Network Working Group	J. F. Reschke
Internet-Draft	greenbytes
Obsoletes: 5987 (if approved)	April 15, 2011
Intended status: Standards Track	
Expires: October 17, 2011	

Character Set and Language Encoding for
Hypertext Transfer Protocol (HTTP) Header Field Parameters
draft-reschke-rfc5987bis-00

Abstract

By default, message header field parameters in Hypertext Transfer
Protocol (HTTP) messages cannot carry characters outside the ISO-8859-1
character set. RFC 2231 defines an encoding mechanism for use in
Multipurpose Internet Mail Extensions (MIME) headers. This document
specifies an encoding suitable for use in HTTP header fields that is
compatible with a profile of the encoding defined in RFC 2231.

Editorial Note (To be removed by RFC Editor before publication)

Distribution of this document is unlimited. Although this is not a work
item of the HTTPbis Working Group, comments should be sent to the
Hypertext Transfer Protocol (HTTP) mailing list at ietf-http-wg@w3.org,
which may be joined by sending a message with subject "subscribe" to
ietf-http-wg-request@w3.org.
Discussions of the HTTPbis Working Group are archived at http://
lists.w3.org/Archives/Public/ietf-http-wg/.
XML versions, latest edits and the issues list for this document are
available from http://greenbytes.de/tech/webdav/#draft-reschke-
rfc5987bis. A collection of test cases is available at http://
greenbytes.de/tech/tc2231/.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

By default, message header field parameters in HTTP ([RFC2616])
messages cannot carry characters outside the ISO-8859-1 character set
([ISO-8859-1]). RFC 2231 ([RFC2231]) defines an encoding mechanism for
use in MIME headers. This document specifies an encoding suitable for
use in HTTP header fields that is compatible with a profile of the
encoding defined in RFC 2231.
This document obsoletes [RFC5987]; the changes are summarized in
[Appendix Appendix A](#).

     *Note: in the remainder of this document, RFC 2231 is only
      referenced for the purpose of explaining the choice of features
      that were adopted; they are therefore purely informative.

     *Note: this encoding does not apply to message payloads
      transmitted over HTTP, such as when using the media type
      "multipart/form-data" ([RFC2388]).

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].
This specification uses the ABNF (Augmented Backus-Naur Form) notation
defined in [RFC5234]. The following core rules are included by
reference, as defined in [RFC5234], Appendix B.1: ALPHA (letters),
DIGIT (decimal 0-9), HEXDIG (hexadecimal 0-9/A-F/a-f), and LWSP (linear
whitespace).
Note that this specification uses the term "character set" for
consistency with other IETF specifications such as RFC 2277 (see

[RFC2277], Section 3). A more accurate term would be "character encoding" (a mapping of code points to octet sequences).

3. Comparison to RFC 2231 and Definition of the Encoding

RFC 2231 defines several extensions to MIME. The sections below discuss if and how they apply to HTTP header fields.
In short:

   *Parameter Continuations aren't needed (Section 3.1),

   *Character Set and Language Information are useful, therefore a
    simple subset is specified (Section 3.2), and

   *Language Specifications in Encoded Words aren't needed (Section
    3.3).

3.1. Parameter Continuations

Section 3 of [RFC2231] defines a mechanism that deals with the length limitations that apply to MIME headers. These limitations do not apply to HTTP ([RFC2616], Section 19.4.7).
Thus, parameter continuations are not part of the encoding defined by this specification.

3.2. Parameter Value Character Set and Language Information

Section 4 of [RFC2231] specifies how to embed language information into parameter values, and also how to encode non-ASCII characters, dealing with restrictions both in MIME and HTTP header parameters.
However, RFC 2231 does not specify a mandatory-to-implement character set, making it hard for senders to decide which character set to use.
Thus, recipients implementing this specification MUST support the character sets "ISO-8859-1" [ISO-8859-1] and "UTF-8" [RFC3629].
Furthermore, RFC 2231 allows the character set information to be left out. The encoding defined by this specification does not allow that.

3.2.1. Definition

The syntax for parameters is defined in Section 3.6 of [RFC2616] (with RFC 2616 implied LWS translated to RFC 5234 LWSP):

```
  parameter     = attribute LWSP "=" LWSP value

  attribute     = token
  value         = token / quoted-string

  quoted-string = <quoted-string, defined in [RFC2616], Section 2.2>
  token         = <token, defined in [RFC2616], Section 2.2>
```

