Network Working Group
Internet-Draft
Obsoletes: [6838, 9694] (if approved)
Intended status: Best Current Practice
Expires: 27 September 2026

M. Nottingham
Cloudflare
P. Resnick
Episteme Technology Consulting

26 March 2026

Media Type Specifications and Registration Procedures
draft-ietf-mediaman-6838bis-08

Abstract

This document defines procedures for the specification and registration of media types for use in HTTP, MIME, and other Internet protocols.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-mediaman-6838bis/.

information can be found at https://datatracker.ietf.org/wg/mediaman/about/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-mediaman/6838bis/.

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 27 September 2026.

Table of Contents

1. Introduction

Internet application protocols (including but not limited to HTTP [RFC9110] and MIME [RFC2045]) are capable of carrying arbitrary labeled content.

Those labels are known as media types. A media type consists of a top-level type (Section 3) and a subtype (Section 4), which is further structured into a tree (identified by a prefix). A subtype can also be associated with a structured syntax (identified by suffix). Optionally, a media type can be defined to allow companion data, known as parameters.

Section 2 defines the criteria for registering media types. Section 5 outlines the procedures used to do so. The location of the media type registry is:

https://www.iana.org/assignments/media-types/

Section 6 outlines the procedures for managing the registry for structured syntax suffixes. It is located at:

https://www.iana.org/assignments/media-type-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.

This specification makes use of the Augmented Backus-Naur Form (ABNF) [RFC5234] notation, including the core rules defined in Appendix B of that document.

2. Media Type Registration Requirements

Media type registrations are expected to conform to various requirements laid out in the following sections. Note that specific requirements can vary depending on the registration tree (Section 4.1).

Other than IETF registrations in the standards tree, the registration of a media type does not imply endorsement, approval, or recommendation by the IANA or the IETF, and does not indicate that the specification is adequate for any particular purpose.

Additional requirements specific to the registration of XML media types are specified in [RFC7303].

2.1. Functionality

Media types MUST function as actual media formats. Registration of things that are better thought of as a transfer encoding, as a charset, or as a collection of separate entities of another type, is not allowed. For example, although applications exist to decode the base64 transfer encoding [RFC2045], base64 cannot be registered as a media type.

This requirement applies regardless of the registration tree involved.

2.1.1. Specification Availability

A permanent and readily available public specification of the format for the media type MUST exist for all types registered in the standards tree. This specification needs provide sufficient detail so that interoperability between independent implementations using the media type is possible. If not part of the media type registration proposal, this specification needs to be referenced by it.

A specification need not be publicly available for media types registered in the vendor and personal trees. Note, however, that the public availability of a specification will often make the difference between having a name reserved and having the potential for useful interoperation.

2.1.2. Intellectual Property

The registration of media types involving patented technology is permitted. However, the restrictions set forth in BCP 79 [RFC8179] and BCP 78 [RFC5378] on the use of patented technology in IETF Standards Track protocols must be respected when the specification of a media type is part of a Standards Track protocol. In addition, other standards-related organizations making use of the standards tree may have their own rules regarding intellectual property that must be observed in their registrations.

Intellectual Property Rights (IPR) disclosures for registrations in the vendor and personal trees are encouraged but not required.

Copyright on the registration template MUST allow the IANA to copy it into the IANA registry.

2.2. Canonicalization and Interoperability

All registered media types MUST employ a single, canonical data format, regardless of registration tree.

Ideally, media types will be defined so they interoperate across as many systems and applications as possible. However, some media types will inevitably have problems interoperating across different platforms. For example, problems with different versions, byte ordering, and specifics of gateway handling can arise.

Universal interoperability of media types is not required, but known interoperability issues should be identified whenever possible. Publication of a media type does not require an exhaustive review of interoperability, and the interoperability considerations section is subject to continuing evaluation.

Universal support and implementation of a media type are NOT a requirement for registration.

The recommendations in this subsection apply regardless of the registration tree involved.

2.3. Naming

All registered media types MUST be assigned top-level type and subtype names. The combination of these names serves to uniquely identify the media type, and the subtype name facet (or the absence of one) identifies the registration tree. Both top-level type and subtype names are case-insensitive.

Type and subtype names MUST conform to the following ABNF:

```
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
```

Note that this syntax is more restrictive than what is allowed by Section 5.1 of [RFC2045] or Section 4.2 of [RFC4288]. Also note that while this syntax allows type and subtype names of up to 127 characters, implementation limits may make such long names problematic. For this reason, 'type-name' and 'subtype-name' SHOULD be limited to 64 characters.

Although this syntax treats "." as equivalent to any other character, characters before any initial "." always specify the registration facet. Note that this means that facet-less standards tree registrations cannot use periods in the subtype name.

Similarly, the final "+" in a subtype name introduces a structured syntax specifier suffix. Structured syntax suffix requirements are specified in Section 4.2.

While it is possible for a given media type to be assigned more than one name, the use of different names to identify the same media type is discouraged.

These requirements apply regardless of the registration tree involved.

