logo

[Global Technical Committee]

FIXML Technical Standard Version 1.2

2015-7-15

Rev.0

Proposal Status: Draft

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).

Copyright 2003-2015 FIX Protocol Limited, all rights reserved.

Table of Contents

[Document History 5](#_Toc428518073)

[1 Introduction 6](#_Toc428518074)

[1.1 Authors 6](#_Toc428518075)

[2 Business Requirements 6](#_Toc428518076)

[2.1 New Requirements 6](#_Toc428518077)

[3 Issues and Discussion Points 7](#_Toc428518078)

[3.1 Should base64Binary encoding be used? 7](#_Toc428518079)

[3.2 Should encoded data be stored in elements or attributes? 7](#_Toc428518080)

[3.3 Should the length field be included in FIXML? 7](#_Toc428518081)

[3.4 Should the encoded data fields be extended to include an IETF standard content type? 8](#_Toc428518082)

[3.5 Are current FIXML fields of type data affected by these changes? 8](#_Toc428518083)

[3.6 Should fields of type NumInGroup also be included 9](#_Toc428518084)

[4 References 9](#_Toc428518085)

[5 Relevant and Related Standards 10](#_Toc428518086)

[6 Intellectual Property Disclosure 10](#_Toc428518087)

[7 Definitions 10](#_Toc428518088)

[8 FIXML Support for Encoded Data Fields 10](#_Toc428518089)

[8.1 Change of data XML datatype (xs:base64Binary) 10](#_Toc428518090)

[8.2 FIXML patterns for data and optional Length fields 11](#_Toc428518091)

[8.2.1 Field references (for data fields) as elements 11](#_Toc428518092)

[8.2.2 Fields of type data defined as complexType 12](#_Toc428518093)

[8.2.1 Fields of type Length defined as simpleType 13](#_Toc428518094)

[8.2.2 Field references (for Length fields) are not included as attributes 14](#_Toc428518095)

[8.2.3 Fields of type NumInGroup as optional attributes 14](#_Toc428518096)

[Appendix A - Usage Examples 14](#_Toc428518097)

[Appendix B – Compliance Strategy 15](#_Toc428518098)

Table of Figures

A Table of Figures is not required. If used, use styles to tag the captions and auto-generate the list here. If not used, remove this section.

# Document History

| **Revision** | **Date** | **Author** | **Revision Comments** |
| --- | --- | --- | --- |
| 0 | Dec 31, 2014 | Jim Northey | Initial draft. |
|  | Feb 15, 2015 | Rich Shriver | Revised business requirements to align with the needs of EP197. |
|  | May 25, 2015 | Rich Shriver | Drafted section 8 to align with prototyping work completed with Jim Northey. |
|  | July 6, 2015 | Rich Shriver | Completed initial revision of Section 8 to identify the FIXML implementation details. |
|  | August 7, 2015 | Jim Northey | Added content to discussion section and other minor edits. |
|  | August 28, 2015 | Rich Shriver | Added sections to include NumInGroup fields in the FIXML schema as optional attributes.  Added sections to include removal of the OptimsedImplicitBlockRepeating component type mapping to FIXML. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

The above document history section, including date, author, and comments, is required to track editing changes to the document. List revisions in **ascending order**. Please insert additional rows in the table as needed.

Template version information:

R0:0 2013-03-13 Initial draft

# Introduction

To meet new requirements for FIXML this technical standard proposal calls for FIXML, the XML encoding of FIX, to be extended to include all encoded data fields as elements with content of type base64binary. This proposal is further extended with the recommendation that all encoded data fields also be able to include a MIME type as specified in the IETF rfc 2046.

