

Workgroup: MEDIAMAN
Internet-Draft: draft-ietf-mediaman-suffixes-07
Updates: [6838](#) (if approved)
Published: 2 March 2024
Intended Status: Standards Track
Expires: 3 September 2024
Authors: M. Sporny A. Guy
 Digital Bazaar Digital Bazaar
 Media Types with Multiple Suffixes

Abstract

This document updates RFC 6838 "Media Type Specifications and Registration Procedures" to describe how to interpret subtypes with multiple suffixes.

Status of This Memo

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 <https://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 3 September 2024.

Copyright Notice

Copyright (c) 2024 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
 - [1.1. Conventions Used in This Document](#)
- [2. Media Types with Multiple Suffixes](#)
 - [2.1. Processing Multiple Suffixes](#)
 - [2.2. Fragment Identifiers](#)
 - [2.3. Structured Syntax Name Suffixes](#)
 - [2.4. Structured Syntax Suffix Registration Template](#)
 - [2.5. Security Considerations](#)
 - [2.5.1. Document Validity for Suffixes](#)
 - [2.5.2. Fragment Semantics for Suffixes](#)
 - [2.5.3. Security Characteristics for Suffixes](#)
 - [2.5.4. Partial Processing of Suffixes](#)
- [3. Normative References](#)
- [Appendix A. IANA Considerations](#)
- [Appendix B. Acknowledgements](#)
- [Authors' Addresses](#)

1. Introduction

As written, RFC 6838 [RFC6838] permits the registration of media type subtype names which contain any number of occurrences of the "+" character. RFC 6838 defines the characters following the first "+" character to be a structured syntax suffix, but does not define anything further about how to interpret subtype names containing more than one "+" character.

This document updates RFC 6838 to clarify how to interpret subtype names containing more than one "+" character as subtypes with multiple suffixes.

As registration of media types which use a structured suffix has become widely supported, this enables further specialization of media types that build on already registered and well-defined media types which themselves use a structured suffix.

1.1. Conventions Used in This Document

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.

2. Media Types with Multiple Suffixes

This section is an addition to RFC 6838.

Media types MAY be registered with more than one structured suffix appended to the base subtype name. Characters on the left-most side of the left-most "+" in a subtype name specify the base subtype name. The entire structured suffix is all of the characters to the right of the left-most "+" sign in the media type, including the left-most "+" sign itself. The entire structured suffix MAY be composed of one or more broader structured suffixes. As an example, given the "application/foo+bar+baz" media type: "application" is the top-level type, "foo" is the base subtype name, "+bar+baz" is the entire structured suffix, and "+baz" is the broader structured suffix contained in the entire structured suffix.

2.1. Processing Multiple Suffixes

This section is an addition to RFC 6838.

Registered media types have clear processing rules. In cases where specific handling of the exact media type is not required, receivers of the media type MAY do generic processing on the underlying representation according to their ability to process any subset of the suffix(es) from left to right inclusive. In other words, an application can choose to ignore the base subtype name from a media type with multiple suffixes, and process according to the remaining media type suffix(es).

This sort of generic processing of a portion of the media type MAY be utilized by a processor that is capable of applying decoding rules associated with the portion of the structured syntax suffix in the [Structured Syntax Suffixes Registry](#).

When the entire structured suffix is composed of multiple structured suffixes, those structured suffixes MUST be considered in left-to-right order. This is done by considering the entire structured suffix first, and then dropping the left-most suffix for each iteration of the structured suffix variations that are considered. Note that "considering" a structured suffix variation does not mean that an implementation has to process the data associated with each variation, but it does mean that the variations that are considered do have a valid order in which they are considered.

For example, for the media type "application/vc+ld+json", applications can choose to process the underlying representation according to any one of the following processing models, in the following order:

1. application/vc+ld+json (as specified in the [Media Type Registry](#))
2. +ld+json (as specified in the [Structured Syntax Suffixes Registry](#))

3. +json (as specified in the [Structured Syntax Suffixes Registry](#))

This means that a processor considers processing the entire media type first, and then considers variations of the structured suffixes in most specific to least specific (left-to-right) order; for example, "+ld+json" (more specific) and then "+json" (less specific). The processor is never expected to process "+ld" alone when presented with a "application/vc+ld+json" media type for two reasons 1) doing so would be considered interpreting multiple structured suffixes out of order, and 2) "+ld" is not a registered structured suffix type.

