Network Working Group
Draft
Category: Informational
draft-ietf-hekkala-backhans-mhtml

Yvonne Backhans
Tina Hekkala
Stockholm University/KTH
February 2004 Expires August 2004

Examining, implementing and testing of RFC2557 (MHTML)

Status of this Document

This document provides information for the Internet community. This document does not specify an Internet standard of any kind. Distribution of this document is unlimited.

Abstract

In order to send a web page with all or some referenced resources in an e-mail message, the web page and its resources need to be aggregated in a MIME formatted structure.
The receiver of such a message need to know how to unpack the structure to display the web page as an email message. The standard RFC2557, MIME Encapsulation of aggregate documents, such as HTML (MHTML) specifies methods for achieving this.

The purpose of this document[1] is to examine RFC2557 and implement an e-mail client that sends MHTML using the Content-Location MIME header field, specified in RFC2557, for referencing resources.
The e-mail client has been used to send MHTML messages to five commercial e-mail clients to see if they can display such messages.

The conclusions drawn from our tests show that all, except one, of the tested e-mail clients can correctly display the simplest form of MHTML messages using Content-Location.

This document can be downloaded in plain text, Microsoft Word and PDF formats from http://dsv.su.se/jpalme/ietf/mhtml.html#testprogs. The PDF version is a little more neatly formatted than the plan text version, but the content is the same.

[1] This is an abbreviated translation of a thesis submitted to Stockholm University in partial fulfillment of the requirements for the degree of Master of Science in Computer and Systems Sciences.

Table of Contents

1. Introduction

MHTML (as specified in RFC2557) was developed in order to facilitate sending HTML or other multi-resource documents in e-mail (via SMTP). MHTML is a way of aggregating a multi-resource document in one single file by embedding the files that make up the multi-resource document in a MIME multipart/related structure. This format may also be used for archiving multi-resource documents or retrieving such documents via protocols other than SMTP (for example HTTP or FTP).

The purpose of this report is to examine RFC2557 and implement an e-mail client (called MHTMLMailer) that sends MHTML using the Content-Location MIME header field, specified in RFC2557, for referencing resources. The mailer has then been used to send MHTML messages to five commercial e-mail clients to see how well they can display such messages.

The goal was to develop a mailer that is unconditionally compliant with RFC2557 and that our work would aid IETF in their work to revalue the status of RFC2557 and examine whether the MHTML standard can be elevated from the proposed standard level to the draft standard level in the Internet Standards Track.

2. MHTMLMailer – an implementation of RFC2557

2.1 Overview

The mailer that was implemented using JavaMail API, for the purpose of sending MHTML using a Content-Location header, consists of two Java classes. The classes are called MHTMLCreator and MHTMLSender. MhtmlCreator takes the HTML source code of the web page, looks for referenced objects such as images and style sheets, retrieves them, and creates body parts of all objects. This is achieved by creating instances of the JavaMail class MimeBodyPart. An instance of MhtmlSender is then created which assembles the MimeBodyParts into an e-mail message, having the media type multipart/related, and sends it.

2.2 The structure of a MHTML message

We use the terms **MHTML message** and **multipart/related structure** as synonyms for a MIME-encoded multi-resource document.

Figures 2.1 and 2.2 show the logical and real structure of a MHTML message created by our mailer, MHTMLMailer. Figure 2.1 does not show all the headers in the MIME parts but focuses on the relations between the different parts by marking the references in bold type. Figure 2.2 shows what the MHTML message looks like as plain text.

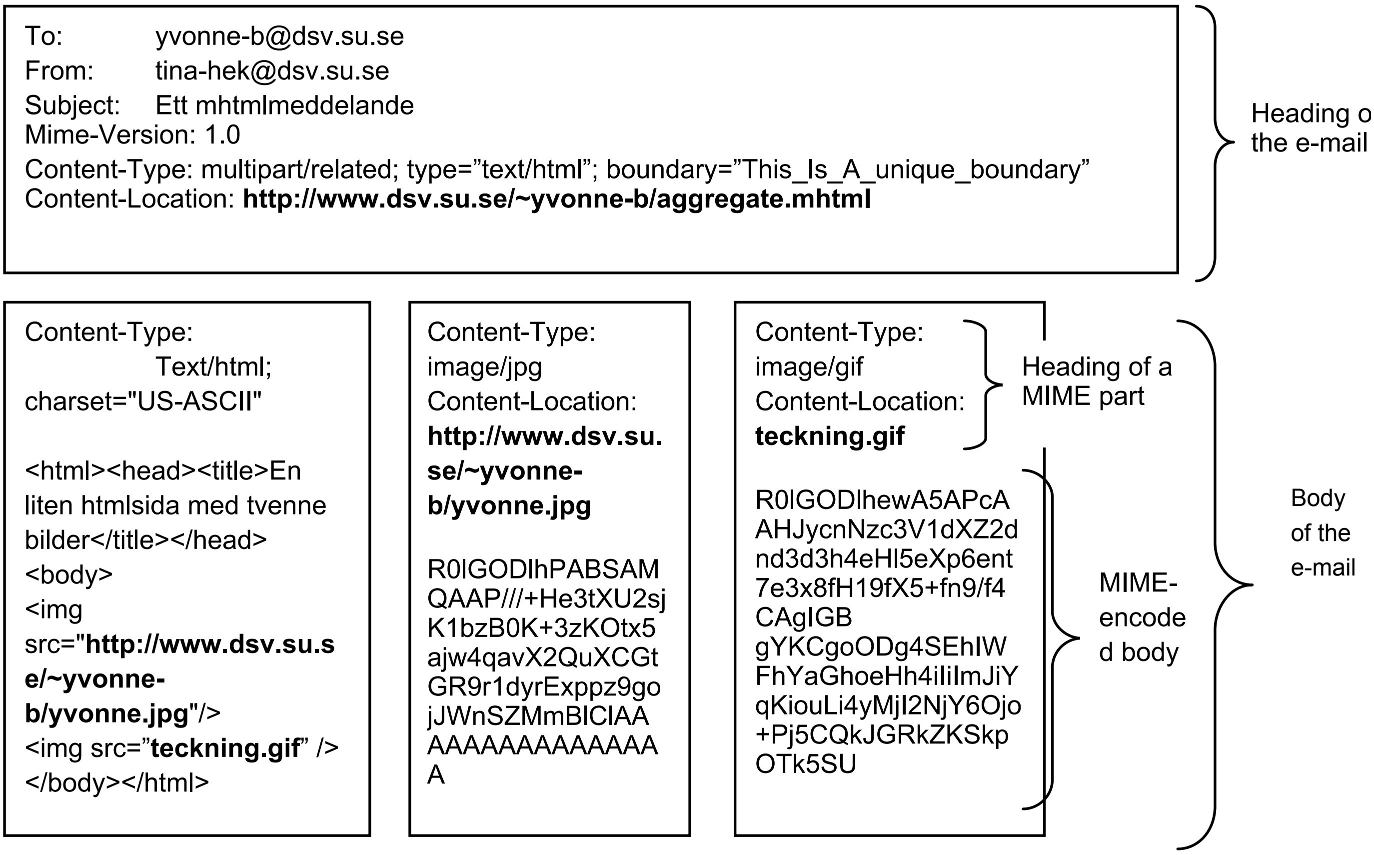

Figure 2.1

The message in this example is made up of three body parts: an HTML file, a jpg image and a gif image. The HTML file and the two referenced image files are embedded in a structure with the media type multipart/related. The media type is shown in the Content-Type header field in the heading of the e-mail. Apart from the value of the field being multipart/related the Content-Type header field also has two parameters, type and boundary. The type parameter specifies the media type of the multipart/related start object.

