Network Working Group Jacob Palme
Internet Draft Stockholm University/KTH
draft-ietf-mhtml-info-0~~0~~1.txt
Category-to-be: Informational
Expires: ~~November 1996~~ ~~May 1996~~
January 1997 July 1996

Sending HTML in E-mail, an informational supplement to RFC ???:
MIME E-mail Encapsulation of Aggregate HTML Documents (MHTML)

Status of this Memo

Abstract

The memo "MIME E-mail Encapsulation of Aggregate HTML Documents (MHTML)" (draft-ietf-mhtml-spec-00.txt) specifies how to send packaged aggregate HTML objects in MIME e-mail. This memo is an accompanying informational document, intended to be an aid to developers. This document is not an Internet standard.

Issues discussed are implementation methods, caching strategies, making messages suitable both for mailers which can and which cannot handle Multipart/related and handling recipients which do not have full Internet connectivity.

Table of Contents

Mailing List Information

Further discussion on this ~~memo~~document should be done through the mailing list MHTML@SEGATE.SUNET.SE.

To subscribe to this list, send a message to
   LISTSERV@SEGATE.SUNET.SE
which contains the text
SUB MHTML <your name (not your e-mail address)>

Archives of this list are available by anonymous ftp from
   FTP://SEGATE.SUNET.SE/lists/m~~html~~HTML/
The archives are also available by e-mail. Send a message to LISTSERV@SEGATE.SUNET.SE with the text "INDEX MHTML" to get a list of the archive files, and then a new message "GET <file name>" to retrieve the archive files.
~~or~~
~~GET MHTML DIGEST~~
Comments on less important details may also be sent to the editor, Jacob Palme <jpalme@dsv.su.se>.

More information may also be available at URL:
HTTP://www.dsv.su.se/~jpalme/ietf/jp-ietf-home.HTML

1. Introduction

[MHTML] specifies how to send packaged aggregate HTML objects in MIME e-mail. This memo is an accompanying informational document, intended to be an aid to developers. This document is not an Internet standard.

2. Implementation methods

The [MHTML] standard has been intentionally written to be implementable both in cases where the web browser and e-mail program is combined, and when they are separate programs. Implementation is of course no problem if the web browser is combined with the e-mail client.

2.1 Method 1: Combining web browser and e-mail program

This is the architecturally simplest approach. A web-browser with a built in e-mail program will be able to use its own web browser capa~~i~~bilities to display HTML-formatted messages.

2.2 Method 2: Rewriting the HTML

```
+---------+                                  +--------+
| Web     |                                  | Mail   |
| browser |                                  | client |
+-------+-+                                  +-+------+
        |                                      |
     +--+--------------------------------+--+
     | +----------+  +--+  +--+              |
     | | Start    |  |  |  |  | Related      |
     | | HTML     |  |  |  |  | body part    |
     | | document |  |  |  |  | parts        |
     | +----------+  +--+  +--+              |
     +---------------------------------------+
```

Figure 1

If the web browser is separate from the e-mail client, the e-mail client might turn over the HTML body part to the web browser and ask it to display it (Figure 1). One way of doing this is to store the HTML body part in a file, and ask the web browser to display this file. If multipart/related is used, this can be implemented by storing all the body parts within the multipart/related in an otherwise empty folder/directory.

The mail client may have to rewrite the HTML, replacing URL-s with shorter relative URL-s which the Web browser can resolve as file names in the same directory/folder where the HTML document itself is stored when turning it over to the Web browser.

2.3 Method 3: Using a translation table

```
+----------+                                   +---------+
| Web      |                                   | Mail    |
| browser  |                                   | client  |
+--------+-+                                   +-+-------+
         |                                       |
     +---+-------------------------------+---+
     | +---------+  +---+  +---+                 |
     | | Trans-  |  |   |  |   | Related         |       Figure 2
     | | lation  |  |   |  |   | body part       |
     | | table   |  |   |  |   | parts           |
     | +---------+  +---+  +---+                 |
     +-------------------------------------+
```

An alternative to rewriting the HTML file before turning it over to the Web browser may be to use a translation table, in case the Web browser has the capability to use such a table to rewrite URL-s on the fly while displaying the document (Figure 2). This requires that the Web browser is capable of receiving CID: URL-s and resolving them using this translation table in the same way as for other URL-s.

2.4 Method 4: Using a proxy HTTP server do retrieve referenced body parts

```
+---------+        +------------+       +---------+
| Proxy   |        | Data base  |       | Mail    |
| web     |--------| of cached  |-------| server  |
| server  |        | objects    |       |         |
+----+----+        +------------+       +----+----+
     |                                       |
+----+-----+                             +----+----+    Figure 3
| Web      |                             | Mail    |
| browser  |                             | client  |
+--------+-+                             +-+-------+
         |                                 |
     +---+-------------------------------+---+
     |          Start HTML object            |
     +-------------------------------------+
```

Yet another method is to use a proxy web server, to which the web browser requests are sent, and which will then use the cached body parts instead of normal web retrieval from the network (Figure 3). If the Web browser is set to use this proxy server for all URL-s, also for CID URL-s, no rewriting of the HTML will be necessary.

