Guidelines for the Definition of New Top Level Media Types

draft-duerst-mediaman-toplevel-00

Abstract

The goal of this document is to identify best practices for defining new top-level media types. It updates RFC 6838 [RFC6838], when approved. Comments and discussion about this document should be directed to media-types@ietf.org, the mailing list of the Media Type Maintenance (mediaman) WG.

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 7 September 2022.

Copyright Notice

Copyright (c) 2022 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.
1. Introduction

This document defines best practices for defining new top-level media types. RFC 6838 [RFC6838] very consisely defines the conditions for defining additional top-level media types. This document expands this and therefore updates RFC 6838 [RFC6838].

This document is currently a personal draft, but is intended for adoption by the Media Type Maintenance (mediaman) IETF WG. Comments and discussion about this document should be directed to that WG’s mailing list at media-types@ietf.org.

Currently, this document is a collection of information, ideas, and text snippets that may be helpful in creating the actual specification. None of the current language is intended to be final.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. Top-Level Media Type History

This section shortly describes the history of top-level media types, with a particular emphasis on the (rather rare) adoption of new top-level types.

RFC 1341 [RFC1341] first defined the structuring of content types into (top-level) type and subtype, and introduced the 'text', 'multipart', 'message', 'image', 'audio', 'video', and 'application' top-level types. That specification also allowed top-level types starting with 'X-'. With respect to new top-level types, it said the following:
An initial set of seven Content-Types is defined by this document. This set of top-level names is intended to be substantially complete. It is expected that additions to the larger set of supported types can generally be accomplished by the creation of new subtypes of these initial types. In the future, more top-level types may be defined only by an extension to this standard. If another primary type is to be used for any reason, it must be given a name starting with "X-" to indicate its non-standard status and to avoid a potential conflict with a future official name.

The first time an additional top-level type was defined was in RFC 1437 [RFC1437], but this was purely for entertainment purposes (please check date).

RFC 2046 [RFC2046] discouraged the use of "X-" for (new) top-level types, with the following words:

In general, the use of "X-" top-level types is strongly discouraged. Implementors should invent subtypes of the existing types whenever possible. In many cases, a subtype of "application" will be more appropriate than a new top-level type.

RFC 2048 [RFC2048], published at the same time as RFC 2046 [RFC2046], defined requirements for the definition of new top-level types:

In some cases a new media type may not "fit" under any currently defined top-level content type. Such cases are expected to be quite rare. However, if such a case arises a new top-level type can be defined to accommodate it. Such a definition must be done via standards-track RFC; no other mechanism can be used to define additional top-level content types.

RFC 4735 [RFC4735] introduced the 'example' top-level type for use in documentation examples.

At some point, the 'model' top-level type was introduced. Any pointers to the defining document are greatly appreciated.

The 'font' top-level media type was defined in RFC 8081 [RFC8081], a work of the 'justfont' IETF WG, in 2017.

There is ongoing work on defining a new 'haptics' top-level media type in draft-ietf-mediaman-haptics [HAPTICS].

Wikipedia (at https://en.wikipedia.org/wiki/Media_type) reports the unofficial use of a 'chemical' top-level type.
The document currently defining the requirements for new top-level media types is RFC 6838 [RFC6838]. Because we are trying to update what it says, we are citing the two relevant sections, Section 4.2.5 and Section 4.2.7, here.

4.2.5. Application Media Types

The "application" top-level type is to be used for discrete data that do not fit under any of the other type names, and particularly for data to be processed by some type of application program. This is information that must be processed by an application before it is viewable or usable by a user. Expected uses for the "application" type name include but are not limited to file transfer, spreadsheets, presentations, scheduling data, and languages for "active" (computational) material. (The last, in particular, can pose security problems that must be understood by implementors. The "application/postscript" media type registration in [RFC2046] provides a good example of how to handle these issues.)