In order to include character set and language information, this
specification modifies the RFC 2616 grammar to be:

```
parameter     = reg-parameter / ext-parameter

reg-parameter = parmname LWSP "=" LWSP value

ext-parameter = parmname "*" LWSP "=" LWSP ext-value

parmname      = 1*attr-char

ext-value     = charset  "'" [ language ] "'" value-chars
              ; like RFC 2231's <extended-initial-value>
              ; (see [RFC2231], Section 7)

charset       = "UTF-8" / "ISO-8859-1" / mime-charset

mime-charset  = 1*mime-charsetc
mime-charsetc = ALPHA / DIGIT
              / "!" / "#" / "$" / "%" / "&"
              / "+" / "-" / "^" / "_" / "`"
              / "{" / "}" / "~"
              ; as <mime-charset> in Section 2.3 of [RFC2978]
              ; except that the single quote is not included
              ; SHOULD be registered in the IANA charset registry

language      = <Language-Tag, defined in [RFC5646], Section 2.1>

value-chars   = *( pct-encoded / attr-char )

pct-encoded   = "%" HEXDIG HEXDIG
              ; see [RFC3986], Section 2.1

attr-char     = ALPHA / DIGIT
              / "!" / "#" / "$" / "&" / "+" / "-" / "."
              / "^" / "_" / "`" / "|" / "~"
              ; token except ( "*" / "'" / "%" )
```

Thus, a parameter is either a regular parameter (reg-parameter), as
previously defined in Section 3.6 of [RFC2616], or an extended
parameter (ext-parameter).
Extended parameters are those where the left-hand side of the
assignment ends with an asterisk character.
The value part of an extended parameter (ext-value) is a token that
consists of three parts: the REQUIRED character set name (charset), the
OPTIONAL language information (language), and a character sequence
representing the actual value (value-chars), separated by single quote
characters. Note that both character set names and language tags are
restricted to the US-ASCII character set, and are matched case-

insensitively (see [RFC2978], Section 2.3 and [RFC5646], Section
2.1.1).
Inside the value part, characters not contained in attr-char are
encoded into an octet sequence using the specified character set. That
octet sequence is then percent-encoded as specified in Section 2.1 of
[RFC3986].
Producers MUST use either the "UTF-8" ([RFC3629]) or the "ISO-8859-1"
([ISO-8859-1]) character set. Extension character sets (mime-charset)
are reserved for future use.

> *Note: recipients should be prepared to handle encoding errors,
> such as malformed or incomplete percent escape sequences, or non-
> decodable octet sequences, in a robust manner. This specification
> does not mandate any specific behavior, for instance, the
> following strategies are all acceptable:
>
> > -ignoring the parameter,
> >
> > -stripping a non-decodable octet sequence,
> >
> > -substituting a non-decodable octet sequence by a replacement
> > character, such as the Unicode character U+FFFD (Replacement
> > Character).
>
> *Note: the RFC 2616 token production ([RFC2616], Section 2.2)
> differs from the production used in RFC 2231 (imported from
> Section 5.1 of [RFC2045]) in that curly braces ("{" and "}") are
> excluded. Thus, these two characters are excluded from the attr-
> char production as well.
>
> *Note: the <mime-charset> ABNF defined here differs from the one
> in Section 2.3 of [RFC2978] in that it does not allow the single
> quote character (see also RFC Errata ID 1912 [Err1912]). In
> practice, no character set names using that character have been
> registered at the time of this writing.

3.2.2. Examples

Non-extended notation, using "token":

  foo: bar; title=Economy

Non-extended notation, using "quoted-string":

  foo: bar; title="US-$ rates"

Extended notation, using the Unicode character U+00A3 (POUND SIGN):

  foo: bar; title*=iso-8859-1'en'%A3%20rates

Note: the Unicode pound sign character U+00A3 was encoded into the
single octet A3 using the ISO-8859-1 character encoding, then percent-
encoded. Also, note that the space character was encoded as %20, as it
is not contained in attr-char.
Extended notation, using the Unicode characters U+00A3 (POUND SIGN) and
U+20AC (EURO SIGN):

    foo: bar; title*=UTF-8''%c2%a3%20and%20e2%82%ac%20rates

Note: the Unicode pound sign character U+00A3 was encoded into the
octet sequence C2 A3 using the UTF-8 character encoding, then percent-
encoded. Likewise, the Unicode euro sign character U+20AC was encoded
into the octet sequence E2 82 AC, then percent-encoded. Also note that
HEXDIG allows both lowercase and uppercase characters, so recipients
must understand both, and that the language information is optional,
while the character set is not.

3.3. Language Specification in Encoded Words

Section 5 of [RFC2231] extends the encoding defined in [RFC2047] to
also support language specification in encoded words. Although the
HTTP/1.1 specification does refer to RFC 2047 ([RFC2616], Section 2.2),
it's not clear to which header field exactly it applies, and whether it
is implemented in practice (see http://tools.ietf.org/wg/httpbis/trac/
ticket/111 for details).
Thus, this specification does not include this feature.