2.4. Parameters

Media types can be defined to allow or require use of media type parameters. Additionally, some parameters may be automatically made available to the media type by virtue of being a subtype of a content type that defines a set of parameters applicable to any of its subtypes.

In either case, the names, values, and meanings of any parameters MUST be fully specified when a media type is registered in the standards tree, and should be specified as completely as possible when media types are registered in the vendor or personal trees.

Parameter names have the same syntax as media type names and values:

```
parameter-name = restricted-name
```

Note that this syntax is more restrictive than what is allowed by the ABNF in [RFC2045] and amended by [RFC2231].

Parameter names are case-insensitive and no meaning is attached to the order in which they appear. It is an error for a specific parameter to be specified more than once.

There is no defined syntax for parameter values; therefore, it needs to be specified upon registration. Additionally, some transports impose restrictions on parameter value syntax, so care needs be taken to limit the use of potentially problematic syntaxes; for example, binary valued parameters, while permitted in some protocols, are best avoided.

Some parameters are reused across multiple media type definitions to provide common functionality. For example, the ’Codecs’ and ’Profiles’ Parameters for "Bucket" Media Types [RFC6381] identify media codecs used inside the container and their parameters. RTP payload formats have several common parameters: see [RFC4855], and [RFC8851].

Note that a protocol can impose further restrictions on parameter value syntax, depending on how it chooses to represent parameters. MIME [RFC2045] [RFC2231] allows binary parameters as well as parameter values expressed in a specific charset, but other protocols may be less flexible. For example, HTTP obsoletes field values containing characters outside the ASCII range (Section 5.5 of [RFC9110]), requiring field definitions to use encoding mechanisms like [RFC8187] to support other characters.

Media types registered in the standards tree MUST NOT subsequently add backwards-incompatible functionality through the addition of parameters. New parameters MAY be used to convey additional information so long as its processing is backwards-compatible, so that existing implementations can still handle the message successfully. Media types registered in the vendor and personal trees SHOULD NOT violate this requirement.

Changes to parameters (including the introduction of new ones) is managed in the same manner as other changes to the media type; see Section 5.5.

2.5. Encoding

Some transports impose restrictions on the type of data they can carry. For example, Internet mail traditionally was limited to 7bit US-ASCII text. Encoding schemes are often used to work around such transport limitations.

An "encoding considerations" field is provided to note what sort of data a media type can consist of as part of its registration. Possible values of this field are:

7bit: The content of the media type consists solely of CRLF-delimited 7bit US-ASCII text.

8bit: The content of the media type consists solely of CRLF-delimited 8bit text.

binary: The content consists of an unrestricted sequence of octets.

framed: The content consists of a series of frames or packets without internal framing or alignment indicators. Additional out-of-band information is needed to interpret the data properly, including but not limited to knowledge of the boundaries between successive frames and knowledge of the transport mechanism. Note that media types of this sort cannot be stored in a file or transported as a stream of octets without further context; therefore, such media types are thus unsuitable for use in many traditional protocols. A commonly used transport with framed encoding is the Real-time Transport Protocol, RTP. Additional rules for framed encodings defined for transport using RTP are given in [RFC4855].

Additional restrictions on 7bit and 8bit text are given in Section 4.1.1 of [RFC2046].

2.6. Fragment Identifiers

Media type registrations can specify how applications should interpret fragment identifiers (specified in Section 3.5 of [RFC3986]) associated with the media type.

Media types that use a structured syntax suffix MUST follow any fragment identifier rules specified for it.

Media types are encouraged to adopt fragment identifier schemes that are used with semantically similar media types.

2.7. Security

All registrations of types in the standards tree MUST include an analysis of security issues. A similar analysis for media types registered in the vendor or personal trees is encouraged but not required.

All descriptions of security issues need to be as accurate as possible regardless of registration tree. In particular, the security considerations MUST NOT state that there are "no security issues associated with this type". Security considerations for types in the vendor or personal tree MAY say that "the security issues associated with this type have not been assessed".

There is no requirement that media types registered in any tree be secure or completely free from risks. Nevertheless, all known security risks need to be identified in the registration of a media type, again regardless of registration tree.

The security considerations section of all registrations is subject to continuing evaluation and modification, and in particular can be extended by use of the "comments on media types" mechanism described in Section 5.4 below.

Issues that need to be described in a security analysis of a media type include:

* Processing of complex media types might modify or delete a recipient’s files or trigger actions on other resources. If unrestricted, this could have devastating effects. See the registration of the application/postscript media type in [RFC2046] for an example of description and handling of these issues.

* Any security analysis MUST state whether or not the format employs such "active content"; if it does, it MUST state what steps have been taken (or are required be taken by applications) of the media type to protect users of the media type.

* Processing of complex media types might institute actions that, while not directly harmful to the recipient, may result in disclosure of information that either facilitates a subsequent attack or else violates a recipient’s privacy in some way. Again, the registration of the application/postscript media type illustrates how such directives can be handled.

* A media type that employs compression may provide an opportunity for sending a small amount of data that, when received and evaluated, expands enormously to consume all of the recipient’s resources. All media types should state whether or not they employ compression; if they do, they should discuss what steps need to be taken to avoid such attacks.