For example, a meeting scheduler might define a standard representation for information about proposed meeting dates. An intelligent user agent would use this information to conduct a dialog with the user, and might then send additional material based on that dialog. More generally, there have been several "active" languages developed in which programs in a suitably specialized language are transported to a remote location and automatically run in the recipient’s environment. Such applications may be defined as subtypes of the "application" top-level type.

The subtype of "application" will often either be the name or include part of the name of the application for which the data are intended. This does not mean, however, that any application program name may simply be used freely as a subtype of "application"; the subtype needs to be registered.

4.2.7. Additional Top-Level Types

In some cases, a new media type may not "fit" under any currently defined top-level type names. Such cases are expected to be quite rare. However, if such a case does arise, a new type name can be defined to accommodate it. Definition of a new top-level type name MUST be done via a Standards Track RFC; no other mechanism can be used to define additional type names.
The two sections above are not strictly aligned, because the first says anything that doesn’t go under a more specific type can go under the 'application' top-level type, while the later section allows for new top-level types.

3. Potential Criteria for New Top-Level Media Types

This section describes potential criteria for new top-level media types, including criteria already defined in RFC 6838 [RFC6838]. Further work is needed to distinguish between required and optional criteria. But it is possible that we end up with just "we didn’t find any objective criteria for new top-level types, and we will stop looking for such criteria".

* New top level types are rare enough and different enough that each application needs to be evaluated separately.

* Need to be documented in a Standards Track RFC.

* This Standards Track RFC should include initial registrations of actual types.

* May (or may not) need an IETF WG for definition.

* Existence of a certain number of subtypes that would be grouped under the new top-level type. At a minimum, one actual subtype should exist. But the existence of a single subtype should not be enough; it should be clear that new similar types may appear in the future.

* Existing wide use of an undefined top-level type may be an indication of a need, and therefore an argument for formally defining this new top-level type.

* On the other hand, the use of undefined top-level types is highly discouraged.

* Top-level types mostly help humans; it is unclear to what extent top-level types are used by applications directly, as opposed to application dispatching and behavior triggered by the type/subtype combination. [More information needed/appreciated here.] Therefore, evaluating how a new top-level type helps humans understand types may be crucial. But as often with humans, opinions may widely differ.

* Need for clear criteria for what types do and don’t fall under the new top-level type.
* Desirability for common parameters: The fact that a group of (potential) types have (mostly) common parameters may be an indication that these belong under a common (new) top-level type.

4. IANA Considerations

There is currently no registry of top-level media types, but the list of top-level types available for registering subtypes is available at https://www.iana.org/assignments/media-types/media-types.xhtml.

There may be a question of whether there is a need for a formal registry of top-level types. Such a registry might contain pointers to the definitions of the top-level types. As a concrete example, the author of this document has not yet been able to find the definition of the 'model' top level type.

5. Security Considerations

This document as such is not expected to introduce any security issues. The security issues of introducing a new top-level media type MUST be evaluated and documented carefully.

Acknowledgements

The initial encouragement for writing this draft came from Harald Alvestrand. Further encouragement was provided by Murray S. Kucherawy. Both Harald and Murray also provided ideas for actual text. Without them, this memo would never have reached even the first draft stage.

References

Normative References


Informative References


Author's Address

Martin J. Dürst
Aoyama Gakuin University
Fuchinobe 5-10-1, Chuo-ku, Sagamihara, Kanagawa
252-5258
Japan
Phone: +81 42 759 6329
Email: duerst@it.aoyama.ac.jp
URI: https://www.sw.it.aoyama.ac.jp/D%C3%BCrst/
Abstract

This memo serves to register and document the 'haptics' top-level media type, under which subtypes for representation formats for haptics may be registered. This document also serves as a registration application for a set of intended subtypes, which are representative of some existing subtypes already in use.

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 23 November 2022.

