Workgroup: MEDIAMAN
Internet-Draft:
draft-ietf-mediaman-suffixes-05
Published: 10 July 2023
Intended Status: Standards Track
Expires: 11 January 2024
Authors: M. Sporny
Digital Bazaar
A. Guy
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 11 January 2024.

**Table of Contents**



## 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 final "+" 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

The following paragraphs are additions 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 first "+" sign in the media type, including the initial "+" sign itself. The entire structured suffix MAY be composed of one or more other 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 other structured suffix contained in the entire structured suffix.

When the entire structured suffix is composed of multiple structured suffixes, those structured suffixes MUST be interpreted as ordered. For example, presume a media type that uses two suffixes, such as "application/foo+bar+baz", where "+bar+baz", "+bar", and "+baz" are registered structured suffixes. A processor is expected to process either the entire media type, or "+baz", or "+bar+baz". The processor is never expected to process "+bar" alone when presented with a "application/foo+bar+baz" media type, as that would be considered interpreting multiple structured suffixes out of order.

Media types containing more than one structured suffix MUST be registered according to the procedure defined in [RFC6838]. A new media type that utilizes a structured suffix MUST only be registered if the entire structured suffix is already registered in the Structured Syntax Suffixes registry. For example, a media type that uses two suffixes, such as "application/foo+bar+baz" is only permitted insofar as "+baz" and "+bar+baz" are already registered structured syntax suffixes.

### 2.1. Processing Multiple Suffixes

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 right to left 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.

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

1. application/did+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).

If an application choses 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.

### 2.2. Fragment Identifiers

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 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 as specified by the specification associated with the full media type.

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

Other advisory information, such as fragment processing not being defined in any existing specification, MAY be provided in the "Fragment Identifier Considerations" column in the IANA Structured Syntax Suffixes registry as long as the text is terse in nature.

### 2.3. Security Considerations

#### 2.3.1. Document Validity

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 or a valid "application/foo+bar+baz" document.

#### 2.3.2. Fragment Semantics

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.

#### 2.3.3. Suffix Security Characteristics

Toolchains cannot assume that the security characteristics of processing based on structured suffixes will be the same for the entire media type or 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 or the same for an "application/foo+bar+baz" document.

#### 2.3.4. Media Type Fibbing

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 would then invoke Microsoft Excel directly, 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 carelessly 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. Acknowledgements

The editors would like to thank the following individuals for feedback on the specification (in alphabetical order): Harald Alvestrand, Martin J. Dürst, Ivan Herman, Graham Klyne, Murray S. Kucherawy, Mark Nottingham, Roberto Polli, Orie 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/