* A media type might be designed for applications that require an assurance of security without providing that assurance. For example, a media type could be defined for storage of sensitive medical information that in turn requires external confidentiality and integrity protection services, or which is designed for use only within a secure environment. Types should always document whether or not they need such services in their security considerations.

2.8. Intended Usage

The "Intended usage" field in the registration template MUST contain one of "COMMON", "LIMITED USE", or "OBSOLETE".

Registrations intended for general use should be registered as COMMON. LIMITED USE accommodates media types that are only intended to be used in closed or specialized systems. OBSOLETE indicates that the media type should no longer be used.

Limited-use media types should also note in the applications list whether or not that list is exhaustive.

The "Restrictions on usage" field can convey additional information about restrictions on where the media type can be used.

2.9. Contact Information

The registration template contains distinct fields for contact information:

* The "Person to contact for further information" is intended to be used by those with questions about the media type and its use. This might be an individual with technical knowledge or a support address.

* The "Author" is the person(s) to be attributed with creation of the media type specification.

* The "Change controller" is the party who will authorise any future changes to the status or details of the registration.

Each should contain a name (of a person or other entity) and e-mail address. Optionally, they may also include a URL.

All three are required for permanent registrations; only the change controller is required for provisional registrations.

2.10. Additional Information

The following optional information should be included in the specification of a media type if it is available:

* Deprecated Aliases. In some cases, a single media type may have been widely deployed using multiple names prior to registration. In such cases, a preferred name MUST be chosen for the media type, and applications are required to use this to be compliant with the type's registration. However, a list of deprecated aliases by which the type is known can be supplied as additional information in order to assist applications in processing the media type properly.

* Magic number(s) (length, octet values). Magic numbers are byte sequences that are always present at a given place in the file, and can be used to identify it as being of a given media type.

* File name extension(s) commonly used on one or more platforms to indicate that some file contains a given media type.

* macOS Uniform Type Identifier (a string), if it makes sense to exchange media of this type through user-triggered exchange mechanisms such as copy-and-paste or drag-and-drop on macOS and related platforms (see [MacOSUTIs] for definitions and syntax).

* Windows clipboard name (a string), if it makes sense to exchange media of this type through user-triggered exchange mechanisms such as copy-and-paste or drag-and-drop on Microsoft Windows and related platforms (see [windowsClipboardNames] for definitions and syntax).

When registering a media type in the standards tree, specification authors can provide this information in the formal specification of the format, by incorporating the IANA media type registration form into the specification itself.

3. Top-Level Media Types

The list of top-level types is maintained in the IANA Top-Level Media Types registry at:

https://www.iana.org/assignments/top-level-media-types/

Top-level types can place various restrictions on the media types that use them. New media types MUST conform to the restrictions (if any) of their top-level type.

3.1. Additional Top-Level Types

In some cases, a new media type may not be easily classified 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 top-level type can be defined to accommodate it.

Registration of a new top-level type requires Standards Action in the IETF and, hence, the publication of a RFC on the Standards Track.

3.1.1. Required Criteria

Definitions of new top-level types are required to fulfil the following criteria:

* The top-level type is defined in a Standards Track RFC (see Section 4.9 of [RFC8126]). This will make sure there is sufficient community interest, review, and consensus.

* The IANA Considerations section of that RFC requests that IANA add this new top-level type to the registry of top-level types.

* The criteria for what types do and do not fall under the new top-level type are defined clearly. This will help the Designated Expert(s) to evaluate whether a subtype belongs below the new type or not, and whether the registration template for a subtype contains the appropriate information. If the criteria cannot be defined clearly, this is a strong indication that whatever is being talked about is not suitable as a top-level type.

* The RFC clearly documents security considerations applying to all or a significant subset of subtypes.

* At the minimum, one subtype (not including a potential 'example' subtype) is described. A top-level type without any subtype serves no purpose. The only exception is the 'example' top-level type, which disallows registration of subtypes.

3.1.2. Additional Considerations

Additional considerations for the definition of a new top-level type include:

* Existing wide use of an unregistered top-level type may be an indication of a need, and therefore an argument for formally defining a new top-level type. On the other hand, the use of unregistered top-level types is highly discouraged.

* Use of an IETF Working Group to define a new top-level type is not needed, but may be advisable in some cases. There are examples of new top-level type definitions without a Working Group ([RFC2077]), with a short, dedicated WG ([RFC8081]), and with a Working Group that included other related work ([RFC9695]).

* The document defining the new top-level type should include initial registrations of actual subtypes. The exception may be a top-level type similar to 'example'. This will help to show the need for the new top-level type, will allow checking the appropriateness of the definition of the new top-level type, will avoid separate work for registering an initial slate of subtypes, and will provide examples of what is considered a valid subtype for future subtype registrations.

* The registration and actual use of a certain number of subtypes under the new top-level type should be expected. The existence of a single subtype should not be enough; it should be clear that new similar types may appear in the future. Otherwise, the creation of a new top-level type is likely unjustified.

* The proposers of the new top-level type and the wider community should be willing to commit to emitting and consuming the new top-level type in environments that they control.

