**ISO/IEC 23090-37:2025(E)**

ISO/IEC JTC 1/SC 29/WG 03

Secretariat: XXXX

**Information technology — Coded representation of immersive media — Part 37: Conformance and reference software for carriage of haptics data**

CD stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Fax: +41 22 749 09 47

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

Contents

[Foreword iv](#_Toc204183850)

[Introduction v](#_Toc204183851)

[1 Scope 1](#_Toc204183852)

[2 Normative references 1](#_Toc204183853)

[3 Terms and definitions 1](#_Toc204183854)

[4 Abbreviated terms 1](#_Toc204183855)

[5 Reference software for carriage of haptics data 1](#_Toc204183856)

[5.1 General 1](#_Toc204183857)

[5.2 Overview 2](#_Toc204183858)

[5.3 Feature list 3](#_Toc204183859)

[5.4 Haptics carriage library API 4](#_Toc204183860)

[5.5 Using the Command-line Applications 4](#_Toc204183861)

[5.5.1 HapticsMultiplexerApp 4](#_Toc204183862)

[5.5.2 HapticsDemultiplexerApp 4](#_Toc204183863)

[6 Conformance for ISO/IEC 23090-32 5](#_Toc204183864)

[Annex A (informative) Reference software build process 7](#_Toc204183865)

[A.1 General 7](#_Toc204183866)

[A.1.1 Dependencies 7](#_Toc204183867)

[A.1.2 Building 7](#_Toc204183868)

[Bibliography 9](#_Toc204183869)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html) or [www.iec.ch/members\_experts/refdocs](https://www.iec.ch/members_experts/refdocs)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)) or the IEC list of patent declarations received (see <https://patents.iec.ch>).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see [www.iso.org/iso/foreword.html](https://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](https://www.iec.ch/understanding-standards).

This document was prepared by Technical Committee ISO/IEC/JTC 1 *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO 23090 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html) and [www.iec.ch/national-committees](https://www.iec.ch/national-committees).

Introduction

The conformance and reference software in this document serves two main purposes:

— Validation of the written specification of ISO/IEC 23090-32;

— Conformance testing for checking interoperability for the various applications against the reference software which aims to be compliant with ISO/IEC 23090-32.

The reference software is structured as an extension of the ISOBMFF reference software library and provides additional functions required for ISO/IEC 23090-32. In addition, it includes a small command line application that uses the library to perform some basic file operations such as multiplexing and demultiplexing a file.

Furthermore, this document is accompanied by a collection of conformance files. These files provide practical demonstrations of various features of ISO/IEC 23090-32, aiding in a more comprehensive understanding and application of ISO/IEC 23090-32.

Information technology — Coded representation of immersive media — Part 37: Conformance and reference software for carriage of haptics data

# Scope

This document specifies the reference software for carriage of haptics data as specified in ISO/IEC 23090-32. The information provided describes the reference software modules and the features that it supports. It also provides a description of how the reference software can be utilized. Finally, it also provides a description of conformance test vectors.

# Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document.

* ISO/IEC 23090-32, Information technology — Coded representation of immersive media — Part 32: Carriage of Haptics data

# Terms and definitions

No terms and definitions are listed in this document.

[Editor’s Note: We may revisit in the future.]

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at [https://www.iso.org/obp](https://www.iso.org/obp/ui)

— IEC Electropedia: available at <https://www.electropedia.org/>

# Abbreviated terms

ISOBMFF ISO base media file format

MIHS MPEG-I haptics stream

# Reference software for carriage of haptics data

## General

The source code of the reference software for the carriage of haptics data is available in the following project URL on MPEG’s GitLab server and is accessible to MPEG members.

https://git.mpeg.expert/MPEG/Systems/FileFormat/hapticscarriageconfrefsoft

## Overview

Figure 1 shows a simplified overview of the reference software implementation architecture for the carriage of haptics data. The reference software implementation implements the features defined in ISO/IEC 23090-32 and is based on the reference software for ISOBMFF (*libisomediafile*) specified in ISO/IEC 14496-32 [3], the reference software for haptics coding [2], and a number of 3rd party open-source libraries. Boxes with a white background in Figure 1 are part of the scope of the reference software implementation and are described in more detail in the following sections.

