NGA.STND.0076-01_V1.0.0_GIMI
2024-10-23

NGA STANDARDIZATION DOCUMENT

GEOINT Imagery Media for Intelligence, Surveillance, and Reconnaissance (ISR) (GIMI)

Volume 1

Profile of ISOBMFF & HEIF

(2024-10-23)

Version 1.0.0

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Table of Contents

Table of Figures

Table of Tables

Introduction

This standard is for use by the GEOINT Enterprise, including all departments and agencies of the Department of Defense (DoD) and Intelligence Communities (IC). The Geospatial-Intelligence (GEOINT) Imagery Media for Intelligence, Surveillance, and Reconnaissance (ISR) standard is a profile of International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 14496-12 ISO Base Media File Format (ISOBMFF) and the ISO/IEC 23008-12 High Efficiency Image File Format (HEIF). NGA.STND.0076-01 enables formatting files of digital Still Imagery, image sequences, Motion Imagery, associated metadata, and imagery related products and exchanging them among members of the GEOINT Enterprise.

This standard describes a set of capabilities and requirements for a broad range of imagery forms and types. This standard is the result of an ongoing collaboration between the US Government and industry, for standardizing storing and exchanging media content comprising of imagery, metadata, and audio information.

When pictures appear in this standard, credit for the picture occurs with an annotation next to the first use of the picture. Pictures without a credit annotation in the first use are from sources within the Government and credit goes to the GEOINT Media Standards Board (GMSB).

The GMSB developed this standard using modern technical industry standards and information. Any comments or recommendations regarding this standard may be emailed to gmsbchair@nga.mil.

Revision History

Table 1: Change Log

Date	Version	Description	Developer
23 Oct 2024	1.0	Initial version.	NGA/TAES

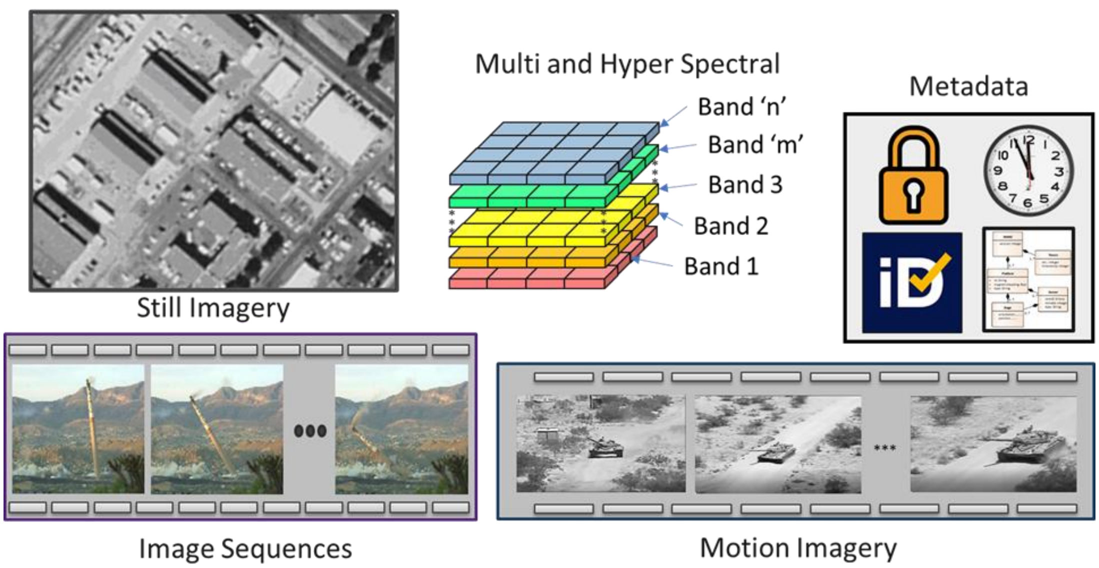
1 Technical Introduction

This standard defines a profile of both the ISOBMFF video/audio standard [1] and the HEIF Still Imagery standard [2]. The Geospatial-Intelligence Standards Working Group (GWG) provides this standard to harmonize Still and Motion Imagery file formats for use within the GEOINT Enterprise community [3], which includes U.S. and foreign government, commercial, and academic entities that produce, use, or contribute to one or more of the three elements of GEOINT (imagery, imagery intelligence, and geospatial data).

1.1 Purpose

The purpose of this standard is to define a modular container format for Still Imagery and time-based media (e.g., Motion Imagery, image sequences, and audio), and the media's supporting metadata. Mandatory inclusion of International Atomic Time (TAI) nanosecond formatted timestamps, information security markings of a dataset, and content identification (Content ID) information supports enterprise-wide uniqueness, traceability, effective search/discovery, and file releasability. The consistent labeling of time and identification with media items facilitate correlating individual pieces of content with other sources of information within the GEOINT ecosystem of ISR imagery and metadata.

This standard supports implementation of imagery solutions meeting stringent technical requirements in a common, interoperable industry format. At a high level, Figure 1 identifies the primary broad classes of media content covered by this standard.



Still Imagery, Image Sequences, and Motion Imagery graphics credit: White Sands Missile Range

Figure 1: Primary forms of media content covered in NGA.STND.0076-01

This standard places constraints on ISOBMFF and HEIF with specific implementation details addressing GEOINT/ISR needs. Requirements for declaring and encoding of metadata, along with requirements restricting the types and usage of coder-decoders (codecs) per type of media content ensure files conformant to this standard maximize interoperability. NGA.STND.0076-01 requires the ISOBMFF branding mechanism to specify file compatibility with

NGA.STND.0076-01. To implement solutions based on this standard, understanding the ISOBMFF and HEIF standards is necessary. Development of software libraries and applications must conform to the ISOBMFF and HEIF standards per limitations and restrictions as found in this document.

ISOBMFF defines a "base" set of extensible file format capabilities addressing a wide variety of media use cases, but with a focus on video and other sequential media. Other derivative documents extend the base to meet evolving needs of media workflows. The industry adoption, extensibility, and wide range of interoperable capabilities makes ISOBMFF an ideal media standard for the GEOINT Enterprise. Figure 2 lists the suite of ISO/IEC Imagery Standards addressing the storage (File Formats) of media content (both Motion Imagery and Still Imagery) and media coding (Codecs).

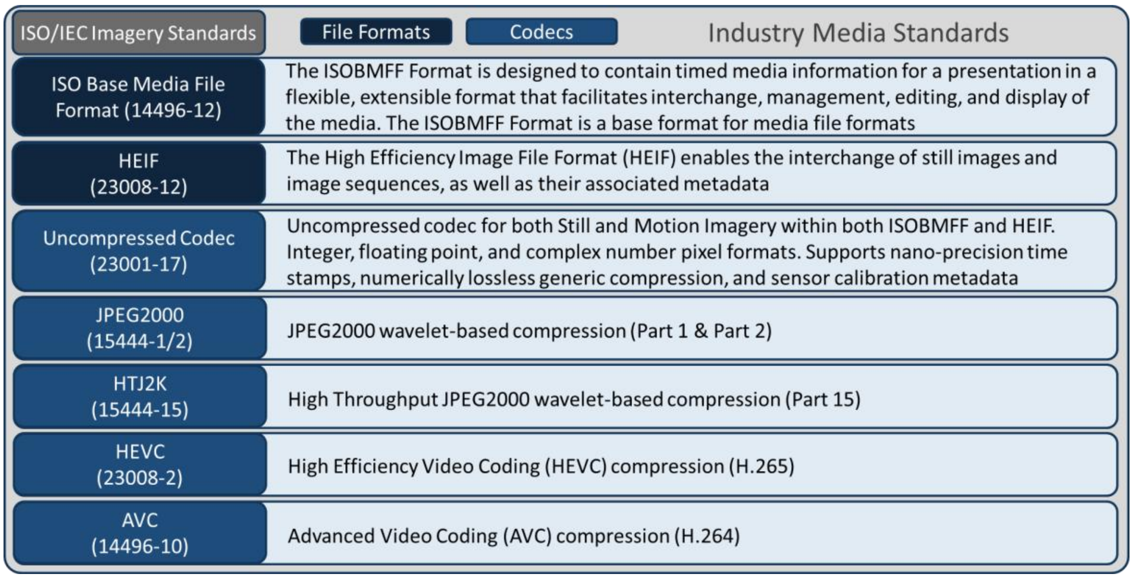


Figure 2: ISOBMFF has a broad range of extensible capabilities

ISOBMFF meets a wide variety of GEOINT use cases, such as high-throughput recording, processing, exploitation, analysis, cloud-based integration, automation, dissemination, and efficient operation with common tools and applications. The ability to define derivative profiles enables the creation of a standard tailored to GEOINT applications. ISOBMFF and its associated software libraries apply to a wide range of GEOINT use cases providing a framework for a high level of interoperability across many users and applications. Such a framework enables both Still Imagery and Motion Imagery content to use the same file container and workflow architecture, thereby simplifying the software infrastructure.

To facilitate interoperability with GEOINT Enterprise ontology structured metadata systems, this standard defines mechanisms for the declaration, carriage, and referencing of Resource Description Framework (RDF) content encoded in the Turtle syntax [4]. To facilitate use cases requiring extensibility and bit efficient encoding, this standard defines mechanisms for the use of SMPTE 336:2017 [5] Key-Length-Value (KLV) encoding for metadata. Certain application-specific metadata implementations may specify a different encoding scheme, such as XML coding. Information security markings, for instance, utilize the Office of the Director of National Intelligence (ODNI) specification for ISM.XML [6].

1.2 Scope

This standard supports the storage and containment of uncompressed imagery and a constrained set of compressed forms of imagery, with supporting metadata, for all types of GEOINT Enterprise ISR applications. The scope includes:

- Containment of imagery with a broad range of array sizes, frame rates, number of components (e.g., bands), variability in color formats, bit depths, and forms of coding. NGA.STND.0076-01 addresses numerous forms of two dimensional, rectangular forms of imagery, including:
    - Still Imagery
        - Panchromatic
        - Color
        - Infrared
        - Multispectral and Hyperspectral
        - Synthetic Aperture Radar (SAR) – including implementations with complex data type
    - Motion Imagery (Motion Imagery Standards Profile (MISP) [7] conformant)
        - Class 0 (uncompressed)
        - Class 1 (broadcast format compressed)
        - Class 2 (non-broadcast format compressed – machine vision, infrared, scientific)
    - Image Sequences
    - Image Collections (as defined by MPEG)
- Metadata
    - NGA.STND.0076-01 mandates core requirements for metadata, including:
        - Defining authorized industry and government metadata formats
        - Methods for encoding and carriage of metadata internal to a file
        - Methods for referencing metadata external to a file (e.g., a “sidecar” file)
    - NGA.STND.0076-01 defines GIMI-essential metadata (content identification and high precision timing), security metadata, and generalized application metadata. The metadata framework allows for the definition, encoding, and carriage of these metadata elements in a GIMI file.
    - NGA.STND.0076-01 supports the carriage of single instance and sequential metadata
- Audio
    - NGA.STND.0076-01 supports audio but does not mandate its use.

Topics outside the scope of this document and addressed in separate documentation include:

- ISOBMFF and HEIF Introductory material
    - NGA Standard Information/Guidance document NGA.SIG.0045-01 ISOBMFF Handbook [8] provides an overview of ISOBMFF
- Application profiles for Streaming
    - The transmission of Motion Imagery over networks, in both low latency, real-time environments and post-capture adaptive bit rate streaming environments
- Application profile for Large Volume Motion Imagery (LVMI)

2 Approach to Conformance

This standard utilizes the Easy Approach to Requirements Syntax (EARS) [9] to explicitly state the requirements for conformance to NGA.STND.0076-01. EARS reduces ambiguity, vagueness, and complexity in requirements writing, thereby facilitating conformance, and testing in an efficient and effective manner. A conformant implementation in accordance with this standard is one including all mandatory provisions (i.e., requirement "shall" statements).

2.1 Order of Precedence

Unless otherwise specified, the order of precedence for the types of normative information in this standard is as follows:

1. Requirements
2. Tables
3. Normative Text
4. Figures

All text in this document is normative unless explicitly labeled as 'Informative'. In the event of a conflict between the text of this standard (NGA.STND.0076-01) and the text in the ISOBMFF standard (ISO/IEC 14496-12) or the HEIF standard (ISO/IEC 23008-12), the ISO/IEC standards take precedence. Nothing in this standard supersedes applicable laws and regulations unless a government program obtains a specific exemption.