The boundary parameter is a string of arbitrary US-ASCII characters. The string is used to separate the different body parts in the multipart/related structure. [RFC2557] This string can be seen in figure 2.2.

The body parts of the MHTML message are located in the body of the e-mail (the body is separated from the heading by an empty line

(CRLFCRLF)). Every body part has its own header and a body. Each header has a Content-Type header field specifying the media type of that body part.
The Content-Type field in the body part containing the HTML file also has a charset parameter specifying the character set of the web page.

In the header of each body part (apart from the text/html body part) there is a Content-Location header field. The value of this field is an URI which is used to locate the object by the referring HTML file. The heading of the e-mail also has a Content-Location field specifying an URI that can be used as a reference to the MHTML message.
[RFC2557]

```
To: yvonne-b@dsv.su.se
From: tina-hek@dsv.su.se
Subject: Ett mhtmlmeddelande
Mime-Version: 1.0
Content-Type: multipart/related; type="text/html";
boundary="This_Is_A_unique_boundary "
Content-Location: http://www.dsv.su.se/~yvonne-b/aggregate.mhtml

-- This_Is_A_unique_boundary
Content-Type: Text/html;charset="US-ASCII"
Content-Transfer-Encoding: 7bit

<html><head><title>En liten htmlsida med tvenne bilder</title></head>
<body><img src="http://www.dsv.su.se/~yvonne-b/yvonne.jpg" />
<img src="teckning.gif" /></body></html>

-- This_Is_A_unique_boundary
Content-Type: image/jpg
Content-Transfer-Encoding: base64
Content-Location: http://www.dsv.su.se/~yvonne-b/yvonne.jpg

R0lGODlhPABSAMQAAP///+He3tXU2sjK1bzB0K+3zKOtx5ajw4qavX2QuXCGtGR9r1dyrEx
ppz9gojJWnSZMmBlClAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAACH5BAEAAAEALAAAAAA8AFIAAAX/YCCOZGmewbKgbOu+55IkK2zfsEzjfF8KDofAR8QtH
AaDo1ZsmgaKBCSSjEASioFze3g4IAxFJKJgeB+Hre/QeDwW7wCj4WCk6l5GWv0SKB5jD3ND
DSoNAQJ0gBBxfC00EAgNUQ4iDQ1zIg5YDQhuCY4oAw0RaYUKhwGXl5YKC4cHEEyhAQMEAQk
ODyJzDEIBdUeIggx2AYIKAglD
jioDYcYIYDQFCkFZDAkNkbwNMwfGagVjArGFBnoODQZyuisGdAcLBq8NBQRjjkAIiFh+A0G
8bAsCQVEQKH6W4ZpFRIAvXwpGEHjlYIzFixjrMLgVgMCXgr94CDhy6UsqEQIA/80hViBIAY
qYIjxgFqBASTAOIuKgQaecgQQEtIhYoIDANkAYHxScyKTAswMGBjDaiAPBgnsNAIopJULBl
DcJDhQYWwABjUU6

-- This_Is_A_unique_boundary
```

```
Content-Type: image/gif
Content-Transfer-Encoding: base64
Content-Location: teckning.gif

R0lGODlhewA5APcAAHJycnNzc3V1dXZ2dnd3d3h4eHl5eXp6ent7e3x8fH19fX5+fn9/f4C
AgIGBgYKCgoODg4SEhIWFhYaGhoeHh4iIiImJiYqKiouLi4yMjI2NjY6Ojo+Pj5CQkJGRkZ
KSkpOTk5SUlJWVlZaWlpeXl5iYmJmZmZqampubm5ycnJ2dnZ6enp+fn6CgoKGhoaKioqOjo
6SkpKWlpaampqenp6ioqKmpqaqqqqurq6ysrK2tra6urq+vr7CwsLGxsbKysrOzs7S0tLW1
tba2tre3t7i4uLm5ubq6uru7u7y8vL29vb6+vr+/v8DAwMHBwcLCwsPDw8TExMXFxcbGxsfHx
Hx8jIyMnJycrKysvLy8zMzM3Nzc7Ozs/Pz9DQ0NHR0dLS0tPT09TU1NXV1dbW1tfX19jY2N
nZ2dra2tvb29zc3N3d3d7e3t/f3+Dg
-- This_Is_A_unique_boundary --
```

Figure 2.2

2.3 Sending MHTML - requirements in RFC2557

The requirements for mailers implementing RFC2557 are listed and, when needed, discussed, below. The compliance of MHTMLMailer with RFC2557 is also discussed. The goal has been for MHTMLMailer to be unconditionally compliant with RFC2557.

2.3.1 The media type multipart/related

1. “If a message contains one or more MIME body parts containing URIs and also contains as separate body parts, resources, to which these URIs (as defined, for example, in HTML 2.0 [HTML2]) refer, then this whole set of body parts (referring body parts and referred-to body parts) SHOULD be sent within a multipart/related structure as defined in [REL].” [RFC2557]

The reason why this requirement only SHOULD be satisfied is unclear. Is there another structure that can be used while still complying with RFC2557? It seems odd that this requirement is not a MUST requirement since the use of multipart/related is the whole point of RFC2557. [RFC2557]

Fulfilled by MHTMLMailer.
MHTMLMailer always sends web pages, both referring body parts and referred-to body parts, within a multipart/related structure.

2. “When the start body part of a multipart/related structure is an atomic object, such as a text/html resource, it SHOULD be employed as the root resource of that multipart/related structure. When the start body part of a multipart/related structure is a multipart/alternative structure, and that structure contains at least one alternative body part which is a suitable atomic object,

such as a text/html resource, then that body part SHOULD be employed as the root resource of the aggregate document." [RFC2557]

Fulfilled by MHTMLMailer.
The text/html body part is always employed as root body part. Multipart/alternative structures are never generated by MHTMLMailer.

3. "If the multipart/related start object is not the first body part in a multipart/related structure, [REL] further requires that its Content-ID MUST be specified as the value of a start parameter in the "Content-Type: multipart/related" header." [RFC2557]

Implicitly fulfilled by MHTMLMailer.
This is an implicit implementation since the start object is always the first body part.

2.3.1.1 Sending of web pages retrieved from the web

4. "When a sending MUA sends objects which were retrieved from the WWW, it SHOULD maintain their WWW URIs. It SHOULD not transform these URIs into some other URI form prior to transmitting them. This will allow the receiving MUA to both verify MICs included with the message, as well as verify the documents against their WWW counterpoints, if this is appropriate." [RFC2557]

Our interpretation of the above us that it means that the WWW URIs in the HTML code should not be transformed. (This, in turn, means that Content-ID cannot be used.)

Fulfilled by MHTMLMailer.
The HTML source is never transformed.

5. "..if a sender wishes a recipient to always retrieve an URI referenced resource from its source, an URI labeled copy of that resource MUST NOT be included in the same multipart/related structure." [RFC2557]

Implicitly fulfilled by MHTMLMailer.
Implicitly implemented since the referenced resources are not meant to be retrieved via HTTP.