A diagram of a software

AI-generated content may be incorrect.

Figure 1 Reference software overview

The *libHapCarriageIso* library is implemented in the C programming language and extends the *libisomediafile* specified in ISO/IEC 14496-32 [3] by implementing all code points defined in ISO/IEC 23090-32 and providing a public API which can be used by applications in order to write and parse the corresponding syntax elements.

[Editor’s Note: The initial version includes a public API which can be used by applications to write and parse ISOBMFF files with the brand 'mih1', as specified in ISO/IEC 23090-32. Support for the ‘mhb1’ brand will be added.]

The Haptics Carriage library *libHapticCarriage* enables the packaging and extraction of MIHS bitstreams using MPEG Haptics reference model specified in ISO/IEC 23090-33 [2] and manages boxes declared in *libHapticCarriageIso*.

The *HapticsMultiplexerApp* and *HapticsDeMultiplexerApp* are command-line applications which use *libHapCarriage* and the library in the reference software for haptics coding [2] to multiplex and demultiplex haptics MIHS bitstreams to and from an ISOBMFF container. These command-line applications use a number of 3rd party open-source libraries such as spdlog [4], json [5], and cxxopts to facilitate logging and parsing of JSON and command-line arguments.

## Feature list

Table 1 summarizes a list of features defined in ISO/IEC 23090-32 and indicates which features are currently supported by the reference software.

Table 1 Reference software feature list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | | **4CCs** | **Version** | **Status** |
| Haptics media handler type | | hapt |  | \* |
| Haptics media header (NullMediaHeaderBox) | | nmhd | 0 | \* |
| Haptics sample entry | |  | 0 | \* |
| Single track | MIHS sample entry | mih1 | 0 | ü |
| MIHS configuration Box | mh1C | 0 | ü |
| Haptics Experience Description Box | hexd | 0 | ü |
| Haptic Experience Description Header Box | hexh | 0 | ü |
| Haptic Avatar Description Box | havd | 0 | ü |
| Haptic Perception Description Box | hprd | 0 | ü |
| Haptic Perception Description Header Box | hprh | 0 | ü |
| Haptic Reference Device Description Box | hrdd | 0 | ü |
| Haptic Channel Description Box | hchd | 0 | ü |
| Haptic Channel Description Header Box | hchh | 0 | ü |
| Multiplexer | N/A |  | ü(\*) |
| Demultiplexer | N/A |  | ü(\*) |
| Common | Haptic Band Description Box | hbnd | 0 | ü |
| Haptics Presentation Dep Group Entry | mhpg | 0 | û |
| Track Reference Type (synchronization with other media) | rsyn | - | \*\* |
| Haptics Silent Unit Sample Group Entry | mhsu | 0 | û |
| MIHS sample format |  |  | ü |
| MIHS sync sample |  |  | û |
| Multi-track | MIHS band sample entry | mhb1 | 0 | û |
| Haptic Band Configuration Box | mibC | 0 | û |
| Multiplexer | N/A |  | û |
| Demultiplexer | N/A |  | û |
| MIHS Band Group Box  (Entity To Group Box) | mhbd |  | û |
| Track Reference Type  (Referring MIHS band track) | mhbd | - | û |
| **Legend:**  û : not implemented  ü : implemented  ü (\*) : implemented in a separate branch  \* : implemented by libisomediafile | | | | |

[Editors’ Note: Status \*\* indicates that this may be implemented by libisomediafile and needs to be checked.]

[Editors’ Note: Status ü (\*) in the table above indicate that the feature is implemented in a separate branch named m73531.]

## Haptics carriage library API

(TBD)

## Using the Command-line Applications

### HapticsMultiplexerApp

The HapticsMultiplexerApp application can be used to package MIHS bitstreams into an ISOBMFF (MP4) file. The command-line options for this application are shown in Table 2.