2.2 Conformance Testing & Certification

This standard references multiple consumer commercial, Government commercial, and Government media standards. To ensure maximum interoperability across systems and solutions, a testing and certification process must account for validating conformance to this standard, as well as the relevant normative references for a specific implementation of this standard. The accumulation of all requirements within this standard and the supporting normative ISO documents defines the scope of conformance testing to this standard. Implementations must be conformant to all requirements or to a subset of requirements defined by a specific acquisition program.

2.3 Abstract Test Suite (ATS)

An Abstract Test Suite (ATS) document is under development for NGA.STND.0076-01. The GEOINT Media Standards Board (GMSB) will publish the NGA.STND.0076-02 ATS document as a standalone document and it will address NGA.STND.0076-01 specific conformance topics. In addition to the NGA.STND.0076-02 ATS, implementers need to address conformance activities related to the foundational ISO/IEC ISOBMFF and HEIF standards as well as the various codecs adopted by a specific implementation.

3 References

The following documents, in whole or in part, and unless specifically labeled as informative, are normative references for this standard. Dated references indicate only the specific version of the document applies. Undated references indicate the latest approved version, including amendments, of the reference applies.

[1] *ISO/IEC 14496-12:2022 - Information technology - Coding of audio-visual objects - Part 12: ISO base media file format.*

[2] *ISO/IEC 23008-12:2022 - Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 12: Image File Format.*

[3] *Geospatial Intelligence (GEOINT) Basic Doctrine, Publication 1.0*, 2018-04.

[4] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux and Gavin Carothers, *RDF 1.1 Turtle: Terse RDF Triple Language. W3C Recommendation*, 2014-02-25. (http://www.w3.org/TR/2014/REC-turtle-20140225/)

[5] *SMPTE ST 336:2017 - Data Encoding Protocol Using Key-Length-Value.*

[6] *ODNI, XML Data Encoding Specification for Information Security Markings (ISM.XML), Version 2021-NOVr2022-NOV.* (https://www.dni.gov/index.php/who-we-are/organizations/ic-cio/ic-technical-specifications/information-security-marking-metadata)

[7] Motion Imagery Standards Board, *MISP-2023.2 - Motion Imagery Standards Profile*, 2023-03.

[8] *NGA.SIG.0045 Standard Information/Guidance (SIG) ISO Base Media File Format (ISOBMFF) Handbook for GEOINT Enterprise Community Applications [Informative],* 2024-10-23.

[9] Alistair Mavin, Philip Wilkinson, Adrian Harwood and Mark Novak, *Proceedings of the 2009 17th IEEE International Requirements Engineering Conference - EARS (Easy Approach to Requirements Syntax).*

[10] *ISO/IEC 23001-17:2024 - Information technology - MPEG Systems technologies - Part17: Uncompressed video and images in ISO Base Media File Format.*

[11] *ISO/IEC 15444-1:2024 - Information technology - JPEG 2000 image coding system: Part 1: Core coding system.*

[12] *ISO/IEC 15444-2:2023 - Information technology - JPEG 2000 image coding system: Part 2: Extensions.*

[13] *ISO/IEC 15444-15:2019 - Information technology - JPEG 2000 image coding system - Part 15: High-Throughput JPEG 2000.*

[14] *ISO/IEC 15444-16:2021 - Information technology - JPEG 2000 image coding system - Part 16 - Encapsulation of JPEG 2000 images into ISO/IEC 23008-12.*

[15] *ISO/IEC 23008-2:2025 - Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding.*

[16] *ITU-T H.265 (v10) - High efficiency video coding,* 2024-07.

[17] *ISO/IEC 14496-10:2022 - Information Technology - Coding of audio-visual objects - Part 10: Advanced Video Coding.*

[18] *ITU-T H.264 (v15) - Advanced video coding for generic audiovisual services,* 2024-08.

[19] *ISO/IEC 14496-15:2022 - Information technology - Coding of audio-visual objects - Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format.*

[20] *MISB ST 0107.5 - KLV Metadata in Motion Imagery,* 2021-10-21.

[21] T. Edwards, *RFC 5119 - A Uniform Resource Name (URN) Namespace for the Society of Motion Picture and Television Engineers (SMPTE).* (https://www.rfc-editor.org/info/rfc5119)

[22] *MISB ST 1902.2 - Motion Imagery Metadata (MIMD): Model-to-KLV Transmutation Instructions*, 2022-06-23.

[23] Z. Szabadka J. Alakuijala, *RFC 7932 - Brotli Compressed Data Format.* (https://www.ietf.org/rfc/rfc7932.txt)

[24] Motion Imagery Standards Board, *MISP-2023.2 - Motion Imagery Handbook*, 2023-03.

[25] *ODNI, XML Data Encoding Specification for Intelligence Community Identifier (IC-ID.XML), Version 2021-NOV.* (https://www.dni.gov/index.php/who-we-are/organizations/ic-cio/ic-technical-specifications/intelligence-community-identifier)

[26] *ISO/IEC 9834-8:2014 - Information technology - Procedures for the Operation of Object Identifier Registration Authorities - Part 8: Generation of Universally Unique Identifiers (UUIDs) and their use in Object Identifiers.*

[27] *NGA NGA.RP.0001_1.0.0 - NSG Recommended Practice for Universally Unique Identifiers.*

[28] *DoD Manual, DoD-M 5200.01 Volume 2: Marking of Information*, 2020-07-28.

[29] *Intelligence Community Directive 710, Classification Management and Control Markings System.*

[30] *ISO/IEC 13818-7:2006 - Information technology - Generic coding of moving pictures and associated audio information, Part 7: Advanced Audio Coding (AAC).*

[31] *ISO/IEC 14496-3:2019 - Information technology - Coding of audio-visual objects - Part 3: Audio.*

[32] *MISB ST 1001.1 - Audio Encoding*, 2014-02-27.

[33] *NIST.SP.330:2019 - The International System of Units (SI) 8th Edition.*

4 Terminology

Similar (or sometimes identical) terms denote different meanings across different communities of practice. This section clarifies the meaning and intent of terms used in this standard. This section gives the reader instruction and context on how to interpret the terms, with an emphasis on ambiguous and overloaded terms.

4.1 Acronyms & Initialisms

The following lists the acronyms and initialisms used in this standard.

ATS	Abstract Test Suite
AVC	Advanced Video Coding
DoD	Department of Defense
DS	Distribution Statement
EARS	Easy Approach to Requirements Syntax
GEOINT	Geospatial-Intelligence
GI&S	Geospatial Information & Services
GIMI	GEOINT Imagery Media for ISR
GMSB	GEOINT Media Standards Board
GWG	Geospatial-Intelligence Standards Working Group
HEIF	High Efficiency Image File Format
HEVC	High Efficiency Video Coding
HSI	Hyperspectral Imagery
HTJ2K	High Throughput JPEG 2000
IC	Intelligence Community
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IRI	Internationalized Resource Identifier
ISO	International Organization for Standardization
ISOBMFF	ISO Base Media File Format
ISR	Intelligence-Surveillance-Reconnaissance
ITU	International Telecommunication Union
J2K	JPEG 2000
JPEG	Joint Photographic Experts Group

KLV	Key Length Value
LSB	Least Significant Byte
LVMI	Large Volume Motion Imagery
MIMD	Motion Imagery MetaData
MISB	Motion Imagery Standards Board
MISP	Motion Imagery Standards Profile
MP4RA	MP4 Registration Authority
MPEG	Moving Picture Experts Group
MSB	Most Significant Byte
MSI	Multispectral Imagery
NAL	Network Abstraction Layer
NALuFF	NAL unit File Format
NGA	National Geospatial-Intelligence Agency
NITF	National Imagery Transmission Format
NITFS	National Imagery Transmission Format Standard
NSG	National System for Geospatial-Intelligence
NTB	NITFS Technical Board
ODNI	Office of the Director of National Intelligence
RDF	Resource Description Framework
SAI	Sample Auxiliary Information
SAR	Synthetic Aperture Radar
SDO	Standards Development Organization
SEMi	Scientific & Engineering Media
SI	Système International (International System of Units(E))
SIMA	Supplemental Image Metadata Array
SMPTE	Society of Motion Picture and Television Engineers
ST	Standard
STND	Standard
TAI	Temps Atomique International (International Atomic Time(E))
Turtle	Terse RDF Triple Language EBNF (Extended Backus-Naur Form)

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
XML	Extensible Markup Language
YUV	(Y) luma, (U) blue projection, (V) red projection

4.2 Terms and Definitions

When multiple normative references define the same term but with different definitions, this standard explicitly lists both definitions and intended usage of the term from each reference. This helps avoid confusion when definitions are not compatible. An example is the term *sample*. MPEG defines sample to indicate the data associated with a single time, such as a collection of pixels in a frame of video. The MISB defines sample to refer to the measurement of a single band of a pixel. The definitions for this situation qualify these terms as follow:

- Sample (MPEG): MPEG-sourced definition
- Sample (MISB): MISB-sourced definition

In addition, to support the context of usage, certain terms include a label to identify traceability to the relevant standards body the definition is sourced from. If there is a discrepancy between a definition listed here and the referenced source document, the definition of the source document takes precedence.

NGA.STND.0076-01 uses the following terms and definitions:

4CC	(MPEG) An ASCII encoded "four-character code" indicating data formatting requirements. Applications implement the 4CC code as a 32-bit big-endian unsigned integer with the first character in the MSB, last character in the LSB. There is no string termination.
Adjunct-content	Media content declared inside a GIMI file but residing outside the file, e.g., in a sidecar file.
Advisory timing	(MPEG) In HEIF files containing sequences, timing may be associated with the timing at collection, or the timing intended for display/playout. HEIF viewers may ignore timing and display sequence content in a static gallery. MPEG refers to this nature of sequence timing as advisory.
Application metadata	Use case specific information within a GIMI file. Application metadata is separate from GIMI-essential and Security metadata. For example, geospatial information, such as platform and sensor information, is Application Metadata.
Asynchronous	Not synchronous, e.g., measurements not occurring together in time. See definition of synchronous for details.
Audio	Sequential audio samples (MPEG), typically sound information for an accompanying video track.
Band	(MISB) An array of samples (MISB) where all phenomena are of the same type.
Block	(MPEG) Consecutive bytes within the sample data containing one or more component values for one or more pixels and any possible padding. (NTB) A rectangular array of pixels. (Synonymous with tile).
Box	(MPEG) An object-oriented container with a unique type identifier and length.

Broadcast format	Commercial forms of video using broadcast industry standards, such as SMPTE. For example, HD or 4K.
Class 0 Motion Imagery	(MISB) *Class 0 Motion Imagery* represents the collective requirements for uncompressed Motion Imagery, Metadata, Audio, and suitable Containers. Class 0 Motion Imagery allows for unconstrained Image Characteristics (see MISP Sec. 2.1.4.1), such as unrestricted Pixel Value Range, Number of Bands, Number of Pixels per Image, and Number of Images per second. Examples include 14-bit infrared, Bayer, 3-Band High Definition. Class 0 Motion Imagery converts to Class 1 Motion Imagery or Class 2 Motion Imagery by applying compression.
Class 1 Motion Imagery	(MISB) *Class 1 Motion Imagery* represents the collective requirements for compressed Motion Imagery, Metadata, Audio, and suitable Containers. Class 1 Motion Imagery is applicable when delivering monochrome and color Motion Imagery in cases where the transmission bandwidth prohibits the use of Class 0 Motion Imagery. MISP Section 2.1.4.1 constrains the Image Characteristics of Class 1 with limits in Pixel Value Range, Number of Bands, and Number of Images per second. Example: H.264/AVC compressed airborne Motion Imagery. Class 1 Motion Imagery, based on standards from commercial broadcast SDO's, uses three Bands of color.
Class 2 Motion Imagery	(MISB) *Class 2 Motion Imagery* represents the collective requirements for compressed Motion Imagery, Metadata, Audio, and suitable Containers. Unlike Class 1 Motion Imagery, Class 2 Motion Imagery allows for a non-constrained set of Image Characteristics. Examples include high frame rate scientific imaging, Large Volume Motion Imagery, and high bit-depth compressed infrared.
Class 3 Motion Imagery	(MISB) *Class 3 Motion Imagery* represents sources external to the NSG, such as cell phones, mobile devices, and surveillance cameras, which may use formats, compression, containers, and other technologies that do not conform to the MISP. The requirements specified for Class 3 Motion Imagery address conversion of Motion Imagery, Metadata, Audio and Containers to meet Class 1 Motion Imagery requirements. See the MISP for additional details about Class 3 Motion Imagery.
Codec	Shorthand for coder-decoder. A hardware or software component providing both an encode and decode function.
Collection	A set of items without any required relationships in time, space, and format.
Commercial	Non-Governmental, profit seeking organizations and activities involved with producing, buying, and selling media technology and/or products. (See additional definitions for Consumer Commercial and Government Commercial.)