* 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.

* Top-level types can help humans with understanding and debugging. Therefore, evaluating how a new top-level type helps humans understand types may be crucial.

* Common restrictions may apply to all subtypes of a top-level type. Examples are the restriction to CRLF line endings for subtypes of type 'text' (at least in the context of electronic mail), or on subtypes of type 'multipart'.

* Top-level types are also used frequently in dispatching code. For example "multipart/*" is frequently handled as multipart/mixed, without understanding of a specific subtype. The top-level types 'image', 'audio', and 'video' are also often handled generically. Documents with these top-level types can be passed to applications handling a wide variety of image, audio, or video formats. HTML generating applications can select different HTML elements (e.g. <img> or <audio>) for including data of different top-level types. Applications can select different icons to represent unknown types in different top-level types.

3.1.3. Negative Criteria

Negative indicators for creation of a new top-level type include:

* Media types are not a general type system. A top-level type whose main or only purpose is to map other type systems, protocol elements, or registration spaces is not appropriate. Examples of such discouraged uses include mapping media types to programming language primitives, ontologies, object identifiers, URIs and URI schemes, and file extensions. That said, media types can use parameters to carry such information. For example, information on a file extension '.dcat' can be encoded as 'application/octet-string; filename=foo.dcat'.

* A new top-level type should not generate aliases for existing widely used types or subtypes.

* Top-level types with an "X-" prefix cannot be registered, and ought not be used. See [RFC6648].

4. Media Subtypes

4.1. Registration Trees

To increase the efficiency and flexibility of the registration process, different structures of subtype names can be registered in "trees," distinguished with faceted prefixes.

For example, a subtype that is recommended for wide support and implementation by the Internet community would be registered in the standards tree and not have a prefix, while a subtype that is used to move files associated with proprietary software would be registered in the vendor tree, and so its subtype name would begin with a "vnd." prefix.

4.1.1. Standards Tree

The standards tree is intended for those media types that require a substantive review and approval process in a recognized standards-related organization (as defined by https://www.iana.org/assignments/iesg-recognized-organizations (https://www.iana.org/assignments/iesg-recognized-organizations)). For media types that do not require such a process, see the vendor and personal trees.

Registrations in the standards tree are either:

1. approved directly by the IESG, or

2. registered by a recognized standards-related organization using the "Specification Required" IANA registration policy Section 4.6 of [RFC8126] (which implies Expert Review), or

3. approved by the Designated Expert(s) as identifying a "community format", as described in Section 4.1.1.1.

The first procedure is used for registrations from IETF consensus documents on the IETF stream, and can be used for RFCs from other streams.

In the second case, the IESG makes a one-time decision on whether the registration submitter represents a recognized standards-related organization; after that, registration requests are performed as specified in Section 5.3. The format is required to be described by a formal specification produced by the submitting standards-related organization.

The third case is described in Section 4.1.1.1.

Media types registered by the IETF in the standards tree MUST be published as RFCs. Standards-tree registrations for media types defined by other standards-related organizations MUST be described by a formal specification produced by that organization. Note that in both cases, the early allocation process described in [RFC7120] is available.

Media types in the standards tree do not have faceted subtype names.

The change controller of a media type registered in the standards tree is assumed to be the standards-related organization itself. In the case of IETF standards and community formats (see Section 4.1.1.1), the change controller is normally the IETF.

Modification or alteration of the specification uses the same level of processing (e.g., a registration submitted on Standards Track can be revised in another Standards Track RFC, but cannot be revised in an Informational RFC) required for the initial registration.

4.1.1.1. Community Formats in the Standards Tree

Some formats are interoperable (i.e., they are supported by more than one implementation), but their specifications are not published by a recognized standards-related organization. To accommodate these cases, the Designated Expert(s) are empowered to approve registrations in the standards tree that meet the following criteria:

* There is a well-defined specification for the format
* That specification is not tied to or heavily associated with one implementation
* The specification is freely available at a stable location
* There are multiple interoperable implementations of the specification, or they are likely to emerge
* The requested media type name is appropriate to the use case, and not so generic that it may be considered ’squatting’
* There is no conflict with IETF work or work at other recognised SDOs (present or future)
* There is evidence of broad adoption

The Designated Expert(s) have discretion in applying these criteria; in rare cases, they might judge it best to register an entry that fails one or more. The intent is to assure that successfully deployed community formats have registered media types. As such, the criteria above are designed to preclude anticipatory registrations.

Note that such registrations still go through preliminary community review (Section 5.1), and decisions can be appealed (Section 5.3).

4.1.2. Vendor Tree

The vendor tree is intended for media types associated with publicly available products. "Vendor" and "producer" are construed very broadly in this context, and are considered equivalent.

A registration may be placed in the vendor tree by anyone who needs to interchange data associated with some product or set of products. However, the registration properly belongs to the vendor or organization producing the software that employs the type being registered, and that vendor or organization can at any time elect to assume change control of a registration done by a third party in order to correct or update it. See Section 5.5 for additional information.

