

Network working group
Internet draft

G. Klyne, Clearswift
9 April 2002
Expires: October 2002

An XML format for mail and other messages
[<draft-klyne-message-rfc822-xml-03.txt>](#)

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/1id-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright Notice

Copyright (C) The Internet Society 2001. All Rights Reserved.

Abstract

This document describes a coding of email and other messages in XML. This coding is intended for use by XML applications that exchange information about such messages.

Discussion of this document

Send comments to ietf-message-xml@research.mimesweeper.com. To subscribe to this list, send a message with the body 'subscribe' to ietf-message-xml-request@research.mimesweeper.com.

Table of contents

1.	Introduction.....	3
1.1	Structure of this document	3
1.2	Document terminology and conventions	4
1.3	About MIME and XML	5
2.	Message structures.....	6
2.1	Message header overview	6
2.2	Multipart/related message structure	7
2.3	Inline XML message structure	8
2.4	Content type Message/Email+XML	8
3.	Message header.....	9
3.1	The <Message> element	9
3.2	Content of <Message> element	10
3.3	Use of XML namespaces	10
3.4	The <content> element	11
3.5	General form of header field elements	12
3.6	RFC822-derived header elements	12
3.7	Header fields containing addresses	13
3.7.1	Header fields containing address groups.....	14
3.8	Header elements containing human readable text	15
3.9	MIME header fields	15
3.10	Other header fields	15
3.10.1	Mandatory extensions.....	16
4.	Summary of RFC822-derived header elements.....	17
5.	IANA considerations.....	17
6.	Internationalization considerations.....	18
6.1	International URIs in XML	19
7.	Security considerations.....	19
8.	Acknowledgements.....	20
9.	References.....	20
10.	Author's address.....	23
	Appendix A: Message/Email+XML content-type registration.....	24
	Appendix B: DTD for Email+XML message format.....	24
	Appendix C: XML schema for Email+XML message format.....	24
	Appendix D: RDF representation of Email+XML message.....	24
	Appendix E: RDF schema for Email+XML message format.....	25
	Appendix F: Amendment history.....	25
	Appendix G: Outstanding issues.....	26
	Full copyright statement.....	26

1. Introduction

This document describes a coding of email and similar messages (such as [RFC822](#) [1]) using XML [2], described here as the Email+XML message format.

The present document is presented as a design that can be used by XML applications that deal with email and similar messages.

The XML coding is designed to address the following goals:

- o to fully capture the semantics of Internet email messages, per [RFC822](#) [1]. However it is not intended to provide a loss-less coding of [RFC822](#) syntax.
- o to extend the scope of address information that can be conveyed to arbitrary URIs [3].
- o to take account of 8-bit clean transfer environments.
- o to fully support, where applicable, international character sets and languages within the message header and content [4,5].
- o to be usable in MIME [6] and pure XML [2] transfer environments.
- o to be fully compliant with the XML [2] and XML namespace [9] specifications.
- o to allow header information to be compatible with RDF format [10], for use by generalized metadata processing applications.

1.1 Structure of this document

[Section 2](#) describes the overall message structure, showing how the message header and message content can be conveyed in MIME and XML transfer environments.

[Section 3](#) describes the message header in greater detail, with particular reference to differences in the value of individual fields compared their [RFC822](#) counterparts.

[Section 4](#) discusses issues that may arise when converting between traditional [RFC822](#) and the Email+XML message format described here.

[Appendix A](#) contains a MIME content-type registration for Message/Email+XML.

[Appendix B](#) contains a DTD for the Email+XML message format.

[Appendix C](#) contains an XML schema for the Email+XML message format. (XML schema are set to replace DTDs are the preferred way to describe XML document content.)

[Appendix D](#) briefly discusses the RDF representation [[10](#)] and its applicability to the Email+XML message format.

[Appendix E](#) contains an RDF schema [[23](#)] description for the Email+XML message format.

[1.2](#) Document terminology and conventions

Message an assemblage of information that constitutes a communication of information from a sender to one or more recipients. Consists of a message header and message content.

Message header
 contains information about the message that is conveyed between message user agents, and not used by the message transfer mechanisms. This may include who the message is from, who it is addressed to, other parties to whom it has been copied, subject of the message, date the message was composed, etc.