Component	(MPEG) Part of the image data representing a single channel (or dimension) of the image. Note: a component may describe visual information such as luminance or chroma, or other information usually not intended for direct display such as depth or transparency
Component type	Assignment of meaning to a component using an association of an ontology class with the component's Content ID. In ISO/IEC 23001-17, component types indicate a nominal characterization of component data and how the data should be displayed. When this standard refers to the ISO/IEC 23001-17 usage of component type (per ComponentDefinitionBox), this standard uses the phrase “MPEG component type” to avoid confusion with the ontologically precise component type usage.
Component value	The numerical value of a component at a specific location in an image.
Compression	A transformation to convert uncompressed media content into compressed media content, occupying less storage size (i.e., byte count).
Consumer commercial	Commercial products specifically structured and implemented for the consumer marketplace. E.g., a 'heic' image from an iPhone, or a video stream of a movie delivered from Netflix.
Content ID	(MISB) A universally unique identifier for a specific sequence of bits representing a piece of media content. Content IDs provide enterprise-wide search and discovery ability.
Decode	To convert an encoded block of data back into its native form.
Decompression	A transformation to restore compressed media content, as closely as possible, to the media content's uncompressed state before being compressed.
Degrading image transformation	Modification to component values and color formatting where the uncompressed nature of an image is maintained, but the conversion is lossy, and some component value information is lost. Note: An uncompressed input to a degrading image transformation results in a lossy-uncompressed result. An RGB conversion to a YUV 4:2:0 image maintains an uncompressed status but results in a degrading image transformation as information in the U and V bands has been binned and the direct inspection at pixel locations for the U and V bands has reduced information.
Disadvantaged user	A user with the one or more limited resources (processing, network speeds, display resolution, software capabilities, etc.), limiting the ability to effectively handle visually eccentric imagery and possibly forms of visually normal imagery.
Dynamic media	Temporally varying media content. Dynamic media uses a temporal assignment (i.e., timestamp) to correlate with other dynamic media and for determining when events occur.

Dynamic metadata	Metadata parameters which vary over time. Dynamic metadata are normally stored as tracks but can be stored as Single-Article metadata when there is only one value in a dataset. Dynamic metadata are associated with timestamps to enable correlation with other information.
Empty string	(MPEG) a single Unicode NUL.
Encode	To convert data from its native, or natural form, into another form.
Entity	(MPEG) An item instantiated in a MetaBox, or a track instantiated in a TrackBox.
Entity group	(MPEG) A grouping of items and/or tracks sharing a particular characteristic or having a particular relationship.
Essential	(MPEG) In ISOBMFF, a property association may be marked as either essential or non-essential. When a writer marks an item property as essential, the reader does not process the associated item if the reader does not support the property. When a writer marks an item property as non-essential, the reader may process the associated item without using the property. Item property box definitions include essential and non-essential markings.
Flipbook	A sequence of independently coded image frames with fixed or smoothly varying scene content, each frame with required metadata, playable and displayed via user interactive control or defined sequencing, with no artificial constraints placed on parameters such as resolution, frame rate, bit depth, number of bands, compression type, interleaving mode, overlays, and quality. The time interval between frames may vary. The consistency of scene content differentiates from a slide show.
Frame	(MPEG) Two-dimensional rectangular array of pixels contained in the sample data. (MISB) A two-dimensional array of regularly spaced pixels in the shape of a rectangle indexed by rows and columns along with a start time and an end time of each pixel.
GEOINT Enterprise	The combination of technology, data, people, policies, capabilities, doctrine, activities, and organizations necessary to produce GEOINT in any government, academic, or commercial environment in the U.S. and partner countries.

Geospatial-Intelligence (GEOINT)	*Statutory definition codified in Title 10 of the U.S. Code, Section 467 (5):* *The exploitation and analysis of imagery and geospatial information to describe, assess, and visually depict physical features and geographically referenced activities on the earth. Geospatial intelligence consists of imagery, imagery intelligence, and geospatial information.* *From DoD Instruction (DoDI) 5000.56:* *All GI&S, imagery or geodesy data, product, or services are components of GEOINT and, for the purposes of this Instruction, shall hereafter be referred to as GEOINT. Imagery, imagery intelligence, and geospatial information, individually or in combination may be considered GEOINT.*
GIMI file	A file conformant to NGA.STND.0076-01. Interpreted to be equivalent to an "NGA.STND.0076-01 file".
GIMI-essential metadata	GIMI file metadata addressing content identification and TAI timing information.
Government commercial	Commercial products specifically structured and implemented to meet Government requirements. E.g., commercial satellite imagery delivered in a Government specified format.
GWG Focus Group	An NSG community forum for issues relating to a particular GEOINT standards area (e.g., MISB – Motion Imagery, NTB – Still Imagery).
Image collection	(MPEG) A set of images stored as items of a single file according to ISO/IEC 23008-12.
Image item	(MPEG) The storage of a single coded image or a single derived image.
Image sequence	(MPEG) An ordered series of coded images which may be associated with advisory timing and in which images may use inter prediction.
Image transformation	A manipulation of an input image, resulting in an output image.
Imagery specialist	A user with the necessary resources (processing, network speeds, software capabilities, etc.) to effectively handle visually eccentric forms of imagery.
Interframe compression	A form of compression leveraging dependencies across multiple images. Commonly uses inter-prediction across multiple frames, resulting in a dependency where decoders must decode some frames first before being able to decode the remaining frames.
International Atomic Time (TAI)	A high-precision, absolute time scale derived from hundreds of precise atomic clocks from around the world and maintained as closely as possible to the *Système International* (SI) second. Current practice achieves a maximum deviation of approximately one second every 100 million years.
Interoperability	(GWG) Ability of a system or a product to correctly function with other systems or products without special effort on the part of the user. Interoperability is made possible by the implementation of standards.

Inter-prediction	(MPEG) prediction derived in a manner that is dependent on data elements (e.g., sample values or motion vectors) of images other than the current image.
Interpret	To discern the meaning or inherent nature of some content.
Intraframe compression	A form of compression leveraging only the dependencies within a specific image.
Item	(MPEG) Data which does not require sequential processing, as opposed to sample data. Items may be an image or an instance of metadata.
Key-Length-Value (KLV)	A form of metadata encoding based on SMPTE 336M.
Large Volume Motion Imagery (LVMI)	(NTB) LVMI is a particular form of Motion Imagery characterized by large datasets with significant storage costs and unique dissemination and exploitation requirements. LVMI systems may utilize multiple cameras necessitating the need to compose multiple individual camera frames into a larger single frame. LVMI systems may operate at frame rates independent of broadcast video rates. LVMI implementations may use Class 0, 1, and 2 forms of Motion Imagery
Lossy compression	A form of image compression where decompression results in an image containing differences, to some degree, from the original.
Lossy uncompressed	Output from a degrading image transformation on an uncompressed image; or the output from a lossy process involving a decompression transformation on a compressed image. Transformation examples include conversion of uncompressed 4:4:4 to uncompressed 4:2:0, scaling a 14-bit image to 8-bits, or the result of the decompression of a lossy compressed image.
Media content	Digital data in an ISOBMFF file which portrays real-world information to an end user. E.g., Still Imagery, Motion Imagery, Image Sequences, Audio, sensor data, etc.
Metadata	(MISB) Data providing descriptive information about other data (including imagery), thereby providing context about the other data.
Metadata item	(MPEG) The storage of metadata as an (MPEG) item.
Monochromatic	Consisting of a single band of imagery.
Motion Imagery	(MISB) A sequence of images, that when viewed (e.g., with a media player) must have the potential for providing informational or intelligence value. This implies the images composing the Motion Imagery are: (1) from sensed data, and (2) relate to each other both in time and in space. The sequence of images enables detecting motion of the sensor and/or objects within.
Motion Imagery Metadata (MIMD)	(MISB) Contextual information, including temporal, platform, payload, sensor, command, automated processes, exploitation, and security, about Motion Imagery.
Movie	(MPEG) A sequential presentation, which may consist of one or more of imagery, audio, and metadata.

National System for Geospatial-Intelligence	NSG Members are IC and DoD organizations that fall under GEOINT Functional Management authorities, as outlined in EO 12333, DoD Directive 5105.60, and Intelligence Community Directive 113. NSG Members include the IC, the Joint Staff, the Military Departments (including the Services), and Combatant Commands (CCMDs).
Network abstraction layer	(MPEG) Enables the partitioning and containment of media content into units formatted for delivery by a streaming server.
Numerically lossless compression	A form of compression where the decompression process results in a bit-wise identical result to the original.
Panchromatic	(NTB) A single band covering a broad range of EO wavelengths, usually used in context of collecting information from much of the visible spectrum and sometimes part of the near infrared spectrum.
Pixel	(MISB) For a given single addressable location in an image, the combination of all band samples. (MPEG) Smallest element of an image, comprised of one or more components.
Primary-media-content	Media content in a GIMI file, configured as items (declared in a file-level, movie, or movie fragment MetaBox), tracks, or track portions, having content identifiers.
Reader	An application which processes GIMI media for display or other purposes.
Residual-media-content	Information related to, or adding context to, media content in a file, without a Content ID (in contrast to primary-media-content), but relevant to the security markings in the file. E.g., a non-uniformity correction box in the sample entry of an uncompressed image.
Resource Description Framework (RDF)	A semantic web data format standard for representing interconnected data.
Sample	(MISB) Data representing a measured phenomenon such as light intensity. (MPEG) All the data associated with a single time (e.g., tracks are composed of samples, i.e., a sample can be a frame of imagery). When 'sample' is used in the text without a (MISB) or (MPEG) qualifier, readers are to interpret it as an MPEG sample.
Sample auxiliary information	(MPEG) Metadata or other information associated with each sample in a track and carried using the 'saiz' and 'saio' boxes.
Security metadata	GIMI file metadata containing classification markings.
SEMi	Scientific & Engineering Media (SEMi): Imagery where the pixel count, the sample (MISB) value range, the number of color/spectral bands, and the frame rate are all unconstrained. Imagery may be coded as uncompressed or coded using numerically lossless compression or visually lossless compression.

Sensor data	A device's measurement information, representing some physical phenomena. For example, photons from an image sensor, time from a clock, rotation angles from a gimbal encoder, geodetic location from a GPS, slant range from a Laser Range Finder (LRF), pressure, temperature, etc. In this standard, the term "data" is synonymous with "sensor data." Sensor data can also be metadata when providing information about other data.
Sequence	A set of related items (MPEG) in a specific order.
Sequence track	A sequence of items (MPEG) placed on a timeline, via a track.
Sequential Application Metadata	A sequence of dynamic metadata captured with timestamps facilitating alignment with other media tracks.
Sequential-Packet Application Metadata	A sequence of individually coded metadata packets, with one or more elements inside, placed on a timeline, via a track.
Sequential-SIMA	A sequence of SIMA samples (MPEG) placed on a timeline, via a track.
Sidecar	A file containing only metadata and pertaining to one or more GIMI files.
Single Instance Application Metadata	A collection of one or more metadata elements stored as a single item (MPEG) in a MetaBox. Elements may be either static metadata or a single value of a parameter of dynamic metadata. The two forms of Single Instance Application Metadata are Single-Article Application Metadata and Single Instance SIMA Application Metadata.
Single-Article Application Metadata	An encoded collection of one or more individual pieces of application metadata, treated together as a single stored unit of metadata.
Single-SIMA	A single instance SIMA item stored in the file level MetaBox.
Slide show	A series of potentially unrelated images played as a sequence.
Static metadata	Metadata values which remain constant over a duration of time deemed sufficient for a given use case. Timestamps are not necessary for static metadata due to lack of change for a given use case duration. Static metadata is a subset of Single Instance Application Metadata.
Still image	(NTB) A multi-dimensional rectangular array of pixels indexed by row and column.
Still Imagery	A category of single images lacking the continuous temporal collection aspect of Motion Imagery. Without additional images, determining motion in a scene is much more difficult.
Supplemental Image Metadata Array	Imagery metadata arranged in one or more components of 2D arrays, where the array values align with individual pixels or groups of pixels in an image. Examples include pixel metrics, geodetics, or cloud cover.
Synchronous	Events or measurements occurring "together in time". The time window defining the state of being synchronous is dependent on application precision and accuracy requirements. For measured events, timestamp precision and uncertainty metadata provide details on the level of two measured data elements or streams being "synchronous".