2.5 Method 5: Putting the mail client into a proxy HTTP server

```
+---------+--------+
| Proxy   | Mail   |
|  HTTP   | client |
| server  |        |
+---------+--------+
          |
   HTTP protocol              Figure 4
          |
    +----+----+
    | Web     |
    | browser |
    +---------+
```

A mail client can also be included in an HTTP server (Figure 4). The user will then not have to install any mail client software in his personal computer, all the mail functionality is mapped on HTTP and HTML elements.

2.6 Other methods

The mail client and the web browser can of course communicate in other ways, such as using inter-process communication.

2.7 Communication between web browser mail client

Many web browsers have API-s to allow other programs to communicate with them. There is however no accepted real or de-facto standard for such API-s, which means that a mail program which relies on such API-s will only be able to use those Web browser, whose API they support.

Note however, that most of the methods described above can be implemented with a very minimal such API. The only API function needed is to be able to tell a Web browser, when it is started, to open a particular file. And this API function is a standardi~~s~~zed part of the operating system on most platforms. In particular, method 1 and 3 above uses the functionality that a relative URL is resolved with the location of the base document as base. This means that if the base document is a file, relative URL-s will be resolved as FILE URL-s in the same directory/folder where the HTML document itself is placed.

There is a need for buttons in the Web page which the user can use to get back to the mail program again after reading the mail with the Web browser. A common technique to achieve this is to define a new MIME data type for this button. The Web browser is then configured to transfer control to the mail client when the user pushes this button, i.e. downloads a file of this new MIME type.

3. Caching of body parts

Suppose a message contains body parts with the Content-Location header as defined in [MHTML]. A receiving agent might then put this body part into a web cache, with the URL in the Content-Location as its name, so that later retrievals of this URL use the cached body parts. There is however no guarantee that such a cached item is correct. Such caching is thus not recommended for use in other ways than for resolution of links within this e-mail message.

4. Recipients which cannot handle the Multipart/related Content-Type

A message sent according to the specifications in [MHTML] may have recipients, whose mailers cannot handle the Multipart/related Content-Type in the way specified in [MHTML].

According to [MIME1] a mailer which encounters an unknown subtype to Multipart, should handle this as Multipart/mixed.

To improve this, Multipart/alternative can be used as ~~specified in section 8~~discussed in section 5 of this memo.

Content-Disposition, as specified in [CONDISP] and in [MHTML], section 10, can also be used as an aid to mailers which do not understand Multipart/related.

5. Use of the Content-Type: Multipart/alternative

If the message is sent to recipients, all of which may not have mailers capable of handling the Text/HTML content-type, then the Content-Type: Multipart/Alternative [MIME1] can be used in two ways:

~~used, for example~~(a) Inside the Content-Type Multipart/related, body parts can be specified with Content-Type: Text/plain as the first choice, and~~ Content- ~~ Content-Type: Text/HTML as the second choice.

~~Both the Content type: Multipart/related, as defined in chapter 6 above and the Content-Type: Multipart/alternative, as defined in chapter 7 above can be combined in the same message. It is then recommended to put the Multipart/alternative inside the Multipart/related.~~

Example:

```
   Content-Type: Multipart/related; boundary="boundary-example-1";
                 type=MULTIPART/ALTERNATIVE

      --boundary-example 1
      Content-Type: MULTIPART/ALTERNATIVE
      Boundary: boundary-example-2

         --boundary-example-2
         Content-Type: Text/plain

         ... plain text version of the document for recipients
         whose mailers cannot handle Text/HTML ...

         --boundary-example-2
         Content-Type: Text/HTML; charset=US-ASCII
         Content-ID: content-id-example@example.host
```

```
         ... text of the HTML document ...

         --boundary-example-2--
      --boundary-example-1
      Content-Type: Image/GIF

      ... a body part, to which the HTML document has a link  ...
      --boundary-example-1--
```

Note that the type parameter of Multipart/related in this case should be Multipart/alternative and not Text/HTML.

~~There are two reasons for the above recommendation:~~

~~(a) All related objects will be inside the multipart/related, as specified in [MHTML] and [REL].~~(b) Outside the Multipart/Related, with Multipart/Related as one alternative and Multipart/Mixed as the other alternative.

When choosing between these two methods of employing multipart/alternative, note the following:

(~~b~~1) Clients which do not support Multipart/related, and which thus will interpret it as multipart/mixed, will ~~display the inline objects.~~ with choice (a) display the inline objects.Thus, a recipient whose mailer can handle Image/gif~~ but not~~

but not multipart/related will still be shown the ~~GIF pictures, they will~~ images, they will not be suppressed by being inside a suppressed branch of the Multipart/alternative.

(2) Choice (b) will not show inline images in the Multipart/Related, unless this information is repeated in both branches of the Multipart/Alternative.

A general warning: Many mailers do not support Content-Type: Multipart/alternative, and may then interpret it as Multipart/mixed.

6. Recipient may not have full Internet connectivity

The recipient of a message sent by e-mail may not always have full Internet connectivity. The recipient may be behind a gateway or firewall which prohibits or restricts Internet connectivity.