2.3.2 Content-Location and Content-ID
When using Content-ID to reference body parts in a multipart/related structure each body part must be given a Content-ID value such as: foo@bar.net, if the sender has the domain bar.net. The value must also be present in the HTML code: <img src="cid:foo@bar.net">. This means that if you want to sent web pages from the web the URIs in the HTML code must be rewritten. [RFC2392]

When using Content-Location the referenced body part is given a Content-Location header field with a value matching the URI in HTML source. See figure 2.1. [RFC2557]

2.3.2.1 Content-Location or Content-ID?

6. "An URI in a Content-Location header need not refer to an resource which is globally available for retrieval using this URI (after resolution of relative URIs). However, URI-s in Content-Location headers (if absolute, or resolvable to absolute URIs) SHOULD still be globally unique." [RFC2557]

Imlicitly fulfilled by MHTMLMailer.
Implicit implementation since MHTMLMailer sends web pages taken from the web. All objects on the web have a globally unique URI.

7. "Content-IDs MUST be globally unique [MIME1]." [RFC2557]

Implicitly fulfilled by MHTMLMailer.
Implicit implementation - does not really apply - since MHTMLMailer never generates Content-IDs.

8. "Within a multipart/related structure, each body part MUST have, if assigned, a different Content-ID header value and *a* Content-Location header field value*s* which resolve to a different URI." [RFC2557]

Since this requirement is formulated incorrectly it is very difficult to interpret.

It is unlikely that it has to do with comparing Content-Location and Content-ID since Content-ID values and Content-Location values never can be identical. Besides, there is a requirement: "When URIs employing a CID (Content-ID) scheme as defined in [URL] and [MIDCID] are used to reference other body parts in an MHTML multipart/related structure, they MUST only be matched against Content-ID header values, and not against Content-Location header with CID: values." [RFC2557]

It is more likely that this requirement means that Content-Location values must be different from one another so there will be no conflicts with identification on receipt.

Fulfilled by MHTMLMailer.
Our interpretation is that every body part must be able to be identified uniquely. Since MHTMLMailer fulfills requirement number 6 this requirement is also satisfied.

2.3.2.2 Base URIs and references to MHTML messages

9. "The Content-Base header, which was present in RFC 2110, has been removed. A conservative implementor may choose to accept this header in input for compatibility with implementations of RFC 2110, but MUST never send any Content-Base header, since this header is not any more a part of this standard." [RFC2557]

Fulfilled by MHTMLMailer.
MHTMLMailer never creates Content-Base header fields.

10. "The URI of an MHTML aggregate is not the same as the URI of its root. The URI of its root will directly retrieve only the root resource itself, even if it may cause a web browser to separately retrieve in-line linked resources. If a Content-Location header field is used in the header of a multipart/related, this Content-Location SHOULD apply to the whole aggregate, not to its root part." [RFC2557]

This header can be used to resolve relative URIs on receipt.[RFC2557]

Fulfilled by MHTMLMailer.
With MHTMLMailer it is possible to choose whether or not to use a Content-Location header with a base URI to the MHTML message. This URI is not the same as the URI of the multipart/related root resource.

2.3.3 URIs in Content-Location header fields

11. **A)** "Some documents may contain URIs with characters that are inappropriate for an RFC 822 header, either because the URI itself has an incorrect syntax according to [URL] or the URI syntax standard has been changed to allow characters not previously allowed in MIME headers. These URIs cannot be sent directly in a message header. If such a URI occurs, all spaces and other illegal characters in it must be encoded using one of the methods described in [MIME3] section 4."
B) "This encoding MUST only be done in the header, not in the HTML text." [RFC2557]

This means that an URI that looks like:
http://www.dsv.su.se/sötvalp.gif,
after encoding the URI with the method above the URI looks like:
=?ISO-8859-1?q?http://www.dsv.su.se/s=F6tvalp.gif?=.

A peculiar thing with this requirement is that it allows an illegal URI as value to a Content-Location header field. This is not allowed according to the definition of Content-Location, where you can read that URIs in Content-Location header fields are restricted to "the syntax for URLs as defined in [URL]" [RFC2557].

11 B is misleading since the referred to method is only used to encode header fields. [MIME3].

11 A is **fulfilled by MHTMLMailer.**
MHTMLMailer encodes URIs containing illegal characters before these are sent in a Content-Location header field. Since space is not an illegal character for RFC2822 headers, spaces are only encoded by MHTMLMailer if there are other illegal characters present.

B is **fulfilled by MHTMLMailer.**
MHTMLMailer never encodes URIs in the HTML code.

12. **A)** "Since MIME header fields have a limited length and long URIs can result in Content-Location headers that exceed this length, Content-Location headers may have to be folded."
B) "Encoding as discussed in clause 4.4.1 MUST be done before such folding. After that, the folding can be done, using the algorithm defined in [URLBODY] section 3.1." [RFC2557]

The referred-to method for folding of Content-Location header fields cannot be used, this will be discussed in chapter 5.

A is **fulfilled by MHTMLMailer.**
MHTMLMailer folds URIs longer than 78 characters (CRLF included). The folding is not done according to the referred to method.

B is fulfilled **by MHTMLMailer.**
Encoding is always done before folding.

2.3.4 Charset

13. A) "The charset parameter value "US-ASCII" SHOULD be used if the URI contains no octets outside of the 7-bit range."
B) "If such octets are present, the correct charset parameter value (derived e.g. from information about the HTML document the URI was found in) SHOULD be used." [RFC2557]

The first problem with this requirement is what charset *parameter* means. Charset *parameter* is a parameter to the Content-Type header field used with the top-level media type "text". [MIME2]

This requirement however has to do with encoding of illegal characters in MIME header fields. The charset used in this encoding is not a charset *parameter* [MIME3].

Requirement **13 A** is peculiar since if an URI contains no illegal characters there is no need for encoding, and no charset must be given.

It is allowed to send US-ASCII as an encoded header field but it is discouraged [MIME3].

> C) “If this cannot be safely established, the value "UNKNOWN-8BIT" [RFC 1428] MUST be used.” [RFC2557]

We have chosen to ignore this requirement since this is a normative reference to an informational RFC. Normative references are supposed to refer to other standards-track RFCs at the same level or higher. (The standard cannot move from Proposed to Draft unless all of the normative references refer to RFCs at Draft or Internet Standard.) [RFC3160].
It is also doubtful if UNKNOWN-8BIT is supposed to be used in this context. More about this in chapter 5.

> D) “Note, that for the matching of URIs in text/html body parts to URIs in Content-Location headers, the value of the charset parameter is irrelevant, but that it may be relevant for other purposes, and that incorrect labeling MUST, therefore, be avoided.” [RFC2557]

A has been **ignored by MHTMLMailer** since MIME header fields containing only US-ASCII does not need to be encoded.
B is fulfilled **by MHTMLMailer.**
C has been ignored **by MHTMLMailer.** See chapter 5.
D is **fulfilled by MHTMLMailer.**

(See requirement 14 how this is done.)

14. “Some transport mechanisms may specify a default "charset" parameter if none is supplied [HTTP, MIME1]. Because the default differs for different mechanisms, when HTML is transferred through e-mail, the charset parameter SHOULD be included, rather than relying on the default.” [RFC2557]

This requirement has to do with the charset parameter used with the text/html body part. This is a parameter to the Content-Type header field in the header of the text/html body part. The parameter specifies the character set used by the web page. [RFC2557]
If the charset used is ISO-8859-1, the Content-type header would look like this:
Content-Type: **text/html**; charset="**ISO-8859-1**".

