



Global Technical Committee AttachmentGrp

December 20, 2012

Revision 0.4.

Proposal Status: SubmittedPublic CommentApproved

For Global Technical Committee Governance Internal Use Only

Submission Date	October 11, 2012	Control Number	<u>EP167</u>
Submission Status	<u>SubmittedPublic CommentApproved</u>	Ratified Date	<u>February 1, 2013</u>
Primary Contact Person	GTC	Release Identifier	<u>5.0 SP3</u>

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 2003-2012 FIX Protocol Limited, all rights reserved.

Table of Contents

Document History.....	5
1 Introduction	6
2 Business Workflow.....	6
3 Issues and Discussion Points	6
4 Proposed Message Flow	7
5 FIX Message Tables	7
5.1 FIX Message TradeCaptureReport.....	7
5.2 FIX Message XMLnonFIX	8
6 FIX Component Blocks	10
6.1 Component AttachmentGrp	10
6.2 Component AttachmentKeywordGrp.....	12
Appendix A - Data Dictionary.....	13
Appendix B - Glossary Entries	16
Appendix C - Abbreviations	16
Appendix D - Usage Examples.....	16
FIXML Trade Capture Report with Base64 Encoding	16
Appendix E - Base64 Encoding.....	18

Table of Figures

Document History

Revision	Date	Author	Revision Comments
0.1	September 28, 2012	Jim Northey, LaSalle Technology Group Ryan Pierce, CME Group Lisa Taikitsadaporn, Brook Path Partners, Inc.	Initial revision.
0.2	2012-10-17	Lisa T., Jim N	Minor revisions to abbreviations, names, and descriptions.
0.3	2012-12-20	Jim N	http://jira.fixprotocol.org/browse/GTC-132 1. There's a typo in the examples. We changed Atch to Atchmnt but that wasn't reflected in the message sample. 2. The "Add to" column in the data dictionary got wiped out. 3. Attachment becomes element content, e.g. CDATA. That needs to be reflected in the proposal and the sample message. 4. Created separate component for AttachmentKeywordGrp
0.4	2012-12-21	Jim N	Incorporated feedback from GTC Dec. 20, 2012 call. Corrected AttachmentKeywordGrp documentation and entries for fields in the AttachmentKeywordGrp in the data dictionary. Documented that FIXML needs to be extended to support Data and Length.
ASBUILT	Feb. 2, 2013	Lisa T.	ASBUILT prepared
	Feb. 6, 2013	Brian D	Tags and enum values assigned
	April 22, 2013	Lisa T.	Made edits to field usage text for AttachmentMediaType(2106) per SPEC-970.

1 Introduction

The Dodd-Frank Act's 17 CFR Part 45 requires clearinghouses, swap dealers and major swap participants to report all swap transactions to Swap Data Repositories (SDRs) whether cleared or uncleared. Part 45 data are to be made accessible to the regulators (i.e. CFTC) by the SDRs.

While FPL has taken considerable effort to enhance the FIX Protocol to support the vast majority of parameters that could define a swap, such as unusual parameters in bespoke deals, one cannot hope to capture everything. Some unusual or custom parameters may appear in deal confirmations or term sheets. Additionally, firms needing to comply with the reporting requirements might not have the capability of translating all aspects of their deals into FIX. Therefore, this Gap Analysis proposes the creation of a method whereby files in some binary format (PDFs, TIFFs, Microsoft Word, etc.) could be attached to a Trade Capture Report message. However, it should be noted that the regulations only allow this reporting method as an interim or short term solution.

Additionally, this same mechanism could be used in the XMLnonFIX message to exchange arbitrary binary files.

2 Business Workflow

Firms that wish to attach a file to a TradeCaptureReport(35=AE) should first populate as many fields in the TradeCaptureReport(35=AE) as they are able. Information such as parties, notional quantity, the instrument (inasmuch as it can be defined in FIX), and identifiers (such as, for Dodd-Frank Part 45 reporting, the USI) should appear in the TradeCaptureReport(35=AE). In this regard, the attached file supplements the information in the TradeCaptureReport(35=AE), and does not replace it.

The XML Schema standard greatly restricts the character set that can appear in messages, therefore one cannot embed binary files directly in a FIXML document. However, Base64 encoding can be used, even though the Base64 representation of a file is approximately 33% larger than the original binary file.