This means that the recipient may not be able to resolve URL-s in an e-mail message, unless the referred-to documents are included in the e-mail message itself. Thus, it is often suitable to include in an e-mail message all documents which are referred to (directly or indirectly) by URL-s in the message. This may of course not always be possible, in some cases the set of referred-to documents (directly or indirectly) may be the whole WWW document space, i.e. millions of documents. A choice must then be made how much to include. Of course, it is most important to include all inline objects, i.e. objects linked by such hyperlinks as IMG, etc., which specify that the linked objects are to be shown to the user immediately.

In the case of ACTION elements in HTML forms, by making these ACTION elements of the "mailto:" URL type, rather than the "http:" URL type, you will enable also recipients without full Internet connectivity to fill in and send in your forms.
The HTML specification [HTML2] allows default action when no ACTION element is included, but this default action may not be suitable when sending the HTML document via e-mail. Thus, it is better to always put an explicit ACTION element into HTML forms sent by e-mail.

7. Encoding of non-ascii characters

Some implementations may have a choice of whether to represent non-ascii characters at the HTML layer (using "&" entity references or numeric character references as defined in [HTML2] section 3.2.1) or at the MIME layer (using Content-Transfer-Encoding as defined in [MIME1] section 5).

In choosing between these two representation methods, note the following effects:

(1) Modifying HTML markup may disrupt security content integrity checks.

(2) The choice of modifying HTML markup may be more suitable for recipients whose mailers do not support MIME.

(3) Using MIME Content-Transfer-Encoding may be more suitable for recipients who have MIME-compliant mailers but do pass the text over to a web browser.

8. Representation of line breaks

Implementors should note that while HTML allows bare CRs, bare LFs and CRLFs as line breaks, MIME requires that all line breaks must be CRLF. If you want the same HTML text to be returned after decoding that was sent before encoding, then you can use BASE64 or Quoted-Printable encoding of bare CRs and bare LFs.

~~7~~9. Acknowledgments

Harald Tveit Alvestrand, Richard Baker, Dave Crocker, Martin J. Duerst, Roy Fielding, Al Gilman, Paul Hoffman, Alexander Hopmann, Mark K. Joseph, Greg Herlihy, Valdis Kletnieks, Daniel LaLiberte, Ed Levinson, Jay Levitt, Albert Lunde, Larry Masinter, Keith Moore, Gavin Nicol, Pete

Resnick, Jon Smirl, Einar Stefferud, Jamie ~~S~~Zawinski and several other people have helped us with preparing this memo. I alone take responsibility for any errors which may still be in the memo.

~~8~~10. References

Temporary note: This list contains some references to Internet drafts. It is anticipated that these Internet drafts will become RFC-s before this memo. The references will then in this memo be changed to refer to the corresponding RFC instead.

Ref.	Author, title
[CONDISP]	R. Troost, S. Dorner: "Communicating Presentation Information in Internet Messages: The Content-Disposition Header", RFC 1806, June 1995.
[HOSTS]	R. Braden (editor): "Requirements for Internet Hosts -- Application and Support", STD-3, RFC 1123, October 1989.
[HTML2]	T. Berners-Lee, D. Connolly: "Hypertext Markup Language - 2.0", RFC 1866, November 1995.
[HTTP]	T. Berners-Lee, R. Fielding, H. Frystyk: "Hypertext Transfer Protocol -- HTTP/1.0", <draft-ietf-http-v10-spec-04.txt>, April 1996.
[MHTML]	J. Palme & A. Hopmann: "Packaging Aggregate HTML Objects in MIME E-mail", <draft-ietf-mhtml-spec-00.txt>, April 1996.
[MIDCID]	E. Levinson: "Message/External-Body Content-ID Access Type", RFC 1873, December 1995.
[MIME1]	N. Borenstein & N. Freed: "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Sept 1993.
[MIME2]	N. Borenstein & N. Freed: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types". draft-ietf-822ext-mime-imt-02.txt, December 1995.
[NEWS]	M.R. Horton, R. Adams: "Standard for interchange of USENET messages", RFC 1036, December 1987.
[REL]	Harald Tveit Alvestrand, Edward Levinson: "The MIME Multipart/Related Content-type", <draft-levinson-multipart-related-00.txt>, January 1995.
[RELURL]	R. Fielding: "Relative Uniform Resource Locators", RFC 1808, June 1995.

[RFC822] D. Crocker: "Standard for the format of ARPA Internet text messages." STD 11, RFC 822, August 1982.

[SMTP] J. Postel: "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[URL] T. Berners-Lee, L. Masinter, M. McCahill: "Uniform Resource Locators (URL)", RFC 1738, December 1994.

[URLBODY] N. Freed and Keith Moore: "Definition of the URL MIME External-Body Access-Type", draft-ietf-mailext-acc-url-01.txt, November 1995.

~~9~~11. Author's Address

Jacob Palme
Stockholm University and KTH
Electrum 230
S-164 40 Kista, Sweden

Phone: +46-8-16 16 67
Fax: +46-8-783 08 29
E-mail: jpalme@dsv.su.se

Working group chairman:

Einar Stefferud <stef@nma.com>