## Authors

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Affiliation** | **Contact** | **Role** |
| Jim Northey | LaSalle Technology Group LLC | [jimn@lasalletch.com](mailto:jimn@lasalletch.com) | GTC Co-chair and co-author |
| Rich Shriver | R. Shriver Associates LLC | [rich@rshriver.com](mailto:rich@rshriver.com) | Co-author |
|  |  |  |  |
|  |  |  |  |

# Business Requirements

The FIX datatype of data was created in FIX 4.2 as part of the technique to support the exchange of binary data where there is a requirement for content to be able to include reserved values of <SOH> 0x01, which is the standard delimiter in the FIX tag=value encoding.

For tag=value encoding, the technique requires two fields to be in sequence – the length of the raw data field and the raw data field. The length of the raw data field is used to determine the raw data field boundary with a datatype of Length. The raw data field contains the binary (unrestricted) data content and is of datatype data.

The data datatype is a string field containing raw data with no format or content restrictions.

Several fields in FIX such as Issuer(106) have related encoded fields – EncodedIssuerLen(348) and EncodedIssuer(349). The EncodedIssuerLen(348) field value is used to provide the FIX tag=value decoder with information to read the encoded data, as a search for the delimiter (0x01) would not produce reliable results.

Many of the encoded fields have not been included in FIXML and XML content does not have the same restriction as the tag=value encoding. An incremental improvement opportunity exists to provide a means for including binary data content within FIXML.

## New Requirements

Work complete by staff at Standard Bank in London in 2008 identified the need to include the encoded fields within FIXML. In markets where encoded data fields are used, it is often the case that the unencoded fields are also included in the FIX messages. The lack of support for encoded data fields in FIXML means that there is information loss when converting between FIX tag=value encoded data and FIXML.

Similarly, fields of type NumInGroup are not presently mapped to FIXML.

Recently, as part of the extension of FIX to support CFTC Parts 43 – 45 regulations (EP197), there are new requirements with Equity Swaps to include an encoded clip image of a payment formula within FIXML. This formula image is natively encoded using base64 binary data.

The recommendation to better support encoded fields is to use the xs:base64Binary XML Schema datatype for encoded data fields. A survey of other standards, such as FpML, the base64Binary datatype is already an established practice for encoding binary data within XML documents.

## Other Requirements

The proposal for inlined FIXML components was implemented to provide greater efficiency, coverage and flexibility in the FIXML schema generation and was intended to replace the need for the legacy OptimizedImplicitBlockRepeating component type and its associated FIXML mapping. Two (2) components (InstrmtLegGrp and UndInstrmtGrp) were identified as temporary issues for FPL constituents and their revision was postponed. In EP192, the addition of LegFinancingDetails to the InstrmtLegGrp component provided a catalyst for changing the two remaining OptimisedImplicitBlockRepeating components to BlockRepeating components with inlined component references. This proposal includes the removal of FIXML code generation for components of the type OptimisedImplicitBlockRepeating.

# Issues and Discussion Points

The following is a summary list of issues and discussion points which are explored in greater detail in this section below.

1. Should base64binary encoding be used?
2. Should encoded data be stored in elements or attributes?
3. Should the length field be included in FIXML?
4. Should the encoded data fields be extended to include an IETF standard content type?
5. Are current FIXML fields of type data affected by these changes?

## Should base64Binary encoding be used?

The use of base64Binary encoding is a widely used technique for capturing large quantities of information within an XML document. Base64Binary is used within FpML. A potential drawback is that the requirement to use base64Binary will require a conversion routine when transforming between FIXML and other FIX encoding types that are not using base64Binary encoding.

## Should encoded data be stored in elements or attributes?

The current proposal specifies storing encoded data fields within elements as opposed to attributes. This breaks the rule within FIXML regarding fields being stored in attributes. The use of elements is recommended given the likely size of some Encoded Data fields.

## Should the length field be included in FIXML?

In XML, the length attribute is not required for low level data parsing as it is in the FIX tag=value encoding. The length value is not needed in FIXML to determine the field termination as it is enclosed within XML tags. However, the use of the Length field can help support a mechanical translation between FIXML and other FIX encodings by the presence of the encoded field.