This gap analysis proposes a new repeating group component that would allow for the attachments of various media types using Base64 encoding (or other future encodings) within the FIX tag-value or FIXML message. One might specify an AttachmentName(TBD) AttachmentMediaType(TBD), AttachmentEncodingType(TBD)=0 (Base64), and send the Base64 encoded attachment in the EncodedAttachment(TBD) field.

While this could be used to transmit attachments over Tag=Value FIX, one advantage of the Tag=Value encoding is its ability to support direct transmission of binary data using pairs of fields of Length and Data datatypes. The above could instead be sent using AttachmentEncodingType(TBD)=1 (Raw binary) in its native binary format, provided that the length appears in EncodedAttachmentLen(TBD). (Note: the EncodedAttachmentLen(TBD) is not used in FIXML.)

Alternately, a TradeCaptureReport(35=AE) could reference a binary document by specifying an AttachmentExternalURL(TBD) where one could retrieve the document directly. However, this presumes the recipient has the capacity to access the URL, which may be complicated by firewalls, VPN configuration, etc.

The field UnencodedAttachmentLen(TBD) indicates the length of the attached file in its native format *before* it was encoded. It can be used with Base64 encoding as a validation that data was not lost or omitted. This field is redundant for raw binary encoding, as the encoded and unencoded lengths are identical.

Similarly, a firm could send a binary file as an attachment, or as a URL reference, using the XMLnonFIX(35=n) message.

3 Issues and Discussion Points

Support for the AttachmentGrp will require an enhancement to FIXML to support FIX datatypes Data and Length, which are not currently supported in FIXML. The standard practice for specification of encoding of Data fields should be addressed at this time. FIX provides a field in the StandardHeader Component – MessageEncoding(tag

347), whose current definition is “Type of message encoding (non-ASCII characters) used in a message’s “Encoded” fields. Required if any “Encoding” fields are used.”

4 Proposed Message Flow

This proposal does not change the flow of existing TradeCaptureReport(35=AE) or XMLnonFIX(35=n) messages. Rather, it enables them to carry an additional binary attachment.

5 FIX Message Tables

5.1 FIX Message TradeCaptureReport

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	TradeCaptureReport
Message Abbreviated Name (for FIXML)	TrdCaptRpt
Category	TradeCapture
Message Synopsis	The Trade Capture Report message can be: <ul style="list-style-type: none"> - Used to report trades between counterparties. - Used to report trades to a trade matching system - Can be sent unsolicited between counterparties. - Sent as a reply to a Trade Capture Report Request. - Can be used to report unmatched and matched trades.
Message Elaboration	
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	AE
Repository Component ID	64

[Other additional text detailing usage of the message may be entered here]

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>XMLName</i>	<i>FIX Spec Comments</i>	<i>Action</i>	<i>Mappings and Usage Comments</i>
	<i>StandardHeader</i>	Y	BaseHeader	MsgType = AE		
	<i>Component <ApplicationSequenceControl></i>	N	<i>ApplSeqCtrl</i>			
571	TradeReportID	N	RptID	TradeReportID(571) is conditionally required in a message-chaining		

				model in which a subsequent message may refer to a prior message via TradeReportRefID(572). The alternative to a message-chain model is an entity-based model in which TradeID(1003) is used to identify a trade. In this case, TradeID(1003) is required and TradeReportID(571) can be optionally specified.		
1003	TradeID	N	TrdID			
1040	SecondaryTradeID	N	TrdID2			
1041	FirmTradeID	N	FirmTrdID			
1042	SecondaryFirmTradeID	N	FirmTrdID2			
487	TradeReportTransType	N	TransTyp			
856	TradeReportType	N	RptTyp			
939	TrdRptStatus	N	TrdRptStat	Status of the trade report. In 3-party listed derivatives model, this is used to convey status of a trade to a counterparty. Used specifically in a "give-up" (also known as "claim") model.		
568	TradeRequestID	N	ReqID	Identifier for the trade capture report request associated with this trade capture report.		
<...truncated...>						
751	TradeReportRejectReason	N	RejRsn	Indicates the reason that a trade report was rejected.		
1328	RejectText	N	RejTxt			
1664	EncodedRejectTextLen	N				
1665	EncodedRejectText	N				
1329	FeeMultiplier	N	FeeMult			
Component <AttachmentGrp>		N	Attchmnt		NEW	
StandardTrailer		Y	Trlr			

5.2 FIX Message XMLnonFIX

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	XMLnonFIX
Message Abbreviated Name (for FIXML)	XMLnonFIX
Category	Session
Message Synopsis	
Message Elaboration	
To be finalized by FPL Technical Office	
(MsgType(tag 35) Enumeration	n
Repository Component ID	47