Message content
 some arbitrary data carried in a message.

Email+XML
 is the message format defined by this document. (This name uses the XML content type labelling convention [[11](#)].)

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[19](#)].

NOTE: Comments like this provide additional nonessential information about the rationale behind this document. Such information is not needed for building a conformant implementation, but may help those who wish to understand the design in greater depth.

[[[Editorial comments and questions about outstanding issues are provided in triple brackets like this. These working comments should be resolved and removed prior to final publication.]]]

1.3 About MIME and XML

There has been much discussion about the relative merits of MIME and XML. The position of this document is that they serve different purposes, and are complementary rather than alternatives.

MIME is a framework primarily for encapsulating and composing arbitrary data entities, and offers the following capabilities:

- o Content type labelling.
- o Transfer encoding for handling arbitrary data on restricted channels.
- o Assembly of different kinds of data into composite entities.
- o End of data detection without need to parse or understand the data content.

XML is a framework primarily for describing data structures, including semi-structured document data, and offers the following capabilities:

- o Construction of arbitrary data structures based on an annotated tree model.
- o Fine-grained labelling of structure components and data attributes.
- o Cross-linking between data structure components.
- o A standard format for interchange of structured information between diverse systems.

There is, of course, some overlap in capabilities, and reasonable people may disagree about the appropriateness of using MIME and/or XML in particular circumstances.

This document is predicated on the idea that XML is a useful mechanism (in addition to existing facilities) for structuring message header information. It aims to be agnostic with regard to using MIME or some other framework for composing and encapsulating messages.

2. Message structures

A message consists of a message header and message content:

- o The message header contains information about the message: who it was sent by, who it is addressed to, its subject, date it was sent, and many other related pieces of information.
- o The message content is any data that is carried by the message: e.g. a text message, fax image, voice message or arbitrary application data. In principle, any data that can be transferred as a MIME object can be message content, though specific applications may limit the kinds of data that can be transferred.

The Email+XML message format uses a URI-reference [[3](#)] in the message header to reference the message content. Thus, the message content may be completely separate from the message header; the message header is the root information of a message, from which message content may be discovered.

Two specific message structure scenarios are contemplated here:

- o Multipart/related, and
- o An XML element within the message header.

These are described below. Other message structures are possible (e.g. multiple resources on a web server, multiple channels in a multiplexed protocol), but are not described here.

2.1 Message header overview

The message header is an XML document whose root element is <Message>. This contains a number of elements; an initial set of such elements is defined based on [RFC822](#) message headers [[1](#)].

The message content is indicated by an attribute of the <Message> element whose value is a URI-reference for the content.

The message header is discussed in greater detail in [section 3](#) below.

2.2 Multipart/related message structure

A message whose content is formatted as a MIME object [[6](#)] may be sent as a Multipart/related object [[15](#)]:

```
Content-type: multipart/related; boundary="boundary";
              start="<1@100Aker.org>";
              type="message/Email+XML"
```

```
--boundary
Content-type: Message/Email+XML
Content-ID: <1@100Aker.org>
```

```
<emx:Message
  xmlns:emx='urn:ietf:params:email-xml:'
  xmlns:rfc822='urn:ietf:params:rfc822:'
  emx:content='cid:2@100Aker.org'>
  <rfc822:from>
    <emx:Address>
      <emx:adrs>mailto:Pooh@PoohCorner.100Aker.org</emx:adrs>
      <emx:name>Winnie the Pooh</emx:name>
    </emx:Address>
  </rfc822:from>
  <rfc822:to>
    <emx:Address>
      <emx:adrs>mailto:Piglet@BeechTree.100Aker.org</emx:adrs>
      <emx:name>MR SANDERS</emx:name>
    </emx:Address>
  </rfc822:to>
  <rfc822:subject>Woozle Hunting</rfc822:subject>
</emx:Message>
--boundary
Content-Type: text/plain;charset=UTF-8
Content-ID: <2@100Aker.org>
```

```
I have Been Foolish and Deluded
I am a Bear of No Brain at All
--boundary--
```

In this case, the Multipart/related contains two MIME parts:

- o the message header, and
- o the message content.

The Multipart/related content-type header indicates the root of the message by its Content-ID value [6]. In turn, the message header refers to the message content with a <Message> element 'content=' attribute whose value is a 'cid:' URI [16].

