# THE 'Haptics' TOP-LEVEL MEDIA TYPE:
# IETF MEDIAMAN DRAFT v03

Yeshwant Muthusamy

Chris Ullrich

March 27, 2023

# NEW DEVELOPMENTS SINCE IETF 115 (Nov 2022)

## *In IETF*

- **Haptics**

  - All comments raised at IETF 115 as well as those logged into the issue tracker have been incorporated into v03 of the haptics I-D: https://datatracker.ietf.org/doc/draft-ietf-mediaman-haptics/03/ (uploaded Feb 8th, 2023)

- **Top-level**

  - https://datatracker.ietf.org/doc/draft-ietf-mediaman-toplevel/03/ (uploaded March 26th, 2023)

  - From our reading, we believe the haptics I-D v03 satisfies **all** requirements listed in the **latest** Section 2.1 and 2.2

## *In MPEG*

- MPEG-I Haptics Phase 1 Codec (ISO/IEC 23090-31: Haptics Coding)
  - Has passed CD (Committee Draft) stage at MPEG141 in January 2023
  - Expected to move to DIS (Draft International Standard) at MPEG142 in April 2023

- MPEG-I Carriage of haptics data in ISOBMFF (ISO/IEC 23090-32: Carriage of haptics data)
  - Revised Working Draft issued at MPEG141
  - Defines the 'mih1' subtype to refer to the MPEG-I haptic streams

- MPEG-I (Immersive Media) Scene Description
  - Two haptics extensions approved for implementation
  - Working Draft of AMD 2 of ISO/IEC 23090-14 specifies interplay of haptics, interactivity, Avatars in Immersive Media.

- Every new top-level type MUST be defined in a Standards Track RFC. This will make sure there is sufficient community interest, review, and consensus appropriate for a new top-level type.

  ➢ The intended status of https://datatracker.ietf.org/doc/draft-ietf-mediaman-haptics/03/ is indeed a Standards Track RFC.
  ➢ Evidence of community interest received from Tencent America, InterDigital, and Razer, Inc., so far. See emails to media-types@ietf.org sent in November 2022 (just before IETF 115).

- The IANA Considerations section of an RFC defining a new top-level type MUST request that IANA add this new top-level type to the registry of top-level types.

  ➢ The introductory paragraph of Section 4 (IANA Considerations) addresses this requirement.

- The criteria for what types do and do not fall under the new top-level type need to be defined clearly. Clear criteria are expected to help expert reviewers 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.

  ➢ All subtypes representing sub-modalities or formats of haptics (anything pertaining to touch) would fall under this top-level type. There is ZERO scope for confusion here.

- The document defining the new top-level type MUST include initial registrations of actual subtypes. This will help to show the need for the new top-level type, will allow to check the appropriateness of the definition of the new top-level type, and will avoid separate work for registering an initial slate of subtypes.

  ➢ There are two actual sub-types defined in the I-D.

  ➢ As soon as the haptics I-D is approved, registration of actual types defined in the I-D will be done and reflected in the final RFC document.

- The registration and actual use of a certain number of subtypes under the new top-level type SHOULD be expected. 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. Otherwise, the creation of a new top-level type is most probably not justified.

  ➢ Several haptic subtypes exist, all of which are documented in Section 2.5 of the haptics I-D. Specifically:

    ➢ `ahap`: Currently the standard encoding on all iOS devices + iOS connected game peripherals (Apple)

    ➢ `ogg`: Proprietary extension to the OGG format in the latest version of Android 11 (Google).

    ➢ `ivs`: Currently a vendor-specific haptic format that is in use in mobile phones (from LG) and gaming phones (from Asus).

    ➢ `hapt`: Currently a vendor-specific haptic format that is in use in mobile haptic advertising and certain Japanese games.

  ➢ Several haptic subtypes are envisioned as part of various haptics standards in various stages of standardization (some are documented in Section 2.6 of the haptics I-D):

    ➢ `hmpg`: the MPEG Haptics Phase1 coding format (specification will progress to DIS ballot in April 2023)

    ➢ `hjif`: the MPEG Haptics Phase 1 interchange format (specification has progressed to CD ballot in October 2022)

    ➢ `mih1`: the MPEG Haptics Phase 1 streaming format (specification in WD stage as of October 2022)

    ➢ `hiev`: the IEEE P1918.1.1 vibrotactile coding format that is currently under IEEE SA ballot

    ➢ `hiekd`: the IEEE P1918.1.1 kinesthetic (with-delay) coding format that is currently under IEEE SA ballot

    ➢ `hiekn`: the IEEE P1918.1.1 kinesthetic (no-delay) coding format that is currently under IEEE SA ballot

    ➢ `henm`: enumerated effects haptic coding format (based on MIDI)

    ➢ `havc`: audio-to-vibe haptic coding format (automatic audio to vibration conversion algorithms)

- Any RFC defining a new top-level type MUST clearly document the security considerations applying to all or a significant subset of subtypes.