[Other additional text detailing usage of the message may be entered here]

<i>Tag</i>	<i>Field Name</i>	<i>Req'd</i>	<i>XMLName</i>	<i>FIX Spec Comments</i>	<i>Action</i>	<i>Mappings and Usage Comments</i>
	<i>StandardHeader</i>	Y	BaseHeader			
	Component <AttachmentGrp>	N	Attchmnt		NEW	
	<i>StandardTrailer</i>	Y	Trlr			

6 FIX Component Blocks

6.1 Component AttachmentGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	AttachmentGrp
Component Abbreviated Name (for FIXML)	Attchmnt
Component Type	_X_ Block Repeating ___ Block
Category	[enter the category name here]
Component Synopsis	<p>The AttachmentGrp component provides the ability to attach other media type documents to a FIX message for transmission. The media type can be any of the media types (previously referred to as MIME types) that are listed by IANA (www.iana.org) [RFC2046].</p> <p>The AttachmentGrp is intended to be used to attach documents in other formats, such as PDF, TIFF, and Microsoft Word, for example to a FIX message.</p> <p>Note- when the AttachmentGrp is used within a business message, such as the TradeCaptureReport(35=AE), the attachment should supplement the data already contained in the <u>business</u> message. It is not intended to replace the content of the business message. The standard fields within the business message should be populated, even if they duplicate data expressed <u>within</u> the attachment(s).</p>
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2234

Component FIXML Abbreviation: <Attchmnt>						
Tag	Field Name	Req'd	ICR	Action	Mappings and Usage Comments	Comments
TBD 2104	NoAttachments	N		NEW		
→	TBD 2105 AttachmentName	N		NEW		Required if NoAttachments(TBD2104) > 0

→	TBD 2106	AttachmentMediaType	N		NEW		Required if EncodedAttachment(2112) is present. Required if NoAttachments(TBD2104) > 0
→	TBD 2107	AttachmentClassification	N		NEW		
→	TBD 2108	AttachmentExternalURL	N		NEW		Either AttachmentExternalURL(TBD2108) or EncodedAttachment(TBD2112) must be specified if NoAttachments(TBD2104) > 0
→	TBD 2109	AttachmentEncodingType	N		NEW		Required if EncodedAttachment(TBD2112) is present.
→	TBD 2110	UnencodedAttachmentLen	N		NEW		
→	TBD 2111	EncodedAttachmentLen	N		NEW		Must be set if EncodedAttachment(TBD2112) is specified and must immediately precede it.
→	TBD 2112	EncodedAttachment	N		NEW		Either AttachmentExternalURL(TBD2108) or EncodedAttachment(TBD2112) must be specified if NoAttachments > 0
→	Component <AttachmentKeywordGrp>		N		NEW		
</Attchmnt>							

6.2 Component AttachmentKeywordGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	AttachmentKeywordGrp
Component Abbreviated Name (for FIXML)	Keywd
Component Type	_X_ Block Repeating ___ Block
Category	[enter the category name here]
Component Synopsis	The AttachmentKeywordGrp component provides a place to associate keywords with an attachment document to support the current approach of tagging to support metadata.
Component Elaboration	
To be finalized by FPL Technical Office	
Repository Component ID	2235

Component FIXML Abbreviation: <Keywd>						
Tag	Field Name		Req'd	ICR	Action	Mappings and Usage Comments
→	TBD2113	NoAttachmentKeywords	N		NEW	
→	→	TBD2114 AttachmentKeyword	N		NEW	Required if NoAttachmentKeywords(TBD2113) > 0
</Attch>						

Appendix A - Data Dictionary

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
TBD2 104	NoAttachments	NEW	NumInGroup	The number of attached files.		AttachmentGrp
TBD2 105	AttachmentName	NEW	String	Specifies the file name of the attachment.	@Name	AttachmentGrp
TBD2 106	AttachmentMediaType	NEW	String	<p>The MIME media type (and optional subtype) of the attachment. The values used are those assigned, listed and maintained by IANA (www.iana.org) [RFC2046]. See http://www.iana.org/assignments/media-types/index.html for available types.</p> <p>Examples values (RFC number provided for reference here only): "application/pdf" (see [RFC3778]) "application/msword" (for .doc files) "multipart/signed" (see [RFC1847]) "application/vnd.openxmlformats-officedocument.wordprocessingml.document" (for .docx files)</p>	@MediaTyp	AttachmentGrp