2.3 Inline XML message structure

When the message content can be expressed as simple text or XML, it may be included within the message header using a <content> element containing the message content instead of a 'content=' attribute.

Content-type: Message/Email+XML

```
<emx:Message
  xmlns:emx='URN:ietf:params:email-xml:'
  xmlns:rfc822='URN:ietf:params:rfc822:'>
  <rfc822:from>
    <emx:Address>
      <emx:adrs>mailto:Christopher.Robin@GreenDoor.org</emx:adrs>
      <emx:name>Christopher Robin</emx:name>
    </emx:Address>
  </rfc822:from>
  <rfc822:to>
    <emx:Address>
      <emx:adrs>mailto:Pooh@PoohCorner.100Aker.org</emx:adrs>
      <emx:name>Winnie the Pooh</emx:name>
    </emx:Address>
  </rfc822:to>
  <rfc822:subject>Re: Woozle hunting</rfc822:subject>
  <emx:content type='text/plain'>
    You're the Best Bear in All the World
  </emx:content>
</emx:Message>
```

This example shows the message contained within a single Message/Email+XML MIME object.

The <content> element indicates the message content. When present, this element MUST be the last element contained in a <Message> element.

2.4 Content type Message/Email+XML

This specification defines a new MIME content-type called Message/Email+XML.

A Message/Email+XML entity contains an XML document conforming to the DTD known by the SYSTEM identifier 'urn:ietf:params:xml:dtd:email-xml:', per [24]. The document may contain <?XML?> and <!DOCTYPE> declarations, but these are not required.

The body of the document is a <Message> element, as described below.

The character set encoding used in a Message/Email+XML entity is UTF-8.

A Content-type registration template for Message/Email+XML is contained in [Appendix A](#) of this document.

3. Message header

The Email+XML message header contains header fields based on [RFC822](#), and coded in XML.

The message header contains information about the message that is conveyed between message user agents, and not used by the message transfer mechanisms. This may include who the message is from, who it is addressed to, other parties to whom it has been copied, subject of the message, date the message was composed, etc.

The message header also contains a reference to the message content, as described in the previous section.

3.1 The <Message> element

The <Message> element contains the message header, and references the message content.

Possible attributes are:

- o 'xmlns=' or 'xmlns:tag=' is used to indicate a default XML namespace or XML namespace tag [9] that applies to the entire <Message> element.
- o 'content=' specifies a URI-reference [3] that references the message content, if such content is not contained inline in a '<content>' element. Typically, the value is a 'cid:' URI as described in the previous section. Other message content URI values are possible, but such use is beyond the scope of this specification.

- o 'xml:lang=' [2] may be used, in which case it specifies the language of any text in the message header, except where overridden by an 'xml:lang=' attribute of an enclosed element.

[3.2](#) Content of <Message> element

The content of a <Message> element is:

- o a sequence of zero or more header field elements, and
- o an optional <content> element.

Header field elements may appear in any order. When present, the <content> element MUST be the last one in the <Message>.

The <Message> element MUST contain either a 'content=' attribute or a single <content> element. It must not contain both.

[3.3](#) Use of XML namespaces

The <Message> element, <Address> and related element names, the <content> element and <Message-content> element names are all associated with a namespace called 'URN:ietf:params:email-xml:'. [RFC822](#) header element names are associated with a namespace called 'URN:IANA:namespace:[rfc822](#):'. (These namespace identifiers are based on "A URN Sub-namespace for Registered Protocol Parameters" [20].)

The namespaces must be declared, either as a default namespace or using a namespace prefix (which is an arbitrary local name). The namespace declaration may appear as an attribute of the <Message> element, or in the surrounding XML context.

The message examples in [section 2](#) use namespace prefixes 'emx:' and '[rfc822](#)', but any prefix could be used here. Here is a different message example using a default namespace rather than a namespace prefix for the non-RFC822-derived names:

Content-type: Message/Email+XML

```
<Message
  xmlns='URN:ietf:params:email-xml:'
  xmlns:rfc822='URN:ietf:params:rfc822:'>
<rfc822:from>
  <Address>
    <adrs>im:Eeyore@ThistlyCorner.100Aker.org</adrs>
    <name>Eeyore</name>
  </Address>
</rfc822:from>
<rfc822:to>
  <Group>
    <name>Anyone</name>
  </Group>
</rfc822:to>
<rfc822:subject>Why?</rfc822:subject>
<content type='text/plain'>
  Wherefore?
  Inasmuch as which?
</content>
</Message>
```

[3.4](#) The <content> element

The <content> element is used to include the message content as text or XML data in the message header. It is present when the <Message> element does not have a 'content=' attribute.