Question – does the length value include the mandatory <SOH> character at the end of the field in tag=value encoding? The length does not include the <SOH> Would it be a different value in FIXML?

## Should the encoded data fields be extended to include an IETF standard content type?

The current proposal raises the possibility of using an extension to XML that specifies a mime type. The advantage of this is to provide meta data for message consumers to be able to more intelligently process payload.

While this does not sound like a bad approach, there is no equivalent mechanism available in other FIX encodings, such as FIX tag=value.

## Are current FIXML fields of type data affected by these changes?

Some of the fields of type data are currently included in FIXML with a data type of data which is restricted by xs:string. These fields and the associated length fields are included as component attributes if they are coded in the FIX repository for inclusion in FIXML. The table below lists all fields of type data and indicates whether or not these fields are included in FIXML. The encoded data and length fields currently included in FIXML were the result of errors in the FIX Repository. The potential does exist that implementations may have used these fields.

Table 1 - fields of type data in FIXML

| Fields that are currently included in FIXML | Fields NOT currently included in FIXML |
| --- | --- |
| RawData(96) | Signature(89) |
| EncodedIssuer(349) | SecureData(91) |
| EncodedSecurityDesc(351) | XmlData(213) |
| EncodedListExecInst(353) | EncodedText(355) |
| EncodedSubject(357) | DerivativeSecurityXML(1283) |
| EncodedHeadline(359) | EncodedSecurityListDesc(1469) |
| EncodedAllocText(361) | InstrumentScopeEncodedSecurityDesc(1621) |
| EncodedUnderlyingIssuer(363) | EncodedRejectText(1665) |
| EncodedUnderlyingSecurityDesc(365) | EncodedFirmAllocText(1734) |
| EncodedListStatusText(446) | EncodedEventText(1579) |
| EncodedLegIssuer(619) | EncodedUnderlyingEventText(2073) |
| EncodedLegSecurityDesc(622) | EncodedLegEventText(2075) |
| DerivativeEncodedIssuer(1278) | EncodedDocumentationText(1527) |
| DerivativeEncodedSecurityDesc(1281) | EncodedOptionExpirationDesc(1679) |
| EncodedMktSegmDesc(1398) | EncodedLegOptionExpirationDesc(2180) |
| EncryptedPassword(1402) | EncodedUnderlyingOptionExpirationDesc(2288) |
| EncryptedNewPassword(1404) | EncodedLegDocumentationText(2493) |
| EncodedAttachment(2112) | EncodedWarningText(2521) |
| EncodedTradeContinuationText(2371) | EncodedAdditionalTermBondDesc(40005) |
| EncodedComplianceText(2352) | EncodedAdditionalTermBondIssuer(40009) |
| EncodedMDStatisticDesc(2482) | EncodedLegStreamText(40979) |
|  | EncodedLegProvisionText(40981) |
|  | EncodedStreamText(40983) |
|  | EncodedPaymentText(40985) |
|  | EncodedProvisionText(40987) |
|  | EncodedUnderlyingStreamText(40989) |
|  | EncodedDeliveryStreamCycleDesc(41084) |
|  | EncodedMarketDisruptionFallbackUnderlierSecurityDesc(41102) |
|  | EncodedExerciseDesc(41108) |
|  | EncodedStreamCommodityDesc(41257) |
|  | EncodedLegAdditionalTermBondDesc(41321) |
|  | EncodedLegAdditionalTermBondIssuer(41325) |
|  | EncodedLegDeliveryStreamCycleDesc(42459) |
|  | EncodedLegMarketDisruptionFallbackUnderlierSecurityDesc(41477) |
|  | EncodedLegExerciseDesc(41483) |
|  | EncodedLegStreamCommodityDesc(41654) |
|  | EncodedUnderlyingDeliveryStreamCycleDesc(41807) |
|  | EncodedUnderlyingExerciseDesc(41812) |
|  | EncodedUnderlyingMarketDisruptionFallbackUnderlierSecurityDesc(41874) |
|  | EncodedUnderlyingStreamCommodityDesc(41970) |
|  | EncodedUnderlyingAdditionalTermBondDesc(41711) |
|  | EncodedUnderlyingAdditionalTermBondIssuer(42026) |
|  | EncodedUnderlyingProvisionText(42172) |