  ➢ Section 3 (Security Considerations) addresses this requirement in detail.

- 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.

  ➢ Immersion has specified two subtypes already in use (`ivs` and `hapt`) in the I-D.
  ➢ Razer has indicated strong interest in registering two subtypes (`haps` and `chaps`) as soon as haptics is approved as a top-level media type
  ➢ The MPEG subtypes listed on previous slide (`mih1, hmpg, hjif`) will become part of the MPEG standard in due course.
  ➢ The IEEE subtypes (`hiev, hiekd, hiekn`) will be used in 5G Tactile Internet applications once the standard is approved.

- 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.

  ➢ No undefined top-level haptic types are currently in use

- Use of an IETF WG to define a new top-level type is not be needed but may be advisable in some cases. There are examples of new top-level type definitions without a WG (RFC 2077), with a short, dedicated WG (RFC 8081), and with a WG that included other related work (draft-ietf-mediaman-haptics).

  ➢ The haptics I-D is under discussion in the mediaman WG.

- 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.
  ➢ All the data formats (subtypes) described in the haptics I-D pertain to the sense of touch and need to be under the proposed haptics top-level type.
  ➢ All the subtypes described in the haptics I-D require a 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.

- 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. But as often with humans, opinions may widely differ.

  ➢ The proposed haptics top-level type pertains to the human sense of touch, just as the existing 'audio' and 'video' top-level types pertain to the senses of hearing and vision, respectively.

- 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'.

  ➢ All subtypes of the `haptics` top-level type must pertain to the sense of touch in some fashion.

- 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 media types 'image', 'audio', and 'video' are also often handled generically. Documents with these top-level media 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.

  ➢ Files containing just haptics, without accompanying audio and video, will be directed to the appropriate haptic output device. For example, bodysuits, haptic chairs (e.g., D-Box chairs) can process haptics-only files.
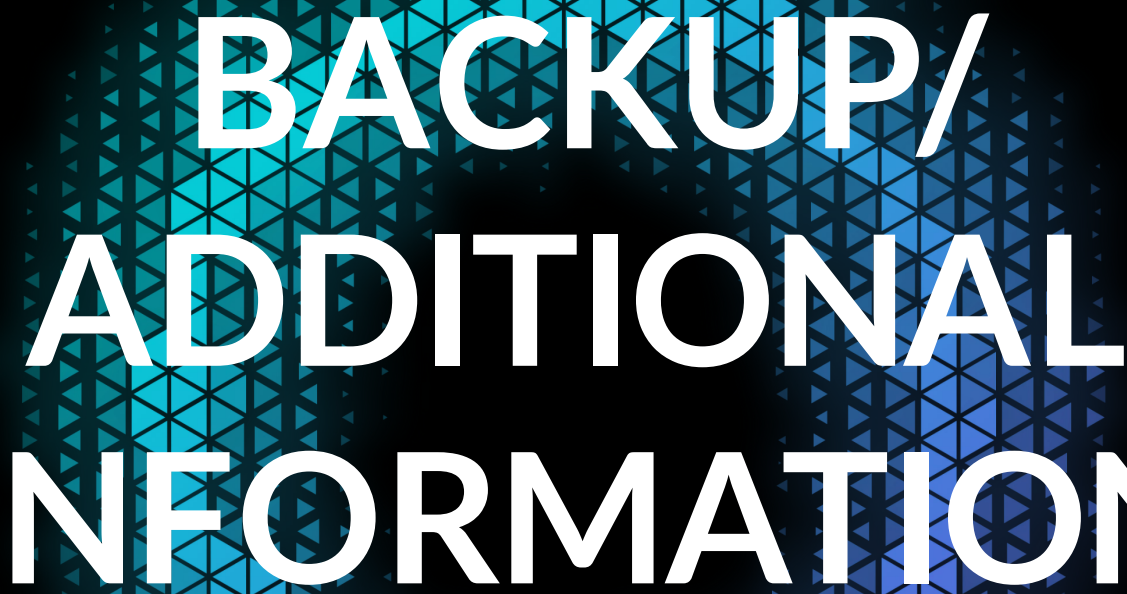
- A top-level type is not a pointer into another registration space that offers duplicate registrations for existing media types. Example: a top-level type of 'oid', leading to types of the form oid/nnnnn, where nnn is an OID designating a specific media format.

  ➢ There are no such instances in the haptics I-D v03.