Synthetic Aperture Radar A method of radar imaging for creating 2D images or 3D reconstructions of the earth's surface.

Synthetic data Data resulting from a software simulation of real-world sensors. For example, AI generation of imagery, depicting life-like scenes, diagrams, or other types of fabricated image content. Not associated with synthetic aperture radar.

TAI clock A clock capable of synchronizing to a source of TAI time and generating TAI timestamps.

Tile (MPEG) Two-dimensional rectangular array of pixels within a frame. Synonymous with block (NTB).

Track (MPEG) A timed sequence of related media samples.

Turtle A compact, plain text format for serializing RDF datasets.

Uncompressed Not compressed.

Uncompressed frame (MPEG) Frame for which each value of each component is coded independently from any other component value in the same frame or any other frame.

Uncompressed image (MPEG) Single uncompressed frame stored as an image item.

Uncompressed video (MPEG) Sequence of one or more uncompressed frames.

Unitary image An image arrangement with all pixels in a single contiguous block (i.e., without tiling).

Video (MISB) An ordered series of image frames with each frame assigned an increasing presentation time, where the presentation time is a relative time.

Visually lossless compression A form of image compression where decompression results in an image which is visually indistinguishable from the original to the average user.

Visually-eccentric image An image not readily displayable, using common libraries and applications, due to it being complex, large, or based on an uncommon encoding form (high bit depth, extremely high pixel count, number of bands greater than three, etc.). There is gray area between visually-normal and visually-eccentric based on resources, access to technology, etc. (see the definitions for imagery specialist and disadvantaged user). The line between visually-eccentric and visually normal is not hard and fast and may change over time due to technological advances (e.g., standard definition to high definition to 4K, etc.). These terms are provided to help implementers and users label and understand requirements and capabilities in a general way.

Visually-normal image An image readily displayable with common image viewers, i.e., an RGB or monochrome image with a general device-friendly resolution and bit depth.

Writer Applications which create or edit GIMI files.

YUV A color encoding system where Y carries brightness information and U and V carry color information.

5 General Overview

NGA.STND.0076-01 provides an integral foundation for GEOINT ISR imaging workflows, addressing file-based imagery activities, and facilitating applications ranging from collection to processing, analysis, exploitation, dissemination, and archive. Additional companion profiles, formats and containers address other aspects of media handling (such as streaming and metadata "sidecar" files) to create a comprehensive capability for the GEOINT Enterprise.

NGA.STND.0076-01 is a container for multiple types of content, such as imagery, audio, and metadata. NGA.STND.0076-01 builds upon the ISOBMFF structure which supports containment of a wide range of imagery types, categorized by:

- Temporal makeup – still, motion, sequences, slide shows, flip books, etc.
- Spatial makeup – unitary images (one tile or block), tiled/gridded images, overviews/image pyramids, etc.
- Spectral makeup – monochromatic, panchromatic, color, patterned, MSI, HSI, ultraviolet, visible, near infrared, short wave infrared, mid-wave infrared, and long-wave infrared, etc.
- Numerical formatting – integer, floating point, and complex number
- Image coding – uncompressed, lossless compression, lossy compression, intraframe compression, and interframe compression

NGA.STND.0076-01 specifies how to implement and utilize ISOBMFF and HEIF capabilities for both producer and consumer GEOINT applications. Unless specifically restricted by a requirement in this standard, all features described in the ISOBMFF family of standards, are available for implementation and use.

NGA.STND.0076-01 supports the carriage of metadata to facilitate GEOINT Enterprise needs. This standard defines three primary categories of metadata, GIMI-essential, Security, and Application.

5.1 GIMI-Essential Metadata

GIMI-essential is mandatory metadata providing universally recognized attributes about the media content within a GIMI file. There are two types of GIMI-essential metadata:

1. Content Identification (Content ID)
2. High precision timing information

Content IDs are identifiers, enabling enterprise-wide uniqueness for each piece of content within a GIMI file. NGA.STND.0076-01 defines mechanisms for the labeling of content with Content IDs to support the linkage of ontology structured metadata with media content it describes.

High precision timing information is the combination of a media timestamp and clock details. Each media timestamp is a count of nanoseconds since the TAI epoch. The clock details include the clock pedigree and per timestamp status metadata. NGA.STND.0076-01 requires dynamic media content to include a timestamp.

Associating Content IDs and timing information to GIMI content enables global search and discovery and provides external files and systems an ability to reference (or “point to”) content within a GIMI file. Reference-able content is primary-media-content and includes Still Image items, image sequence and Motion Imagery tracks, Motion Imagery frames, groups of image items and/or tracks, image components, and user defined regions of interest.

5.2 Security Metadata

For applications within the domain of the NSG, NGA.STND.0076-01 mandates each GIMI file include information security markings. For other applications within the GEOINT Enterprise, but outside the scope of the NSG, the implementation of security markings is outside the scope of this standard. For these GEOINT Enterprise applications, NGA.STND.0076-01 provides generic methods for declaring metadata, enabling other bodies, such as NATO, with the ability to carry other forms of security markings.

Within the NSG, this standard invokes the ODNI ISM.XML specification and directly encodes and carries security information in the form of GIMI Security XML. NSG applications producing unclassified files must include the GIMI Security XML markings. NGA.STND.0076-01 specifies how to link GIMI content to a classification and where to insert the security information into a GIMI file. The ODNI standard evolves independently, and existing tools supporting the ODNI standard can directly support NGA.STND.0076-01 applications within the NSG.

5.3 Application Metadata

Application metadata addresses a large scope of contextual information about the content and includes details about the systems and components which originally captured the imagery, e.g., cameras, sensor balls, range finders, platforms, etc. Application metadata enables the instantiation of rigorous sensor models for all types of Still and Motion Imagery sensors allowing for imagery exploitation, and research and development of secondary reference products.

NGA.STND.0076-01 optionally carries application metadata in the form of RDF or KLV encoded metadata. Focus Groups within the GWG define ontologies for use within the imagery domain. Applications store ontology structured metadata using Turtle encoded RDF. Turtle supports querying and inferencing so backend applications can easily extract metadata. Turtle either embeds within a GIMI file or the Turtle may be a stand-alone file, here after called a "sidecar file". Typically, systems merge Turtle into graph databases to facilitate analysis and exploitation.

The Focus Groups additionally define imagery domain KLV sets and packs, which this standard supports. KLV is a compact binary encoding format, so real-time collection use cases, where bandwidth and storage constraints are a primary concern, use KLV.

To address constraints in real-time and address functionality required during processing, analysis, and exploitation activities, workflows may consider using each specific metadata encoding method (i.e., KLV or Turtle) at different stages of an overall mission process.

Carriage of application metadata in separate files is a primary design feature of the NGA.STND.0076-01 architecture which enables metadata additions without disturbance of the original imagery content.

6 Detailed Description and Requirements

6.1 File Format Interoperability Indicators

The ISOBMFF standard defines a "branding" method for identifying what types of content an ISOBMFF file contains. NGA.STND.0076-01 relies on, and extends, the existing branding method to identify specific NGA.STND.0076-01 capabilities. The ISOBMFF Handbook (NGA.SIG.0045) describes the ISOBMFF file structure and the box structure (i.e., FileTypeBox) which contains the branding list.

6.1.1 FileTypeBox and Branding

Per ISOBMFF, the FileTypeBox ('ftyp') carries branding information identifying the specifications to which the file complies. Brands dictate requirements for writers when generating files and for readers when decoding, interpreting, and presenting content. Brands indicate conformance to the requirements of the brand's specification. Brands may indicate the use of a specific media encoding method or constraints placed on a specific encoding method.

The FileTypeBox carries a major brand, a minor version, and a list of compatible brands. The major brand indicates the primary intended use of the file. NGA.STND.0076-01 does not use the minor version. The list of compatibility brands defines the requirements for a reader when accessing content within a file. The complete list of brands (major plus compatibility brands) defines the full set of requirements and constraints needed for a reader to open and access all the file's content.

Readers conforming to NGA.STND.0076-01 may ignore:

- Features not addressed by this standard, such as out-of-scope metadata encoding formats (MPEG-7, etc.).
- Features non-conformant to this standard, such as the use of codecs not approved for use by this standard.

Readers may be fully or partially conformant to the NGA.STND.0076-01 standard. Fully conformant readers must comply with all requirements of this standard, as well as all requirements associated with brands delineated in the requirements of this standard. The requirements in NGA.STND.0076-01 related to security markings are specifically directed at NSG applications. Files generated outside the scope of the NSG may have unique security marking requirements, and therefore a different implementation for security markings.

Readers may claim partial conformance to this standard by clearly stating conformance to a specified subset of the listed media coding brands. For example, a GIMI reader application which only decodes JPEG 2000 imagery is considered partially conformant when it complies with this standard's requirements, the requirements of the JPEG 2000 brand, and not the requirements of the other GIMI imagery coding brands.

Readers only need to be conformant with and recognize one of the brands listed in a file to determine if the reader can make use of the file. Scanning the list of brands on file open enables reader applications to determine the level of interoperability expected with the content in the file. Per ISOBMFF requirements, readers must properly read and interpret all box content associated with a conformant brand and ignore unrecognized boxes and media (i.e., readers do not crash from unrecognized boxes or media).

To meet requirements, readers must support specific brands for Motion Imagery as listed in Annex E of ISO/IEC 14496-12, and specific brands for Still Imagery as listed in Section 10 of ISO/IEC 23008-12. These sections list the evolution of the brands, and the features enabled with each new version. Requirements in this standard specify approved 4CC brands. Readers conformant with versions of ISO brands listed in this standard (or more recent versions of the brands) are capable of reading and understanding files conformant with this standard.

GIMI file writers must generate files conformant to the NGA.STND.0076-01 brand's requirements plus all requirements derived from additional media branding the writer adds to the compatible brands list. For example, a writer's application only needs to write JPEG 2000 media, so the writer implements all the NGA.STND.0076-01 requirements and only the JPEG 2000 requirements, i.e., the writer does not need to implement other media requirements such as H.264. To address situations involving the ingest of Class 3 Motion Imagery content into a GIMI workflow, writers may add content to GIMI files using brands not on the NGA.STND.0076-01 brands list. In these cases, applications transcode the original content into a GIMI conformant form to achieve workflow interoperability and the application generates GIMI conformant metadata (timestamps, etc.) in the best manner possible. Applications may include the original content (based on non-GIMI brands) to maintain a copy of the original content in the same file. This provides one method for retaining original source data while ensuring interoperability with GIMI workflow tools.

Every brand defines an identifier or code-point, which is in the form of a four-character code or 4CC, e.g., "mif2". The MP4 Registration Authority (https://mp4ra.org/#/) maintains and manages a list of all registered code-points on behalf of ISO in support of the MPEG and JPEG communities. The MP4 Registration Authority code-points include brands, boxes, codecs, handlers, etc. As the GMSB generates brands and other code-points for this standard, the Government submits the code-points to the MP4RA for registration.

This specification includes a brands list which enables GIMI files to meet the needs of GEOINT Enterprise ISR and GEOINT applications. As ISOBMFF and HEIF evolve, NGA may update the NGA.STND.0076-01 brands list to exercise emerging capabilities.