When a third party registers a type on behalf of someone else, both entities SHOULD be noted in the Change Controller field in the registration. One possible format for this would be "Foo, on behalf of Bar".

Vendor tree registrations are distinguished by the leading facet "vnd.". That may be followed, at the discretion of the registrant, by either a subtype name from a well-known producer (e.g., "vnd.mudpie") or by an IANA-approved designation of the producer's name that is followed by a media type or product designation (e.g., vnd.bigcompany.funnypictures).

While public exposure and review of media types to be registered in the vendor tree are not required, requesting review on the media-types@ietf.org mailing list is encouraged, to improve the quality of those specifications.

Registrations in the vendor tree may be submitted directly to the IANA, where they will undergo Expert Review Section 4.5 of [RFC8126] prior to approval.

4.1.3. Personal Tree

The personal tree is intended for media types created experimentally or as part of products that are not distributed commercially. This tree is sometimes referred to as the "vanity" tree.

Personal tree registrations are distinguished by the leading facet "prs.".

The change controller of a "personal" registration is the person or entity making the registration, or one to whom responsibility has been transferred as described below.

While public exposure and review of media types to be registered in the personal tree are not required, requesting review on the media-types@ietf.org mailing list is encouraged, to improve the quality of those specifications.

Registrations in the personal tree may be submitted directly to the IANA, where they will undergo Expert Review Section 4.5 of [RFC8126] prior to approval.

4.1.4. Unregistered x. Tree

Subtype names with "x." as the first facet are intended exclusively for use in private, local environments. Subtypes using this tree cannot be registered and are intended for use only with the active agreement of the parties exchanging them.

The low barrier to registration in the vendor and personal trees means it should rarely, if ever, be necessary to use unregistered types. Therefore, use of types in the "x." tree is strongly discouraged.

Note that types with subtype names beginning with "x-" are no longer considered to be members of this tree (see [RFC6648]). Also note that if a generally useful and widely deployed type incorrectly uses an "x-" subtype name prefix, it can be registered in an alternative tree by following the procedure defined in Section 4.1.1.1.

4.1.5. Additional Registration Trees

New top-level registration trees may be created by IETF Standards Action.

In general, the quality of review of specifications for one of these additional registration trees is expected to be equivalent to registrations in the standards tree by a recognized standards-related organization.

When the IETF performs such review, it needs to consider the greater expertise of the requesting organization with respect to the subject media type.

4.2. Structured Syntax Suffixes

Media types can be identified as using a well-known structured syntax (for example, XML or JSON) with a "+suffix" convention.

A structured syntax suffix is defined as all of the characters to the right of the right-most "+" sign in a media type, including the right-most "+" sign itself. A structured syntax suffix MUST NOT contain more than one "+" sign.

For example, in the "application/foo+bar" media type "application" is the top-level type, "foo" is the subtype name, and "+bar" is the structured syntax suffix. A media type such as "application/foo+bar+baz" is not registrable, but if it nevertheless used, its suffix is "+baz".

Structured syntax suffixes MUST be registered before use; see Section 6. Media types that make use of a structured syntax SHOULD use the appropriate suffix, and MUST NOT use suffixes for structured syntaxes that they do not actually employ.

Media types that make use of a structured syntax, or similar separator such as a dash "-", SHOULD be semantically aligned, from a data model perspective, with existing 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 subtype name "application/foo" are the same between both media types. Registrations are expected to align with existing subtype or suffix names in the media type registry; see Section 5.3.

A party requesting registration of a media type that adds a suffix to an existing subtype is expected to coordinate with the change controller(s) for the already registered media type.

4.2.1. Use Cases

Common use cases for media types that employ structured syntax suffixes include:

* Identifying use of a structured data format; for example "+xml", "+json", "+yaml", and "+cbor"

* Flagging compression with a format such as "+zip" or "+gzip"

* Flagging encoding in a digital signature format such as "+jwt" or "+cose"

While it might be desirable to use a compound suffix (e.g., "+xml+zip"), experience shows that suffixes are a poor means of indicating more than one such convention; the combinations of suffixes quickly multiply, and there is not a well-specified processing model that can handle them safely. Therefore, multiple suffixes are disallowed from use.

4.2.2. Fragment Identifiers and Structured Syntax Suffixes

Structured syntax suffixes can specify fragment identifier handling for all subtypes that utilise them, as indicated in the "Fragment Identifier Considerations" column of the Structured Syntax Suffixes registry.

Individual subtypes can specify additional handling. To ensure consistent processing, precedence is determined by the following rules (first match winning):

1. When the structured syntax suffix defines fragment identifier handling and it successfully resolves the fragment identifier, that determines fragment identifier handling;

2. Otherwise, the specific media type determines fragment identifier handling.

4.2.3. Security Considerations for Structured Syntax Suffix Processing

Processors of the information in structured syntax suffixes encounter the following security considerations.

4.2.3.1. Relationships Between Types

The relationship between a media type that employs a structured syntax suffix and the type (if any) that results from removing that suffix cannot be known merely by examining the types. For example, content marked "application/foo+bar" may or may not be processable or valid as "application/foo" content. It may be possible to derive one from the other, but that is specific to the structured syntax suffix and/or media type itself.