- A top-level type MUST NOT be defined for the mapping of other protocol elements to media types. For example, while there may be some merit to a mapping from media types to URIs, e.g. in the context of RDF, there is very limited merit in a reverse mapping, and even less merit in creating a top-level type for such a mapping. The same applies to other protocol elements such as file extensions or URI schemes. The recommended solution in case a mapping is needed is to choose a single type/subtype and put the additional information in an appropriately named parameter. As an example, information on a file extension '.dcat' can be encoded as 'application/octet-string; filename=foo.dcat'.

  ➢ There are no such mappings in the haptics I-D v03.

- Media types are not a general type system. A top-level type MUST NOT be defined if its main or only purpose is to map other type systems, e.g. in programming languages or ontologies.

  ➢ The main purpose of defining a `haptics` top-level media type is to provide a logical top-level type for all data formats (subtypes) pertaining to the sense of touch. There is no mapping to other type systems.

- A new top-level type SHOULD NOT generate aliases for existing widely used types or subtypes.

  ➢ The proposed `haptics` top-level media type does not generate any such aliases.

- Top-level types with an "X-" prefix cannot be registered, and SHOULD NOT be used. This is in line with RFC [RFC6648].

  ➢ The proposed `haptics` top-level media type does not have an "X-" prefix.

THANK YOU!
ANY QUESTIONS?

# BACKUP/ ADDITIONAL INFORMATION

# RECAP: BASICS

- 'Haptics' refers to the generation of touch-related sensations in a device or interface
- Haptics is widely used in consumer devices to provide touch-based feedback:
  - Mobile devices
  - Automobiles
  - Gaming
- Haptic technologies require some form of actuation in order to create a tactile sensation
  - In mobile devices and game controllers, the actuators are typically small vibrating motors
  - In large automotive touchscreens, the actuators are specialized piezo-electric materials
- Haptic capabilities are now part of every modern smartphone, gaming, and VR controller
  - Examples: Sony PS5 DualSense controller, all late-model iPhones and Android phones

# JUSTIFICATION – MPEG ISOBMFF

- Haptic signals provide an additional layer of entertainment and sensory immersion for the user.

- The user experience and enjoyment of media content can be significantly enhanced by adding haptics to audio/video content in:

  - ISO Base Media File Format (ISOBMFF) files

  - Media streams such as ATSC 3.0 broadcasts

  - Streaming games, and

  - Mobile advertising

- To date, there has not been a registration of formats for haptics

- We envision the following designations for haptics in .mp4 files, once 'haptics/' is registered:

  - haptics/mp4 - mp4 files with just haptic tracks in them (e.g., streaming games, haptics files for haptic vests, belts, gloves, etc.)

  - video/mp4 - mp4 files with video, audio, and haptics (ensures consistency with existing video mp4 files)

  - audio/mp4 - mp4 files with audio and haptics (ensures consistency with existing audio-only mp4 files)

# JUSTIFICATION – ANOTHER HUMAN SENSE

- For the human sense of hearing, we have the top-level media type `audio`

- For the human sense of seeing, we have the top-level media type `video`

- For the (equally important) human sense of touch, makes perfect sense to have the top-level media type `haptics`

- Placing `haptics` under `audio` or `video` or `application` 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 or applications

# JUSTIFICATION – 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)

- 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`.

# JUSTIFICATION – HAPTIC DATA FORMATS IN USE

- The following haptic data formats, prevalent in a large number of devices around the world, would live as subtypes under the proposed `haptics` top-level media type:

  - **`ahap:`** [AHAP](#) haptic data format, currently the standard encoding on all iOS devices + iOS connected game peripherals.  Decoders also 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.  Enables haptic media to be stored in OGG files.

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

  - **`hapt:`** The HAPT haptic data format is 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:
      - KLAB: https://www.klab.com/en/
      - Craft&Meister: http://www.crafts-meister.co.jp/pc/company_en.html

- The following standardized haptic coding formats are likely to be registered soon and will also live as subtypes under `haptics`

  - **`hmpg:`** the selected coding format from the MPEG Call for Proposals on the Coded Representation of Haptics

  - **`hiee:`** IEEE P1918.1.1 vibrotactile coding standard (currently under ballot) – part of the IEEE Tactile Internet standard.

# JUSTIFICATION – 'application/' NOT SUITABLE

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

  - haptics connects to a sensory system, touch/motion, directly, and is more specific than the abstract `application` type,

  - `application` has historically been used for applications, i.e., code, which means it is viewed and treated with great care for security (i.e., viruses and other active code).

  - `haptics` is <u>not</u> code, just as `audio` and `video` are <u>not</u> 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.

# UPDATES IN Version 02 of the I-D

- Version 02 uploaded on May 17, 2021: https://datatracker.ietf.org/doc/draft-muthusamy-dispatch-haptics/02/

- Specifically:

  - Updated the URL in the MPEG-Haptics-CfP reference to point to the latest version of the MPEG Haptics CfP document, issued at MPEG134.

