Network Working Group	J. Reschke
Internet-Draft	greenbytes
Intended status: Standards Track	May 19, 2009
Expires: November 20, 2009	

TOC

Application of RFC 2231 Encoding to Hypertext Transfer Protocol (HTTP) Headers
draft-reschke-rfc2231-in-http-02

Status of this Memo

Copyright Notice

Abstract

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

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

There are multiple HTTP headers that already use RFC 2231 encoding in practice (Content-Disposition) or might use it in the future (Link). The purpose of this document is to provide a single place where the generic aspects of RFC 2231 encoding in HTTP headers are defined.
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-rfc2231-in-http. A collection of test cases is available at http://greenbytes.de/tech/tc2231/.

Table of Contents

1. Introduction
2. Notational Conventions
3. A Profile of RFC 2231 for Use in HTTP
    3.1. Parameter Continuations
    3.2. Parameter Value Character Set and Language Information
        3.2.1. Examples
    3.3. Language specification in Encoded Words
4. Guidelines for Usage in HTTP Header Definitions
    4.1. When to Use the Extension
    4.2. Error Handling
5. Security Considerations
6. IANA Considerations
7. Acknowledgements
8. References
    8.1. Normative References
    8.2. Informative References
Appendix A. Change Log (to be removed by RFC Editor before publication)
    A.1. Since draft-reschke-rfc2231-in-http-00
    A.2. Since draft-reschke-rfc2231-in-http-01
Appendix B. Open issues (to be removed by RFC Editor prior to publication)
    B.1. edit
§ Author's Address

1. Introduction

TOC

By default, message header parameters in HTTP ([RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.)) messages can not carry characters outside the ISO-8859-1 character set ([ISO-8859-1] (International Organization for Standardization, “Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1,” 1998.)). RFC 2231 ([RFC2231] (Freed, N. and K. Moore, “MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations,” November 1997.)) defines an escaping mechanism for use in MIME headers. This document specifies a profile of that encoding for use in HTTP.

2. Notational Conventions

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

This specification uses the ABNF (Augmented Backus-Naur Form) notation defined in [RFC5234] (Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF,” January 2008.). The following core rules are included by reference, as defined in [RFC5234] (Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF,” January 2008.), Appendix B.1: ALPHA (letters), DIGIT (decimal 0-9), HEXDIG (hexadecimal 0-9/A-F/a-f) and LWSP (linear white space).

Note that this specification uses the term "character set" for consistency with other IETF specifications such as RFC 2277 (see [RFC2277] (Alvestrand, H., “IETF Policy on Character Sets and Languages,” January 1998.), Section 3). A more accurate term would be "character encoding" (a mapping of code points to octet sequences).

3. A Profile of RFC 2231 for Use in HTTP

TOC

RFC 2231 defines several extensions to MIME. The sections below discuss if and how they apply to HTTP.

In short:

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

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

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

3.1. Parameter Continuations

TOC

Section 3 of [RFC2231] (Freed, N. and K. Moore, “MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations,” November 1997.) defines a mechanism that deals with the length limitations that apply to MIME headers. These limitations do not apply to HTTP ([RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.), Section 19.4.7).
Thus in HTTP, senders MUST NOT use parameter continuations, and therefore recipients do not need to support them.

3.2. Parameter Value Character Set and Language Information

TOC

Section 4 of [RFC2231] (Freed, N. and K. Moore, “MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations,” November 1997.) 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 encoding, 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] (International Organization for Standardization, “Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1,” 1998.) and "UTF-8" [RFC3629] (Yergeau, F., “UTF-8, a transformation format of ISO 10646,” November 2003.).
Furthermore, RFC 2231 allows leaving out the character encoding information. The profile defined by this specification does not allow that.
The syntax for parameters is defined in Section 3.6 of [RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.) (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>
```

This specification extends the grammar to:

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

     reg-parameter = attribute LWSP "=" LWSP value

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

     ext-value     = charset  "'" [ language ] "'" value-chars
                   ; extended-initial-value,
                   ; defined in [RFC2231], Section 7

     charset       = %x55.54.46.2D.38 ; "UTF-8"
                   / %x49.53.4F.2D.38.38.35.39.2D.31 ; "ISO-8859-1"
                   / ext-charset

     ext-charset   = token ; see IANA charset registry
                   ; (<http://www.iana.org/assignments/character-sets>)

     language      = <Language-Tag, defined in [RFC4646], Section 2.1>

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

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

     attr-char     = ALPHA / DIGIT
                   / "-" / "." / "_" / "~" / ":"
                   / "!" / "$" / "&" / "+"
```

Thus, a parameter is either regular parameter (reg-parameter), as previously defined in Section 3.6 of [RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.), 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 a character sequence

representing the actual value (value-chars), separated by single quote characters.
Inside the value part, characters not contained in attr-char are encoded into an octet sequence using the specified character set. That octet sequence then is percent-encoded as specified in Section 2.1 of [RFC3986] (Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” January 2005.).
Producers MUST NOT use character sets other than "UTF-8" ([RFC3629] (Yergeau, F., “UTF-8, a transformation format of ISO 10646,” November 2003.)) or ISO-8859-1 ([ISO-8859-1] (International Organization for Standardization, “Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1,” 1998.)). Extension character sets (ext-charset) are reserved for future use.