Possible <content> attributes are:

- o 'type=' is optional, and indicates the MIME content-type of the message content. If not specified, a content type of "text/xml" is assumed.

(Whatever MIME content-type may be declared, the message content must be well-formed XML or character data. In practice, this means the content must be some character-based data representation.)

- o 'xml:lang=' [\[2\]](#) may be used, in which case it specifies the language of the message content.

[<draft-klyne-message-rfc822-xml-03.txt>](#)

The character encoding for the message content is the same as that used for the surrounding XML. This is typically UTF-8, from the character set encoding of the MIME content-type Message/Email+XML.)

The message content may be any well-formed XML, which includes simple character data. Characters '<' and '&' that are not part of XML markup MUST be represented as '<' and '&,' respectively. The character '>' appearing in the sequence ']]>', other than at the end of a CDATA section, MUST be represented as '>,'.

[3.5](#) General form of header field elements

Each header field is represented by an XML element that identifies the field.

The element content is the header field value. For [RFC822](#) and MIME header fields, the field value is character data in which the characters '<', '&' and '>' are represented as for character data in <Message-content> (see above).

[3.6](#) [RFC822](#)-derived header elements

For representing information about email messages, this specification introduces message header elements with names and semantics based on [RFC822](#) header fields [[1](#)]. The intent is that the semantics of any [RFC822](#) header field is easily represented in an Email+XML header element; it is not a goal to capture the detailed syntax of any particular [RFC822](#) message, or to construct a corresponding [RFC822](#) message from any Email+XML message.

[RFC822](#)-derived header elements have names based on [RFC822](#) header names, using all lower-case characters (noting that XML element names are case sensitive).

[RFC822](#)-derived header elements are associated with an XML namespace, as noted above at [section 3.3](#), and may need to be combined with a namespace prefix if it is not the default namespace. (See examples in sections [2.2](#) and [2.3](#).)

[RFC822](#)-derived header element contents have the same syntax and meaning as corresponding [RFC822](#) header field values, except that:

- o Characters are not limited to US-ASCII. UTF-8 character set encoding is typically used.
- o Encoded words ('=?...?') are not needed, and no special processing is defined for sequences of this form.

[<draft-klyne-message-rfc822-xml-03.txt>](#)

- o Special considerations apply to fields containing address values (from, to, etc.) -- see section below.
- o Special considerations apply to fields containing human-readable text values (subject, comments, etc.) -- see section below.

[3.7](#) Header fields containing addresses

Parts of an [RFC822](#) address value are separated out into separate elements, all contained within an <Address> element. The element types defined here are <adrs> and <name>.

A major change from [RFC822](#) is that all addresses are presented as URIs, rather than as [RFC822](#) 'addr-spec' values. Email addresses (the only kind that appear in [RFC822](#) headers) are expressed as 'mailto:' URLs [[21](#)]. Address URIs are enclosed in an <adrs> element.

This change anticipates that XML-based message headers may be used with a variety of different protocols with different addressing schemes.

Finally, only one address per message header element is allowed (or an address group: see below). Where permitted, multiple values are represented by repeating the header element for each value.

Note that characters in URIs are drawn from a limited repertoire; the URI '%' escape sequence may be used to represent other characters that are legal for the URI scheme used [[14](#)].

The [RFC822](#) address structures using 'phrase' are supported. The 'phrase' is a "formal name", and is enclosed in a <name> element.

The [RFC822](#) structures using source-route values (i.e. 'route' in 'route-addr') are not supported. [RFC822](#) 'comment' values within addresses are not supported. Thus, [RFC822](#) e-mail addresses that might be expressed as:

Piglet@TrespassersW.100Aker.org (MR SANDERS)

which is generally equivalent to:

MR SANDERS <Piglet@TrespassersW.100Aker.org>

must be presented in the form:

```
<emx:Address>  
  <emx:adrs>mailto:Piglet@TrespassersW.100Aker.org</emx:adrs>
```



```
<emx:name>MR SANDERS</emx:name>
</emx:Address>
```

Any '<', '&' and certain '>' characters appearing in a formal name (<name> element) MUST be represented using '<', '&' or '>' as noted previously in [section 3.4](#).