## Should fields of type NumInGroup also be included

< Perhaps Jim N. could add some commentary on this discussion point >>

## Are fields of type NumInGroup in the FIX Repository affected?

Presently, no fields of the datatype NumInGroup are generated in the FIXML schema. Some of the fields include an abbreviated name and most, but not all of the fields are encoded with the attribute NotReqXML = 1 in the FIX Repository. This meta data will need attention and cleanup in support of implementation.

# References

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference** | **Version** | **Relevance** | **Normative** |
| XML Schema Part 0: Primer Second Edition  [http://www.w3.org/TR/xmlschema-0/#DefnDeclars](http://www.w3.org/TR/xmlschema-0/%23DefnDeclars) |  | Provides background on declaring simple and complex types | Incorporates |
| Describing Media Content of Binary Data in XML  [http://www.w3.org/TR/xml-media-types/#contentType](http://www.w3.org/TR/xml-media-types/%23contentType) |  | Recommended approach for specifying MIME types for binary data transmitted within an XML element. | Incorporates |
|  |  |  |  |
|  |  |  |  |

# Relevant and Related Standards

|  |  |  |  |
| --- | --- | --- | --- |
| **Related Standard** | **Reference location** | **Relationship** | **Normative** |
| XML Schema |  | Uses | Yes |
| Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies | <https://tools.ietf.org/html/rfc2045> |  | No |
| Multipurpose Internet Mail  (MIME) Part Two: Media Types | <https://tools.ietf.org/html/rfc2046> | Uses | Yes |
| FIXML Specification 1.0 | Not yet published | Extends | Yes |
| FpML |  | References | No |

# Intellectual Property Disclosure

There are no intellectual property issues identified at this time.

# Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |

# FIXML Support for Encoded Data and NumInGroup Fields

This section identifies the changes to FIXML datatypes and mapping patterns recommended for support of encoded data fields and fields of type NumInGroup.

## FIXML Datatype Changes

The proposal includes recommendations for changes to two (2) datatypes (data and NumInGroup). The XML mapping of the data datatype is recommended to be changed from xs:String to xs:base64Binary. The NumInGroup datatype is also being recommended to establish an XML datatype mapping to xs:nonNegativeInteger. Details of these changes are included in the subsections below.

### Change of data XML datatype (xs:base64Binary)

This proposal is recommending the change of the xml datatype attribute of the FIX data datatype to be changed from xs:string to xs:base64Binary. The table below lists the current information in the FIX Repository for the datatype of data and the recommended changes to the datatype metadata. The XML base type is recommended to change from xs:string to xs:base64Binary. The recommendation is to identify this datatype as an XML builtin datatype (set value = "1"). The data type description is also recommended to be replaced with a description that indicates all fields of type data are to be encoded using the base64Binary encoding.

Table 2 - data Datatype

|  |  |
| --- | --- |
| **Type name** | data |
| **Base type** | String |
| **Description** | string field containing raw data with no format or content restrictions. Data fields are always immediately preceded by a length field. The length field should specify the number of bytes of the value of the data field (up to but not including the terminating SOH).  Caution: the value of one of these fields may contain the delimiter (SOH) character. Note that the value specified for this field should be followed by the delimiter (SOH) character as all fields are terminated with an "SOH". |
| **XML builtin** | 1 (builtin) |
| **XML base type** | xs:base64Binary |
| **XML minInclusive** |  |
| **XML pattern** |  |
| **XML Description** | base64Binary In FIXML, all data type fields are encoded using base64Binary encoding. |

**Example:**

The FIXML shema for the data datatype should appear as follows:

<xs:simpleType name="data">

<xs:annotation>

<xs:documentation>base64Binary In FIXML, all data type fields are encoded using base64Binary encoding.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:base64Binary"/>

</xs:simpleType>

### Change of NumInGroup XML datatype

This proposal is recommending the addition of a FIXML mapping for the FIX datatype NumInGroup.

