MAIL ACCOUNT XML SCHEMA

How Internet Messages can be stored as XML

David Minor
North Carolina State Archives

Mail Account Schema

- This XML schema specifies the structure and organization of an XML file that stores a set of e-mail messages in a hierarchical set of folders. Usually each XML file will store all of the e-mail messages belonging to a single e-mail account.

- Any product that needs to store e-mail, such as e-mail servers and e-mail clients, could use XML files of this type as the centerpiece of its storage service.

Accessible

- E-mail messages are transmitted using the Internet Message Format. This standard was developed before the advent of XML and is difficult to use.
- Once the message has been converted to XML, each data item receives its own easily addressable element. No longer are complicated parsing routines required to extract data items such as the list of recipients, the date the message was sent, etc.

Accurate

- During the conversion to XML, many transformations are required. Although we have high degree of confidence that the transformations are faithful, we can never be 100% sure.
- For those requiring that the exact original information be provided and in case some day a better or different parser is built, the original message is kept.

Mbox files.

- The original messages are kept in text files external to the main XML file. All messages belonging to a particular folder are kept in a single mbox file.
- The mbox file format is used by some of the early UNIX mail servers, and some e-mail clients for local storage such as Eudora, Firefox, and Outlook Express.

Attachments

- Originally e-mail could only be used to send text messages. In June 1992 the ability to attach binary files was introduced. This allowed people to send, images and word documents, for example.
- Since the infrastructure used to transmit e-mail messages was designed to only accommodate text messages and ASCII text only required 7 bits, the designers being ever so resourceful decided to use the eighth bit of for transmission control purposes.

MIME June 1992

IETF Network Working Group Request for Comments: 1341
Multipurpose Internet Mail Extensions
N. Borenstein, (Bellcore) N. Freed, (Innosoft)

SMTP August 1982

IETF Network Working Group Request for Comments: 821
SIMPLE MAIL TRANSFER PROTOCOL
Jonathan B. Postel
Information Sciences Institute
University of Southern California

Binary Attachments

- Binary attachments and modern text encodings use all 8 bits.
- Attachments are transformed into 7-bit only character strings using the BinHex encoding. This allows them to be included in e-mail messages without breaking the 7-bit rule.
- When the attachment is opened, this encoding is reversed producing the original byte stream.

Unicode

- XML uses Unicode for character encoding.
- Unicode provides a single code page for every alphabet. Instead of reusing the same 7 or 8 bits for different languages differently and then requiring some mechanism to declare which encoding is being used, Unicode uses 21 bits and every character from every language is assigned a unique value.

Unicode characters are defined from 1 – 10FFFF – This range requires only 21 bits.
Reference: http://unicode.org/reports/tr19/tr19-9.html
Reference: http://www.w3.org/TR/2006/REC-xml11-20060816/#charsets

Binary Payloads in XML

- XML cannot directly include raw binary data, since binary data may use any 8-bit value and some code points, most notably 0, is not allowed.
- We have two choices:
  - Transform the binary data to BinHex.
  - Save the attachment as an external file, and save a link to each file.
- The XML Schema allows for both.

Text Payload in XML

- The character encoding used for the message's body and each text attachment are specified in the e-mail message. This specification is required to properly display the correct letters. We must know which glyph belongs to which code point.
- We must be careful when placing these text elements into the XML file.
- We have several choices…

Convert text to Unicode

- Convert the text to Unicode, and store the content within the XML file as character data. When the XML file is viewed in a standard viewer the correct characters will be displayed.
- The consumer of the XML document would not have to pay attention to the original character encoding, except possibly in those cases where the text were to be used in the original application; an application that doesn't support Unicode.

Keep the original encoding

- Keep the original character encoding. And indicate which character encoding is being used.
- The consumer of the XML file will then be responsible for using the correct character encoding as indicated.
- It is possible for some character encodings to produce byte sequences that are not allowed in Unicode. For western hemisphere, single-byte character sets this should never happen, for double-byte character sets that have no affinity with ASCII this is a real possibility.

Reference: http://www.unicode.org/versions/Unicode5.1.0/#Notable_Changes
For a discussion on invalid characters.

Store the text in BinHex

- The XML Schema supports storing binary data encoded in BinHex directly in the XML document.
- Using this facility, we could simply encode the original string using BinHex.
- The consumer would have to convert it back from BinHex and then pay attention to the specified character encoding.

- The Mail Account XML Schema allows for first two options and should probably be modified to allow the third.

