Workgroup: Network Working Group
Internet-Draft:
draft-bormann-core-media-content-type-format-03
Published: 18 February 2021
Intended Status: Standards Track
Expires: 22 August 2021
Authors: C. Bormann — Universität Bremen TZI
H. Birkholz — Fraunhofer SIT

On Media-Types, Content-Types, and related terminology

Abstract

There is a lot of confusion about media-types, content-types, and related terminology.

This memo is an attempt at clearing it up, so we can use consistent terminology in CoRE and related specifications. It also defines some ABNF that can be used in these specifications.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

[RFC1590] introduced media types and their registration. That document took MIME types from [RFC1521] and gave them a new name. At that time, the term "media type" was often used just for the major type ("text", "audio"), and what we call a media-type now was the combination of a type and a subtype. This lives on in [RFC6838], which does not even have an ABNF [RFC5234] production for media type. [RFC6838]'s predecessor, [RFC4288], supplied the ABNF shown in (Figure 1).

```
type-name = reg-name
subtype-name = reg-name

reg-name = 1*127reg-name-chars
reg-name-chars = ALPHA / DIGIT / "!" /
                 "#" / "$" / "&" / "." /
                 "+" / "-" / "^" / "_"
```

Figure 1: ABNF for type and subtype, cited from RFC 4288

[RFC6838], obsoleting [RFC4288], restricts the first character of a reg-name to alphanumeric. It contains the otherwise semantically equivalent ABNF shown in Figure 2, however adding prose comments that further limit the use of "." and "+".

```
type-name = restricted-name
subtype-name = restricted-name

restricted-name = restricted-name-first *126restricted-name-chars
restricted-name-first  = ALPHA / DIGIT
restricted-name-chars  = ALPHA / DIGIT / "!" / "#" /
                         "$" / "&" / "-" / "^" / "_"
restricted-name-chars =/ "." ; Characters before first dot always
                             ; specify a facet name
restricted-name-chars =/ "+" ; Characters after last plus always
                             ; specify a structured syntax suffix
```

Figure 2: ABNF for type and subtype, as defined from RFC 6838

2. Media-Type

Today, the term "media type" is now generally used for a registered combination of a type-name and a subtype-name, as well as for the specification that defines the semantics of this combination. We further disambiguate by calling the former a *media type name*. An ABNF definition of Media-Type-Name:

```
Media-Type-Name = type-name "/" subtype-name
```

Figure 3: Definition of Media-Type-Name

For the purposes of this memo, we define:

Media-Type-Name: A combination of a type-name and a subtype-name registered in [IANA.media-types], conventionally identified by the two names separated by a slash.

(This leaves the term "Media Type" for the actual specification that is registered under the Media-Type-Name.)

3. Content-Type

Media types can have parameters [RFC6838], some of which are defined by the media type specification to be mandatory. In HTTP and many other protocols, media-type-names and parameters are then used together in a "Content-Type" header field. HTTP [RFC7231] uses the ABNF in Figure 4:

```
Content-Type = media-type
media-type = type "/" subtype *( OWS ";" OWS parameter )
type       = token
subtype    = token
token          = 1*tchar
tchar          = "!" / "#" / "$" / "%" / "&" / "'" / "*"
               / "+" / "-" / "." / "^" / "_" / "`" / "|" / "~"
               / DIGIT / ALPHA
OWS        = *( SP / HTAB )
```

Figure 4: Content-Type ABNF from RFC 7231

In the ABNF as established by [RFC2616], parts of which became [RFC7231], the rule name media-type is used for a Media-Type-Name with parameters attached. We don't follow this inclusive use of media-type; note that [RFC2616] was quite confused about this term by claiming (Section 3.7 of [RFC2616]):

> Media-type values are registered with the Internet Assigned Number Authority (IANA [19]).

This clearly reverts to the understanding of Media-Type-Name we use.

Instead of prolonging this confusion, we define as a separate term:

Content-Type: A Media-Type-Name, optionally associated with parameters (separated from the media type name and from each other by a semicolon).