Fulfilled by MHTMLMailer.
The problem is that there is no fully reliable way of finding out the charset of the web page if it has not been specified in the HTML code. The HTTP standard does state that if there is no charset specified, ISO-8859-1 can be used. [HTTP].

We use two methods in an effort to find out the charset of the web page:
1) First we look for the HTML element <meta> tag in the HTML source. This element might be used to specify a charset.
2) If this is unsuccessful a request is sent to the web server the web page was returned from, asking for the charset used.

If neither method is successful we use the default charset specified by HTTP along with a Comments header field:

Comments: The charset parameter uses the Http default charset.

We also make sure that the charset used for the encoding of Content-Location header fields is equal to this one so there will be no error due to discrepancy.

2.3.5 Other requirements

15. "The MIME standard [MIME2] requires that e-mailed documents of "Content-Type: Text/ MUST be in canonical form before a Content-Transfer-Encoding is applied, i.e. that line breaks are encoded as CRLFs, not as bare CRs or bare LFs or something else. This is in contrast to [HTTP] where section 3.6.1 allows other representations of line breaks." [RFC2557]

Probably fulfilled **by MHTMLMailer.**
We assume that this requrement is satisfied since JavaMail implements MIME. [JavaMail tutorial]. We have not made any tests to verify this assumption.

16. "If a document has to be converted in such a way that a checksum based message integrity check becomes invalid, then this integrity check header SHOULD be removed from the document." [RFC2557]

We have not been able to find any information about this header, but we presume that it is a MIME header field.

Implicitly fulfilled **by MHTMLMailer.**
MHTMLMailer does not create such header fields.

2.3.6 Conclusions

Due to difficulties interpreting rfc2557 we can not say with absolute certainty that our mailer is compliant with RFC2557. The requirements that have been most problematic are 8, 13A and 13C. Apart from these requirements, we believe our mailer is unconditionally compliant with RFC2557.

Regarding requirement 8 we have implemented it as we interpreted the requirement.
We have chosen to ignore requirements 13A and 13C since our opinion is that these requirements should not be complied with. Our interpretation of requirement 13 leads us to the conclusion that 13A is unneccessary. Regarding 13C we are doubtful if the referred to RFC is appropriate. It is only an informational RFC and we are not convinced that the charset UNKNOWN-8BIT is suitable in this context.

3. Comparison of MHTMLMailer with Microsoft Outlook Express, version 6

Since we have only found one vendor, Microsoft, providing e-mail clients that uses Content-Location when sending web pages we thought it would be helpful to use one of their mailers, Outlook Express as a reference. Another reason for this decision was the difficulty we had with interpreting RFC2557. It was deemed useful to see how Outlook Express had chosen to resolve the ambiguities.

3.1 Testing

When choosing **Message-> New message using-> Web page** in Outlook Express Outlook does, in effect, do the same thing MHTMLMailer does. It sends an arbitrary web page from the WWW using the Content-Location header to reference body parts in a multipart/related structure.

The comparison between MHTMLMailer and Outlook Express was made by sending each of the web pages mentioned below with the two mailers, respectively. The messages were received with the text based e-mail client Pine and saved as text. They were then opened in a text editor and compared to each other.

3.1.1 Test web pages

Five different web pages were sent in order to investigate the general structure of the generated MHTML messages:

- http://www.dsv.su.se
- http://www.kth.se
- http://www.ietf.org
- http://www.sun.com
- http://www.dsv.su.se/~tina-hek/test/ContentLocation.html

The last web page is a page developed for this particular test.

3.2 Test results

Microsoft Outlook Express version 6 was tested on the operating system Windows 2000.

The questions are derived from the requirements RFC2557 has on an implementation.

3.2.1 The media type multipart/related

- What does the generated multipart/related structure look like?
- Does it comply with the requirements for multipart/related structures?

Outlook Express correctly creates a multipart/related structure. The structure differs from the one created by MHTMLMailer in that its text/html part (root resource) is embedded in a multipart/alternative structure which in turn, is embedded in a multipart/related structure. In the multipart/alternative structure lies, apart from the HTML file, also a plain text version of this HTML file. This text version has the media type text/plain. Due to the fact that the root resource is in a multipart/alternative structure the type parameter (in the Content-Type header in the heading of the multipart/related structure) has the value multipart/alternative. The structure created by MHMTLMailer is simpler: it consists solely of the parts that make up the web page. The type parameter, accordingly, has the value text/html.

3.2.2 Content-Location and Content-ID

- Are URIs in Content-Location fields unique, in absolute form, within the message?
- Are Content-ID fields or Content-Base fields used?

Neither e-mail client use Content-ID fields in the multipart/related-structure, both use Content-Location fields. The URIs in these fields are globally unique (when made absolute) since the web pages sent are taken directly from the web. Neither client uses Content-Base fields.

3.2.3 Content-Location header field values as base

- Is there a Content-Location field in the heading of the MHTML message?
- Does its value differ from the URI that locates the web page?
- Is there a Content-Location field on the text/html part that can be used as a base for resolving relative URIs?

Outlook Express does not use Content-Location fields in the heading of the MHTML message nor in the heading of the text/html part. When using MHTMLMailer the sender can chose to use such a field in the heading of the message. This URI is then different from the URI of the web page.

3.2.4 Encoding of URIs

- Are illegal characters in Content-Location fields encoded according to RFC2047?
- If so, is the same charset, as that of the web page, used?

In order to test this a web page developed especially for this purpose was sent. It contains URIs with illegal characters. We wanted to see if Outlook Express deals with URIs whose syntax is faulty and, if so, if these are encoded.

When the user has typed the URI to a web page to be sent by Outlook Express the mailer displays the page in a window before sending. When trying to send the web page with illegal URIs Outlook Express did not show the images with references containing illegal characters (å, ä, ö). A message stating that one or more images could not be found was displayed. Outlook Express does not send these images but does however send the images referenced by URIs containing space.
Outlook Express does not encode the URIs containing space but in case of illegal characters in The Subject field these are encoded according to RFC2047 (using B encoding).
MHTMLMailer sends the images referenced by URIs containing illegal characters and encodes these in the Content-Location field using the same charset as that of the Content-Type field of the text/html part. This is done to avoid problems with non-English characters using different charset.

3.2.5 Folding of long URIs

- Are URIs longer than 78 characters folded?

This was tested using a web page containing three images whose referencing URIs are longer than 78 characters but shorter than 998 characters. URIs longer than 998 characters were deemed too unlikely a scenario to be tested.
Outlook Express does not fold URIs longer than 78 characters.
MHTMLMailer folds all URIs in Content-Location fields longer than 78 characters.

3.2.6 Analysis of the MIME body part text/html

- Is the HTML code changed with regard to content (tags, URIs etc).

Unlike MHTMLMailer Outlook Express adds a number of elements to the HTML code. Among other things a <meta> element specifying a Windows-specific charset is added (windows-1252). A <base> element containing the URI of the web page is also added. If such a <base> tag already exists Outlook simply adds its own <base> element above the existing one. This is not only illegal HTML but also illegal MHTML.

3.2.7 Comments in Content-Location header fields?

- Are comments generated in Content-Location header fields?