If an application chooses to utilize a portion of the media type that is a structured syntax suffix, the suffix **MUST** exist as an entry in the [Structured Syntax Suffixes Registry](#) and the specification referred to in the "Encoding Considerations" entry of the registry **MUST** be used for both encoding and decoding the byte stream associated with the media type.

In order to gain the most benefit from the media that is presented, and to reduce the security attack surface for processed content, implementations that are able to process multiple suffixes **SHOULD** process the most specific media type or structured syntax suffix that they are able to process.

For processors that choose to process data associated with multiple structured suffixes, if processing data according to any registered structured suffix type fails, further processing on less specific structured suffixes **SHOULD NOT** be performed.

2.2. Fragment Identifiers

This section is an addition to RFC 6838.

The syntax and semantics for fragment identifiers are specified in the "Fragment Identifier Considerations" column in the IANA Structured Syntax Suffixes registry. In general, when processing fragment identifiers associated with a structured syntax suffix, the following rules **SHOULD** be followed:

1. For cases defined for the structured syntax suffix, where the fragment identifier does resolve per the structured syntax suffix rules, then proceed as specified by the specification associated with the "Fragment Identifier Considerations" column in the IANA Structured Syntax Suffixes registry.
2. For cases defined for the structured syntax suffix, where the fragment identifier does not resolve per the structured syntax suffix rules, then proceed as specified by the specification associated with the full media type.

3. For cases not defined for the structured syntax suffix, then proceed as specified by the specification associated with the full media type.

2.3. Structured Syntax Name Suffixes

The following paragraphs are additional guidance to Section 4.2.8 "Structured Syntax Name Suffixes", in RFC 6838.

Media types that make use of a named structured syntax, or similar separator such as a dash "-", MUST ensure that the registration is semantically aligned, from a data model perspective, with existing base subtype names in the media type registry. For example, for the media types "application/foo+bar" and "application/foo+baz", the expectation is that the semantics suggested by the base subtype name "application/foo" are the same between both media types. The Designated Expert MUST reject a registration if they believe the semantics for a media type registration does not align with existing base subtype names in the media type registry.

Registrants MUST prove to the Designated Expert, such as through an email to a public mailing list or issue tracker comment, that they have consent from the existing Change Controller for the associated base subtype name to register the new media type.

2.4. Structured Syntax Suffix Registration Template

This section replaces Section 6.2 "Structured Syntax Suffix Registration Template" in RFC 6838.

Media types containing more than one structured suffix MUST be registered according to the procedure defined in [RFC6838] and this document.

For structured suffixes containing more than one broader structured suffix, the structured suffix to the right of the one being registered MUST also be registered. For example, for the structured suffix "+foo+bar+baz", the registration order is "+baz", then "+bar+baz", and finally "+foo+bar+baz". Structured suffixes containing more than one broader structured suffix MAY register all of the suffixes at the same time.

This template describes the fields that must be supplied in a structured syntax suffix registration request:

Name

Full name of the well-defined structured syntax.

+suffix

Suffix used to indicate conformance to the syntax.

References

Include full citations for all specifications necessary to understand the structured syntax.

Encoding considerations

A full citation to a section in a specification that provides general guidance regarding encoding considerations for any type employing this syntax. The same requirements for media type encoding considerations given in Section 4.8 apply here.

Interoperability considerations

A full citation to a section in a specification that documents any issues regarding the interoperable use of types employing this structured syntax should be given here. Examples would include the existence of incompatible versions of the syntax, issues combining certain charsets with the syntax, or incompatibilities with other types or protocols.

Fragment identifier considerations

A full citation to a section in a specification that documents the generic processing rules of fragment identifiers for any type employing this syntax should be described here.

Security considerations

A full citation to a section in a specification that provides security considerations shared by media types employing this structured syntax must be specified here. The same requirements for media type security considerations given in Section 4.6 apply here, with the exception that the option of not assessing the security considerations is not available for suffix registrations.