[3.7.1](#) Header fields containing address groups

Some [RFC822](#) headers can have address group values as well as just address values. The [RFC822](#) 'group' structure associates a collection of addresses with a name for that collection. The individual addresses in a group may be omitted.

An address group is expressed using a <Group> element containing the name of the group and zero, one or more <member> elements each containing an <Address>:

```
<emx:Group>
  <emx:name>Christopher-Robins-friends</emx:name>
  <emx:member>
    <emx:Address>
      <emx:adrs>mailto:Pooh@PoohCorner.100Aker.org</emx:adrs>
      <emx:name>Winnie the Pooh</emx:name>
    </emx:Address>
  </emx:member>
  <emx:member>
    <emx:Address>
      <emx:adrs>mailto:Piglet@TrespassersW.100Aker.org</emx:adrs>
      <emx:name>MR SANDERS</emx:name>
    </emx:Address>
  </emx:member>
  <emx:member>
    <Address>
      <adrs>im:Eeyore@ThistlyCorner.100Aker.org</adrs>
      <name>Eeyore</name>
    </Address>
  </emx:member>
</emx:Group>
```

Omitting the individual member addresses, this would be:

```
<emx:Group>
  <emx:name>Christopher-Robins-friends</emx:name>
</emx:Group>
```


[3.8](#) Header elements containing human readable text

Header fields that contain human readable text MAY have an 'xml:lang=' attribute of the header element to indicate a language for the contained text.

In the absence of such an attribute, any language applicable to the surrounding XML is to be assumed.

[3.9](#) MIME header fields

MIME content header fields MAY be part of the message header, using the same general format and XML namespace as [RFC822](#)-derived header fields (i.e. element name based on the MIME header field name, and associated with the same XML namespace).

But note that most MIME header fields are not appropriate for use with the Email+XML message format. When the message content is supplied as a separate MIME entity then MIME content header fields SHOULD be applied to that entity.

It is expected that MIME header fields may be useful in the following circumstances:

- o When the message content is included as inline XML, to convey information about it that cannot be conveyed using native XML mechanisms; e.g. the Content-features header [[22](#)].
- o MIME headers, not having an obvious XML counterpart, that express information that might be taken as metadata applying to the message as a whole, in isolation from the specific message content; e.g. the Content-description header field.

[3.10](#) Other header fields

A message header MAY contain header fields that are not derived from [RFC822](#) or MIME. Any such header field names used MUST be associated with a different namespace.

This specification does not define any such additional header fields.

3.10.1 Mandatory extensions

In general, a message handler should ignore any header fields that it does not understand.

But sometimes it is desirable to introduce new header fields that must be understood for proper processing of the message to take place. This specification defines an XML attribute 'mustUnderstand=', which indicates whether or not the element to which it applies must be understood by a message processor:

mustUnderstand='false' is the default case, and indicates that the corresponding element MAY safely be ignored.

mustUnderstand='true' indicates that the element to which it applies MUST be processed, OR processing of the entire message (or message header) MUST be abandoned.

In XML namespace terms [[9](#)], the 'mustUnderstand=' attribute belongs to a "per-element-type namespace partition". Interpretation of the attribute is a property of the element to which it applies. In any case, the DTD or XML schema must declare that the element is allowed on any particular XML element type. It is strongly recommended that any header elements used within an Email+XML message header allow this attribute with the interpretation described here.

Non-validating XML processors used to handle Email+XML message headers MAY interpret the 'mustUnderstand=' attribute appearing on any header field element as described here.

Notwithstanding the presence or absence of a 'mustUnderstand=' attribute, individual applications may require that certain header elements are present or absent from any header that they interpret.