This was tested due to a discussion on the mailing list of mhtml (the ietf group responsible for RFC2557) in January and March 2002. The discussion was about the inappropriateness of comments in the Content-Location header field, and the removal of this possibility from the ABNF definition. Before making such a decision it was decided that existing implementations would have to be examined to rule out that such comments are being used.

Neither e-mail client uses comments in the Content-Location header field.

3.3 Summary - differences between Microsoft Outlook Express and MHTMLMailer

Outlook Express always generates a text version of the HTML file. The text/plain and text/html MIME body parts are embedded in a multipart/alternative structure. This structure, along with images and other linked objects are embedded in a multipart/related structure. MHTMLMailer does not use a multipart/alternative structure; all MIME body parts of the MHTML message are embedded in a multipart/related structure.

Outlook Express always uses the character set Windows-1252 as a value of the charset parameter on the text/html body part as well as on the text/plain body part. MHTMLMailer first examines whether there is a charset specified for the web page or if this information can be obtained from the web server. If this is unsuccessful the default for HTTP is used.
Outlook Express always uses the media type application/octet-stream as the value of The Content-Type field for the images embedded in the multipart/related structure (and never image/jpg if the image is in jpg format). MHTMLMailer adjusts the Content-Type value according to the format of the image.

Unlike MHTMLMailer Outlook Express does not send linked style sheets.

Outlook Express does not send images referenced by URIs containing illegal characters, which means that there is no need for the encoding of such URIs in the Content-Location header field. Nor does Outlook Express fold URIs longer than 78 characters. MHTMLMailer encodes and folds URIs when necessary.

4. Testing and results

4.1 Receipt of MHTML messages

The testing of MHTMLMailer was conducted by sending different web pages to e-mail clients to find out how well these clients could receive and

display MHTML messages. The focus was on testing multipart/related and Content-Location. We hope these results can aid in the work of ietf to decide if the MHTML standard can be elevated from the proposed standard level to the draft standard level in the Internet Standards Track.

4.1.1 Testing methods and clients tested

Three different web pages especially developed for this test was sent from MHTMLMailer:

* http://www.dsv.su.se/~tina-hek/test/ContentLocation.html
* http://www.dsv.su.se/~tina-hek/test/Encoding.html
* http://www.dsv.su.se/~tina-hek/test/Folding.html.

The testing included the following steps:
1. When a web page had been sent, its linked images were removed from the web before the MHTML message was opened by the receiving client. This was done in order to insure that the e-mail client is using the images sent in the message (rather than retrieving linked objects via HTTP).
2. The images were then returned to their place on the web. In case the client did not initially show the images correctly, the message was opened once again to see if the client can retrieve linked object via HTTP (much like a browser).
3. After finding out whether the e-mail client can handle MHTML messages that uses Content-Location, we moved on to test if they can decode such Content-Location header fields.
4. The e-mail clients ability to unfold Content-Location fields was then tested.

The tested email clients were Microsoft Outlook Express 6, MSN Hotmail, Yahoo! Mail, Netscape Messenger 4.77 and Qualcomm Eudora 5.0. We chose the tested clients partly in accordance with an earlier test [Hentze-Muto 2000] conducted at Stockholm University (in the year of 2000) and partly after a discussion with Jacob Palme.
Microsoft Outlook Express 6 was chosen because it is a well-known and much used client and also because Microsoft is the only vendor we know of, which implements Content-Location when sending. MSN Hotmail and Yahoo Mail! were chosen because we wanted to test a couple of well-established web based e-mail clients. Netscape Messenger 4.77 and Qualcomm Eudora 5.0 was chosen because they are well-known clients alongside Outlook Express.

4.1.2 Testing - detailed description of the analysis

Does the receiving e-mail client handle Content-Location on receipt?
This was tested by sending a web page containing both absolute and relative URIs referencing linked images and see how the message was displayed by the receiving client. The message was opened twice, the

first time without the images being present on the web, and the second time with the images on the web. From this we could draw a conclusion as to whether any images are retreived via HTTP.
If all linked objects (images and style sheets) are displayed properly the first time the message is opened it means that the receiver can handle Content-Location. If the receiver does not display every object correctly it means that they cannot handle Content-Location on receipt.

The web page was sent once with a base URI in the Content-Location header field in the heading of the MHTML message and once without such an URI. This means that the above procedure was executed twice.

Does the receiving client retrieve linked objects via HTTP?
In those cases where the recipient did not initially display all linked objects the images were returned to their place on the web and the message was reopened.
If the images not initially displayed are shown it means that the receiving client retrieves these images via HTTP.

Can the receiving client decode encoded URIs in the Content-Location header field?
This was tested by sending a web page containing four images referenced by relative URIs containing illegal characters and observe how this message was displayed by the receiving client. Two of the images have URIs containing illegal characters. One image has an URI containing both illegal characters and space. The forth image has an URI containing space. The reason for testing URIs with space is what RFC2557 says about encoding: 'all spaces and other illegal characters in [the field] must be encoded' [RFC2557]. MHTMLMailer encodes spaces in Content-Location header fields in those cases where the URI also contains other illegal characters, but not otherwise, partly because the method used in JavaMail API behaved this way.

If the images referenced by URIs containing illegal characters are displayed properly it means that the receiving client can decode an encoded URI in a Content-Location header field.

Can the receiving client unfold a folded URI in the Content-Location header field?
This was tested by sending a web page containing three images referenced by relative URIs longer than 80 characters and observing how this message was displayed by the receiving client. One of the URIs also contained illegal characters (å, ä, ö).

Since MHTMLMailer folds URIs longer than 40 characters one can presume that if the images are displayed properly it means that the receiver

can unfold such URIs. (Well, at least the kind of folded URIs that MHTMLMailer generates.)

4.2 Test results

Microsoft Outlook Express version 6
Microsoft Outlook Express version 6 was tested on the operating system Windows 2000.

Outlook Express does not have any problems displaying the MHTML messages sent by MHTMLMailer. The web pages look exactly as they do on the web. There is no difference between how Outlook Express displays the message if there is a base URI in the Content-Location header field in the heading of the message or not.

Outlook Express can decode as well as unfold URIs in Content-Location header fields.

4.2.1 Netscape Messenger version 4.77

Netscape Messenger in the package Netscape Navigator 4.77 was tested with the operating system Debian Gnu/Linux 3.0.

Netscape has some difficulty displaying MHTML messages correctly. The web pages look like they do on the web with the exception that the images are not always displayed correctly.
There is a difference between how Netscape displays MHTML messages depending on if there is a base URI in the Content-Location header field in the heading of the message or not.
When the web page is sent without a base URI all images are displayed correctly. This implies that Netscape can handle Content-Location on receipt.
When the same web page is sent with a base URI the images referenced by relative URIs, are not displayed. This means that the base URI somehow interfere with the resolution of relative URIs in the text/html body part.

Netscape can neither decode nor unfold URIs in Content-Location header fields.

4.2.2 QUALCOMM Eudora version 5.2

QUALCOMM Eudora was tested with the operating system Windows 2000.

The MHTML messages are more or less displayed correctly. The web pages look essentially as they do on the web. Eudora does however add information onto the web page: parts of the heading of the e-mail are shown at the top of the page and a linked style sheet is displayed as text on the bottom of the web page.

There is no difference between how Eudora displays the message if there is a base URI in the Content-Location header field in the heading of the message or if there is not.
Eudora cannot show images referenced by an absolute URI, which means that Eudora does not fully handle Content-Location on receipt.