This uncertainty extends to fragment identifier processing: per the rules in Section 4.2.2, a fragment identifier that might be valid for an "application/foo+bar" document might not be applicable to another "+bar" document, because media-type specific fragment identifier resolution might be used.

Likewise, the security characteristics that a processor needs to consider may change depending upon whether it is solely processing the structured syntax suffix or the entire media type. For example, a processor cannot presume that the security characteristics for a "application/baz+bar" document will be the same as for a "application/foo+bar" document.

4.2.3.2. Partial Processing

An attacker might append structured syntax suffixes in order to trick processors into skipping security checks. For example, an attacker might use an "application/vnd.ms-excel.addin.macroEnabled.12+zip" structured syntax suffix to trigger an unzip process into invoking Microsoft Excel, bypassing anti-virus scanners that would normally block the file from being opened.

Enterprising attackers might take advantage of toolchains that partially process media types in this manner. Processing of media types based only on the presence of a structured syntax suffix needs to ensure that further processing does not blindly trust the decoded data. For example, proper magic header or file structure checking could mitigate this attack.

5. Media Type Registration Procedures

The media type registration procedure is not a formal standards process, but rather an administrative procedure intended to allow community comment and sanity checking without excessive time delay.

Normal IETF processes need to be followed for all IETF registrations in the standards tree. The posting of an Internet Draft is a necessary first step, followed by posting to the media-types@ietf.org list as discussed below.

5.1. Preliminary Community Review

Notice of a potential media type registration in the standards tree MUST be sent to the media-types@ietf.org mailing list for review.

Registrations in other trees can be sent to the list for review as well; doing so is entirely optional, but is strongly encouraged.

The purpose of this notification is to solicit comments and feedback on the choice of type/subtype name, the unambiguity of the references with respect to versions and external profiling information, and a review of any interoperability or security considerations. The submitter may submit a revised registration proposal or abandon the registration at any time.

5.2. Submit Request to IANA

Standards tree registrations by the IETF itself are submitted by the IESG as part of the normal standards process.

Standards-tree registrations by recognized standards-related organizations are submitted directly to the IANA, unless other arrangements were made as part of a liaison agreement.

Standards-tree registrations using the community format process (Section 4.1.1.1) are submitted by the Designated Expert(s) to IANA.

Registrations in the vendor and personal trees are submitted directly to the IANA.

Registration requests can be submitted to iana@iana.org. A web form for registration request submission is also available at:

https://www.iana.org/form/media-types

5.3. Review and Approval

Registrations submitted to the IANA will be first given to the Designated Expert(s), who are appointed by the IESG. When a suffix is present in a registration, IANA will also inform the Designated Expert(s) of any potentially clashing registrations (see Section 4.2).

In the case of standards-tree registrations from other standards-related organizations, IANA will also check that the submitter is in fact a recognized standards-related organization. If the submitter is not currently recognized as such, the IESG will be asked to confirm their status. Recognition from the IESG needs to be obtained before a standards-tree registration can proceed.

The Designated Expert(s) will examine registration requests to make sure they meet the requirements set forth in this document. Decisions made by the Designated Expert(s) may be appealed to the IESG using the procedure specified in Section 6.5.4 of [RFC2026].

Once a media type registration has passed review, the IANA will register the media type and make the media type registration available to the community.

5.4. Comments on Media Type Registrations

Comments on registered media types may be submitted by members of the community to the IANA at iana@iana.org. These comments will be reviewed by the Designated Expert(s) and then passed on to the change controller of the media type if possible.

Submitters of comments may request that their comment be attached to the media type registration itself; if the IANA, in consultation with the Designated Expert(s), approves, the comment will be made accessible in conjunction with the type registration.

5.5. Change Procedures

When a change to a media type registration is requested, the applicable registration procedure for that media type's tree is used to process the request. Changes may be requested by the change controller, or by other parties if IANA verifies that the change controller approves the change.

Media type registrations may not be deleted; media types that are no longer believed appropriate for use can be declared OBSOLETE by a change to their "intended use" field (see Section 2.8).

Significant changes to a media type's definition should be requested only when there are serious omissions or errors in the published specification. When review is required, a change request may be denied if it renders entities that were valid under the previous definition invalid under the new definition.

The change controller of a media type may pass responsibility to another person or agency by informing the IANA; this can be done without discussion or review.

If the Designated Expert(s) find that the change controller is unresponsive or uncontactable for a reasonable period of time and reasonable efforts have been made to contact the change controller, they may recommend to the IESG that the change controller be updated.

5.6. Registration Template

Type name:
[see Section 2.3]

Subtype name:
[see Section 2.3]

Required parameters:
[see Section 2.4]

Optional parameters:
[see Section 2.4]

Encoding considerations:
[see Section 2.5]

Security considerations:
[see Section 2.7]

Interoperability considerations:
[see Section 2.2]

Published specification:
[see Section 2.1.1]

Applications that use this media type:

Fragment identifier considerations:
[see Section 2.6]

Additional information: [see Section 2.10]
Deprecated alias names for this type:

Magic number(s):

File name extension(s):

macOS Uniform Type Identifier(s):

Windows clipboard name(s):

Person & email address to contact for further information:
[see Section 2.9]

Intended usage:
[see Section 2.8]

Restrictions on usage:
[see Section 2.8]

Author:
[see Section 2.9]

Change controller:
[see Section 2.9]

(Any other information that the author deems interesting may be added below this line.)