Table 2 - Command-line options for HapticsMultiplexerApp.

|  |  |
| --- | --- |
| **Option** | **Description** |
| -d, --debug | Enables debugging |
| -i, --inputPath arg | Input MIHS bitstream (.hmpg) filename |
| -o, --outputPath arg | Output MP4 filename |
| -f, --force | Overwrite output file if already exists |
| -F, --fragmented | Generate a fragmented MP4 file |
| -v, --verbose | Verbose output |
| -h, --help | Print usage |

#### Usage examples

The following command-line can be used to create a non-fragmented MP4 file from an MIHS bitstream file named “sample.hmpg”:

$ bin/HapticsMultiplexerApp -i data/sample.hmpg -f

The following command-line can be used to create a fragmented MP4 file from and MIHS bitstream file named “sample.hmpg”:

$ bin/HapticsMultiplexerApp -i data/sample.hmpg -f -F -d

### HapticsDemultiplexerApp

The HapticsDemultiplexerApp application can be used to extract an MIHS bitstream from an ISOBMFF (MP4) file. The command-line options for this application are shown in Table 3.

Table 3 - Command-line options for HapticsDemultiplexerApp.

|  |  |
| --- | --- |
| **Option** | **Description** |
| -d, --debug | Enables debugging |
| -i, --inputPath arg | Input MP4 filename |
| -o, --outputPath arg | Output MIHS bitstream filename |
| -f, --force | Overwrite output file if already exists |
| -v, --verbose | Verbose output |
| -h, --help | Print usage |

#### Usage examples

The following command-line extracts an MIHS bitstream from the MP4 file that was generated above with the HapticsMultiplexerApp:

$ bin/HapticsDemultiplexerApp -i data/sample.mp4 -o data/out.hmpg -d

# Conformance for ISO/IEC 23090-32

(TBD) MP4 Files, fMP4 Files and DASH MPD Files

1. (informative)  
     
   Reference software build process
   1. General

This annex describes the reference software build process.

* + 1. Dependencies

Dependencies are installed automatically using CMake. Currently this project has the following dependencies:

* + [libisomedia](https://github.com/MPEGGroup/isobmff): ISOBMFF reference software [3]
  + [spdlog](https://github.com/gabime/spdlog): C++ logging library [4]
  + [json](https://github.com/nlohmann/json): JSON header only library [5]
  + [cxxopts](https://github.com/jarro2783/cxxopts): Header only command-line options parser [6]
  + MPEG Haptics reference model (RM): MPEG Haptics Coding reference software [2] that supports parsing of ISO/IEC 23090-31:2025 [1] conformant bitstreams.
    1. Building

Git and CMake are needed to be installed on the system. The following steps can be followed to checkout the code from the repository and build it:

1. Clone project

git clone https://git.mpeg.expert/MPEG/Systems/FileFormat/hapticscarriageconfrefsoft.git 23090-32\_reference-software

1. Clone submodules

cd 23090-32\_reference-software

git submodule update --init --recursive

1. Apply a patch on submodules (bug fix, warnings fix). Done only once.

external/applyPatches.sh

1. Build

mkdir build

cd build

cmake ..

cmake --build . --config Release

After building the project, two command-line applications are generated in the in the “bin” directory: HapticsMultiplexerApp and HapticsDemultiplexerApp.

1. Run and test

cd ..

bin/HapticsMultiplexerApp.exe -h

bin/HapticsMultiplexerApp.exe -i data/sample.hmpg

ls -al data/sample.mp4

Note: The “data” directory contains an MIHS bitstream file named “sample.hmpg” that can be used for testing.

Bibliography

1. ISO/IEC 23090-31, Information technology — Coded Representation of Immersive Media — Part 31: Haptics coding
2. ISO/IEC 23090-33, Information technology — Coded Representation of Immersive Media — Part 33: Conformance and reference software for haptics coding
3. ISO/IEC 14496-32, Information technology — Coding of audio-visual objects — Part 32: File format reference software and conformance
4. spdlog, Online: <https://github.com/gabime/spdlog>
5. json, Online: <https://github.com/nlohmann/json>
6. cxxopts, Online: <https://github.com/jarro2783/cxxopts>