Copyright Notice

Copyright (c) 2022 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.
1. Introduction

The term ‘haptics’ refers to the generation of touch-related sensations in a device or interface. Haptics is widely used in consumer devices in order to provide touch-based feedback to users. The most common use of haptics is in mobile devices, where it is used to provide feedback to users interacting with the touchscreen, e.g., typing on a virtual keyboard. Haptic technologies are unlike audio and visual enabling technologies in the sense that they require some form of actuation in order to create a tactile sensation. For mobile phones and game controllers, these actuators are typically small vibrating motors. For large touchscreens in vehicles, these actuators can be specialized piezoelectric materials. Haptic capabilities are found in nearly every modern smartphone and game and virtual reality controller, making these devices an ideal target for enhanced media experiences.

Internet Media Types [RFC6838] are used to label content carried over Internet protocols. This document defines a new top-level type ‘haptics’ according to Section 4.2.7 of [RFC6838]. This top-level type indicates that the content specifies haptic data. Under this top-level type, different representation formats of haptics may be registered.
1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. Background and Justification

Haptic signals provide an additional layer of entertainment and sensory immersion for the user. Haptic tracks, in separate files, can be combined with audio/video files and played back in sync to provide an overall immersive media experience (audio, visual, tactile) for the user. More recently, haptic tracks embedded in standard file formats such as ISOBMFF (ISO Base Media File Format), enable playback of the haptic signals over one or more actuators, simultaneously with audio and video playback. Haptic signals are also part of media streams that use RTP, such as those for streaming games, XR, and wearables.

2.1. MPEG ISOBMFF

Historically, there has not been a registration of formats for haptics. However, haptics has been proposed as a first-order media type (at the same level as audio and video) in ISOBMFF. This proposal was made to the MPEG Systems File Format sub-group in April 2020. The proposal was accepted, has since progressed to International Standard, and was published in January 2022 [ISOBMFF-IS]. Haptics is officially part of the ISO/IEC 14496-12 (ISOBMFF) standard, 7th Edition. Given this development, a strong case can be made for haptics to be added to the list of top-level media types recognized by the IETF.

We envision the following designations for haptics in mp4 files, once the top-level type ‘haptics’ is registered:

1. ‘haptics/mp4’ - mp4 files with just haptic tracks and no audio or video in them (e.g., streaming games, haptics files for haptic vests, belts, gloves, etc.)

2. ‘video/mp4’ - mp4 files with video, audio, and haptics (to ensure consistency with existing mp4 files with video content)

3. ‘audio/mp4’ - mp4 files with audio and haptics (to ensure consistency with existing mp4 files with audio content without any video)
2.2. Haptic Sub-modalities

There are multiple sub-modalities of haptics:

* Vibrotactile (touch, vibration)
* Kinesthetic (force feedback)
* Surface (surface friction)
* Spatial, non-contact (ultrasound)
* Thermal (temperature)

Therefore, designating ‘haptics’ as a top-level media type would enable the definition of data formats pertaining to these sub-modalities in a more streamlined manner. This would not be possible if ‘haptics’ were to be placed under other top-level types like ‘audio’, ‘video’, or ‘application’.

2.3. Another Human Sense

The top-level media type ‘audio’ pertains to the human sense of hearing, the top-level media type ‘video’ pertains to the human sense of seeing, so it only makes sense for the (equally important) human sense of touch to be represented by another top-level media type ‘haptics’. Placing ‘haptics’ under ‘audio’ or ‘video’ is not reflective of the kinds of files or use cases that would need haptics but have nothing whatsoever to do with audio or video.

2.4. Commercial Uptake

Haptics is rapidly becoming a standard feature of consumer electronic devices. For example:

* iPhone (206+ million units sold in 2020): native support for haptic encoded data
* Android (1.38+ billion units sold in 2020): API support of haptic buffers
* W3C (HTML vibration API [W3C-Vibration]): Optionally supported in mobile web browsers. W3C has also defined vibration extensions for gamepads [W3C-Gamepad]
* Game consoles (39+ million units sold in 2019): MS Xbox, Sony PlayStation, Nintendo Switch, etc.
* XR devices (9+ million units sold in 2019): OpenXR haptic API

Haptic media is expected to be commonly exchanged between these devices. Since they represent the majority of CE devices, a strong case can be made for ‘haptics’ as a top-level media type.