When the images are returned to the web, the image referenced by an absolute URI is displayed. This means that Eudora chooses to retrieve this image via HTTP.

Eudora can decode but not unfold URIs in Content-Location header fields.

4.2.3 MSN Hotmail
MSN Hotmail was tested with operating system Windows 2000 using Internet Explorer 6 and Netscape 7.01 as browsers.

Hotmail does not have any problems displaying the MHTML messages correctly. The web pages look as they do on the web apart from the web page with a linked style sheet, where the style sheet is not used.
In the HTML code for the web pages developed especially for this test the URIs referencing each image was placed in <p> elements next to the image. These URIs are distorted by Hotmail into a strange absolute URI. (The only exceptions are relative URIs, containing no illegal characters, that are shorter than 78 characters). This implies that the HTML code has also been changed. Since messages cannot be saved as text in Hotmail this has not further been explored.
There is no difference between how Hotmail displays the message if there is a base URI in the Content-Location header field in the heading of the message or if there is not.

Hotmail can both decode and unfold URIs in Content-Location header fields.

4.2.4 Yahoo! Mail
Yahoo! Mail was tested with operating system Windows 2000 using Internet Explorer 6 and Netscape 7.01 as browsers.

Yahoo! Mail does not show the web page as it looks on the web. No images are displayed and the linked style sheet is not used. The images are displayed as attachments. This means that Yahoo cannot handle MHTML at all.

When the images are returned to their place on the web the images with an absolute URI are displayed correctly which means that Yahoo retrieves these images via HTTP.

Whether Yahoo can decode and unfold URIs in Content-Location header fields cannot be tested since Yahoo does not handle Content-Location on receipt.

4.2.5 Possible source of error

Images removed from the web

Since the testing of all e-mail clients were done in the same physical location we tried to avoid that images would be retrieved from a cache by using different images for different web pages and by removing these images from the web prior to opening the MHTML messages the first time. The Refresh or Reload function in the e-mail client has been used throughout to update the messages.

Unfolding of folded URIs

The method we use to fold URIs may not be the best method to fold Content-Location header field values. Our method entails folding URIs after 40 characters which means that they can be folded anywhere in the string of characters, even in the middle of the file name. This manner of folding a field value is not recommended in RFC2822.

4.2.6 Conclusions

The conclusions from these tests are that most of the tested e-mail clients can handle Content-Location. All of them, with the exception of Yahoo, can display MHTML messages sent by MHTMLMailer more or less without errors. Eudora cannot show images referenced by absolute URIs and Netscape cannot handle a base URI in the Content-Location header field in the heading of the message. It is difficult for us to comment on why these e-mail clients can handle one function but not another although the functions require the same treatment.

All e-mail clients that can handle MHTML with Content-Location can decode URIs with the exception of Eudora. Concerning encoding there was no difference whether the URI contained only space or illegal characters (å, ä, ö). Of the e-mail clients that can handle MHTML with Content-Location, Outlook Express and Hotmail can unfold folded URIs while Eudora and Netscape cannot.

5. Comments on RFC2557

The specification of how to implement the Content-Location header is not always straightforward. We found several issues that were unclear and sometimes even contradictory.

In this chapter, we summarize the different parts of the standard that were problematic when interpreting and implementing the mailer.

A general comment on RFC2557 is that it could be a lot clearer concerning the purpose of having two ways of referencing MIME body parts in a multipart/related structure. Also the thoughts behind developing the standard could be clearly pointed out.

We think that maybe the authors have tried to make the standard "too general" and that this sometimes contributes to making it ambiguous.

5.1 The purpose of developing RFC2557 should be clearer

5.1.1 MUST and SHOULD requirements are neither adequate nor clear

5.1.1.1 Multipart/related

The MIME media type multipart/related is used to aggregate the different parts of a document. RFC2557 mentions no other means of aggregating the referenced objects of an HTML document, and yet the requirement to use multipart/related is a SHOULD requirement. Why? Is not the main purpose of RFC2557 to embed referenced objects in a multipart/related structure?

5.1.1.2 Encoding and folding

Encoding of URIs containing illegal characters and folding of long URIs are not subject to any SHOULD or MUST requirement. It is unclear why. When implementing our mailer we choose to view these as "real" requirements. The reason for not specifying the requirements with capital letters might be the authours wishes to be able to implement MHTML when sending via protocols other than SMTP. If this is the reason it should be mentioned in RFC2557.

5.1.2 Charset parameter

Four different requirements concerning use of the correct charset parameter, are formulated in RFC2557 (chapter 4.4.1 and 4.4.2). This makes the charset parameter seem very important (especially in relation to encoding and folding). The requirements are not well formulated and leave the implementor wondering how they should be managed.

5.1.3 Content-Location and MICs

The last part of chapter one mentions, as a reason for using the Content-Location header, that you are able to send MHTML without rewriting the HTML, the disadvantage with such rewriting being that "security checksums" cannot be validated.
This is, of course, dependent on when MICs are derived. RFC2557 gives the impression that the validation only concerns the web page (the text/html part of the MHTML message) compared to the web page residing on the web. Such a validation might be impossible for other reasons: the sender might not have sent the URI to the web page in the MHTML message since there is no requirement to do so. Another reason might be that line breaks might need to be replaced as described in chapter 10.

If the sender, on the other hand, puts her signature on the entire MHTML message after composing it, there is no special advantage of using Content-Location with regards to security checksum validation. This, of course, requires that the receiver does trust the sender.

Our point is that we find that RFC2557 exaggerates the use of Content-Location for security reasons. Not needing to rewrite URIs is an advantage on its own.

5.2 Badly organized and formulated text

5.2.1 Title related to content of chapters

Our general impression of the disposition in RFC2557 is that it is not very organized but rather messy. The content related to the titles of some chapters can be questioned and the reader is confused about the matters handled. Below are some examples of confusing disposition.

5.2.1.1 Chapter four: Encoding of MIME header fields?

Chapter four is entitled "Encoding of MIME header fields" but it seems to specify the encoding of the text/html body part. This is because it deals with the charset *parameter* that is not used when encoding header fields but when encoding text/html data.

5.2.1.2 Chapter seven: Use of the Content-type multipart/related?

Chapter seven is entitled "Use of the Content-type multipart/related". The first part is fine with a description of how to use multipart/related, what requirements there are and so on. Then, out of nowhere, there is a listing of the advantages of MHTML. We think this information would be better suited as an introduction to the standard. After the listing of the advantages there is a passage that we misunderstood for a long time and that even now seems odd. The passage is:

"When a sending MUA sends objects which were retrieved from the WWW, it SHOULD maintain their WWW URIs. It SHOULD not transform these URIs into some other URI form prior to transmitting them. This will allow the receiving MUA to both verify MICs included with the message, as well as verify the documents against the WWW ***counterpoints****, if this is appropriate."*

Since the text is placed in a chapter entitled "Use of the Content-type multipart/related" it is not obvious that the requirement only deals with the text/html body part.

We interpret this text as when sending HTML documents as MHTML one should not rewrite the HTML source code. If this is a correct interpretation then the implication of this is that one should always

use the Content-Location header and never use the Content-ID header. Again, if this is correct then it could be specified in a better way for example by saying:
When sending web pages taken from the web one should always use the Content-Location header field in the multipart/related structure, in order to be able to send the web page without rewriting the URIs in the HTML source.