"N/A", written exactly that way, can be used in any field if desired to emphasize the fact that it does not apply or that the question was not omitted by accident. Do not use 'none' or other words that could be mistaken for a response.

6. Structured Syntax Suffix Registration Procedures

Structured syntax suffixes must be described by a readily available description, preferably within a document published by an established standards-related organization, for which there's a reference that can be used in a Normative References section of an RFC.

Someone wishing to define a "+suffix" name for a structured syntax for use with a new media type registration should:

1. Check IANA's registry of media type name suffixes to see whether or not there is already an entry for that well-defined structured syntax.

2. If there is no corresponding entry, fill out the template (see Section 6.2) and include that with the media type registration request. The template may be contained in an Internet Draft, alone or as part of some other protocol specification. The template may also be submitted in some other form, but the contents will be treated as an "IETF Contribution" under the guidelines of BCP 78 [RFC5378].

3. Send a copy of the template or a pointer to the containing document (with specific reference to the section with the template) to the mailing list media-types@ietf.org, requesting review. This may be combined with a request to review the media type registration.

4. Respond to review comments and make revisions to the proposed registration as needed.

5. Submit the (possibly updated) registration template (or pointer to the document containing it) to IANA at iana@iana.org.

Upon receipt of a structured syntax suffix registration request:

1. IANA checks the submission for completeness; if sections are missing or citations are not correct, IANA rejects the registration request.

2. IANA checks the current registry for an entry with the same name; if such a registry exists, IANA rejects the registration request.

3. IANA requests Expert Review of the registration request against the corresponding guidelines.

4. The Designated Expert may request additional review or discussion, as necessary.

5. If Expert Review recommends registration, IANA adds the registration to the appropriate registry.

The initial registry content specification [RFC6839] provides examples of structured syntax suffix registrations.

6.1. Change Procedures

Registrations may be updated in each registry by the same mechanism as required for an initial registration. In cases where the original definition of the scheme is contained in an IESG-approved document, update of the specification also requires IESG approval.

6.2. Registration Template

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. Required.

+suffix
: Suffix used to indicate conformance to the syntax. Required.

References
: A citation for all specifications necessary to understand the structured syntax. Required.

Encoding considerations
: A citation to a section in a specification that provides general guidance regarding encoding considerations for any type employing this syntax. The requirements for media type encoding considerations given in Section 2.5 apply.

Interoperability considerations
: A citation to a section in a specification that documents any issues regarding the interoperable use of types employing this structured syntax. For example, 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 citation to a section in a specification that documents the generic processing rules of fragment identifiers for any type employing this syntax.

Security considerations

A citation to a section in a specification that provides security considerations shared by media types employing this structured syntax. The security considerations for a media type registration in the standards tree (per Section 2.7) apply. Required.

Contact
Person (including contact information) to contact for further information.

Author/Change controller.
Person (including contact information) authorized to change this suffix registration. Required.

7. Security Considerations

Security considerations for media types are discussed in Section 2.7. Considerations for structured suffix registrations are discussed in Section 4.2.3.

8. IANA Considerations

8.1. Top-Level Types Registry

In the Top-Level Media Types registry, IANA should link the reference field for each top-level type to the specific subsection in question, rather than just the relevant RFC.

8.2. Recognized Standards Organisations

IANA should notify recognized standards organisations when this document is published (where feasible), and highlight the need to consider how their processes interact with the registration procedure (see eg https://www.w3.org/guide/editor/mediatypes.html#registration-process (https://www.w3.org/guide/editor/mediatypes.html#registration-process)).

8.3. Provisional Registrations

This revision removes provisional registrations from the standards tree. Accordingly, IANA should remove that information from the registry and work with the Experts and the IESG as appropriate to identify an appropriate disposition for registrations that are marked as 'provisional'.

9. Acknowledgments

The current authors would like to acknowledge their debt to the late Dr. Jon Postel, whose general model of IANA registration procedures and specific contributions shaped the predecessors of this document [RFC2048] [RFC4288]. We hope that the current version is one with which he would have agreed but, as it is impossible to verify that agreement, we have regretfully removed his name as a co-author.

Randy Bush, Francis Dupont, Bjoern Hoehrmann, Barry Leiba, Murray Kucherawy, Alexey Melnikov, S. Moonesamy, Mark Nottingham, Tom Petch, Peter Saint-Andre, and Jeni Tennison provided many helpful review comments and suggestions.

