Network Working Group
Request for Comments: 2184
Updates: 2045, 2047, 2183
Category: Standards Track

N. Freed
Innosoft
K. Moore
University of Tennessee

August 1997

MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations

Status of this Memo

1. Abstract

This memo defines extensions to the RFC 2045 media type and RFC 2183 disposition parameter value mechanisms to provide

(1) a means to specify parameter values in character sets other than US-ASCII,

(2) to specify the language to be used should the value be displayed, and

(3) a continuation mechanism for long parameter values to avoid problems with header line wrapping.

This memo also defines an extension to the encoded words defined in RFC 2047 to allow the specification of the language to be used for display as well as the character set.

2. Introduction

The Multipurpose Internet Mail Extensions, or MIME [RFC-2045, RFC-2046, RFC-2047, RFC-2048, RFC-2049], define a message format that allows for

(1) textual message bodies in character sets other than US-ASCII,

(2) non-textual message bodies,

(3) multi-part message bodies, and

(4) textual header information in character sets other than US-ASCII.

MIME is now widely deployed and is used by a variety of Internet protocols, including, of course, Internet email. However, MIME's success has resulted in the need for additional mechanisms that were not provided in the original protocol specification.

In particular, existing MIME mechanisms provide for named media type (content-type field) parameters as well as named disposition (content-disposition field). A MIME media type may specify any number of parameters associated with all of its subtypes, and any specific subtype may specify additional parameters for its own use. A MIME disposition value may specify any number of associated parameters, the most important of which is probably the attachment disposition's filename parameter.

These parameter names and values end up appearing in the content-type and content-disposition header fields in Internet email. This inherently imposes three crucial limitations:

(1) Lines in Internet email header fields are folded according to RFC 822 folding rules. This makes long parameter values problematic.

(2) MIME headers, like the RFC 822 headers they often appear in, are limited to 7bit US-ASCII, and the encoded-word mechanisms of RFC 2047 are not available to parameter values. This makes it impossible to have parameter values in character sets other than US-ASCII without specifying some sort of private per-parameter encoding.

(3) It has recently become clear that character set information is not sufficient to properly display some sorts of information -- language information is also needed [RFC-2130]. For example, support for handicapped users may require reading text string aloud. The language the text is written in is needed for this to be done correctly. Some parameter values may need to be displayed, hence there is a need to allow for the inclusion of language information.

The last problem on this list is also an issue for the encoded words defined by RFC 2047, as encoded words are intended primarily for display purposes.

This document defines extensions that address all of these limitations. All of these extensions are implemented in a fashion that is completely compatible at a syntactic level with existing MIME implementations. In addition, the extensions are designed to have as little impact as possible on existing uses of MIME.

IMPORTANT NOTE: These mechanisms end up being somewhat gibbous when they actually are used. As such, use of these mechanisms should not be used lightly; they should be reserved for situations where a real need for them exists.

2.1. Requirements notation

This document occasionally uses terms that appear in capital letters. When the terms "MUST", "SHOULD", "MUST NOT", "SHOULD NOT", and "MAY" appear capitalized, they are being used to indicate particular requirements of this specification. A discussion of the meanings of these terms appears in [RFC-2119].

3. Parameter Value Continuations

Long MIME media type or disposition parameter values do not interact well with header line wrapping conventions. In particular, proper header line wrapping depends on there being places where linear whitespace (LWSP) is allowed, which may or may not be present in a parameter value, and even if present may not be recognizable as such since specific knowledge of parameter value syntax may not be available to the agent doing the line wrapping. The result is that long parameter values may end up getting truncated or otherwise damaged by incorrect line wrapping implementations.

A mechanism is therefore needed to break up parameter values into smaller units that are amenable to line wrapping. Any such mechanism MUST be compatible with existing MIME processors. This means that

(1) the mechanism MUST NOT change the syntax of MIME media type and disposition lines, and

(2) the mechanism MUST NOT depend on parameter ordering since MIME states that parameters are not order sensitive. Note that while MIME does prohibit modification of MIME headers during transport, it is still possible that parameters will be reordered when user agent level processing is done.