ISOBMFF supports “dual-branding”, allowing for the carriage of both Still Imagery and Motion Imagery content within the same file. A dual-branded file contains both Still Imagery with its associated metadata and Motion Imagery with its associated metadata. Files containing multiple media types include the necessary brands for properly interpreting each type of media content.

6.1.2 Conformant Branding

This standard defines brands for the unique capabilities this standard specifies, including the encoding and carriage of required GEOINT related media content. When writers implement NGA.STND.0076-01, the compatible brands list must include the NGA.STND.0076-01 brand, which then identifies a file as being conformant with NGA.STND.0076-01's unique capabilities.

Expect future versions of this standard to include additional brands to support specific implementation capabilities; for example, including high dynamic range compression via high profile codecs. Future brands will define additional interoperable capabilities within the standard.

Brand names are four characters, so NGA.STND.0076-01 defines the first three characters of the brand name as 'geo' and the fourth character is a version identifier. The 'geo' term reflects the carriage of geospatial imagery. Following the 'geo' prefix is a version number. The first version is 'geo1' and following versions will increment by one until reaching 9, then the version digit will shift to a, b, c, etc. Applications include the 'geo' + version brand in the compatible brands list to indicate the file is conformant to NGA.STND.0076-01's unique capabilities and the file contains geospatial media.

The scope of the 'geo1' brand includes rectangular, uncompressed, and compressed forms of Still Imagery, including monochrome, color visible, panchromatic, MSI, HSI, and SAR. The scope also includes rectangular forms of Class 0 (uncompressed), Class 1 (broadcast, compressed), and Class 2 (SEMi, compressed) Motion Imagery. The presence of the 'geo1' brand ensures conformant software applications can operate with these delineated forms of imagery.

NGA.STND.0076-01 also defines the 'sm01' brand to indicate the presence of GIMI Security XML containing U.S. Government ODNI ISM.XML security markings (see Section 6.8). All NSG activities generating GIMI files must carry the 'sm01' brand in the compatible brands list. All other activities are exempt from the use of the 'sm01' brand and exempt from the inclusion of GIMI Security XML markings. This branding methodology and interpretation supports adding additional security marking brands to accommodate other forms of security markings (e.g., NATO), by creating additional brands (such as 'sm02', etc.). Commercial applications using NGA.STND.0076-01 and not requiring security markings do not carry a security marking brand.

This version of the standard uses the HEIF defined 'mif2' brand for Still Imagery, the HEIF defined 'msf1' brand for image sequences, the ISOBMFF defined 'isob' brand for Motion Imagery, and the ISOBMFF defined 'unif' brand mandating unique IDs for items and tracks in a file. This standard requires fully conformant file readers to be conformant with all features of the specific brand versions listed in Table 2, which summarizes the NGA.STND.0076-01 brands. This standard also requires partially conformant file readers to be conformant with all features of the specific brands and brand versions with which the reader claims conformance. File writers may choose to limit features to a prior version of a specific brand. In this case, the compatible brands list includes the prior brand the file is conformant to, informing readers of the lesser scope of content within the file.

In Table 2, the Brand column states the 4CC code name for the brand. The Source column specifies the standards document which defines the brand. The Purpose column defines when to use the brand, e.g., if the file contains Still Imagery, then declare and follow the requirements for the 'mif2' brand. The “(All)” indicates all GIMI files include the brand. The Description column provides additional information about the brand.

Table 2: Descriptions for Brands for NGA.STND.0076-01 Implementation

Brand	Source	Purpose	Description
'mif2'	HEIF	Still Imagery	The 'mif2' brand represents interoperability requirements for image items and associated metadata items. 'mif2' represents a baseline for Still Imagery support in this standard. The HEIF standard documents the specifics of the branding differences.
'msf1'	HEIF	Imagery Sequences	The 'msf1' brand indicates the presence of a HEIF defined image sequence.

Brand	Source	Purpose	Description
'isob'	ISOBMFF	Media Tracks, including Motion Imagery	The 'isob' brand addresses boxes defined in ISOBMFF, many of which directly address the carriage of video and other media tracks.
'unif'	ISOBMFF	Unique Identifiers (All)	The 'unif' brand indicates the unified implementation and handling of IDs across file-scoped MetaBox items, tracks, track groups, and entity groups.
'geo1'	NGA.STND.0076-01	GIMI File (All)	The 'geo1' brand indicates a file is conformant to version 1 of this standard (NGA.STND.0076-01_V1.0_GIMI). Future versions of the standard will increment the brand number accordingly.
'sm01'	NGA.STND.0076-01	Security Markings	The 'sm01' brand indicates the file carries GIMI Security XML markings.

Along with the brands listed in Table 2, additional brands indicate the presence of media content encoded using a specific codec. Codec brands indicate the requirements for decoding and interpreting encoded content. Many codecs support branding for distinctive features, such as higher-level profiles. The inclusion of uncompressed images and video, via ISO/IEC 23001-17 [10] does not require a special brand. Branding requirements for a specific implementation requires research into the capabilities implemented, the ISO standard which defines the capabilities, and the specific brand version addressing the implemented capabilities. Brands for both formats and codecs evolve over time as new capabilities are added and new editions and amendments are approved. Each revision of ISO/IEC documentation describes the capabilities implemented under each version of a brand.

Table 3 shows an example of a FileTypeBox. The first two columns, Data Type and Element Name, are directly from the ISOBMFF specification and define the type for the 4CC code, an unsigned int(32) value, and the class attribute name. The example value is the string representation of what is in the box, the quotes are not part of the value. This example is for a HEIF conformant file containing a JPEG 2000 encoded image item (indicated by the 'mif2' and 'j2ki' brands) and an HEVC Main Profile encoded image sequence (indicated by the 'msf1' and 'hevc' brands). The 'geo1' brand signals conformance to NGA.STND.0076-01 which requires the inclusion of the 'unif' brand. The 'sm01' brand indicates the presence of GIMI Security XML markings in the file. The Compatible Brands list is unordered.

Table 3: Example payload of a FileTypeBox

Data Type	Element Name	Example Value
unsigned int(32)	Major Brand	'mif1'
unsigned int(32)	Minor version	0
unsigned int(32) [ ]	Compatible Brands	'mif2', 'msf1', 'geo1', 'sm01', 'j2ki', 'hevc', 'unif'

Requirement(s)	
NGA.STND.0076-01_V1.0-01	A version 1.0 NGA.STND.0076-01 file shall include the 'geo1' brand in the compatible brands list.
NGA.STND.0076-01_V1.0-02	An NGA.STND.0076-01 conformant reader shall correctly process the 'geo1' brand's associated box content.
NGA.STND.0076-01_V1.0-03	An NGA.STND.0076-01 file shall include the 'unif' brand in the compatible brands list.
NGA.STND.0076-01_V1.0-04	An NGA.STND.0076-01 file shall conform to the requirements of the ISOBMFF 'unif' brand.
NGA.STND.0076-01_V1.0-05	Where an NGA.STND.0076-01 file contains Still Imagery content, the file shall conform to the 'mif2' brand requirements.
NGA.STND.0076-01_V1.0-06	Where an NGA.STND.0076-01 file contains Still Imagery content, the file shall include the 'mif2' brand.
NGA.STND.0076-01_V1.0-07	Where an NGA.STND.0076-01 file contains the 'mif2' brand, a GIMI reader conformant with the 'mif2' brand shall correctly process the 'mif2' brand's associated box content.
NGA.STND.0076-01_V1.0-08	Where an NGA.STND.0076-01 file contains image sequence content, the file shall conform to the requirements associated with the 'msf1' brand.
NGA.STND.0076-01_V1.0-09	Where an NGA.STND.0076-01 file contains image sequence content, the file shall include the 'msf1' brand.
NGA.STND.0076-01_V1.0-10	Where an NGA.STND.0076-01 file contains the 'msf1' brand, a GIMI reader conformant with the 'msf1' brand shall correctly process the 'msf1' brand's associated box content.
NGA.STND.0076-01_V1.0-11	Where an NGA.STND.0076-01 file contains media track content, the file shall conform to the requirements associated with the 'isob' brand.
NGA.STND.0076-01_V1.0-12	Where an NGA.STND.0076-01 file contains media track content, the file shall include the 'isob' brand.
NGA.STND.0076-01_V1.0-13	Where an NGA.STND.0076-01 file contains the 'isob' brand, a GIMI reader conformant with the 'isob' brand shall correctly process the 'isob' brand's associated box content.
NGA.STND.0076-01_V1.0-14	Where an NGA.STND.0076-01 file contains GIMI Security XML, the file shall conform to the requirements associated with the 'sm01' brand.
NGA.STND.0076-01_V1.0-15	NSG applications generating NGA.STND.0076-01 files shall include the 'sm01' brand in the FileTypeBox.
NGA.STND.0076-01_V1.0-16	Where an NGA.STND.0076-01 file contains the 'sm01' brand, a GIMI reader conformant with the 'sm01' brand shall correctly process the 'sm01' brand's associated security marking content.

Requirement(s)	
NGA.STND.0076-01_V1.0-17	Where implementing a capability from a normative reference, the NGA.STND.0076-01 file shall conform to all requirements for the capability.
NGA.STND.0076-01_V1.0-18	An NGA.STND.0076-01 conformant reader shall ignore boxes the reader does not recognize.

6.1.3 File Extensions

While file extensions frequently signal a specific format and interoperability requirements, an ISOBMFF file extension carries less definitive meaning. Brands in the FileTypeBox communicate interoperability requirements. When an ISOBMFF file includes multiple brands the ISOBMFF guidance is to select a file extension identifying the primary use of the file. For instance, files carrying both Still and Motion Imagery may use a file extension of '.heif' or '.mp4'. The writer application determines the best option for the extension. Reader applications utilize brands to determine which specification(s) a file and its contents conforms to, regardless of the extension. The presence of the NGA.STND.0076-01 'geo1' brand provides a definitive marker, indicating the file is conformant with this specification. Readers treat files without the 'geo1' brand as non-NGA.STND.0076-01 imagery, such as Class 3 in the case of Motion Imagery.

6.2 Imagery Content

The focus of this standard is the storage of media content, the labeling of the content, and identifying media content relationships. This section covers the nature of imagery in general and the forms for uncompressed and compressed imagery when stored in a file. The following sections cover the details of Still Imagery, image sequences, and Motion Imagery. Section 6.3 discusses metadata content and its association with imagery.

6.2.1 Still Imagery and Image Sequence Codecs

This standard addresses a broad range of imagery forms and types found in the GEOINT Enterprise community. This includes one (e.g., IR) and three band imagery (e.g., visible), multi and hyperspectral imagery, SAR imagery, imagery with unusually high resolutions and/or high dynamic range, and imagery with extended definition storage formats, such as floating point and complex numbers.

ISOBMFF and HEIF are agnostic to image encoding methods and support a wide variety of existing codecs. NGA.STND.0076-01 allows a select sub-set of codecs to balance the trade space of functional coverage, supporting legacy needs, and maximizing interoperability. Table 4 lists the approved codecs for Still Imagery and image sequence coding. The first column lists each encoding method, the second column identifies the associated codec standard, and the third column indicates the standard for carriage of the coded item within a HEIF file.

The ISO/IEC 23001-17 uncompressed codec, as well as numerically lossless implementations of other codecs, such as JPEG 2000, are for applications where preservation of content and quality of imagery is of primary concern. For situations where storage and transmission efficiency are of primary consideration, lossy compression options, including those with visually lossless compression settings, are available.

Different codecs, and their various profiles, place constraints on the imagery. The codecs constrain options for arranging component and pixel content, such as tiling, interleaving, padding, sub-sampling, and pixel alignment on defined byte boundaries. Component bit-depth is a primary feature constraint in codecs focused on broad or common use in the consumer media space. Readily available codecs generally support 8/10 bit visually-normal imagery. Codecs with higher level profiles supporting above 10-bits are representative of those supporting visually-eccentric imagery (are not readily displayable in their native form) and tend to be more complex, require more processing capabilities, are less common is general use, and are less interoperable with common industry software implementations. GEOINT Enterprise workflows commonly use imagery with greater than 10-bits of dynamic range.