<p>TBD2 107</p>	<p>AttachmentClassification</p>	<p>NEW</p>	<p>String</p>	<p>Specifies semantically the type of the attached document from a business perspective. The default classification scheme reuses the FIX standard classification scheme of a high level section (pretrade, trade, posttrade, etc.) and a category, then a specific application or document type. The expression follows {"section/category/application type"}.</p> <p>The goal here is to map the attachment into the sections and categories of the FIX business messages if possible. The classification scheme can be expanded or replaced by counterparty agreement. This approach permits the introduction and reference to other business ontologies.</p> <p>Example: posttrade/confirmation/confirm pretrade//termsheet</p>	<p>@Clfn</p>	<p>AttachmentGrp</p>
<p>TBD2 108</p>	<p>AttachmentExternalURL</p>	<p>NEW</p>	<p>String</p>	<p>Used to specify an external URL where the attachment can be obtained.</p>	<p>@URL</p>	<p>AttachmentGrp</p>

TBD2 109	AttachmentEncodingType	NEW	int	<p>The encoding type of the content provided in EncodedAttachment(TBD2112).</p> <p>Valid values:</p> <p>0 = Base64</p> <p>1 = Raw binary (Elaboration: Unencoded binary content.)</p> <p>Reserved100+</p> <p>Elaboration: The AttachmentEncodingType is a distinct and separate concept from MessageEncoding(347) that defines how FIX fields of type Data are encoded. The MessageEncoding(347) is used embed text in another character set (e.g. Unicode or Shift-JIS) within FIX.</p>	@EncTyp	AttachmentGrp
TBD2 110	UnencodedAttachmentLength	NEW	int	Unencoded content length in bytes. Can be used to validate successful unencoding.	@UnencLen	AttachmentGrp
TBD2 111	EncodedAttachmentLen	NEW	Length	Byte length of encoded the EncodedAttachment(TBD2112) field.	@Len	AttachmentGrp
TBD2 112	EncodedAttachment	NEW	Data	The content of the attachment in the encoding format specified in the AttachmentEncodingType(TBD2109) field.	@Attchmnt	AttachmentGrp
TBD2 113	NoAttachmentKeywords	NEW	NumInGroup	The number of attachment keywords.		AttachmentKeywordGrp
TBD2 114	AttachmentKeyword	NEW	String	Can be used to provide data or keyword tagging of the content of the attachment.	@Keywd	AttachmentKeywordGrp

Appendix B - Glossary Entries

Term	Definition	Field where used										
MIME	<p>Multipurpose Internet Mail Extensions – used to specify an email attachment. The MIME is made up of a Media Type and optional Subtype.</p> <p>http://en.wikipedia.org/wiki/MIME</p> <p>Examples:</p> <table border="1"> <tr> <td>Adobe PDF documents</td> <td>application/pdf http://www.rfc-editor.org/rfc/rfc3778.txt</td> </tr> <tr> <td>Microsoft Office Office Open XML Word Document (.docx)</td> <td>application/vnd.openxmlformats-officedocument.wordprocessingml.document</td> </tr> <tr> <td>Microsoft Office Word Document (.doc)</td> <td>application/msword</td> </tr> <tr> <td>Microsoft Office Office Open XML Excel Spreadsheet (.xlsx)</td> <td>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</td> </tr> <tr> <td>Microsoft Office Excel Spreadsheet (.xls)</td> <td>application/ms-excel</td> </tr> </table>	Adobe PDF documents	application/pdf http://www.rfc-editor.org/rfc/rfc3778.txt	Microsoft Office Office Open XML Word Document (.docx)	application/vnd.openxmlformats-officedocument.wordprocessingml.document	Microsoft Office Word Document (.doc)	application/msword	Microsoft Office Office Open XML Excel Spreadsheet (.xlsx)	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	Microsoft Office Excel Spreadsheet (.xls)	application/ms-excel	
Adobe PDF documents	application/pdf http://www.rfc-editor.org/rfc/rfc3778.txt											
Microsoft Office Office Open XML Word Document (.docx)	application/vnd.openxmlformats-officedocument.wordprocessingml.document											
Microsoft Office Word Document (.doc)	application/msword											
Microsoft Office Office Open XML Excel Spreadsheet (.xlsx)	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet											
Microsoft Office Excel Spreadsheet (.xls)	application/ms-excel											