2.5. Haptic Data Formats in Use

There are multiple instances of existing haptic data formats that would live as sub-types under the proposed ‘haptics’ top-level media type. While these subtypes have *not* been registered with IANA or standardized (yet), the prevalence of these haptic data formats in a large number of devices around the world, pre-dating the standardization of haptic tracks in ISOBMFF, provides a compelling argument for ‘haptics’ to be designated as a top-level media type:

* ‘ahap’: The AHAP haptic data format [AHAP] is currently the standard encoding on all iOS devices + iOS connected game peripherals. The format has seen usage and adoption beyond Apple devices as well, with decoders available for Android and other XR systems.

* ‘ogg’: Google has introduced a proprietary extension to the OGG format in the latest version of Android 11. This encoding enables haptic media to be stored in OGG files.

* ‘ivs’: The IVS haptic data format is currently a vendor-specific format that is in use:
  - In mobile phones from LG Electronics (specifically, the models V30, V40, and the newest V50) that are sold worldwide
  - In gaming phones from ASUS (specifically, models ROG, ROG Phone II, ROG Phone 3) that are sold worldwide

* ‘hapt’: The HAPT haptic data format is currently a vendor-specific format that is in use:
  - In mobile haptic advertising (for W3C devices)
  - The following Japanese game developers use the HAPT format as part of Immersion’s TouchSense SDK:

Given the widespread use of these subtypes, it makes sense for 'haptics' to be a top-level media type.

### 2.6. Haptic Subtypes (envisioned standards)

The MPEG ISOBMFF proposal included an informative annex of known haptic coding formats with proposed FourCC codes for them. These codes are not registered yet, but the plan is indeed to standardize these haptic coding formats in the near future. Once standardized, they will also live as subtypes under the proposed 'haptics' top-level media type:

- 'hmpg': the selected coding format from the MPEG Call for Proposals on the Coded Representation of Haptics [MPEG-Haptics-CfP]
- 'hiee': IEEE P1918.1.1 vibrotactile coding standard [IEEE-P191811] being developed under the IEEE Tactile Internet initiative as part of the 5G URLL profile
- 'henm': enumerated effects haptic coding format (based on MIDI)
- 'havc': audio-to-vibe haptic coding format (automatic audio to vibration conversion algorithms)

### 2.7. 'application' top-level type not suitable

From the above arguments, it is clear that haptics does not really belong under any other media type. To reiterate, there are three main reasons why the 'haptics' media type does not fit under the 'application' top-level type:

* haptics connects to a sensory system, touch/motion, directly, and is more specific than the abstract 'application' type, and
* 'application' has historically been used for applications, i.e., code, which means it is viewed and treated with great care for security. 'haptics' is not code, just as 'audio' and 'video' are not code either.
haptics is a property of a media stream, it is not an application under any normal definition. As such, it should be its own type.

3. Security Considerations

Haptics are interpreted data structures that represent collections of different media rendering instructions intended to be decoded and rendered on target device hardware. Haptic data can be represented as collections of signal data and/or descriptive text in XML/JSON or similar format. Signal data is typically not executed by endpoint processors and represents minimal security risk. Descriptive text is typically parsed and represented in memory using standard XML data structures. This data is utilized to construct one or more signals that are sent to the endpoint device hardware.

Because of the media/rendering nature of the data path for haptic coded data the security profile of haptic data is expected to be largely consistent with the security profile of visual and audio media data.