JPEG 2000 (including Part 1 [11], Part 2 [12], and HTJ2K [13]) is a primary set of codec capabilities for GEOINT Still Imagery applications. The underlying wavelet technology supports high dynamic range, high density images, high numbers of components, and unique methods for accessing regions-of-interest without having to load and decompress the entire image. To support the need for greater processing efficiency, this standard supports the use of High-Throughput JPEG 2000 (HTJ2K). In addition to Still Imagery, JPEG 2000 and High Throughput JPEG 2000 coding systems support implementation as timed image sequences via ISO/IEC 15444-16 [14].

While HEVC and AVC are principally video codecs, they do support Still Image and image sequence compression. These codecs find use in applications requiring high interoperability, such as disaster relief, and dissemination to disadvantaged users, as well as applications for extracting and storing a single frame from an HEVC or AVC coded image sequence file.

Table 4: List of codecs approved for use with Still Imagery and Image Sequences*

Encoding Method	Codec Standard	Carriage in HEIF
Uncompressed Video and Images in ISOBMFF	ISO/IEC 23001-17	ISO/IEC 23001-17
JPEG 2000 (Part 1 & Part 2)	ISO/IEC 15444-1, ISO/IEC 15444-2	ISO/IEC 15444-16
High Throughput JPEG 2000	ISO/IEC 15444-15	ISO/IEC 15444-16
HEVC	ISO/IEC 23008-2	ISO/IEC 23008-12
AVC	ISO/IEC 14496-10	ISO/IEC 23008-12

* *Note: Government programs may independently require supporting multiple or all codecs listed and may also constrain support to specific codec profiles, depending on program application needs.*

Requirement(s)	
NGA.STND.0076-01_V1.0-19	Where an NGA.STND.0076-01 file contains Still Imagery content, the coding of the Still Imagery shall conform to one of the codec standards listed in Table 4.
NGA.STND.0076-01_V1.0-20	Where an NGA.STND.0076-01 file contains Still Imagery content coded with one of the options in Table 4, the carriage of the Still Imagery shall conform to the associated carriage standard listed in Table 4.

6.2.1.1 Alternate Versions of Content

In cases of an uncommon form of pixel (high bit depth, floating point, etc.) or codec profile (visually-eccentric imagery), it may be advantageous to re-encode the image using a common codec, such as HEVC, and include this visually-normal imagery as an alternate version of an image item or track. This improves interoperability with commodity devices and software allowing a higher percentage of applications to view some version of imagery in the file. The utility of this approach is dependent on a variety of issues related to workflow, security, stakeholder resources, long term storage cost, etc. In applications requiring the highest level of fidelity, such as exploitation, developers need to take care to ensure users are working with the highest quality image available. Writers include the visually-normal version by creating an entity group of type alternate ('altr'), containing both forms of the image item to inform readers the visually-normal image item is an alternate, more interoperable, and displayable form of the visually-eccentric image item. ISOBMFF stores entity groups in groups list boxes ('grpl').

6.2.1.2 Image Sequences

An image sequence is an ordered series of images. The following traits support image sequences:

1. Sequences are an ordered series of coded images.
2. Sequences 'may' be associated with advisory timing.
3. The coding of images may use inter-prediction.

Image sequences are composed of image items but are playable in a flipbook manner, or like a Motion Imagery track. Readers identify image sequences by their handler type, which is set to 'pict'. Sequences have more flexibility in their implementation than video tracks as their timing is less restrictive. The images in a sequence support playout as a movie, as a slideshow, with pauses and transitions, or displayable as a single composited frame. Image sequence declaration utilizes the MovieBox ('moov') and TrackBox ('trak') structures. This provides a configurable timing structure and options for attaching timestamps, metadata, and audio to a sequence. These tools provide an ability to playout a collection of image items in a sequential presentation. The editing features available within a track facilitate presentation options, such as looping sequences.

Requirement(s)	
NGA.STND.0076-01_V1.0-21	Where an NGA.STND.0076-01 file contains image sequence track content, the image sequence content shall conform to one of the codec standards listed in Table 4.
NGA.STND.0076-01_V1.0-22	Where an NGA.STND.0076-01 file contains image sequence track content coded with one of the options in Table 4, the carriage of the image sequence track shall conform to the associated carriage standard listed in Table 4.

6.2.2 Motion Imagery Codecs

Motion Imagery is categorized in the MISP into the following four defined classes:

- Class 0 MI: represents uncompressed forms of Motion Imagery.
- Class 1 MI: broadcast forms of compressed Motion Imagery, such as SD, HD, 4K, etc.
- Class 2 MI: compressed Motion Imagery with unconstrained image characteristics

- Class 3 MI: forms of compressed Motion Imagery not conformant to the MISP. Common sources include cell phones, security cameras, etc.

GIMI files can natively carry Class 0, 1, and 2 Motion Imagery. NGA.STND.0076-01 declares Motion Imagery in a file using the MovieBox and the TrackBox. To signal a track is carrying Motion Imagery, the handler type for the track is set to 'vide'. The SampleEntry signals the encoding method for the imagery. Approved codecs for the different Motion Imagery classes are:

- Class 0 Motion Imagery applications use the Uncompressed Video and Images in ISOBMFF (ISO/IEC 23001-17) codec.
- Class 1 Motion Imagery applications may use the HEVC (ISO/IEC 23008-2) or AVC (ISO/IEC 14496-10) codecs, with Levels and Profiles as defined by the MISP.
- Class 2 Motion Imagery may use the HEVC and AVC codecs.

Table 5, Table 6, and Table 7 list codecs approved for use with Motion Imagery implementations. The first column lists each encoding method, the second column identifies the associated codec standard, and the third column indicates the standard(s) for carriage in ISOBMFF.

Table 5: Codecs approved for use with Class 0 Motion Imagery*

Encoding Method	Codec Standard	Carriage in ISOBMFF
Uncompressed Video and Images in ISOBMFF	ISO/IEC 23001-17	ISO/IEC 23001-17

Table 6: Codecs approved for use with Class 1 Motion Imagery*

Encoding Method	Codec Standard	Carriage in ISOBMFF
HEVC/H.265	ISO/IEC 23008-2	ISO/IEC 14496-12, ISO/IEC 14496-15
AVC/H.264	ISO/IEC 14496-10	ISO/IEC 14496-12, ISO/IEC 14496-15

* *Note: programs/projects may independently require supporting multiple or all codecs listed and may also constrain support to specific codec profiles, depending on program/project application needs.*

Table 7: Codecs approved for use with Class 2 Motion Imagery*

Encoding Method	Codec Standard	Carriage in ISOBMFF
HEVC/H.265	ISO/IEC 23008-2	ISO/IEC 14496-12, ISO/IEC 14496-15
AVC/H.264	ISO/IEC 14496-10	ISO/IEC 14496-12, ISO/IEC 14496-15

* *Note: programs may independently require supporting multiple or all codecs listed and may also constrain support to specific codec profiles, depending on program application needs.*

The following lists additional details for these codec options:

- ISO/IEC 23001-17 provides flexible mechanisms for the carriage of uncompressed Class 0 Motion Imagery within ISO Base Media File Format. When producing subsampled YUV imagery, ISO/IEC 23001-17 supports 4:4:4, 4:2:2, 4:2:0, and 4:1:1 formats. The degrading image transformation of 4:4:4 to 4:2:2 or 4:2:0 results in a lossy uncompressed image as color information is averaged, resulting in unretrievable loss from the original. As the MISP does not allow the use of interlaced video, implementers must not use the 4:1:1 format.

- High Efficiency Video Coding (HEVC/H.265) is the successor to AVC/H.264 (see next bullet) and provides the benefits of enhanced capabilities and improved compression efficiency. HEVC profiles support bit depths of 8-bits and up to 16-bits at the highest profiles and levels. The MISP provides guidance for which profiles and levels of HEVC to use for a specific Motion Imagery Class and in specific applications. ISO/IEC 23008-2 [15] and ITU-T H.265 [16] jointly specify the HEVC codec.
- Advanced Video Coding (AVC/H.264) is an industry media standard defined within the MPEG-4 suite of standards. AVC profiles support bit depths of 8-bits and up to 14-bits at the highest profiles and levels. The MISP provides guidance for which profiles and levels of AVC to use for a specific Motion Imagery Class and in specific applications. ISO/IEC 14496-10 (MPEG-4 Part 10) [17] and ITU-T H.264 [18] jointly specify the AVC codec.

An additional standard of interest when using AVC and HEVC is ISO/IEC 14496-15 (NALuFF), which defines how to carry AVC and HEVC coded video in Network Abstraction Layer (NAL) unit form.

- ISO/IEC 14496-15 [19] specifies the formatting and carriage of HEVC and AVC content within ISOBMFF. The NALuFF includes Sample Entry and box type definitions for handling both HEVC and AVC compressed video and describes how to encapsulate a video stream into NAL units within the ISOBMFF framework to facilitate delivery of the content by a streaming server.

Requirement(s)	
NGA.STND.0076-01_V1.0-23	NGA.STND.0076-01 implementations shall restrict chroma subsampling of Motion Imagery to 4:2:2 and 4:2:0.
NGA.STND.0076-01_V1.0-24	Where an NGA.STND.0076-01 file contains Class 0 Motion Imagery, the Class 0 Motion Imagery content shall conform to the codec standard in Table 5.
NGA.STND.0076-01_V1.0-25	Where an NGA.STND.0076-01 file contains Class 0 Motion Imagery, the carriage of the Class 0 Motion Imagery in a track shall conform to the associated carriage standard listed in Table 5.
NGA.STND.0076-01_V1.0-26	Where an NGA.STND.0076-01 file contains Class 1 Motion Imagery, the Class 1 Motion Imagery content shall conform to one of the codecs listed in Table 6.
NGA.STND.0076-01_V1.0-27	Where an NGA.STND.0076-01 file contains Class 1 Motion Imagery, the carriage of the Class 1 Motion Imagery in a track shall conform to the associated carriage standards listed in Table 6.
NGA.STND.0076-01_V1.0-28	Where an NGA.STND.0076-01 file contains Class 2 Motion Imagery, the Class 2 Motion Imagery content shall conform to one of the codecs listed in Table 7.
NGA.STND.0076-01_V1.0-29	Where an NGA.STND.0076-01 file contains Class 2 Motion Imagery, the carriage of the Class 2 Motion Imagery in a track shall conform to the associated carriage standards listed in Table 7.

6.3 Metadata Content

ISOBMFF and HEIF provide several foundational metadata structures which GIMI builds-on to support GIMI's core functions. Figure 3 is an illustration introducing the GIMI Metadata architectural components and their dependencies on each other. From the bottom-up, ISOBMFF provides the foundational metadata structures which includes methods for storing and associating metadata to ISOBMFF/HEIF media. GIMI defines three primary categories of metadata, **GIMI-Essential**, **Security**, and **Application Metadata**, all of which use ISOBMFF/HEIF methods to store and associate metadata with ISOBMFF/HEIF media. GIMI-Essential Metadata (green) stores and associates **TAI Timestamps** and **Content IDs** to media. Security Metadata (yellow) stores and associate's security markings to media by linking Content IDs to embedded security markings. Application metadata (blue) stores and associates GEOINT metadata with the media using two different methodologies, **Single Instance** and **Sequential**, both of which use TAI Timestamps and Content IDs. Within Single Instance forms of metadata, the coding of one or more metadata elements into a single block of metadata is a **Single-Article** metadata item. The coding of one or more layers of 2D arrays (e.g., pixel metrics, pixel geodetics) as a single or multi-component metadata item, respectively, using an image codec is a Supplemental Image Metadata Array (**Single-SIMA**). Within Sequential forms of metadata, the coding of dynamic metadata as a KLV set or pack on a timeline is **Sequential-Packet** metadata and the coding of temporally varying SIMA data on a timeline is **Sequential-SIMA**.

