

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 1, 2010

J. Reschke
greenbytes
April 30, 2010

Character Set and Language Encoding for Hypertext Transfer Protocol
(HTTP) Header Field Parameters
draft-reschke-rfc2231-in-http-12

Abstract

By default, message header field parameters in Hypertext Transfer Protocol (HTTP) messages can not 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 which is compatible to a profile of the encoding defined in [RFC 2231](#).

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

There are multiple HTTP header fields 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 header fields 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 [1], which may be joined by sending a message with subject "subscribe" to ietf-http-wg-request@w3.org [2].

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/>.

Note: as of February 2010, there were at least three independent implementations of the encoding defined in [Section 3.2](#): Konqueror (starting with 4.4.1), Mozilla Firefox, and Opera.

Status of This Memo

Internet-Draft

Charset/Language Encoding in HTTP

April 2010

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 1, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	4
2.	Notational Conventions	4
3.	Comparison to RFC 2231 and Definition of the Encoding	4
3.1.	Parameter Continuations	5
3.2.	Parameter Value Character Set and Language Information	5
3.2.1.	Definition	5
3.2.2.	Examples	7
3.3.	Language specification in Encoded Words	8
4.	Guidelines for Usage in HTTP Header Field Definitions	8
4.1.	When to Use the Extension	9
4.2.	Error Handling	9
5.	Security Considerations	10
6.	IANA Considerations	10
7.	Acknowledgements	10
8.	References	10
8.1.	Normative References	10
8.2.	Informative References	11
Appendix A.	Document History and Future Plans (to be removed by RFC Editor before publication)	12
Appendix B.	Change Log (to be removed by RFC Editor before publication)	12
B.1.	Since draft-reschke-rfc2231-in-http-00	12
B.2.	Since draft-reschke-rfc2231-in-http-01	12
B.3.	Since draft-reschke-rfc2231-in-http-02	13
B.4.	Since draft-reschke-rfc2231-in-http-03	13
B.5.	Since draft-reschke-rfc2231-in-http-04	13
B.6.	Since draft-reschke-rfc2231-in-http-05	13
B.7.	Since draft-reschke-rfc2231-in-http-06	13
B.8.	Since draft-reschke-rfc2231-in-http-07	13
B.9.	Since draft-reschke-rfc2231-in-http-08	13
B.10.	Since draft-reschke-rfc2231-in-http-09	13
B.11.	Since draft-reschke-rfc2231-in-http-10	13
B.12.	Since draft-reschke-rfc2231-in-http-11	14
Appendix C.	Resolved issues (to be removed by RFC Editor	

	before publication)	14
C.1.	edit	14
C.2.	nonorm2231	14

[1.](#) Introduction

By default, message header field parameters in HTTP ([\[RFC2616\]](#)) messages can not 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 which is compatible to a profile of the encoding defined in [RFC 2231](#).

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 white space).

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:

- o Parameter Continuations aren't needed ([Section 3.1](#)),
- o Character Set and Language Information are useful, therefore a simple subset is specified ([Section 3.2](#)), and
- o Language Specifications in Encoded Words aren't needed ([Section 3.3](#)).

Reschke

Expires November 1, 2010

[Page 4]

Internet-Draft

Charset/Language Encoding in HTTP

April 2010

[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 leaving out the character set information. 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:

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 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 then is 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 Editor Errata ID 1912 [\[3\]](#)). 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 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