4. Guidelines for Usage in HTTP Header Field Definitions

Specifications of HTTP header fields that use the extensions defined in
Section 3.2 ought to clearly state that. A simple way to achieve this
is to normatively reference this specification, and to include the ext-
value [bnf] production into the ABNF for that header field.
For instance:

    foo-header = "foo" LWSP ":" LWSP token ";" LWSP title-param
    title-param = "title" LWSP "=" LWSP value
                / "title*" LWSP "=" LWSP ext-value
    ext-value   = <see RFC 5987, Section 3.2>

      *Note: The Parameter Value Continuation feature defined in Section
       3 of [RFC2231] makes it impossible to have multiple instances of
       extended parameters with identical parmname components, as the
       processing of continuations would become ambiguous. Thus,
       specifications using this extension are advised to disallow this
       case for compatibility with RFC 2231.

4.1. When to Use the Extension

Section 4.2 of [RFC2277] requires that protocol elements containing
human-readable text are able to carry language information. Thus, the
ext-value [bnf] production ought to be always used when the parameter
value is of textual nature and its language is known.
Furthermore, the extension ought to also be used whenever the parameter
value needs to carry characters not present in the US-ASCII ([USASCII])
character set (note that it would be unacceptable to define a new
parameter that would be restricted to a subset of the Unicode character
set).

4.2. Error Handling

Header field specifications need to define whether multiple instances
of parameters with identical parmname components are allowed, and how
they should be processed. This specification suggests that a parameter
using the extended syntax takes precedence. This would allow producers
to use both formats without breaking recipients that do not understand
the extended syntax yet.
Example:

```
  foo: bar; title="EURO exchange rates";
            title*=utf-8''%e2%82%ac%20exchange%20rates
```

In this case, the sender provides an ASCII version of the title for
legacy recipients, but also includes an internationalized version for
recipients understanding this specification -- the latter obviously
ought to prefer the new syntax over the old one.

> *Note: at the time of this writing, many implementations failed to
> ignore the form they do not understand, or prioritize the ASCII
> form although the extended syntax was present.

5. Security Considerations

The format described in this document makes it possible to transport
non-ASCII characters, and thus enables character "spoofing" scenarios,
in which a displayed value appears to be something other than it is.
Furthermore, there are known attack scenarios relating to decoding
UTF-8.
See Section 10 of [RFC3629] for more information on both topics.
In addition, the extension specified in this document makes it possible
to transport multiple language variants for a single parameter, and
such use might allow spoofing attacks, where different language
versions of the same parameter are not equivalent. Whether this attack
is useful as an attack depends on the parameter specified.

6. Acknowledgements

Thanks to Martin Duerst and Frank Ellermann for help figuring out ABNF details, to Graham Klyne and Alexey Melnikov for general review, to Chris Newman for pointing out an RFC 2231 incompatibility, and to Benjamin Carlyle and Roar Lauritzsen for implementer's feedback.

7. References

7.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
[RFC2978]	Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2978, October 2000.
[RFC3629]	Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
[RFC3986]	Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
[RFC5646]	Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.
[RFC5234]	Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
[ISO-8859-1]	International Organization for Standardization, "Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1", ISO/IEC 8859-1:1998, 1998.
[USASCII]	American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

7.2. Informative References

[Err1912]	RFC Errata, "Errata ID 1912, RFC 2978", .
[RFC2045]	Freed, N. and N.S. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
[RFC2047]	Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.
[RFC2231]	Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.

[RFC2277]	Alvestrand, H.T., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.
[RFC2388]	Masinter, L., "Returning Values from Forms: multipart/form-data", RFC 2388, August 1998.
[RFC5987]	Reschke, J. F., "Character Set and Language Encoding for Hypertext Transfer Protocol (HTTP) Header Field Parameters", RFC 5987, August 2010.

Appendix A. Changes from RFC 5987

This section summarizes the changes compared to [RFC5987]:

Appendix B. Change Log (to be removed by RFC Editor before publication)

Appendix B.1. Since RFC5987

Only editorial changes for the purpose of starting the revision process (obs5987).

Appendix C. Resolved issues (to be removed by RFC Editor before publication)

Issues that were either rejected or resolved in this version of this document.

Appendix C.1. obs5987

Type: change
julian.reschke@greenbytes.de (2011-04-15): Obsolete RFC 5987, summarize differences.

Appendix D. Open issues (to be removed by RFC Editor prior to publication)

Appendix D.1. edit

Type: edit
julian.reschke@greenbytes.de (2011-04-15): Umbrella issue for editorial fixes/enhancements.

Appendix D.2. impls

Type: change
julian.reschke@greenbytes.de (2011-04-15): Add implementation report.

Appendix D.3. iso-8859-1

Type: change

julian.reschke@greenbytes.de (2011-04-15): Remove requirement to support ISO-8859-1? It doesn't really help, and it is not implemented in IE9.

Author's Address

Julian F. Reschke Reschke greenbytes GmbH Hafenweg 16 Muenster, NW 48155 Germany EMail: julian.reschke@greenbytes.de URI: http://greenbytes.de/tech/webdav/