the same information as a FIX message encoded in the Tag=Value encoding.

5 FIX Message Tables

This section does not apply to this proposal.

6 FIX Component Blocks

This section does not apply to this proposal.

7 Category Changes

This section does not apply to this proposal.

Appendix A - Data Dictionary

This section does not apply to this proposal.

Appendix B - Glossary Entries

This section does not apply to this proposal.

Appendix C - Abbreviations

This section does not apply to this proposal.

Appendix D - Usage Examples

This section does not apply to this proposal.