Basic Steps

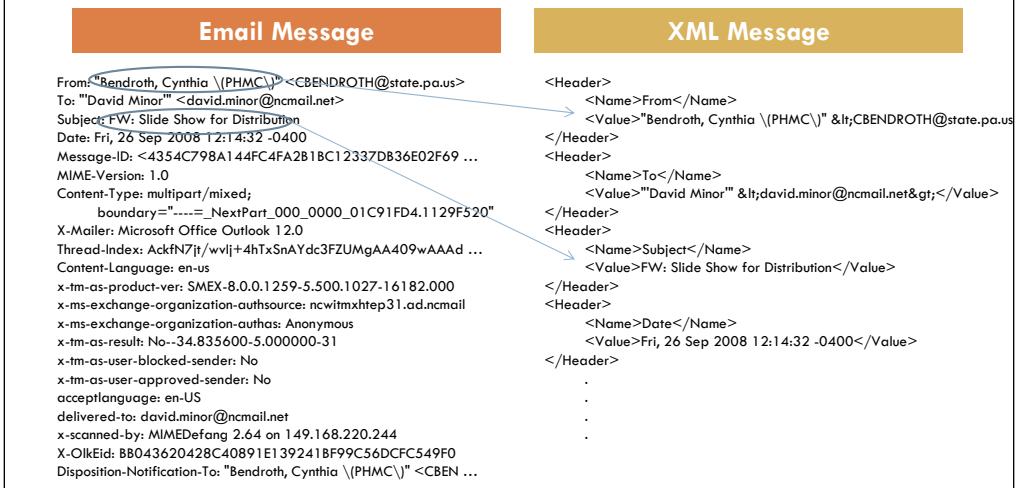
Here are the basic steps that any parse must perform.

Unparsed Headers

- Every header is copied to the XML file without any transformation.
- The name of the header is recorded in an element named “Name,” and the value of the header is recorded in an element named “Value.”

Unparsed Headers (Example)

All message headers are recorded in “plain” name, value pairs.



Parsed Headers

- In addition to storing every header "as is" the following headers, if present, are parsed into named XML elements:

Message-ID	To	References
Date	Cc	Subject
From	Bcc	Comments
Sender	InReplyTo	Keywords

  - Headers that have multiple values, produce one element per value.
  - The Date header is written out as an XML DateTime value.

Parsed Header (Example)

This message had only 3 Named Headers.

Email Message

From: "Bendroth, Cynthia \(PHMC\)" <CBENDROTH@state.pa.us>
To: "'David Minor'" <david.minor@ncmail.net>;groucho.marx@this.net
Subject: FW: Slide Show for Distribution
Date: Fri, 26 Sep 2008 12:14:32 -0400
Message-ID: <4354C798A144FC4FA2B1BC12337DB36E02F69 …
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="----=_NextPart_000_0000_01C91FD4.1129F520"
X-Mailer: Microsoft Office Outlook 12.0
Thread-Index: AckfN7jt/wvlj+4hTxSnAYdc3FZUMgAA409wAAAd …
Content-Language: en-us
x-tm-as-product-ver: SMEX-8.0.0.1259-5.500.1027-16182.000
x-ms-exchange-organization-authsource: ncwitmxhtep31.ad.ncmail
x-ms-exchange-organization-authas: Anonymous
x-tm-as-result: No--34.835600-5.000000-31
x-tm-as-user-blocked-sender: No
x-tm-as-user-approved-sender: No
acceptlanguage: en-US
delivered-to: david.minor@ncmail.net
x-scanned-by: MIMEDefang 2.64 on 149.168.220.244
X-OlkEid: BB043620428C40891E139241BF99C56DCFC549F0
Disposition-Notification-To: "Bendroth, Cynthia \(PHMC\)" <CBEN …

XML Message

.
.
.
<MessageId>
<4354C798A144FC4FA2B1BC12337DB36E02F69…
</MessageId>
<OrigDate>2008-09-26T12:14:32-04:00</OrigDate>
<From>
"Bendroth, Cynthia \(PHMC\)" <CBENDROTH@state.pa.us>
</From>
<To>"'David Minor'" <david.minor@ncmail.net></To>
<To>groucho.marx@this.net</To>
<Subject>FW: Slide Show for Distribution</Subject>
.
.
.

Attachments

Email Message

```
------=_NextPart_000_0000_01C91FD4.1129F520
Content-Type: multipart/alternative;
                boundary="----
=_NextPart_001_0001_01C91FD4.1129F520“

------=_NextPart_001_0001_01C91FD4.1129F520
Content-Type: text/plain;
                charset="us-ascii"
Content-Transfer-Encoding: 7bit

Hi David-
This is one of the messages we tried putting into hmail and it did
not
have the attachment.
Cindy
-----Original Message-----
From:    McKenzie, Kathleen R (GC)
Sent:   Thursday, September 25, 2008 2:45 PM
To:     Akers, Rodney (GC); Sanders, Jeffry, Ph.D.; Guistwite, Kevin
P;
Haynes, Arwilda; Weis, Shawn; Longwell, Scott; Keeler, Catherine;
```

XML Message

```
<MultiBody>
    <ContentType>multipart/mixed</ContentType>
        <BoundaryString>----=_NextPart_ ….</BoundaryString>
            <MultiBody>
                <ContentType>multipart/alternative</ContentType>
                    <BoundaryString>----=_Nex…. </BoundaryString>
                    <SingleBody>
                        <ContentType>text/plain</ContentType>
                        <Charset>us-ascii</Charset>
                        <TransferEncoding>7bit</TransferEncoding>
                        <BodyContent>
                            <Content>
Hi David-
This is one of the messages we tried putting into hmail and it did not
have the attachment.
Cindy
-----Original Message-----
From:    McKenzie, Kathleen R (GC)
Sent:   Thursday, September 25, 2008 2:45 PM
To:     Akers, Rodney (GC); Sanders, Jeffry, Ph.D.; Guistwite, Kevin P;
Haynes, Arwilda; Weis, Shawn; Longwell, Scott; Keeler, Catherine;
```