As with any synthesized media data (audio, video, and haptics), there is a security risk associated with execution of commands based on the descriptive encoding either through its inherent extensibility or through the insertion of arbitrary executable data in the descriptive format itself. Indeed, media rendering systems are normally implemented with a mix of user and kernel space execution since these media must ultimately make their way to a hardware system. In theory, malicious instructions present in descriptive haptic media have the potential to execute arbitrary code in kernel space, effectively bypassing system permissions structures and/or execution sandboxes.

Haptics, audio, and video media have widespread use and careful attention should be paid by operating system and device driver implementors to ensure that synthesis and rendering signal paths do not provide attack surfaces for malicious payloads. Ultimately, any coded representation of haptic media is insufficient to implicitly provide sufficient security and this protection should be enforced by the operating system implementor.

4. IANA Considerations

This specification registers a new top-level type, ‘haptics’, in the standards tree, adds it as an alternative value of "Type Name" in the media types registration form [Media-Type-Registration], and registers several subtypes for it.
4.1. Definition and Encoding

'haptics' as the primary media content type indicates that the content identified by it requires a certain haptics subsystem such as low-level haptics APIs, which in turn will require hardware capabilities such as one or more actuators to render the haptics media. The 'haptics' media type does not provide any specific information about the underlying data format and how the haptics information should be interpreted -- the subtypes defined within a 'haptics' tree name the specific haptic formats. Unrecognized subtypes of 'haptics' should be treated as 'application/octet-stream'. Implementations may still pass unrecognized subtypes to the haptics subsystem and associated rendering hardware.

4.2. Registration Procedure

New haptics formats should be registered using the online form [Media-Type-Registration]. [RFC6838] should be consulted on registration procedures. In particular, the haptics specification should preferably be freely available.

Note that new parameter sub-values may be defined in the future. If an implementation does not recognize a sub-value in the comma-separated list, it should ignore the sub-value and continue processing the other sub-values in the list.

4.3. Subtype Registrations

In this section, the initial entries under the top-level 'haptics' media type are specified. They also serve as examples for future registrations.

4.3.1. IVS Haptics Type

Type name: haptics
Subtype name: ivs
Required parameters: None
Optional parameters: None
Encoding considerations: Text/binary
Interoperability considerations: The IVS format is a device-independent haptic effect coding. It is designed to enable interoperability between distinct physical endpoints. Not all devices may be able to render all effects present in an IVS file.

Applications that use this media type: All applications that are able to create, edit, or display haptic media content.

Additional information:

* File extension(s): Haptic file extensions used for IVS files: .ivs (xml) and .ivt (binary)
* Macintosh file type code(s): (no code specified)
* Macintosh Universal Type Identifier code: None
* Fragment Identifier: None
* Deprecated Alias: None

Person & email address to contact for further information: Yeshwant Muthusamy(ymuthusamy@immersion.com)

Change controller: Immersion Corporation

4.3.2. HAPT Haptics Type

Type name: haptics

Subtype name: hapt

Required parameters: None

Optional parameters: None

Encoding considerations: Text/binary

Interoperability considerations: The HAPT format is a device-dependent haptic effect coding based on the RIFF coding standard. It is designed to enable efficient coding of a device-specific haptic effect.

Published specification: HAPT is a logical extension of the RIFF standard [RIFF].

Applications that use this media type: All applications that are able to create, edit, or display haptic media content.
Additional information:

* File extension(s): Haptic file extensions used for HAPT files: .hapt

* Macintosh file type code(s): (no code specified)

* Macintosh Universal Type Identifier code: None

* Fragment Identifier: None

* Deprecated Alias: None

Person & email address to contact for further information: Yeshwant Muthusamy (ymuthusamy@immersion.com)

Change controller: Immersion Corporation

5. Normative References


6. Informative References


[MPEG-Haptics-CfP]
"MPEG Evaluates Responses to the Haptics Phase 1 Call for Proposals",
<https://www.mpegstandards.org/meetings/mpeg-136/>.

[W3C-Vibration]
"W3C Vibration API (Second Edition)",
<https://www.w3.org/TR/vibration/>.