The table below lists the current information in the FIX Repository for the datatype of NumInGroup and the recommended changes to the datatype metadata. The XML base type is recommended to be xs:nonNegativeInteger. The data type description is also recommended to be replaced with a description that is consistent with the FIX datatype description.

Table 3 - NumInGroup Datatype

|  |  |
| --- | --- |
| **Type name** | NumInGroup |
| **Base type** | int |
| **Description** | int field representing the number of entries in a repeating group. Value must be positive inclusive of zero. |
| **XML builtin** | 0 |
| **XML base type** | xs:nonNegativeInteger |
| **XML minInclusive** |  |
| **XML pattern** |  |
| **XML Description** | int field representing the number of entries in a repeating group. Value must be positive inclusive of zero. |

**Example**

The FIXML datatype schema for the NumInGroup datatype should appear as follows:

<xs:simpleType name="NumInGroup">

<xs:annotation>

<xs:documentation>int field representing the number of entries in a repeating group. Value must be positive inclusive of zero.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:nonNegativeInteger"/>

</xs:simpleType>

## FIXML patterns for data and optional Length fields

Given the size of the image value, the frequent encoding of base64 data with 76 byte lines to support Internet email standards, the recommendation is to place the data within an element. This breaks with FIXML, which controversially stores data within attributes. The use of an element in this case permits us to include metadata as attributes to the encoded data field. One piece of metadata that can be included is the length of the field to assist in mapping between FIX tag=value and FIXML. Another option that the element approach supports is the ability to optionally specify the MIME type of the data being encoded, such as PDF, JPEG, PNG, etc.

This proposal recommends the following FIXML implementations:

* Field references (for data fields) as elements
* Fields of type data defined as complexType for base64Binary content
* Fields references (for Length fields) as an optional attribute of data field element
* xmlmime type as an optional attribute of the data field element
* Fields of type Length defined as simpleType
* Field references for fields of type data and Length are not included in the component or message attribute group.

### Field references (for data fields) as elements

FIX Repository field references for fields coded with the datatype "data" are included in FIXML as elements related to their corresponding message or component.

* The name attribute is the field's abbreviated name.
* The type attribute is set to the data field complexType (described below). The type name is a concatenation of the field name and "\_t" ("EncodedIssuer\_t" in the examples).
* For the minOccurs attribute, if the field reference is coded in the repository as not required, the minOccurs attribute of the element is set to "0"; otherwise the minOccurs attribute is set to "1" for required fields.
* The maxOccurs value of the element is always set to "1".

In the example below, the field EncodedIssuer is included within the InstrumentEmelements group.

**Example:**

<xs:group name="InstrumentElements">

<xs:sequence>

<xs:element name="EncIssr" type="EncodedIssuer\_t" minOccurs="0" maxOccurs="1"/>

</xs:sequence>

</xs:group>

This element is included in FIXML when the notReqXML attribute of the field reference and the field are both set to "0". The element is not generated if the field reference or the data attribute notReqXML is set to "1".

### Fields of type data defined as complexType

FIX Repository fields coded with a type "data" are included in FIXML as complexTypes declarations.

* The name attribute is set to the field name appended with the token "\_t" to identify the field type.
* The annotation of the complexType is the same as is included for all FIX Repository fields.
* The type is simpleContent extended by the "data" datatype which when configured as described above, results in a base64Binary content.
* The optional Length attribute and xmime contentType are described in sections below.

The example below is for the complexType definition of the EncodedIssuer(349) field.