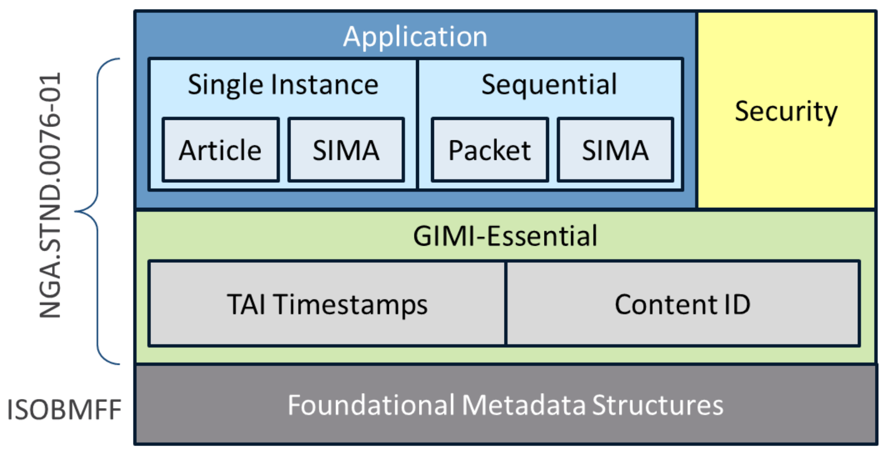


Figure 3: GIMI Metadata Architecture uses ISOBMFF's Metadata Infrastructure

6.3.1 ISOBMFF Foundational Metadata

The ISOBMFF standard defines a metadata architecture which supports internal housekeeping of file contents, as well as storing and cross-linking application metadata to media. ISOBMFF and its derivative standards define metadata elements directly associated with the media formatting, media location (e.g., where it goes in a file), and intended display of imagery content. This architecture aligns with the needs of industry and consumer media applications; however, NGA.STND.0076-01 uses and extends the ISOBMFF metadata architecture to support additional NGA.STND.0076-01 metadata for GEOINT Enterprise applications. ISOBMFF and HEIF both define tools for carrying media support information in a variety of ways. Some are generic, supporting custom applications, and others have predefined content parameters to address relatively common needs. The latter includes item properties for still images, and various boxes which carry media support information, such as color boxes.

6.3.2 GIMI-Essential Metadata

GIMI-essential metadata establishes an identity foundation for every element of media content and provides two capabilities: universal identification of media content and timestamp labeling of media content. Universal identification requires assigning a unique Content Identifier (Content ID) to unique primary-media-content (see Section 6.7). Timestamp labeling requires assigning an absolute timestamp, i.e., an International Atomic Time timestamp (TAI timestamp), (see Section 6.6) to specific elements of media content. A conformant GIMI file storing media content will have GIMI-essential metadata including a Content ID and a timestamp assigned to the media content. In the case of metadata, certain situations require writers to assign timestamps using ISOBMFF mechanisms, and in others they embed timestamps in an encoded collection of metadata elements, such as per MISP KLV standards. NGA.STND.0076-01 requires the use of GIMI-essential metadata in all GIMI files.

6.3.3 Security Metadata

Security metadata provides security labeling of data within a GIMI file. Security labeling of content media data uses the Content ID to assign valid security markings for each element of primary-media-content in a GIMI file (see Section 6.8). In NSG applications, writers implement security markings using the GIMI Security XML defined in Section 6.8.2.

6.3.4 Application Metadata

Application metadata is the GEOINT data necessary to understand and work with GIMI media content for a particular use case. Application metadata is any metadata relating to the media's lifecycle processes (i.e., tasking, collection, processing, exploitation, distribution, and archive). Application metadata has a temporal assignment, such as a device's measurement time, allowing correlation with other sequential media. There are two methods of organizing application metadata, either temporally on a timeline or as a collection of elements encoded together and treated as one block of data in a file. When organizing temporally, writers store metadata in a track. When organizing metadata as a collection of elements, writers encode the metadata elements into a single block of data and store as a metadata item. When storing as a collection of metadata elements in an item, writers include timing information for the individual metadata elements inside the encoded item.

This standard defines two different storage methods for application metadata depending on the organization method: Sequence and Single Instance. For temporally organized metadata, writers use the Sequential Application Metadata method which uses ISOBMFF track data. When storing a collection of metadata elements in a single encoded item, writers use the Single Instance Application Metadata method which uses ISOBMFF's MetaBoxes.

In traditional GEOINT workflows, metadata critical to the imagery content resides within the same file as the content; however, NGA.STND.0076-01 allows for storing application metadata: within a GIMI file (Section 6.4 and Section 6.5), in an external sidecar file, or a combination of both. Using a sidecar file is possible because the enterprise-wide unique content identifier (Content ID) enables the sidecar's metadata to reference the media content. External sidecar storage enables simplifying the imagery files by only including necessary and required GIMI-essential metadata and supports optimized access, modification, and transmission of application metadata content.

6.3.5 Encoding Methods

ISOBMFF provides the ability to utilize application specific metadata encoding methods. This standard constrains the choice of methods and requires the following forms of encoding:

- GIMI-essential metadata
  - Content Identification (see Section 6.7)
  - Timestamps use a TAI Timing structure (see Section 6.6)
- Security metadata
  - Security markings (within the NSG) with XML encoding (see Section 6.8)
- Application metadata
  - Single Instance
    - Single-Article Metadata with KLV encoding (see Section 6.4.1.1)
    - Single-Article Metadata with Turtle encoding (see Section 6.4.1.2)
    - Supplemental Image Metadata Arrays (see Section 6.4.1.3)
  - Sequential
    - Packet Metadata with KLV encoding (see Section 6.5.1.1)
    - Supplemental Image Metadata Arrays (see Section 6.5.2)

6.4 Single Instance Application Metadata

There are two types of Single Instance Application Metadata, Single-Article and Single-SIMA. Writers declare Single-Article metadata in a MetaBox at either the file, movie, movie fragment, track, or track fragment levels, and associate a Content Identifier with the Single-Article. Timestamps and other timing information for Single-Article metadata is carried inside the encoded article of metadata. Encoding methods for Single-Article metadata include KLV and Turtle.

A Single-SIMA is one or more homogeneous collections of metadata in the form of a single or multi-component two-dimensional matrix. Examples of Single-SIMA are an array of pixel metrics, or the geodetics (latitude, longitude, HAE) of every pixel in an image. The high-level diagram in Figure 4 shows a declaration of Single Instance Application Metadata items at the file-scoped level and inside a track. In the figure, the MediaDataBox ('mdat') stores the Single Instance Application Metadata content.

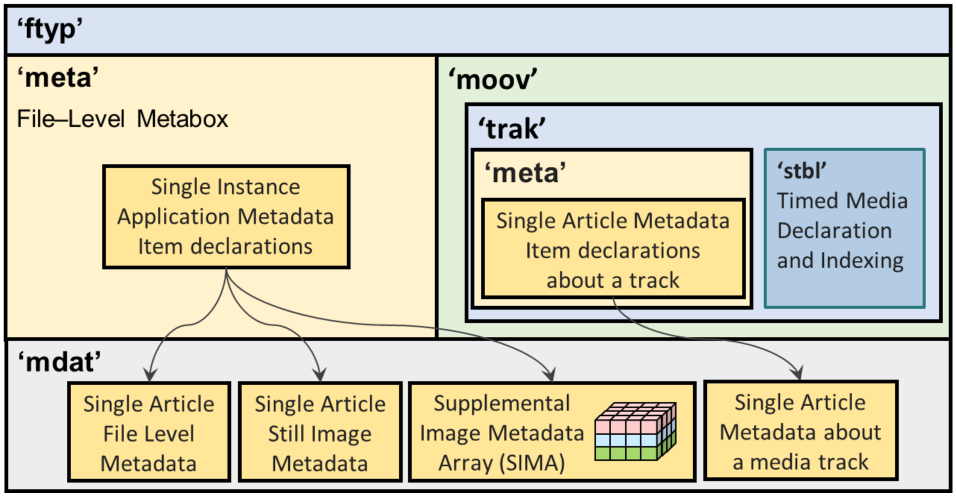


Figure 4: Declaration and storage of item metadata

6.4.1 Single Instance Application Metadata Encoding

ISOBMFF and HEIF both support specific types of metadata encoding (e.g., EXIF, XMP, MPEG-7, etc.), as well as generalized methods for identifying other user-implemented encoding types (e.g., MIME and URI defined types, such as XML and KLV, respectively). While these types and encodings provide great flexibility, allowing too many encoding methods hampers interoperability.

This standard limits the encoding methods for Single Instance Application Metadata, i.e., Single-Article and Single-SIMA. For Single-Article metadata this standard allows Key-Length-Value and Turtle encoding methods. For Single-SIMA, this standard supports storing arrays as an image using allowed codecs. Table 8 summarizes the options for encoding Single Instance Application Metadata.

Table 8: Encoding methods approved for use with Single Instance Application Metadata

Encoding Method	Encoding Standard
Coding methods for Single-Article Application Metadata	
Data Encoding Protocol Using Key-Length-Value	SMPTE ST 336:2017
Turtle serialization format for RDF graphs	W3C RDF 1.1 Turtle
Coding methods for Single-SIMA Application Metadata	
Uncompressed Video and Images in ISOBMFF	ISO/IEC 23001-17
JPEG 2000 (Part 1 & Part 2)	ISO/IEC 15444-1, ISO/IEC 15444-2
High Throughput JPEG 2000	ISO/IEC 15444-15
HEVC	ISO/IEC 23008-2
AVC	ISO/IEC 14496-10

As per ISOBMFF rules, the file-scoped MetaBox contains all declarations for Single Instance Application Metadata applicable to all media within a file. The MetaBox in the MovieBox contains declarations for Single Instance Application Metadata scoped to a sequential movie presentation (collection of tracks). Finally, the MetaBox in a TrackBox contains declarations for Single Instance Application Metadata scoped to a single sequential media track. In cases where there is a conflict between metadata elements at different levels of scope, the metadata at the narrowest scope takes precedent for the media within that narrow scope.

Requirement(s)	
NGA.STND.0076-01_V1.0-30	Where writers are storing Single Instance Application Metadata, the writer shall only use an encoding method from Table 8.

6.4.1.1 Single-Article Key-Length-Value Item Encoding

To support flexible and binary efficient storage of Single-Article metadata items, this standard allows the Key-Length-Value encoding method for Single-Article metadata items stored internal to an NGA.STND.0076-01 conformant file. MISB ST 0107 [20] provides baseline requirements for implementing KLV metadata in NSG applications. Additional MISB standards define packs and sets of specific metadata elements addressing various application use cases.

Writers use the following steps to declare a Single-Article metadata item with KLV encoding:

1. Writers declare individual blocks of metadata in the ItemInfoBox ('iinf') with the associated ItemInfoEntry ('infe') configured with its item type set to 'uri '.
2. To declare KLV encoding using a specific local set (or pack), the item type is set to 'uri ', which then requires an additional item URI type string (utf8string).
3. A writer codes the string as a SMPTE URN, as defined in RFC 5119 [21], for a specific 16-byte GMSB-defined KLV key as 'urn:smpte:ul:[QUADBYTE encoded KLV key]', where the 16-byte key is declared using QUADBYTE notation with each successive 4-bytes separated by a '.' (period). Note: the brackets are not included in the string; they just illustrate where to place the QUADBYTE encoded 16-byte value.
4. A writer declares and associates an ItemContentID (see 6.7.1) with the Single-Article KLV encoded item.

Figure 5 shows an example using the MISB ST 1902.2 Motion Imagery Metadata (MIMD) Model [22] defined length pack key. The 'item_uri_type' string in the figure is a declaration element from the ItemInfoEntryBox when the item_type is 'uri '.

ST1902.2 MIMD Defined Length Pack Key: 06.0E.2B.34.02.05.01.01.0E.01.05.04.00.00.00.00

item_uri_type: 'urn:smpte:ul:060E2B34.02050101.0E010504.00000000'

Figure 5: Example of an 'item_uri_type' string generation using the MIMD key

For a KLV encoded Single-Article Application Metadata item, the 'item_uri_type' field entry stores the 'key'. The KLV 'length' is the totality of 'extent_lengths' for the item in the ItemLocationBox. Writers store the KLV ‘value’ in a data box of choice (typically in the MediaDataBox) and set the media content storage location. The value portion contains the encoded metadata information from an approved KLV set (or pack) standard. This method for geospatial metadata declaration applies to any MetaBox in a file. When using multiple extents (see NGA.SIG.0045 ISOBMFF Handbook) to store a block of metadata, care must be taken to reassemble the extents before parsing the KLV packets.