4. Summary of [RFC822](#)-derived header elements

[RFC822](#) fields containing a simple address:

- return-path
- from
- sender
- resent-from
- resent-sender

[RFC822](#) fields containing an address or group:

- to
- cc
- bcc
- reply-to
- resent-to
- resent-cc
- resent-bcc
- resent-reply-to

[RFC822](#) fields containing human-readable text:

- keywords
- subject
- comments

Other [RFC822](#) fields:

- received
- date
- resent-date
- message-id
- resent-message-id
- in-reply-to
- references
- encrypted

5. IANA considerations

This specification calls for the registration of the new MIME content-type Message/Email+XML. The registration template is at [appendix A](#).

[[[XML document identifier -- URN from IANA space?]]]

[[[XML namespace identifier -- URN from IANA space?]]]

[[[Waiting on [\[20\]](#)...]]]

6. Internationalization considerations

This specification attempts to relax the restriction of international data imposed by [RFC822](#).

[RFC822](#) limits characters in address local parts to US-ASCII. This specification uses URIs and XML-based address format, relaxing that constraint so that foreign language personal names can be represented. Character restrictions apply to URIs, and the %-escape mechanism defined by [RFC2396](#) must be followed for representing non-URI characters. The character encoding used is dependent on the URI scheme, but UTF-8 is the strongly recommended choice. [[[todo: cite IRI work, and charmod?]]]

Similarly, the characters that can be used in domain names are currently severely constrained. Work is under way to define international forms for domain names.

Message content is tagged using standard MIME capabilities (charset parameter for text data [\[13\]](#), and Content-language header for language tagging [\[22\]](#)). Mandating handling of international data formats is a matter for particular applications; it is recommended that applications using the Email+XML message format be required to process UTF-8 coded character data. That does not necessarily mean that all characters received can be displayed.

For content included in an XML element, language tagging can be achieved by including an 'xml:lang=' attribute [\[16\]](#) in the <Message-content> element (subject to appropriate DTD or XML schema permission to use that attribute).

6.1 International URIs in XML

This sub-section is commentary, not part of this specification:

In a message to the W3C URI mailing list (<http://lists.w3.org/Archives/Public/uri/2000Oct/0008.html>), Martin Duerst wrote:

The original XML spec says (<http://www.w3.org/TR/1998/REC-xml-19980210#sec-external-ent>):

An XML processor should handle a non-ASCII character in a URI by representing the character in UTF-8 as one or more bytes, and then escaping these bytes with the URI escaping mechanism (i.e., by converting each byte to %HH, where HH is the hexadecimal notation of the byte value).

This says that the XML processor should do this for you, and therefore it should be okay for you to put in the original characters. But there are three problems here:

- o It says 'should', not must.
- o It's not clear whether it applies to all URIs, or just to the URIs used in System Identifiers, and in the former case, it's not clear how an XML processor would find all URIs in a document (without e.g. Schema information).
- o The text in the second edition of XML (<http://www.w3.org/TR/REC-xml#sec-external-ent>) is much clearer about how the conversion has to take place; unfortunately, it doesn't make clear who should do this conversion (the original document producer or the XML processor). The idea was not to change this for the second edition, but somehow it got lost. I'm following up on this.

7. Security considerations

This document for the most part describes an alternative coding of an existing message structure, and is not believed to introduce any new security exposure not already inherent in existing systems.

MIME based messages may be protected using existing MIME security frameworks, such as S/MIME [[12](#)], OpenPGP [[13](#)], etc.

Using a non-MIME, pure XML message format means that alternative security frameworks may be applicable, such as XML digital signatures [[14](#)].

Note that this framework is not designed to allow the conversion of message formats (e.g. between [RFC822](#) and XML) while preserving signatures or other security information. If a signature is applied in a MIME body part, and that body part is moved to a message with a different header format, then the signature may be expected to remain intact.

8. Acknowledgements

The author thanks the following for their comments and/or contributions: Harald Alvestrand, Dave Crocker, Simon Josefsson, [[[...]]].

9. References

- [1] Crocker, D.,
"Standard for the format of ARPA Internet text messages",
[RFC 822](#), STD 11,
August 1982.
 - [2] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen,
"Extensible Markup Language (XML) 1.0",
W3C recommendation: <<http://www.w3.org/TR/REC-xml>>,
10 February 1998.
 - [3] Berners-Lee, T., Fielding, R.T. and L. Masinter,
"Uniform Resource Identifiers (URI): Generic Syntax",
[RFC 2396](#),
August 1998.
 - [4] Weider, C., Preston, C., Simonsen, K., Alvestrand, H., Atkinson,
R., Crispin, M., Svanberg, P.,
"Report from the IAB Character Set Workshop",
[RFC 2130](#),
April 1997.
- Alvestrand, H,
"IETF Policy on Character Sets and Languages",
[RFC 2277](#), [BCP 18](#),
January 1998.