[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 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 RFCxxxx, Section 3.2>
```

[rfcno: Note to RFC Editor: in the figure above, please replace

"xxxx" by the RFC number assigned to this specification.]]

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 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 could be used by 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.](#) IANA Considerations

There are no IANA Considerations related to this specification.

[7.](#) Acknowledgements

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

[8.](#) References

[8.1.](#) Normative References

- [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.
- [RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", [BCP 19](#), [RFC 2978](#), October 2000.

Reschke

Expires November 1, 2010

[Page 10]

Internet-Draft

Charset/Language Encoding in HTTP

April 2010

- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", [RFC 3629](#), STD 63, November 2003.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", [RFC 3986](#), STD 66, January 2005.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", [BCP 47](#), [RFC 5646](#), September 2009.
- [USASCII] American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

[8.2.](#) Informative References

- [RFC2045] Freed, N. and N. 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., "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.

URIs

- [1] <<mailto:ietf-http-wg@w3.org>>
- [2] <<mailto:ietf-http-wg-request@w3.org?subject=subscribe>>
- [3] <http://www.rfc-editor.org/errata_search.php?eid=1912>

Reschke

Expires November 1, 2010

[Page 11]

Internet-Draft

Charset/Language Encoding in HTTP

April 2010

[Appendix A](#). Document History and Future Plans (to be removed by RFC Editor before publication)

Problems with the internationalization of the HTTP Content-Disposition header field have been known for many years (see test cases at <<http://greenbytes.de/tech/tc2231/>>).

During IETF 72 (<<http://tools.ietf.org/wg/httpbis/minutes?item=minutes72.html>>), the HTTPbis Working Group shortly discussed how to deal with the underspecification of (1) Content-Disposition, and its (2) internationalization aspects. Back then, there was rough consensus in the room to move the definition into a separate draft.

This specification addresses problem (2), by defining a simple subset of the encoding format defined in [RFC 2231](#). A separate specification, [draft-reschke-rfc2183-in-http](#), is planned to address problem (1). Note that this approach was chosen because Content-Disposition is just an example for an HTTP header field using this kind of encoding. Another example is the currently proposed Link header field ([draft-nottingham-http-link-header](#)).

This document is planned to be published on the IETF Standards Track, so that other standards-track level documents can depend on it, such as the new specification of Content-Disposition, or potentially

future revisions of the HTTP Link Header specification.

Also note that this document specifies a proper subset of the extensions defined in [RFC 2231](#), but does not normatively refer to it. Thus, [RFC 2231](#) can be revised separately, should the email community decide to.

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

[B.1](#). Since [draft-reschke-rfc2231-in-http-00](#)

Use [RFC5234](#)-style ABNF, closer to the one used in [RFC 2231](#).

Make [RFC 2231](#) dependency informative, so this specification can evolve independently.

Explain the ABNF in prose.

[B.2](#). Since [draft-reschke-rfc2231-in-http-01](#)

Remove unneeded [RFC5137](#) notation (code point vs character).

Reschke

Expires November 1, 2010

[Page 12]

Internet-Draft

Charset/Language Encoding in HTTP

April 2010

[B.3](#). Since [draft-reschke-rfc2231-in-http-02](#)

And and resolve issues "charset", "repeats" and "[rfc4646](#)".

[B.4](#). Since [draft-reschke-rfc2231-in-http-03](#)

And and resolve issue "charsetmatch".

[B.5](#). Since [draft-reschke-rfc2231-in-http-04](#)

Add and resolve issues "badseq" and "tokenquotcharset".

[B.6](#). Since [draft-reschke-rfc2231-in-http-05](#)

Say "header field" instead of "header" in the context of HTTP.

[B.7](#). Since [draft-reschke-rfc2231-in-http-06](#)

Add an appendix discussing document history and future plans, to be removed before publication.

B.8. Since [draft-reschke-rfc2231-in-http-07](#)

Add and resolve issues "impl" and "rel-2388".

B.9. Since [draft-reschke-rfc2231-in-http-08](#)

Editorial improvements. Add and resolve issues "attrcharvstoken" and "tokengrammar".

B.10. Since [draft-reschke-rfc2231-in-http-09](#)

Add issues "i18n-spoofing", "iso8859", "parameter-abnf", and "when-ext-value". Add and resolve issues "[rfc2978](#)-normative", "[rfc3986](#)-normative" and "usascii-normative".

B.11. Since [draft-reschke-rfc2231-in-http-10](#)

Resolve issues "i18n-spoofing", "iso8859", "parameter-abnf", and "when-ext-value".

Add and resolve issue "charset-registered", "handling-multiple", "multiple-inst-spoofing", "repeated-param" and "value-abnf".

Update the KDE implementation note.

B.12. Since [draft-reschke-rfc2231-in-http-11](#)

In the prose in [Section 3.2](#), "ext-charset" -> "mime-charset". In [Section 4](#), avoid the use of "should" and "recommended". In [Section 4.1](#) clarify that the [RFC 2277](#) requirement is about human-readable text. Clarify parts that made it look as if this spec has a normative dependency on [RFC 2231](#) (new issue "nonorm2231").

[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.

C.1. edit

Type: edit

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

C.2. nonorm2231

Type: edit

julian.reschke@greenbytes.de (2010-04-23): It's not totally clear that the mentions of [RFC 2231](#) really are all informative.

Resolution (2010-04-28): Clarify title of the spec, plus text talking about [RFC 2231](#). Avoid saying "profile" in general.

Author's Address

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany

EMail: julian.reschke@greenbytes.de
URI: <http://greenbytes.de/tech/webdav/>