[W3C-Gamepad]
"W3C Gamepad Extensions",

[IEEE-P191811]
"P1918.1.1 - Haptic Codecs for the Tactile Internet",
<https://standards.ieee.org/project/1918_1_1.html>.

[Media-Type-Registration]
"IANA, Application for a Media Type",
<http://www.iana.org/form/media-types>.

[RIIFF]
"Resource Interchange File Format",

Authors’ Addresses

Yeshwant K. Muthusamy
Immersion Corporation
2999 N.E. 191st Street, Suite 610
Aventura, FL 33180
United States of America
Phone: +1 469-583-2171
Email: ymuthusamy@immersion.com

Chris Ullrich
311 Court Ave
Ventura, CA 93003
United States of America
Phone: +1 805-320-0774
Email: chrisullrich@gmail.com
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 5 August 2022.

Copyright Notice

Copyright (c) 2022 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.
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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. They may also appear in lower or mixed case as plain English words, without any normative meaning.

2. Media Types with Multiple Suffixes

The following paragraphs are additions to RFC 6838.
Media types MAY be registered with more than one suffix appended to the base subtype name. The suffixes MUST be interpreted as ordered. Valid media type names containing a structured suffix are built from right to left (not left to right). Characters on the left-most side of the left-most "+" in a subtype name specify the base subtype name. Characters to the right of each "+" in a subtype name denote additional structured syntax suffixes.

Media types with more than one suffix MUST be registered according to the procedure defined in [RFC6838]. A new base subtype name MUST only be registered with suffix combinations that are already registered in their own right in the Structured Syntax Suffixes registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml).

For example, a media type that uses two suffixes, such as "application/foo+xml+gzip" is only permitted insofar as "+gzip" and "+xml" 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 and left-most "+" from a media type with multiple suffixes, and process according to the remaining media type suffix(es).

This sort of generic processing MAY be utilized in a processing pipeline where each segment of the pipeline handles a particular structured syntax suffix by applying decoding rules associated with the structured syntax suffix in the Structured Syntax Suffixes Registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml). The segment of the pipeline could then remove the structured syntax suffix from the media type and then pass the output of the decoding operation as well as the modified media type further down the pipeline.

For example, for the media type "application/did+ld+json", applications can choose to process the underlying representation according any of the following processing models: 1) application/did+ld+json (as specified in the Media Type Registry (https://www.iana.org/assignments/media-types/media-types.xhtml)), 2) +ld+json (as specified in the Structured Syntax Suffixes Registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml)), or 3) +json (as specified in the
Structured Syntax Suffixes Registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml). As a further example, for the media type "image/svg+xml+gzip", applications can choose to process the underlying representation according any of the following processing models: 1) image/svg+xml+gzip (as specified in the Media Type Registry (https://www.iana.org/assignments/media-types/media-types.xhtml)), 2) +gzip (as specified in the Structured Syntax Suffixes Registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml)), and then +xml (as specified in the Structured Syntax Suffixes Registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml)).

If an application chooses to utilize a portion of the media type that is a structured syntax suffix, the specification referred to in the the "Encoding Considerations" entry of the Structured Syntax Suffixes Registry (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml) MUST be used for both encoding and decoding the byte stream associated with the media type.

2.2. Security Considerations

2.2.1. Media Type Fibbing

It is possible for 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.

While the likelihood of these sorts of attacks are low, they are not zero and enterprising attackers might take advantage of applications that carelessly register themselves in a structured suffix processing toolchain. These sorts of toolchains need to ensure that the incoming media type is not blindly trusted and that proper magic header or file structure checking is performed before allowing the encoded data to drive operations that might negatively impact the application environment or operating system.

3. Normative References


Appendix A. Acknowledgements

The editors would like to thank the following individuals for feedback on the specification (in alphabetical order): Martin J. Duerst, Ivan Herman, Graham Klyne, Murray S. Kucherawy, 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/