**Example:**

<xs:complexType name="EncodedIssuer\_t">

<xs:annotation>

<xs:documentation>Encoded (non-ASCII characters) representation of the Issuer field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Issuer field.</xs:documentation>

<xs:appinfo>

<fm:Xref Protocol="FIX" name="EncodedIssuer" ComponentType="Field" Tag="349" Type="data" AbbrName="EncIssr"/>

</xs:appinfo>

</xs:annotation>

<xs:simpleContent>

<xs:extension base="data">

<xs:attribute name="EncIssrLen" type="EncodedIssuerLen\_t"/>

<xs:attribute ref="xmime:contentType" use="optional"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

This complexType is included in FIXML when the notReqXML attribute of the field is set to "0"; otherwise this complexType is not included in FIXML when the notReqXML attribute of the field is set to "1".

#### Inclusion of the optional Length field

FIX tag=value requires the inclusion of a length field so that a decoder can determine how many bytes to read for an encoded data field. XML, because of its more robust encoding, does not require this length field. A strong argument can be made for the exclusion of the length field from the FIXML encoding. As of this first draft the Length field associated with the encoded data field is included as an optional attribute on the encoded data element.

* The name attribute is the field abbreviated name for the length field.
* The type attribute is set to the field simpleType declaration. The name of the simpleType is derived by concatenating the field name with the token "\_t" ("EncodedIssuerLen\_t" in the examples).

**Example:**

<xs:attribute name="EncIssrLen" type="EncodedIssuerLen\_t"/>

This attribute is included in FIXML when the notReqXML attribute of the field is set to "0" and the associatedDataTag of the Length field is set to the tag value for the corresponding data field.

#### Inclusion of optional mime type.

The media types of the Multipart Internet Mail (MIME) have become the standard way of specifying the content type. An approach for specifying the content type of an xs:base64binary field within XML was developed by the W3C. A discussion and example can be found in [Describing Media Content of Binary Data in XML](http://www.w3.org/TR/xml-media-types/#contentType) available from the W3C via the Internet. This proposal strongly recommends this be included as an optional part of FIXML encoding.

**Example:**

<xs:attribute ref="xmime:contentType" use="optional"/>

The namespace for xmlmime must be added to the schema name spaces

xmlns:xmime="http://www.w3.org/2005/05/xmlmime"

### Fields of type Length defined as simpleType

FIX Repository fields coded with a type "Length" are included in FIXML as simpleType declarations.

* The name attribute is set to the field name appended with the token "\_t" to identify the field type ("EncodedIssuerLen\_t" in the examples).
* The annotation of the simpleType is the same as is included for FIX Repository fields.
* The simpleType is restricted by the Length datatype.

The example below is for the simpleType of the EncodedIssuerLen(348) field.

**Example:**

<xs:simpleType name="EncodedIssuerLen\_t">

<xs:annotation>

<xs:documentation>Byte length of encoded (non-ASCII characters) EncodedIssuer(349) field.</xs:documentation>

<xs:appinfo>

<fm:Xref Protocol="FIX" name="EncodedIssuerLen" ComponentType="Field" Tag="348" Type="Length" AbbrName="EncIssrLen"/>

</xs:appinfo>

</xs:annotation>

<xs:restriction base="Length"/>

</xs:simpleType>

This simpleType is included in FIXML when the notReqXML attribute of the Length field is set to "0"; otherwise this simpleType is not included in FIXML when the notReqXML attribute is set to "1".

### Field references (for Length fields) are not included as attributes

Neither the Length nor the data field reference should be included in the attribute group along with other field references for a component or message (as they exist in FIXML 1.1).

## FIXML patterns for fields of type NumInGroup

To increase the coverage and functionality of FIXML, fields of type NumInGroup are recommended to support a mapping into FIXML. This proposal recommends the following FIXML implementations to support mapping of NumInGroup fields:

* Change the NumInGroup datatype to map to the XML datatype xs:nonNegativeInteger (see 8.1.2).
* Add an attribute with the abbreviated name of the field (e.g. "NoDts") to the parent component attribute group with type of the simpleType listed below.
* Add a simpleType with the name of the field and suffix "\_t" (e.g. "NoDates\_t") and the restriction of the NumInGroup datatype.