3.2.1. Examples

TOC

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 using ISO-8859-1 into the single octet A3, 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%20%e2%82%ac%20rates
```

Note: the unicode pound sign character U+00A3 was encoded using UTF-8 into the octet sequence C2 A3, 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 lower-case and upper-case character, 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

TOC

Section 5 of [RFC2231] (Freed, N. and K. Moore, “MIME Parameter Value and Encoded Word Extensions:
Character Sets, Languages, and Continuations,” November 1997.) extends the encoding defined in [RFC2047] (Moore, K., “MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text,” November 1996.) to also support language specification in encoded words. Although the HTTP/1.1 specification does refer to RFC 2047 ([RFC2616] (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.), 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, the RFC 2231 profile defined by this specification does not include this feature.

4. Guidelines for Usage in HTTP Header Definitions

TOC

Specifications of HTTP headers that use the extensions defined in Section 3.2 (Parameter Value Character Set and Language Information) should clearly state that. A simple way to achieve this is to normatively reference this specification, and to include the ext-value production into the ABNF for that header.
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 RFCxxxx, Section 3.2>
```

[rfcno] (Note to RFC Editor: in the figure above, please replace "xxxx" by the RFC number assigned to this specification.)

4.1. When to Use the Extension

TOC

Section 4.2 of [RFC2277] (Alvestrand, H., “IETF Policy on Character Sets and Languages,” January 1998.) requires that protocol elements containing text can carry language information. Thus, the ext-value production should always be used when the parameter value is of textual nature.

Furthermore, the extension should also be used whenever the parameter value needs to carry characters not present in the US-ASCII ([USASCII] (American National Standards Institute, “Coded Character Set -- 7-bit American Standard Code for Information Interchange,” 1986.)) 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

TOC

Header specifications that include parameters should also specify whether same-named parameters can occur multiple times. If repetitions are not allowed (and this is believed to be the common case), the specification should state whether regular or the extended syntax takes precedence. In the latter case, this could be used by producers to use both formats without breaking recipients that do not understand the syntax. [anchor6] (Does not work as expected, see <http://greenbytes.de/tech/tc2231/#attfnboth> and <http://greenbytes.de/tech/tc2231/#attfnboth2>.)
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 should prefer the new syntax over the old one.

5. Security Considerations

TOC

This document does not discuss security issues and is not believed to raise any security issues not already endemic in HTTP.

6. IANA Considerations

TOC

There are no IANA Considerations related to this specification.

TOC

7. Acknowledgements

Thanks to Frank Ellermann for help figuring out ABNF details, and to Roar Lauritzsen for implementer's feedback.

8. References

TOC

8.1. Normative References

TOC

[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.
[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.
[RFC3629]	Yergeau, F., “UTF-8, a transformation format of ISO 10646,” RFC 3629, STD 63, November 2003.
[RFC4646]	Phillips, A. and M. Davis, “Tags for Identifying Languages,” BCP 47, RFC 4646, September 2006.
[RFC5234]	Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF,” STD 68, RFC 5234, January 2008.

8.2. Informative References

TOC

[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., “IETF Policy on Character Sets and Languages,” BCP 18, RFC 2277, January 1998.
[RFC3986]	Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” RFC 3986, STD 66, January 2005.

[USASCII] American National Standards Institute, “Coded Character Set -- 7-bit American Standard Code for Information Interchange,” ANSI X3.4, 1986.

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

TOC

A.1. Since draft-reschke-rfc2231-in-http-00

TOC

Use RFC5234-style ABNF, closer to the one used in RFC 2231.
Make RFC 2231 dependency informative, so this specification can evolve independantly.
Explain the ABNF in prose.

A.2. Since draft-reschke-rfc2231-in-http-01

TOC

Remove unneeded RFC5137 notation (code point vs character).

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

TOC

B.1. edit

TOC

Type: edit
julian.reschke@greenbytes.de (2009-04-17): Umbrella issue for editorial fixes/enhancements.

Author's Address

TOC

Julian F. Reschke
greenbytes GmbH
Hafenweg 16

	Muenster, NW 48155
	Germany
Email:	julian.reschke@greenbytes.de
URI:	http://greenbytes.de/tech/webdav/