Contact

Person or organization (including contact information) to contact for further information.

Author/Change controller

Person or organization (including contact information) authorized to change this suffix registration.

2.5. Security Considerations

This section is an addition to Section 7 "Security Considerations" in RFC 6838.

2.5.1. Document Validity for Suffixes

If a toolchain chooses to process a provided media type by using the selected structured suffix processing rules, it cannot presume that a document that is valid per the decoding rules associated with the structured suffix will be valid for a recognized subset of the

structured suffix. For example, presuming a media type of "application/foo+bar+baz", a toolchain cannot presume that a valid "+baz" document will also be a valid "+bar+baz" document nor a valid "application/foo+bar+baz" document. On the other hand, presuming a media type of "application/foo+bar+baz", a toolchain *can* presume that a valid "application/foo+bar+baz" document will also be a valid "+bar+baz" document and a valid "+baz" document.

2.5.2. Fragment Semantics for Suffixes

If a toolchain chooses to process a provided media type by using the selected structured suffix processing rules, it cannot presume that fragment identifier semantics will be the same across a recognized subset of the structured suffix. For example, presuming a media type of "application/foo+bar+baz", a toolchain cannot presume that the fragment semantics for a "+baz" document will be the same as for a "+bar+baz" document or the same as for an "application/foo+bar+baz" document.

2.5.3. Security Characteristics for Suffixes

Toolchains cannot assume that the security characteristics of processing based on structured suffixes will be the same for the entire media type nor for any combination of recognized structured suffixes. For example, presuming a media type of "application/foo+bar+baz", a toolchain cannot presume that the security considerations for a "+baz" document will be the same as for a "+bar+baz" document nor the same for an "application/foo+bar+baz" document.

2.5.4. Partial Processing of Suffixes

It is possible for an attacker to utilize multiple structured suffixes in a way that tricks unsuspecting toolchains into skipping important security checks and allowing viruses to propagate. For example, an attacker might utilize an "application/vnd.ms-excel.addin.macroEnabled.12+zip" structured suffix to trigger an unzip process that might then directly invoke Microsoft Excel, bypassing anti-virus tooling that would otherwise block a macro-enabled MS Excel file containing a virus of some kind from being scanned or opened.

Enterprising attackers might take advantage of toolchains that partially process media types in this manner. Toolchains that process media types based purely on a structured suffix need to ensure that further processing does not blindly trust the decoded data, and that proper magic header or file structure checking is performed, before allowing the decoded data to drive operations that might negatively impact the application environment or operating system.

3. Normative References

- [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>>.
- [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>>.
- [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>>.

Appendix A. IANA Considerations

[RFC6838] established the Registration Procedure for the Structured Syntax Suffixes Registry as "Expert Review". However, since the inception of the registry, the Designated Experts have been operating as if the Registration Procedure is "Specification Required" given that a specification is required in the registration template for the "References" entry, which defines how the structured suffix is to be used. Every entry in the Structured Syntax Suffixes Registry contains at least one reference to a specification. Furthermore, this document updates the Structured Syntax Suffixes Registry Registration Template to include links to specifications for most fields. Therefore, there is a clear requirement for at least one specification when performing a Structured Syntax Suffix registration.

This section updates the Registration Procedure for the Structured Syntax Suffixes Registry to "Specification Required" and instructs IANA to update the existing registry to reflect this change.

Appendix B. Acknowledgements

The editors would like to thank the following individuals for feedback on the specification (in alphabetical order): Harald Alvestrand, Amanda Baber, Martin J. Dürst, Ivan Herman, Graham Klyne, Murray S. Kucherawy, Darrel Miller, Mark Nottingham, Roberto Polli, Orië Steele, and Ted Thibodeau Jr.

Authors' Addresses

Manu Sporny
Digital Bazaar
203 Roanoke Street W.
Blacksburg, VA 24060
United States of America

Email: msporny@digitalbazaar.com

URI: <http://manu.sporny.org/>

Amy Guy

Digital Bazaar

203 Roanoke Street W.

Blacksburg, VA 24060

United States of America

Email: rhiaro@digitalbazaar.com

URI: <https://rhiaro.co.uk/>