<[draft-klyne-message-rfc822-xml-03.txt](#)>

Freed, N., and J. Postel,
"IANA Charset Registration Procedures",
[BCP 19](#), [RFC 2278](#),
January 1998.

[[[Is there a more definitive reference?]]]

- [5] Alvestrand, H.,
"Tags for the Identification of Languages",
[RFC 1766](#),
March 1995.
(Defines Content-language header.)
- [6] Freed, N. and N. Borenstein,
"Multipurpose Internet Mail Extensions (MIME) Part One: Format of
Internet Message Bodies",
[RFC 2045](#),
November 1996.
- [7] Freed, N. and N. Borenstein,
"Multipurpose Internet Mail Extensions (MIME) Part Two: Media
Types",
[RFC 2046](#)
November 1996.
- [8] Freed, N., Klensin, J., and J. Postel,
"Multipurpose Internet Mail Extensions (MIME) Part Four:
Registration Procedures",
[RFC 2048](#), [BCP 13](#),
November 1996.
- [9] Tim Bray, Dave Hollander, and Andrew Layman
"Namespaces in XML",
W3C recommendation: <<http://www.w3.org/TR/REC-xml-names>>,
14 January 1999.
- [10] Lassila, O. and R. Swick,
"Resource Description Framework (RDF) Model and Syntax
Specification",
W3C recommendation: <<http://www.w3.org/TR/REC-rdf-syntax>>,
22 February 1999.
- [11] Kohn, D., Murata, M. and S. St.Laurent,
"XML Media Types",
[draft-murata-xml-09.txt](#),
September 2000.
(Introduces '+XML' content-type naming convention.)

[<draft-klyne-message-rfc822-xml-03.txt>](#)

- [12] Ramsdell, B.,
"S/MIME Version 3 Message Specification",
[RFC 2633](#),
June 1999.
- [13] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer,
"OpenPGP Message Format",
[RFC 2440](#),
November 1998.
- [14] Eastlake, D., Reagle, J. and D. Solo,
"XML-Signature Syntax and Processing",
Work in progress: [<draft-ietf-xmldsig-core-09.txt>](#),
August 2000.
- [15] Levinson, E.,
"The MIME Multipart/Related Content-type",
[RFC 2387](#),
August 1998.
- [16] Levinson, E.,
"Content-ID and Message-ID Uniform Resource Locators",
[RFC 2392](#),
August 1998.
- [17] Daniel, R., DeRose, S. and E. Maler
"XML Pointer Language (XPointer) Version 1.0",
W3C Candidate Recommendation: [<http://www.w3.org/TR/xptr>](#)
7 June 2000.
- [18] Fallside, D.,
"XML Schema Part 0: Primer",
W3C Working Draft: [<http://www.w3.org/TR/xmlschema-0/>](#),
22 September 2000.
- Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn
"XML Schema Part 1: Structures",
W3C Working Draft: [<http://www.w3.org/TR/xmlschema-1/>](#)
22 September 2000.
- Biron, P. and A. Malhotra,
"XML Schema Part 2: Datatypes",
W3C Working Draft: [<http://www.w3.org/TR/xmlschema-2/>](#)
22 September 2000.

<[draft-klyne-message-rfc822-xml-03.txt](#)>

- [19] Bradner, S.,
"Key words for use in RFCs to Indicate Requirement Levels",
[RFC 2119](#),
March 1997.

- [20] Mealling, M., Masinter, L., Hardie, T., and G. Klyne,
"A URN Sub-namespace for Registered Protocol Parameters",
[draft-mealling-iana-urn-01.txt](#) (work in progress),
August 2001.

- [21] Hoffman, P., Masinter, L., and J. Zawinski,
"The mailto URL scheme",
[RFC 2368](#),
July 1998.

- [22] Klyne, G.,
"Indicating Media Features for MIME Content",
[RFC 2912](#),
September 2000.