Appendix C - Abbreviations

Term	Proposed Abbreviation	Proposed Messages, Components, Fields where used
Keyword	Keywd	AttachmentKeyword
Media	Media	AttachmentMediaType
Unencoded	Unenc	UnencodedAttachmentLen
Attachment	Attchmnt	Attachment

Appendix D - Usage Examples

FIXML Trade Capture Report with Base64 Encoding

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXML>
  <TrdCaptRpt TransTyp="0" LastQty="15000000.0" QtyTyp="0"
  LastPx="97.0400000" RptID="4FC77E9A1464008D" PxNeg="4" RptTyp="0">
```



```
ExecID2="4FC77E9A1464008D" TxnTm="2012-05-31T14:22:18.467+00:00" TrdTyp="22"
VenuTyp="S" RegRptTyp="4" Clrd="2">
  <Hdr TID="CME" SSub="SenderUserID" TSub="CPAPI" SID="XXX"/>
  <RegTrdID ID="1342376676" Typ="0" Src="someone_Namespace" Evnt="0"/>
  <Pty R="73" ID="LEI of the SEF" Src="N"/>
  <Instrmt Src="104" MMY="201606" CpnRt="5.00000" SecTyp="CDS"
ID="2I65BRIV4" Exch="CMD"/>
  <Amt Ccy="USD" Typ="IPMT" Amt="-283159.00"/>
  <TrdRegTS Typ="1" TS="2012-05-31T14:22:18.520536+00:00"/>
  <RptSide InptSrc="XXX" Side="1" ClOrdID="4FC77E9A1464009E">
    <Pty Src="D" R="24" ID="CCOLCMECDSGM1">
      <Sub Typ="3" ID="XXX"/>
    </Pty>
    <Pty Src="D" R="36" ID="8198134">
      <Sub Typ="9" ID="SOME PERSON"/>
    </Pty>
  </RptSide>
  <RptSide InptSrc="XXX" Side="2" ClOrdID="4FC77E9A146400AD">
    <Pty Src="D" R="24" ID="MSAUCT001">
      <Sub Typ="3" ID="BMG"/>
      <Sub Typ="49" ID="Y"/>
    </Pty>
    <Pty Src="D" R="36" ID="8198134">
      <Sub Typ="9" ID="Trader Joe"/>
    </Pty>
  </RptSide>
  <Attchmnt Name="HelloWorld.pdf" MediaTyp="application/pdf" EncTyp="0"
UnencLen="89762"><![CDATA[
JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC9UeXB1L0NhDGfSb2cvUGFnZXMGMiAwIFIVtGFu
Zyhlbi1VUykgL1N0cnVjdFRyZWVsb290IDewIDAgUi9NYXJrSW5mbzw8L01hcmtlZCB0cnVlPj4+
Pg0KZW5kb2JqDQoyIDAgb2JqDQo8PC9UeXB1L1BhZ2VzL0NvdW50IDevS2lkclsgMyAwIFJdID4+
DQplbmRvYmoNCjMgMCAvYmoNCjw8L1R5cGUvUGFnZS9QYXJlbnQgMiAwIFIVtGFuVzVzY2VzPDdw
WE9iamVjdDw8L0ltYWdlNSA1IDAgUj4+L0V4dEdTdGF0ZTw8L0dTNiA2IDAgUj4+L0ZvbnQ8PC9G
MSA3IDAgUj4+L1Byb2NTZXRB1BERi9UZXB0L0ltYWdlQi9JbWFnZUMvSW1hZ2VJXSA+Pi9NZWRp
YUJveFsgMCAwIDYxMiA3OTJdIC9Db250ZW50cyA0IDAgUi9Hcm91cDw8L1R5cGUvR3JvdXAvUy9U
cmFuc3BhcnVuY3kvQ1MvRGV2aWNlUkdCPj4vVGficy9TL1N0cnVjdFBhcnVudHMgMD4+DQplbmRv
YmoNCjQgMCAvYmoNCjw8L0ZpbHRlcj9GbgF0ZURlY29kZS9MZW5ndGggMjA4Pj4NCnN0cmVhbQ0K
...
MDAwIG4NCnRyYWlsZXINCjw8L1NpemUgMjQvUm9vdCAxIDAgUi9JbmZvIDkgMCAvS2lEZWw1Q0Q0
OTI5RDBFNDA0NTQ0QTlDMUQ2QkJFNkNBQkM5MT48NUNENDEyOUQwRTQwNDU0NEE5QzFENkJCRTZD
QUJDOTE+XSA+Pg0Kc3RhcncR4cmVmdQo4ODk0OA0KJSVFT0YncnhyZWYncjAgMA0KdHJhaWxlcg0K
PDwvU2l6ZSAyNC9Sb290IDegMCAvS2lEZWw1Q0Q0OTI5RDBFNDA0NTQ0QTlDMUQ2QkJFNkNBQkM5MT5dIC9QcmV2IDg4
OTQ4L1hSZWZTdG0gODg2NjE+Pg0Kc3RhcncR4cmVmdQo4OTU4NA0KJSVFT0Y=
... most of document omitted ...
]]>/>
... remainder of message omitted ...
</TrdCaptRpt>
</FIXML>
```

Appendix E - Base64 Encoding

Base64 encoding, as used in MIME, is defined in [RFC 1521] which is available at:
<http://www.ietf.org/rfc/rfc1521.txt>

Relevant text from section 5.2 is quoted below:

The Base64 Content-Transfer-Encoding is designed to represent arbitrary sequences of octets in a form that need not be humanly readable. The encoding and decoding algorithms are simple, but the encoded data are consistently only about 33 percent larger than the unencoded data. This encoding is virtually identical to the one used in Privacy Enhanced Mail (PEM) applications, as defined in RFC 1421. The base64 encoding is adapted from RFC 1421, with one change: base64 eliminates the "*" mechanism for embedded clear text.

A 65-character subset of US-ASCII is used, enabling 6 bits to be represented per printable character. (The extra 65th character, "=", is used to signify a special processing function.)