This motivation for using Content-Location is in our meaning not a very good one, as discussed in the chapter "Content-Location and MICs" above.

5.2.1.3 Should URIs in Content-Location field follow the URI syntax or not?

Chapter 4 contains the following subchapter:

"4.4.1 Encoding of URIs containing inappropriate characters

Some documents may contain URIs with characters that are inappropriate for an RFC 822 header, either because the URI itself has an incorrect syntax according to [URL] or the URI syntax standard has been changed to allow characters not previously allowed in MIME headers. These URIs cannot be sent directly in a message header. If such a URI occurs, all spaces and other illegal characters in it must be encoded using one of the methods described in [MIME3] section 4".

The text above allows URIs with illegal syntax to be sent in MHTML messages while the ABNF specification of the Content-Location header field in chapter 4.1 says the URI should follow the syntax for URIs. We think it should be clearly motivated if URIs with illegal syntax is allowed or not.

5.2.1.4 Terminology

5.2.1.4.1 HTML aggregate objects

Chapter 2.2 (Other Terminology) defines terms that are never used again. An example is HTML aggregate objects.

5.2.2 Aggregate document
Another term whose meaning is unclear is "aggregate document". Does it mean a web page with all referenced objects or a MHTML message?
The title of the standard is MIME encapsulation of aggregate documents, **such as** HTML (MHTML). This implies that the web page (HTML) is the aggregate document.
Then in chapter three there is the following passage: *"An aggregate document is a MIME-encoded message that contains a root resource (object) as well as other resources linked to it via URIs."* This on the

other hand makes it sound as if an aggregate document is a MHTML message, which in turn makes the title of RFC2557 ambiguous. In chapter seven the text/html MIME part is referred to as a *"root resource of the aggregate document"*.

Chapter 4.3 entitled "URIs of MHTML aggregates", where *MHTML aggregate* is solely used in this chapter and is not defined in the terminology chapter.

Aggregate document, *MHTML aggregate*, *multipart/related structure* and *MHTML multipart/related structure* are probably used as synonyms. None of these terms are defined in the terminology chapter.

Sometimes, the word "message" is used in a strange way. Chapter seven begins:

"If a message contains one or more MIME body parts containing URIs and also contains as separate body parts, resources, to which these URIs (as defined, for example, in HTML 2.0 [HTML2]) refer, then this whole set of body parts (referring body parts and referred-to body parts) SHOULD be sent within a multipart/related structure as defined in [REL]."

It sounds a bit odd that a **message** contains body parts since it is not a message before putting it in a multipart/related structure and sending it.

We recommend checking the terminology in RFC2557 to reduce the number of terms used and to define every important term. The standard should also be checked concerning semantics and editorial errors that, unnecessarily, confuse the reader. Also, a definition of MHTML should be added.

5.2.2.1 Editorial errors?

Another peculiar passage is found in chapter 4.4.1 that states reasons for encoding URIs in Content-Location headers:

"...either because the URI itself has an incorrect syntax according to [URL] or the URI syntax standard has been changed to allow characters not previously allowed in MIME headers."

The word "previously" should not be present. We assume that the meaning is to encode characters that are not allowed in MIME header fields regardless whether these previously where not allowed in URIs.

The text continues:

"These URIs cannot be sent directly in a message header. If such a URI occurs, all spaces and other illegal characters in it must be encoded using one of the methods described in [MIME3] section 4.
…
Receiving clients MUST decode the [MIME3] encoding in the heading before comparing URIs in body text to URIs in Content-Location headers."

The first sentence concerns only *message* headers. The latter sentence makes it clear that also headers of MIME parts should be encoded (or else it would be unnecessary to decode these). In other parts of RFC2557 there is a clear distinction between the message heading and the headers of MIME parts so the reader is left wondering whether to encode all or only message headers.

5.3 Techniques that should not or can not be used

5.3.1 Unknown-8bit

Chapter 4.4.1 specifies:
"The charset parameter value "US-ASCII" SHOULD be used if the URI contains no octets outside of the 7-bit range. If such octets are present, the correct charset parameter value (derived e.g. from information about the HTML document the URI was found in) SHOULD be used. If this cannot be safely established, the value "UNKNOWN-8BIT" [RFC 1428] MUST be used."

We have identified two problems concerning the above specification:

1. RFC1428 referred-to by the MUST requirement above, is an informational RFC and does not participate in the standards track. Normative references (like MUST requirements) must refer to standards on a higher level than the referring standard or else the referring standard cannot advance in the standards track [RFC3160].

2. Is UNKNOWN-8bit really meant to be used like this?

   The appendix of RFC1428 says:
   "This character set is not intended to be used by mail composers. It is assumed that the mail composer knows the character set in use and will mark it with a character set value…
   The use of the "unknown-8bit" label is intended only by mail gateway agents which cannot determine via out-of-band information the intended character set."

   One could draw the conclusion that "UNKNOWN-8BIT" should not be used when implementing an e-mail client. There is no explanation for this requirement in RFC2557 and we think there should be a motivation why this is a requirement that MUST be met.

5.3.2 Folding of long URIs in Content-Location headers
The following is what RFC2557 says about folding:

"Since MIME header fields have a limited length and long URIs can result in Content-Location headers that exceed this length, Content-Location headers may have to be folded.
Encoding as discussed in clause 4.4.1 MUST be done before such folding. After that, the folding can be done, using the algorithm defined in [URLBODY] section 3.1." [RFC2557]

The method of encoding that RFC2557 specifies states that encoded fields cannot be longer than 75 characters, and if they are they will be folded. We assume that this means that no other folding (as specified in RFC2557) should be needed. It should not be necessary to fold an encoded URI as specified in the last sentence of the passage.

Only when there is an URI that does not need to be encoded but needs to be folded, would you need to do a separate folding. The referred-to standard RFC2017 (URLBODY), can not be applied on an URI in a Content-Location header field. We give our analysis of why this is so, next.

5.3.2.1 RFC2017 - Definition of the URL MIME External-Body Access-Type

This standard specifies a MIME part having the media type "message/external-body" that can be referred to by URIs. The purpose of message/external-body is to have a way of NOT sending the object referred-to. [MIME2] Which is exactly the opposite of the purpose of RFC2557, which seems odd.

In RFC2017, the algorithm to fold URIs is intended to be used to fold a *parameter* to a *Content-Type field* with the value *message/external-body*. The value of the parameter is in this case an URI as it is in a Content-Location header field. [RFC2017] But that seems to be the only similarity.

Before the algorithm in RFC2017 is specified, the authors state the following:

"The syntax of an actual URL string is given in RFC 1738. URL strings can be of any length and can contain arbitrary character content. This presents problems when URLs are embedded in MIME body part headers that are wrapped according to RFC 822 rules. For this reason they are transformed into a URL-parameter for inclusion in a message/external-body content-type specification as follows:"

Our impression is that the authors have identified problems with sending URIs in MIME header fields and consequently solves the problems by putting the URIs in a parameter to the Content-Type field.

We have not made any efforts to find out exactly what the problems, identified by the authors of RFC2017, are nor why the problem is solved when sending the URI in a parameter. This is an issue for the authors of RFC2557 to deal with. We just thought it would be worth mentioning since if there is a problem then this problem also concerns the Content-Location header.