- [23] Brickley, D. and R. V. Guha,
"Resource Description Framework (RDF) Schema Specification",
W3C recommendation: <<http://www.w3.org/TR/PR-rdf-schema>>,
27 March 2000.

- [24] Mealling, M.,
"The IETF XML Registry",
[draft-mealling-iana-xmlns-registry-02.txt](#) (work in progress),
June 2001.

10. Author's address

Graham Klyne
MIMESweeper Group
Clearswift Corporation
1310 Waterside
Arlington Business Park
Theale
Reading, RG7 4SA
United Kingdom

Telephone: +44 11 8903 8903
E-mail: Graham.Klyne@MIMESweeper.com

Appendix A: Message/Email+XML content-type registration

[[[TBD]]]

Appendix B: DTD for Email+XML message format

[[[TBD]]]

Appendix C: XML schema for Email+XML message format

[[[TBD]]]

Appendix D: RDF representation of Email+XML message

The message header format described here is designed to be compatible with RDF [[10](#)]. To prepare a message header for presentation to an RDF processor, it should be enclosed in an <rdf:RDF> element having an appropriate RDF namespace declaration.

In RDF terms, the message header is a resource, having a property arc for each header element and also one for the message content.

Here is an informal representation of the RDF graph corresponding to the message example from [section 2.3](#):

```
[<Message>]
|
|--rfc822:from--> [<Address>]
|                   |
|                   -----
|                   |
|                   +--adrs-->"im:Eeyore@ThistlyCorner.100Aker.org"
|                   +--name-->"Eeyore"
|
+--rfc822:to-----> [<Group>]
|                   |
|                   +--name--> "Anyone"
|
+--rfc822:subject--> "Why?"
|
+--content--> "Wherefore?
              Inasmuch as which?"
```

There is a subtle difference in the RDF form of a message with inline content and one that references a separate content object:

<[draft-klyne-message-rfc822-xml-03.txt](#)>

both have a 'content' property whose value is a resource; if the content is defined externally, the value of the 'content' property is an RDF resource containing the content; when the content is inline, the property value is an RDF literal.

If inline message content contains XML markup, to ensure complete RDF compatibility the 'content' element should have a property 'parseType="Literal"', to prevent the RDF processor from trying to interpret the content as RDF.

Appendix E: RDF schema for Email+XML message format

[[[TBD]]]

Appendix F: Amendment history

- 00a 13-Oct-2000 Memo initially created.
- 00b 16-Oct-2000 Add reference to XML spec note about non-ASCII text in a URI.
- 00c 18-Oct-2000 Change [RFC822](#)|XML to [RFC822](#)+XML (per later XML-MIME spec).
- 01a 04-Jan-2001 Change draft title and message format name. Indicate that this is not an exact coding of [RFC822](#) messages, but an attempt to capture their essential semantics. Change syntax of address elements to be RDF compliant.
- 01b 10-Jan-2001 Add [RFC822](#) group structure to address format. Distinguish between headers that allow group values and those that allow simple addresses. Use separate namespaces for message structure and headers derived from [RFC822](#). Add brief discussion of RDF compatibility.
- 01c 12-Jan-2001 Add discussion list details.
- 02a 19-Jan-2001 Add clarification to security considerations that message signatures are not generally expected to survive any message format conversion.
- 02b 10-Sep-2001 Update contact details. Update proposed namespace names in line with [\[20\]](#). Update proposed DTD name, per [\[24\]](#) (new reference).

03a 09-Apr-2002 Update contact details. Change name of
'seeNoEvil' attribute to 'mustUnderstand'.

Appendix G: Outstanding issues

- o Review namespace URIs.
- o Review MIME type name. (Message/XML? Application/Message+XML?)
- o Allow more flexible use of RDF syntax to reduce verbosity (but increase number of different ways of expressing some constructs in XML; e.g. adrs and name attributes for <Address>)?
- o Clarify effect of namespaces (or not) on element attribute names. XML attributes do not follow the same default namespace rules as elements.
- o Define DTD, XML schema and RDF schema.
- o Finalize IANA considerations.

Full copyright statement

Copyright (C) The Internet Society 2001. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

XML coding of [RFC822](#) messages

9 April 2002

[<draft-klyne-message-rfc822-xml-03.txt>](#)

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