The obvious solution, then, is to use multiple parameters to contain a single parameter value and to use some kind of distinguished name to indicate when this is being done. And this obvious solution is exactly what is specified here: The asterisk character ("*") followed by a decimal count is employed to indicate that multiple parameters are being used to encapsulate a single parameter value. The count starts at 0 and increments by 1 for each subsequent section of the parameter value. Decimal values are used and neither leading zeroes nor gaps in the sequence are allowed.

The original parameter value is recovered by concatenating the various sections of the parameter, in order. For example, the content-type field

```
Content-Type: message/external-body; access-type=URL;
 URL*0="ftp://";
 URL*1="cs.utk.edu/pub/moore/bulk-mailer/bulk-mailer.tar"
```

is semantically identical to

```
Content-Type: message/external-body; access-type=URL;
 URL="ftp://cs.utk.edu/pub/moore/bulk-mailer/bulk-mailer.tar"
```

Note that quotes around parameter values are part of the value syntax; they are NOT part of the value itself. Furthermore, it is explicitly permitted to have a mixture of quoted and unquoted continuation fields.

4. Parameter Value Character Set and Language Information

Some parameter values may need to be qualified with character set or language information. It is clear that a distinguished parameter name is needed to identify when this information is present along with a specific syntax for the information in the value itself. In addition, a lightweight encoding mechanism is needed to accomodate 8 bit information in parameter values.

Asterisks ("*") are reused to provide the indicator that language and character set information is present and encoding is being used. A single quote ("'") is used to delimit the character set and language information at the beginning of the parameter value. Percent signs ("%") are used as the encoding flag, which agrees with RFC 2047.

Specifically, an asterisk at the end of a parameter name acts as an indicator that character set and language information may appear at the beginning of the parameter value. A single quote is used to separate the character set, language, and actual value information in the parameter value string, and an percent sign is used to flag octets encoded in hexadecimal. For example:

```
Content-Type: application/x-stuff;
 title*=us-ascii'en-us'This%20is%20%2A%2A%2Afun%2A%2A%2A
```

Note that it is perfectly permissible to leave either the character set or language field blank. Note also that the single quote delimiters MUST be present even when one of the field values is omitted. This is done when either character set, language, or both are not relevant to the parameter value at hand. This MUST NOT be done in order to indicate a default character set or language -- parameter field definitions MUST NOT assign a default character set or lanugage.

4.1. Combining Character Set, Language, and Parameter Continuations

Character set and language information may be combined with the parameter continuation mechanism. For example:

```
Content-Type: application/x-stuff
 title*1*=us-ascii'en'This%20is%20even%20more%20
 title*2*=%2A%2A%2Afun%2A%2A%2A%20
 title*3="isn't it!"
```

Note that:

(1) Language and character set information only appear at the beginning of a given parameter value.

(2) Continuations do not provide a facility for using more than one character set or language in the same parameter value.

(3) A value presented using multiple continuations may contain a mixture of encoded and unencoded segments.

(4) The first segment of a continuation MUST be encoded if language and character set information are given.

(5) If the first segment of a continued parameter value is encoded the language and character set field delimiters MUST be present even when the fields are left blank.

5. Language specification in Encoded Words

RFC 2047 provides support for non-US-ASCII character sets in RFC 822 message header comments, phrases, and any unstructured text field. This is done by defining an encoded word construct which can appear in any of these places. Given that these are fields intended for display, it is sometimes necessary to associate language information with encoded words as well as just the character set. This specification extends the definition of an encoded word to allow the inclusion of such information. This is simply done by suffixing the character set specification with an asterisk followed by the language tag. For example:

```
From: =?US-ASCII*EN?Q?Keith_Moore?= <moore@cs.utk.edu>
```

6. IMAP4 Handling of Parameter Values

IMAP4 [RFC-2060] servers SHOULD decode parameter value continuations when generating the BODY and BODYSTRUCTURE fetch attributes.