NOTE: This subset has the important property that it is represented identically in all versions of ISO 646, including US ASCII, and all characters in the subset are also represented identically in all versions of EBCDIC. Other popular encodings, such as the encoding used by the uuencode utility and the base85 encoding specified as part of Level 2 PostScript, do not share these properties, and thus do not fulfill the portability requirements a binary transport encoding for mail must meet.

The encoding process represents 24-bit groups of input bits as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8-bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the base64 alphabet. When encoding a bit stream via the base64 encoding, the bit stream must be presumed to be ordered with the most-significant-bit first. That is, the first bit in the stream will be the high-order bit in the first byte, and the eighth bit will be the low-order bit in the first byte, and so on.

Each 6-bit group is used as an index into an array of 64 printable characters. The character referenced by the index is placed in the output string. These characters, identified in Table 1, below, are selected so as to be universally representable, and the set excludes characters with particular significance to SMTP (e.g., ".", CR, LF) and to the encapsulation boundaries defined in this document (e.g., "-").

Table 1: The Base64 Alphabet

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1

3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

The output stream (encoded bytes) must be represented in lines of no more than 76 characters each. All line breaks or other characters not found in Table 1 must be ignored by decoding software. In base64 data, characters other than those in Table 1, line breaks, and other white space probably indicate a transmission error, about which a warning message or even a message rejection might be appropriate under some circumstances.

Special processing is performed if fewer than 24 bits are available at the end of the data being encoded. A full encoding quantum is always completed at the end of a body. When fewer than 24 input bits are available in an input group, zero bits are added (on the right) to form an integral number of 6-bit groups. Padding at the end of the data is performed using the '=' character. Since all base64 input is an integral number of octets, only the following cases can arise: (1) the final quantum of encoding input is an integral multiple of 24 bits; here, the final unit of encoded output will be an integral multiple of 4 characters with no "=" padding, (2) the final quantum of encoding input is exactly 8 bits; here, the final unit of encoded output will be two characters followed by two "=" padding characters, or (3) the final quantum of encoding input is exactly 16 bits; here, the final unit of encoded output will be three characters followed by one "=" padding character.

Because it is used only for padding at the end of the data, the occurrence of any '=' characters may be taken as evidence that the end of the data has been reached (without truncation in transit). No such assurance is possible, however, when the number of octets transmitted was a multiple of three.

Any characters outside of the base64 alphabet are to be ignored in base64-encoded data. The same applies to any illegal sequence of characters in the base64 encoding, such as "===="

Note that it is customary within MIME to break the encoded Base64 stream into lines of exactly 76 bytes of data (plus CR and LF or other whitespace), except for the last line, which has <= 76 bytes of data. This is convenient, as it is less than 80 columns, a historically significant number, and it is also a multiple of 4 bytes. (3 binary bytes of input data are encoded into 4 binary bytes of output data.) Breaking lines is optional. Firms should accept attachments with and without line breaks, as well as other white space characters.