5.3.2.2 The algorithm in RFC2017, referred by RFC2557

The first step is to encode illegal characters according to RFC2396 [RFC2017], but this is not in line with what RFC2557 specifies since RFC2557 states to encode illegal characters according to RFC2047.

If an implementor would encode according to what RFC2017 specifies this would mean that the receiver of such a message would not be able to match Content-Locations to the URIs in the HTML source code, since RFC2557 specifies not to decode characters encoded according to RFC2396 before matching the URIs.

The next step is to fold the URI in strings that are 40 characters or less. This step is not a problem in itself.

The third step is to put the strings separated by LWSP between quotation marks in an URL parameter to a Content-Type field with the value message/external-body.

LWSP is, if we are correct, the opposite of FWSP and FWSP allows folding but LSWP does not. If this is true then this method does not even fold the URI.

One commentator on the mhtml mailing list wrote that RFC2231 deals with specific methods for folding and encoding a *parameter* to a *Content-Type* field. We have choosed not to investigate this since RFC2557 has nothing to do with parameters in Content-Type header fields but Content-Location header fields without parameters.

5.3.2.3 So, how should folding be implemented?

How the algorithm in RFC2017 is supposed to be applied to a Content-Location header field is not known to us. Nor is it clear why RFC2557 refers to RFC2017 instead of RFC2822. We think RFC2822 should be applied, given that Content-Location fields are also in a sense RFC2822 fields.

It is then unclear how to fold URIs in Content-Location fields. According to the definition of Content-Location, FWS can only be used before or after the URI. This would mean that if an URI makes the Content-Location field longer than 78 characters (following the SHOULD requirement of RFC2822) you cannot fold it to make the line shorter.

We will not dig any deeper into this matter, the purpose is to present the problems we have had in implementing RFC2557. It is up to the authors to specify how to fold so implementors easily can follow the specification.

5.3.3 Use of Thismessage:/

We have not been able to see the point in using "thismessage:/", as specified in RFC2557, to resolve relative URIs when a MHTML message has no base URI in the Content-Location field in the heading. What is the difference between using thismessage:/ and resolving them directly against each other? The authors should also make it clear that the use of thismessage:/ does not mean that thismessage:/ should be added to relative URIs, by writing it, but that the use is implicit.

From the description in RFC2557, of how to unpack received MHTML messages, it is clear that the sender can put a Content-Location header with an absolute URI in the message heading or in the heading of the text/html MIME part.

RFC2557 gives an example of an MHTML message in which a relative URI in the HTML source has been transformed to an absolute Content-Location URI. A receiver of such a message would need to resolve the relative URI in the source to an absolute URI before being able to match that body part.

5.3.3.1 Problem

There is a problem with this example. If there is no absolute URI in the message heading or on the text/html part then the MIME part would not be matched using "thismessage:/". When this is the case the receiver is expected to retrieve the referenced object via HTTP. Which, of course, is not possible. This is a clear error in RFC2557.

Apart from not being usable under all circumstances, the specification of the use of "thismessage:/" is not as clear as it should be.

5.4 How to view the Content-Location header

We have not been able to get answers concerning how to view the Content-Location header with regards to other e-mail header fields. Is Content-Location a structured or unstructured header field as defined in RFC822? And does this have implications for encoding and folding of URIs in Content-Location fields?

Content-Location is a MIME field and as such should follow the syntax for fields according to [RFC2822].

When implementing our mailer, we chose to follow the specification of Content-Location in RFC2557, since we did not have the experience nor knowledge to decide whether Content-Location is structured or not nor if this has any consequences.

5.5 Techniques more difficult than necessary

5.5.1 Receiving MHTML messages

The necessity that all relative URIs in a MHTML message using Content-Location, should be made absolute before matching HTML source and Content-Locations, is not a very good one.

Besides the information about this being divided into two different chapters that do not seem well synchronized, we feel that this might be an unneccessary step that receivers of MHTML must take.

It might have its cause in a desire to make the standard general. We have not seen any reasons for a mailer sending MHTML to ever need to make URIs in Content-Location headers any different from the URIs in the HTML source code.

If this is true, then there would not be any reasons for not being able to match URIs directly against each other without making all relative URIs absolute.

The reason for having receivers make all relative URIs absolute might be that two MIME parts should not be able to have the same URI. With web pages taken from the web this is never an issue. With other webpages there is always the possibility to use Content-ID instead.

6. Acknowledgments

Jacob Palme
Fredrik Kilander
Lars Enderin
Sven Olofsson

7. References

Ref.	Author, title
[RFC2110]	J. Palme, A. Hopmann: "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)", RFC2110, March 1997.

[RFC2557] J. Palme, A. Hopmann: "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)", RFC2557, March 1999.

[RFC2119] S. Bradner: "Key words for use in RFCs to Indicate Requirement Levels", RFC2119, March 1997.

[RFC2821] J. Klensin, Editor: "Simple Mail Transfer Protocol", RFC2821, April 2001.

[HTTP] T. Berners-Lee, R. Fielding, H. Frystyk: "Hypertext Transfer Protocol -- HTTP/1.0. ", RFC 1945, May 1996.

[RFC2119] S. Bradner: "Key words for use in RFCs to Indicate Requirements Levels. " RFC 2119, March 1997.

[RFC2822] P. Resnick, Editor: "Internet Message Standard", RFC2822, April 2001.

[RFC3160] S. Harris: "The Tao of IETF - A Novice's Guide to the Internet Engineering Task Force", RFC3160, August 2001

[RFC2392] E. Levinson: "Content-ID and Message-ID Uniform Resource Locators", RFC2392, August 1998.

[MIME1] N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, December 1996.
.

[MIME2] N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, December 1996.

[MIME3] K. Moore, "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, December 1996.

[RFC2026] S. Bradner: "The Internet Standards Process", RFC2026, October 1996.

[RFC2387] Edward Levinson: "The MIME Multipart/Related Content-Type", RFC 2387, August 1998.

[RFC2028] R. Hovey, S. Bradner: "The Organizations Involved in the IETF Standards Process", RFC2028, October 1996.

[RFC3160] S. Harris: "The Tao of IETF - A Novice's Guide to the Internet Engineering Task Force", RFC3160, August 2001.

[RFC2396] T. Berners-Lee: "Uniform Resource Identifiers", RFC 2396, August 1998.

[URLBODY] N. Freed and Keith Moore: "Definition of the URL MIME External-Body Access-Type", RFC 2017, October 1996.

[Hentze-Muto 2000] R. Hentze and A. Muto: Sending HTML in E-mail - Status Report 2000, May 2000, http://dsv.su.se/jpalme/ietf/mhtml.html#testprogs

8. Author's Addresses

Electrum 230
S-164 40 Kista, Sweden Fax: +46-8-783 08 29

Yvonne Backhans Phone: +46-8-6672600
Emmylundsvägen 3:921 Email: yvonnebackhans@hotmail.com
171 72 Solna, Sweden

Tina Hekkala Email: tinahek@yahoo.se
Albacken Phone: +46-8-55089893
152 93 Hölö, Sweden

Send questions and comments on this document also to:

Jacob Palme Email: jpalme@dsv.su.se
Skeppargatan 73 Phone: +46-8-16 16 67
115 30 Stockholm, Sweden

or join the MHTML mailing list
(http://dsv.su.se/jpalme/ietf/mhtml.html#mailing-list) and thereafter
send comments to that list.