7. Modifications to MIME ABNF

The ABNF for MIME parameter values given in RFC 2045 is:

```
parameter := attribute "=" value

attribute := token
             ; Matching of attributes
             ; is ALWAYS case-insensitive.
```

This specification changes this ABNF to:

```
parameter := regular-parameter / extended-parameter

regular-parameter := regular-parameter-name "=" value

regular-parameter-name := attribute [section]

attribute := 1*attribute-char

attribute-char := <any (US-ASCII) CHAR except SPACE, CTLs,
                   "*", "'", "%", or tspecials>

section := initial-section / other-sections

initial-section := "*1"
```

```
other-sections := "*" (("2" / "3" / "4" / "5" /
                        "6" / "7" / "8" / "9") *DIGIT) /
                       ("1" 1*DIGIT))

extended-parameter := (extended-initial-name "="
                       extended-value) /
                      (extended-other-names "="
                       extended-other-values)

extended-initial-name := attribute [initial-section] "*"

extended-other-names := attribute other-sections "*"

extended-initial-value := [charset] "'" [language] "'"
                          extended-other-values

extended-other-values := *(ext-octet / attribute-char)

ext-octet := "%" 2(DIGIT / "A" / "B" / "C" / "D" / "E" / "F")

charset := <registered character set name>

language := <registered language tag [RFC-1766]>
```

The ABNF given in RFC 2047 for encoded-words is:

```
encoded-word := "=?" charset "?" encoding "?" encoded-text "?="
```

This specification changes this ABNF to:

```
encoded-word := "=?" charset ["*" language] "?" encoded-text "?="
```

8. Character sets which allow specification of language

In the future it is likely that some character sets will provide facilities for inline language labelling. Such facilities are inherently more flexible than those defined here as they allow for language switching in the middle of a string.

If and when such facilities are developed they SHOULD be used in preference to the language labelling facilities specified here. Note that all the mechanisms defined here allow for the omission of language labels so as to be able to accomodate this possible future usage.

9. Security Considerations

This RFC does not discuss security issues and is not believed to raise any security issues not already endemic in electronic mail and present in fully conforming implementations of MIME.

10. References

[RFC-822]
Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

[RFC-1766]
Alvestrand, H., "Tags for the Identification of Languages", RFC 1766, March 1995.

[RFC-2045]
Freed, N. and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, Innosoft, First Virtual Holdings, December 1996.

[RFC-2046]
Freed, N. and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, Innosoft, First Virtual Holdings, December 1996.

[RFC-2047]
Moore, K., "Multipurpose Internet Mail Extensions (MIME) Part Three: Representation of Non-ASCII Text in Internet Message Headers", RFC 2047, University of Tennessee, December 1996.

[RFC-2048]
Freed, N., Klensin, J., Postel, J., "Multipurpose Internet Mail Extensions (MIME) Part Four: MIME Registration Procedures", RFC 2048, Innosoft, MCI, ISI, December 1996.

[RFC-2049]
Freed, N. and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, Innosoft, FIrst Virtual Holdings, December 1996.

[RFC-2060]
Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, December 1996.

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

[RFC-2130]
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.

[RFC-2183]
Troost, R., Dorner, S., and Moore, K., "Communicating Presentation Information in Internet Messages: The Content-Disposition Header", RFC 2183, August 1997.

11. Authors' Addresses

Ned Freed
Innosoft International, Inc.
1050 East Garvey Avenue South
West Covina, CA 91790
USA
tel: +1 818 919 3600 fax: +1 818 919 3614
email: ned@innosoft.com

Keith Moore
Computer Science Dept.
University of Tennessee
107 Ayres Hall
Knoxville, TN 37996-1301
USA
email: moore@cs.utk.edu