10. Contributors

Much of the text of this document is directly taken from [RFC6838] and [RFC9694]. We acknowledge the following authors of those documents as contributors to this:

Ned Freed

John C. Klensin 1770 Massachusetts Ave, #322 Cambridge, MA 02140 USA EMail: john+ietf@jck.com

Tony Hansen AT&T Laboratories 200 Laurel Ave. Middletown, NJ 07748 USA EMail: tony+mtsuffix@maillennium.att.com

Martin J. D端rst 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端rst/

11. References

11.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <https://www.rfc-editor.org/rfc/rfc2045>.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <https://www.rfc-editor.org/rfc/rfc2046>.

[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/rfc/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/rfc/rfc4855>.

[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/rfc/rfc5234>.

[RFC5378] Bradner, S., Ed. and J. Contreras, Ed., "Rights Contributors Provide to the IETF Trust", BCP 78, RFC 5378, DOI 10.17487/RFC5378, November 2008, <https://www.rfc-editor.org/rfc/rfc5378>.

[RFC6648] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, DOI 10.17487/RFC6648, June 2012, <https://www.rfc-editor.org/rfc/rfc6648>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <https://www.rfc-editor.org/rfc/rfc7120>.

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <https://www.rfc-editor.org/rfc/rfc7303>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[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/rfc/rfc8174>.

[RFC8179] Bradner, S. and J. Contreras, "Intellectual Property Rights in IETF Technology", BCP 79, RFC 8179, DOI 10.17487/RFC8179, May 2017, <https://www.rfc-editor.org/rfc/rfc8179>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

11.2. Informative References

[MacOSUTIs] Apple Computer, Inc., "Framework: Uniform Type Identifiers", March 2024, <https://developer.apple.com/documentation/uniformtypeidentifiers>.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, DOI 10.17487/RFC2026, October 1996, <https://www.rfc-editor.org/rfc/rfc2026>.

[RFC2048] Freed, N., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", RFC 2048, DOI 10.17487/RFC2048, November 1996, <https://www.rfc-editor.org/rfc/rfc2048>.

[RFC2077] Nelson, S., Parks, C., and Mitra, "The Model Primary Content Type for Multipurpose Internet Mail Extensions", RFC 2077, DOI 10.17487/RFC2077, January 1997, <https://www.rfc-editor.org/rfc/rfc2077>.

[RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, DOI 10.17487/RFC2231, November 1997, <https://www.rfc-editor.org/rfc/rfc2231>.

[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/rfc/rfc4288>.

[RFC6381] Gellens, R., Singer, D., and P. Frojdh, "The ’Codecs’ and ’Profiles’ Parameters for "Bucket" Media Types", RFC 6381, DOI 10.17487/RFC6381, August 2011, <https://www.rfc-editor.org/rfc/rfc6381>.

[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/rfc/rfc6838>.

[RFC6839] Hansen, T. and A. Melnikov, "Additional Media Type Structured Syntax Suffixes", RFC 6839, DOI 10.17487/RFC6839, January 2013, <https://www.rfc-editor.org/rfc/rfc6839>.

[RFC8081] Lilley, C., "The "font" Top-Level Media Type", RFC 8081, DOI 10.17487/RFC8081, February 2017, <https://www.rfc-editor.org/rfc/rfc8081>.

[RFC8187] Reschke, J., "Indicating Character Encoding and Language for HTTP Header Field Parameters", RFC 8187, DOI 10.17487/RFC8187, September 2017, <https://www.rfc-editor.org/rfc/rfc8187>.

[RFC8851] Roach, A.B., Ed., "RTP Payload Format Restrictions", RFC 8851, DOI 10.17487/RFC8851, January 2021, <https://www.rfc-editor.org/rfc/rfc8851>.

[RFC9694] D端rst, M.J., "Guidelines for the Definition of New Top-Level Media Types", BCP 13, RFC 9694, DOI 10.17487/RFC9694, March 2025, <https://www.rfc-editor.org/rfc/rfc9694>.

[RFC9695] Muthusamy, Y. K. and C. Ullrich, "The ’haptics’ Top-Level Media Type", RFC 9695, DOI 10.17487/RFC9695, March 2025, <https://www.rfc-editor.org/rfc/rfc9695>.

[windowsClipboardNames] MicroSoft Inc., "Clipboard Formats", August 2020, <https://learn.microsoft.com/en-us/windows/win32/dataxchg/clipboard-formats>.

Appendix A. Historical Note

The media type registration process was initially defined for registering media types for use in the context of the asynchronous Internet mail environment. In this mail environment, there is a need to limit the number of possible media types, to increase the likelihood of interoperability when the capabilities of the remote mail system are not known. As media types are used in new environments in which the proliferation of media types is not a hindrance to interoperability, the original procedure proved excessively restrictive and had to be generalized. This was

initially done in [RFC2048], but the procedure defined there was still part of the MIME document set. The media type specification and registration procedure is now a separate document, to make it clear that it is independent of MIME.

It may be desirable to restrict the use of media types to specific environments or to prohibit their use in other environments. This specification incorporates such restrictions into media type registrations in a systematic way. See Section 2.8.

Authors' Addresses

Mark Nottingham
Cloudflare
Melbourne
Australia
Email: mnot@mnot.net
URI: https://www.mnot.net/

Pete Resnick
Episteme Technology Consulting
Email: resnick@episteme.net