



THE 'HAPTICS' TOP-LEVEL MEDIA TYPE: IETF DISPATCH INTERNET DRAFT V01

Yeshwant Muthusamy, Ph.D. | Senior Director, Standards

Chris Ullrich, CTO

IETF 109

November 16, 2020

INTRODUCTION

- '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: feedback to the users interacting with the touchscreen, e.g., virtual keyboard, notifications
 - Automobiles: buttons on dashboard replaced by a large touchscreen that provides a bi-directional touch interface
- 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 and game/VR controllers
 - Examples: Sony PS5 DualSense controller, all late-model iPhones and Android phones
 - Considered essential for enhanced media experiences

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 ISO/BMFF
 - 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 (1/8)

- 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
- Low-hanging fruit: 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.

JUSTIFICATION (2/8) - MPEG ISOBMFF

- To date, there has not been a registration of formats for haptics
- Haptics has been proposed as a first-order media type (at the same level as audio and video) in ISOBMFF
 - Accepted by the MPEG Systems File Format sub-group in April 2020
 - Progressed to DAM in October 2020 after passing the CD ballot
 - Expected to issue as an AMD (Amendment) in October 2021
 - Will become part of ISO/IEC 14496-12, 6th Edition
- 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 (to ensure consistency with existing mp4 files with video content)
 - `audio/mp4` - mp4 files with audio and haptics (to ensure consistency with existing mp4 files with audio content without any video)

JUSTIFICATION (3/8) – 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 (4/8) – 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` 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

JUSTIFICATION (5/8) – COMMERCIAL UPTAKE

- Haptics is rapidly becoming a standard feature of consumer electronic devices. For example:
 - iPhone (191+ million units sold in 2019): native support for haptic encoded data
 - Android (1.18+ billion units sold in 2019): 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): 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.

JUSTIFICATION (6/8) – 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. 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 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 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

JUSTIFICATION (7/8) – 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 not 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 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)
- Once standardized, these formats will also live as subtypes under the proposed `haptics` top-level media type

JUSTIFICATION (8/8) – ‘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 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,
 - `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.

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: 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.
 - 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: Text/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

THANK YOU!
QUESTIONS?