Removing the legacy HTAB characters now shunned in polite conversation, as well as some other cobwebs, we define the conventional textual representation of a Content-Type with the ABNF in Figure 5:

```
Content-Type   = Media-Type-Name *( *SP ";" *SP parameter )
parameter      = token "=" ( token / quoted-string )

token          = 1*tchar
tchar          = "!" / "#" / "$" / "%" / "&" / "'" / "*"
               / "+" / "-" / "." / "^" / "_" / "`" / "|" / "~"
               / DIGIT / ALPHA
quoted-string  = %x22 *qdtext %x22
qdtext         = SP / %x21 / %x23-5B / %x5D-7E
```

Figure 5: Definition of Content-Type

Note that there is a slight inconsistency between the "token" used here and the "reg-name"/"restricted-name" used above; since media type parameters probably will be defined within the guard rails set by [RFC7231], we need to use HTTP's more comprehensive definition here.

4. Content-Coding

Section 3.5 of [RFC2616] also introduced the term Content-Coding, a registered name for an encoding transformation that has been or can be applied to a representation:

```
content-coding   = token
```

Figure 6: Definition of content-coding as in RFC 2616

Confusingly, in HTTP the Content-Coding is then given in a header field called "Content-Encoding"; we **never** use this term (except when we are in error). Instead we define:

Content-Coding: a registered name for an encoding transformation that has been or can be applied to a representation.

Content-Codings are registered in the HTTP Content Coding Registry, a subregistry of [IANA.http-parameters]. We often use the "identity" Content-Coding, which is the identity transformation, and often fail to identify that Content-Coding by name, instead calling it "no Content-Coding".

5. Content-Format

CoAP, in Section 1 of [RFC7252], defines a Content-Format as the combination of a Content-Type and a Content-Coding, identified by a numeric identifier defined in the "CoAP Content-Formats" registry (a subregistry of [IANA.core-parameters]), but in more confusing words (it did not have the benefit of the present specifications).

Content-Format: the combination of a Content-Type and a Content-Coding, identified by a numeric identifier defined by the "CoAP Content-Formats" subregistry of [IANA.core-parameters].

Note that there has not been a conventional string representation of just the combination of a Content-Type and a Content-Coding; Content-Formats so far always are identified by their registered Content-Format numbers. However, there are applications where that is useful [I-D.keranen-core-senml-data-ct], so we define:

```
Content-Format = "0" / (POS-DIGIT *DIGIT)
Content-Format-String   = Content-Type ["@" content-coding]
```

Figure 7: Definition of Content-Format/-String

This allows the use of Content-Format-Strings such as "application/json@deflate" in place of the less self-describing content-format "11050", or other combinations that do not have a content-format number defined yet.

Content-Format-Strings MUST NOT explicitly use the content-coding value of "identity" (i.e., if an identity content-coding is desired, the entire optional part including the "@" sign is left out).

Note that a quoted string inside a content-type parameter might contain an "@" sign, so the parsing of Content-Format-Strings cannot be done in a too simplistic way.

6. Remaining ABNF

This specification uses the ABNF given in Figure 8, as originally defined in [RFC5234] and [RFC8866]:

```
DIGIT     =  %x30-39           ; 0 – 9
POS-DIGIT =  %x31-39           ; 1 – 9
ALPHA     =  %x41-5A / %x61-7A ; A – Z / a – z
SP        =  %x20
```

Figure 8: Commonly Used ABNF Definitions

7. Abbreviations

Media type names are sometimes abbreviated as "mt", and Content-Types as "ct". We propose not to use those abbreviations: Where the long form of the values can be used, the long form "Content-Type" can also be used to name them.

For historical reasons, both [RFC6690] and [RFC7252] use the abbreviation "ct" for Content-Format (think first and last character).

For Content-Coding, the abbreviation "cc" can be used.

8. Discussion

The ABNF given here is provisional and may need some more cleanup, such as unifying the various forms of reg-name, token, etc.