  - Added a new reference, MPEG-Haptics-Encoder-Format, to the Encoder Input Format document and updated its N-number (from N 13 to N 72) to reflect the latest version of this document, issued at MPEG134.

# UPDATES IN Version 01 of the I-D

- Version 01 uploaded on November 15, 2020: https://datatracker.ietf.org/doc/draft-muthusamy-dispatch-haptics/

- Addressed the comments received on Version 00 on the DISPATCH mailing list: https://mailarchive.ietf.org/arch/msg/dispatch/Tv-4_ZUwASBjs-DrgOR5I0oSDcQ/

- Specifically:

  - Reformulated prose to dispel the inadvertent misconception (in Version 00) that the following haptic subtypes
    - `haptics/ahap`
    - `haptics/ogg`
    - `haptics/ivs`
    - `haptics/hapt`

    were already in use. **They are not**. `ahap, ogg, ivs,` and `hapt` are haptic data formats widely in use that **would live** under the proposed `haptics/` top-level type. Made that point more explicit.

  - Added sections on subtype registrations for `haptics/ivs` and `haptics/hapt` to illustrate how subtype registrations would be managed, once the `haptics/` top-level type was registered.

  - Added the following new references and associated prose
    - ISO/IEC 14496-12 AMD 1 document and timeline showing the progression of haptics standardization in ISOBMFF
    - Encoder Input Format for MPEG Haptics – published document describing the IVS format in detail
    - AHAP – Apple Haptic and Audio Pattern – public reference to the AHAP data format
    - RIFF – Resource Interchange File Format – the HAPT format is based on this standard
    - MPEG Draft Call for Proposals on the Coded Representation of Haptics – ongoing standardization of an MPEG haptic coding format

# JUSTIFICATION – W3C HAPTICS

- W3C Defines a Vibration API (https://www.w3.org/TR/vibration/) which is implemented on Chrome and Firefox for Android.
- W3C recently defined a vibration extension for gamepads (https://w3c.github.io/gamepad/extensions.html).
- Both APIs assume a vibration buffer and do not define a media format
- Vibration buffers are insufficiently portable across haptic devices due to large differences in mechanical performance – a well-defined haptic format would be portable
- Haptic media types are certain to become common for these APIs in the future.

# JUSTIFICATION – ENVISIONED SUBTYPE STANDARDS

- The MPEG ISOBMFF proposal included an informative annex of known haptic coding formats with proposed FourCC codes for them.

- These codes are <u>not</u> registered yet, but the plan is indeed to standardize these haptic coding formats in the near future:

  - `hmpg:` the selected coding format from the MPEG Call for Proposals on the Coded Representation of Haptics

  - `hiee:` IEEE P1918.1.1 vibrotactile coding standard (currently under ballot) – part of the IEEE Tactile Internet standard.

  - `henm:` enumerated effects haptic coding format (based on MIDI)

  - `havc:` audio-to-vibe haptic coding format (automatic audio-to-vibration conversion algorithms)

- Once standardized, these formats will also live as subtypes under the proposed `haptics` top-level media type

# JUSTIFICATION – 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): Optionally supported in mobile web browsers
  - Game consoles (39+ million units sold in 2019): MS Xbox, Sony PlayStation, Nintendo Switch, etc.
  - XR devices (9+ million units sold in 2019): Khronos OpenXR haptic API
- Haptic media is expected to be commonly exchanged between these devices.
- Since they represent the majority of CE devices in use around the world, a strong   case can be made for `haptics` as a top-level media type.

# 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.

- 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.

- Any coded representation of haptic media is insufficient to implicitly provide sufficient security and this protection should be enforced by the operating system implementor.

# IANA CONSIDERATIONS – 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.

# IANA CONSIDERATIONS – REGISTRATION PROCEDURE

- New haptics formats should be registered using the online form at http://www.iana.org/form/media-types

- 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.

# IANA CONSIDERATIONS – SUBTYPE REGISTRATIONS (1/2)

- Two initial entries under the proposed `haptics` media type are specified. They also serve as examples for future registrations:

- **IVS Haptic Type**

  - Type name: haptics

  - Subtype name: ivs

  - Required parameters: None

  - Optional parameters: None

  - Encoding considerations: 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.

  - Published specification: ISO/IEC JTC 1/SC 29/WG 2 N 13 "Encoder Input Format for Haptics" being developed by ISO/IEC JTC1/SC29 WG 2.

  - 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

# IANA CONSIDERATIONS – SUBTYPE REGISTRATIONS (2/2)

- **HAPT Haptic Type**

  - Type name: haptics

  - Subtype name: hapt

  - Required parameters: None

  - Optional parameters: None

  - Encoding considerations: binary

  - Interoperability considerations: The HAPT format is a device-dependent haptic effect coding based on the RIFF 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

  - 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