### Field references (for NumInGroup fields) as attributes

Field references in the FIX repository should result in an attribute added to the parent component of the repeating group. The NumInGroup attribute is to be added to the parent component attribute group to provide reference for the number of entries in the repeating group (outside of the group).

**Example:**

<xs:attribute name="NoDts" type="NoDates\_t" use="optional"/>

### Fields of type NumInGroup defined as simpleType

FIX Repository fields coded with a type "NumInGroup" are included in FIXML as simpleType declarations.

* The name attribute is set to the field name appended with the token "\_t" to identify the field type ("NoDates\_t" in the examples).
* The annotation of the simpleType is the same as is included for FIX Repository fields.
* The simpleType is restricted by the NumInGroup datatype

**Example:**

<xs:simpleType name="NoDates\_t">

<xs:annotation>

<xs:documentation>Number of Date fields provided in date range</xs:documentation>

<xs:appinfo>

<fm:Xref Protocol="FIX" name="NoDates" ComponentType="Field" Tag="580" Type="NumInGroup" AbbrName="NoDts"/>

</xs:appinfo>

</xs:annotation>

<xs:restriction base="NumInGroup"/>

</xs:simpleType>

# Appendix A - Usage Examples

The following is an example FIXML document using the EncodedIssuer field. The EncodedIssuer field includes base64Binary encoded data,

<?xml version="1.0" encoding="UTF-8"?>

<FIXML v="FIX.5.0SP2" xv="192" s="2014-05-07" xmlns="http://www.fixprotocol.org/FIXML-5-0-SP2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.fixprotocol.org/FIXML-5-0-SP2 file:///E:/FPL/Build/EP197/fixmlschema\_FIX.5.0SP2\_EP197/fixml-main-5-0-SP2.xsd">

<Order ID="123456" Side="2" TxnTm="2001-09-11T09:30:47-05:00" Typ="2" Px="93.25" Acct="26522154">

<Instrmt Sym="IBM" ID="459200101" Src="1">

<EncIssr EncIssrLen="2943">/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwcG

BwcICQsJCAgKCAcHCg0KCgsMDAwMBwkODw0MDgsMDAz/2wBDAQICAgMDAwYDAwYMCAcIDAwMDAwM

DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAARCAApAGEDASIA

AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA

AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3

ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm

p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA

AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx

BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK

U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3

uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9/KKK

/PDSv+DjDwj4z8ReJrLwd+zL+2R8QrTwnrl34evtV8KfDqHVtON3bSFJUWaK8Iz0OGw21lJUZpcy

5uTra/yVl+q+8dny83Tb8/8AJn6H0V5f+yF+07/w1z8G4fGP/Cvfih8MfOu5rX+w/H+g/wBi6zH5

ZA8xrfe+I2zlW3cgGvUKqUXF2ZMZJq6CiiikMKKKKACiiigAorgvjr8ALH9oKw0yx1bxH480XStP

ne4mtfDHiW78PPqLFCqeddWTxXYRMlgkc6IxPzhwFA8D/wCCOvjbxP4z+DHxKXU/E+u+NfBOi/Er

XNJ+H2va1dyX97qOgwSIkZa7kLSXaJci6jSeRnd0jXLtgUQ96UodUub5JxX33louqTd9LBP3YqXR

u3zak/utF697K3U+uaKKKACvw+/4I6f8FQv+GN/BPx08K/8ADO/7UnxV+0fGXxLqH9sfDvwF/bml

R75o18hp/PjxMuzLJt4Doc81+4NfjF/wS/8A24PF3/BM/wANfGTwT4x/ZN/bI8U3eu/FbxB4ksdR

8KfDCa+064tLiVFiZZZZYSxPlFsqpUqykMc1FN2xD1tenJevv0nb8L/IdRXorS9pxfp7lRX/ABt8

z9bfgD8Xv+F+fBrw94x/4Rfxh4L/AOEgtBdf2H4q03+ztZ03JI8u5t9zeVJxnbuPBFefftv/APDS

n/CNaF/wzb/wo7+2PtUn9tf8LK/tT7N5G0eX9n+wfN5m7O7fxjGOa2/2Of2qP+GwvhCfF3/CuPir

8LcX8tj/AGL8QvD/APYmrnywh87yPMf902/Ctu5KtxxWJ+2/+xD/AMNt+GtC03/hb3xx+EX9hXUl

z9q+Gvir+wLnUN6hfLuH8qTzEXGVXAwSTV4mLv7qtqnv0un+X37CoSVtXfdbddV+Z85R/wDD0vzF

3f8ADAW3PzY/4S7OPavvu33+Qnm7fM2jfs+7nvj2r4Ej/wCCB3lyK3/Daf7fpwc4Pxe4P1/0Svvu

3i+zwIm532KF3Ocs2O5PrWl1yJdb/wCXUnXmufl58X/2n/2u/jj/AMFkfjF+zr8G/G/hXwh4M0jw

7omqSeJNZ0S11GTwPFLAGmls7baj3tzPK6BVuJGhRUfhSykcV8Nv2k/299O/bn8ffsbP8Rvhr4w8

TwaZa+K7T4y6n4bgsbnw/ocwEUrpo9vi3uLoTOghjc7VZJPNaRJFMX0J+zF8HfF2gf8ABwR+0x4z

vvC3iOy8H694E8O2ema7Ppk0em6jPEkfmxQ3BURyOmDuVWJXHIFN+G/wd8XWP/ByF8RvHM3hXxHD

4JvvgzY6XbeIH0yZdKuLtb+N2tkuSvlNKFBYoG3ADOKwwUbwoKevOqnNf+77WUdd1rCKumm07bNW

rFSadVx+z7Ll+apKXr8Ur3urq+9zI/4J/ftRfHz4M/8ABT3xx+yl8fvH+l/GOb/hD4fH/hPxtbeG

rfw/cy2xljtprOe1tv3ICybypG5vlYs5DqkfHfDD4/8A7U3/AAWF+MXxT1f4H/G3QP2cPg18LfFF

34L0m7TwPaeKtV8Y3luI/tFzOLxljhhXIMXlEHErBwxUMPRtQ+Dni9/+DlWw8djwr4kPghPgW+kN

4hGmTf2Ut7/axk+ym62+V52z5vL3btvOMV4R+yZ8Svin/wAEKfib8Y/hb4p/Z4+Ovxk+G/i/xpf+

N/A/ij4XeHf+EiAgvdnmWd3D5iG2eMooy7bnbzCqlNrlqSkqUqu/JU20vKNXlje2l/ZqTS6tXs2V

Ui4OoqW3NT87RdNOVr/9PGk7apNrRXPfvi/8JP20f2mv2LvCXgXX7f4TaN4mn1a6074lXFh4tv8A

Qv8AhKNHhl2wjTLq3srt7QX0Q/fM8ayRqXVFQyfuvqT9k3w74t8C/DC18N+I/h/8OvhxpnhyGHTt

B0rwd4nudcs4rOOMKqkz6dYmLbgAKqOCOSwPFV/2K/iZ8UPjF8BLLxL8XfAFn8LvFuq3d1KnhiHU

U1GXSrLzWFss86Eo87RBXfaFAL4KqQVHrFatODkpLWVm9t0vLTTXRe6m5cqV2Z+7KzjtG6Xpd9++

mu8ko3bsFFFFQUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH/9k=</EncIssr>

</Instrmt>

<OrdQty Qty="1000"/>

</Order>

</FIXML>

# Appendix B – Compliance Strategy

The FIXML Schemas are validated using both Apache and Altova software products. Further testing should be done by member firms using various language bindings, such as JAXB, JOBX, .Net, etc.