(ABNF just shown for illustration is centered, in a blockquote, and tagged with <artwork type="abnf;old"...> in the XML, while the normative ABNF of this memo is left-aligned and tagged with <sourcecode type="abnf"...>.)

The XPath expression //sourcecode[@type='abnf']/text() can be used on the XML form of this specification to extract the ABNF defined here.

We need to discuss case-insensitivity at some point, which is usually rather insensitive.

9. Suggested usage

9.1. COSE

Section 3.1 of [RFC8152] defines a common COSE header parameter (number 3) called "content type" in the description, to indicate the type of the data in the payload or ciphertext fields.

This header can either be an unsigned integer, indicating a CoRE Content-Format number, or a text string that is only defined in general terms. It points to Section 4.2 of [RFC6838] for 'text values following the syntax of "<type-name>/<subtype-name>"...', but also discusses the use of parameters and subparameters; no ABNF or similar detail specification is provided. The text does not discuss the use of Content-Coding in the text string form, probably because nothing like the present document existed at the time, creating a weird gap compared with numeric Content-Format-Strings. The text only has trivial changes in Section 3.1 of [I-D.ietf-cose-rfc8152bis-struct-15].

The present specification suggests using the production Content-Format-String as a more formal definition of the text string that can go into the "content type" (number 3) common header parameter in COSE.

9.2. SenML

As discussed above, Section 3 of [I-D.keranen-core-senml-data-ct] makes use of the present specification.

9.3. ...

(to be filled in along further use cases)

10. IANA Considerations

While this memo talks a lot about IANA registries, it does not require any action from IANA.

11. Security Considerations

Confusion about terminology may, in the worst case, cause security problems, as can loosely defined syntax elements of a specification. No other security considerations are known to be raised by the present specification.

12. References

12.1. Normative References

[IANA.core-parameters] IANA, "Constrained RESTful Environments (CoRE) Parameters", <http://www.iana.org/assignments/core-parameters>.

[IANA.http-parameters] IANA, "Hypertext Transfer Protocol (HTTP) Parameters", <http://www.iana.org/assignments/http-parameters>.

[IANA.media-types] IANA, "Media Types", <http://www.iana.org/assignments/media-types>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

[I-D.ietf-cose-rfc8152bis-struct-15]
Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", Work in Progress, Internet-Draft, draft-ietf-cose-rfc8152bis-struct-15, 1 February 2021, <https://www.ietf.org/archive/id/draft-ietf-cose-rfc8152bis-struct-15.txt>.

[I-D.keranen-core-senml-data-ct]
Keranen, A. and C. Bormann, "SenML Data Value Content-Format Indication", Work in Progress, Internet-Draft, draft-keranen-core-senml-data-ct-02, 8 July 2019, <https://www.ietf.org/archive/id/draft-keranen-core-senml-data-ct-02.txt>.

[RFC1521] Borenstein, N. and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC

1521, DOI 10.17487/RFC1521, September 1993, <https://www.rfc-editor.org/info/rfc1521>.

[RFC1590] Postel, J., "Media Type Registration Procedure", RFC 1590, DOI 10.17487/RFC1590, March 1994, <https://www.rfc-editor.org/info/rfc1590>.

[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, DOI 10.17487/RFC2616, June 1999, <https://www.rfc-editor.org/info/rfc2616>.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, DOI 10.17487/RFC4288, December 2005, <https://www.rfc-editor.org/info/rfc4288>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/info/rfc6690>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <https://www.rfc-editor.org/info/rfc8866>.

Acknowledgements

Matthias Kovatsch forced the authors to make up their minds about this. Ari Keränen forced them to write it up, then, and created a convincing use case of Content-Format-Strings. John Mattsson alerted us to a mistake. Alexey Melnikov suggested reviving this draft after a year of dormancy.

Authors' Addresses

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
64295 Darmstadt
Germany

Email: henk.birkholz@sit.fraunhofer.de