Figure 6 shows the instantiation of a Single-Article Metadata item containing a KLV encoded MIMD Defined Length Pack metadata item and the HEVC coded still image item the metadata associates with. In this case, the ItemLocationBox assigns the storage location for both items to the MediaDataBox. A 'cdsc' item reference links the KLV metadata item to the still image item. The ItemPropertiesBox includes IC-ID formatted ItemContentID item properties, with corresponding assignments to the Single-Article KLV item and the HEVC still image item. The diagram highlights (light blue) the location of the MIMD defined length pack key in the ItemInfoEntry declaration, and the MIMD Defined Length Pack KLV-length term (light yellow) in the ItemLocationBox; the KLV-length is encapsulated in the extent_length(s). Additional information, such as information security markings, is not shown in the diagram but is present in a fully populated example. As Single-Article KLV items carry timestamps internally, writers do not associate a TAI Timestamp item property with the KLV item.

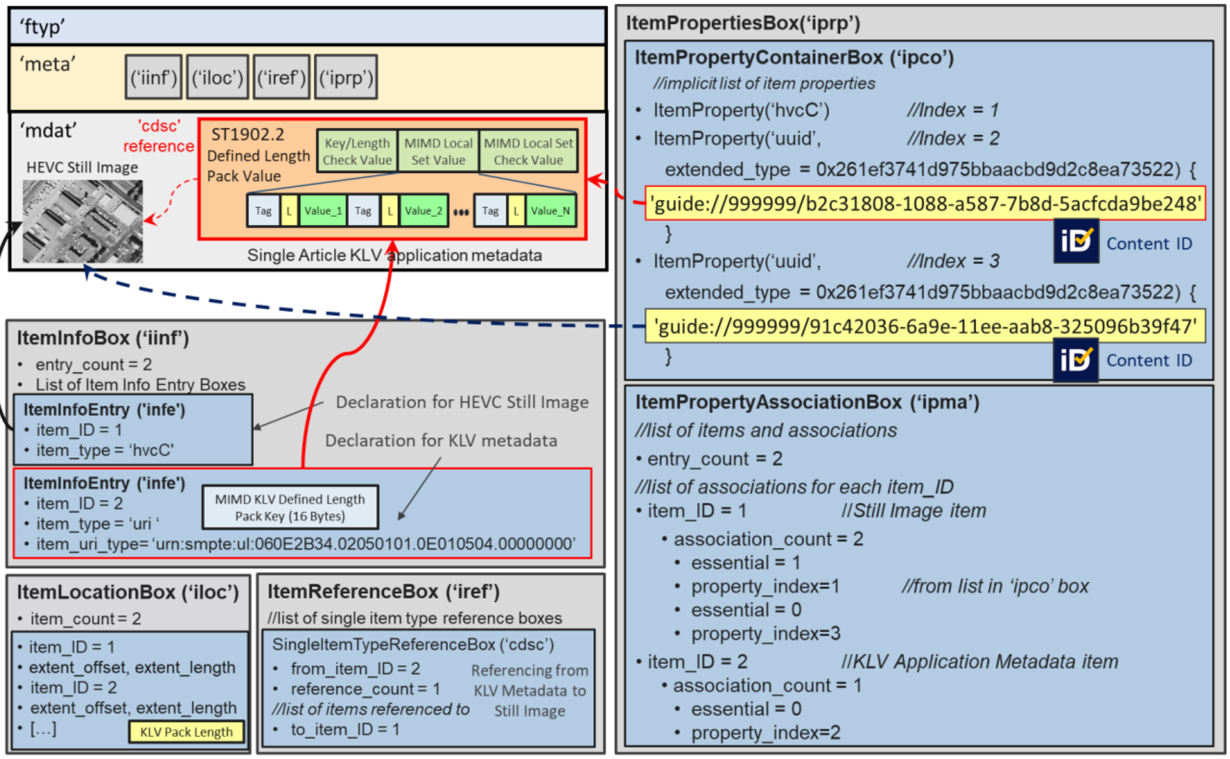


Figure 6: Declaration of Single-Article metadata item with KLV encoding

Requirement(s)	
NGA.STND.0076-01_V1.0-31	Metadata encoded using Key-Length-Value shall conform to MISB ST 0107.
NGA.STND.0076-01_V1.0-32	Encoding of Single-Article metadata items using Key-Length-Value (KLV) shall use GWG approved KLV sets and packs.

6.4.1.2 Single-Article Ontology Structured Metadata Item Encoding

To support ontology structured metadata applications, NGA.STND.0076-01 supports formatting Single-Article metadata as Turtle encoded RDF. A GIMI file either carries the Turtle ontology-structured content inside the imagery file or alongside the imagery file in an Turtle sidecar file.

This standard operates seamlessly with ontology structured information contained in GIMI-embedded Turtle, external Turtle sidecar files, and graph databases. Content IDs in a GIMI file (see Section 6.7) provide a mechanism for linking ontological information to media content. NGA.STND.0076-01 supports declaring metadata items containing RDF content formatted in the Turtle syntax.

Writers use the following steps to declare a Single-Article Ontology Structure Metadata item.

1. Writers declare individual blocks of metadata in the ItemInfoBox ('iinf') with the associated ItemInfoEntry ('infe').
2. The item_type is set to 'mime'
3. The content_type is set to 'text/turtle'
4. A writer optionally implements numerically lossless compression by setting the content_encoding parameter to 'br', for Brotli encoding [23]. An empty string for the

content_encoding parameter indicates no additional content encoding other than the Turtle syntax formatting.

Note: While ISOBMFF supports gzip and deflate compression, this standard only authorizes Brotli encoding for numerically lossless compression of Turtle metadata objects inside a GIMI file.

5. A writer declares and associates an ItemContentID (see 6.7.1) with the Single-Article Turtle encoded item.

A writer indicates whether the Turtle content is stored internal to a GIMI file or in an external resource, such as a Turtle sidecar file, and provides storage location and addressing information using the DataInformationBox and the data-reference-index and other information inside the ItemLocationBox.

Figure 7 shows the instantiation and encoding of ontology structured metadata using Single-Article Turtle encoding with Brotli compression. In this case, the ItemLocationBox assigns the storage location to the MediaDataBox. An ItemContentID item property is declared and associated with the Turtle metadata information. As Single-Article Turtle items carry timestamps internally, writers do not associate a TAI Timestamp item property with the Turtle item.

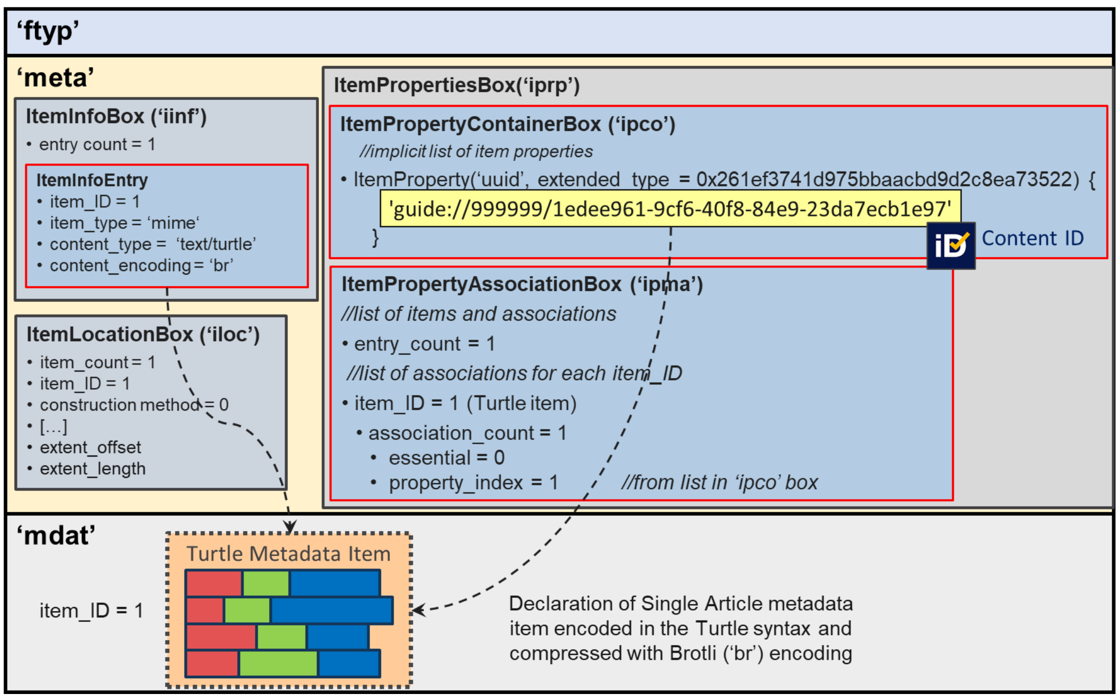


Figure 7: Declaration of Single-Article metadata item as a Turtle encoded item

Requirement(s)	
NGA.STND.0076-01_V1.0-33	Writers shall encode ontology structured Single-Article metadata items declared in an NGA.STND.0076-01 file in the Turtle syntax.
NGA.STND.0076-01_V1.0-34	Where writers compress a Turtle metadata item in an NGA.STND.0076-01 file, writers shall use Brotli encoding.

Requirement(s)	
NGA.STND.0076-01_V1.0-35	When a writer declares a Brotli compressed Turtle item in an ItemInfoEntry, writers shall set the content_encoding parameter to 'br'.

6.4.1.3 Supplemental Image Metadata Array (SIMA) Item Encoding

Writers declare Single-SIMA in the file-level MetaBox and associate a Content Identifier with the Single-SIMA. When the Single-SIMA content is dynamic metadata, writers associate a TAI Timestamp item property with the Single-SIMA item (see Section 6.6). When the Single-SIMA content is static metadata, writers do not associate a TAI Timestamp item property with the Single-SIMA item.

Writers use any of the image codecs approved by this standard to encode and carry Single-SIMA content. The use of image codecs enables storing arrays as uncompressed, numerically lossless compressed, and lossy compressed.

When encoding Single-SIMA as uncompressed content, ISO/IEC 23001-17 supports the encoding and carriage of:

- 1 to 256-bit unsigned integer values,
- 16, 32, 64, 128, and 256-bit floating point values,
- 16, 32, 64, and 128-bit real and imaginary terms for complex number values.

Where writers desire to compress Single-SIMA content, whether lossless or lossy, writers utilize an appropriate image codec from the list previously defined in Table 4. The nature of the data and the application dictate which form of coding is appropriate. The carriage of Single-SIMA applies to use cases involving pixel level metadata, such as cloud cover, image quality metrics, geodetics (latitude, longitude, height above ellipsoid), etc.

For still applications, writers declare Single-SIMA content as an item using one of the allowed codecs. When writers require more than one layer of Single-SIMA content, writers declare each layer in separate items, or writers stack the layers using the component feature of the chosen codec. Each codec has its own limits on the number of components a single encoded item can carry. The uncompressed codec and JPEG 2000 support very large numbers of components to address most if not all practical applications. The carriage of multiple layers in a single item is useful when multiple arrays share a relationship, the same timing information, and the same dimensions. When two components of Single-SIMA content do not share the same measurement timing information or the same dimensions, writers declare each component as a separate item.

Writers never declare and carry Single-SIMA content and imagery data in the same item. Writers can use grouping constructs and item references to show an association between a Single-SIMA item and imagery item, and between multiple Single-SIMA items when necessary. When a Single-SIMA item applies to an image item, such as pixel metrics, a cloud cover grid, or pixel level geodetics, writers use a 'cdsc' item reference to show the relationship.

Figure 8 illustrates geodetic data, consisting of latitude, longitude, and height above ellipsoid information which a writer may carry as a single item with three components. When declaring multiple components in a single uncompressed item, a writer assigns each component with its own numerical format (integer, floating point, etc.). All components may have the same numerical format, or they may be different, depending on use case needs. Implementers should consider the tradeoff of a more complex parsing of a single multi-component item with different types and lengths versus having multiple, stand-alone homogenous items. For each uncompressed component, a writer assigns an MPEG component type using the ComponentDefinitionBox ('cmpd'). The intent of the MPEG component type is to guide default display behavior and rendering of the component. Writers should not use the MPEG component type for other purposes or to convey other meaning. Writers populating a ComponentDefinitionBox assign MPEG component types using only the enumerated values with specified entries in ISO/IEC 23001-17. NGA.STND.0076-01 forbids writers from assigning the optional user-defined MPEG component types.

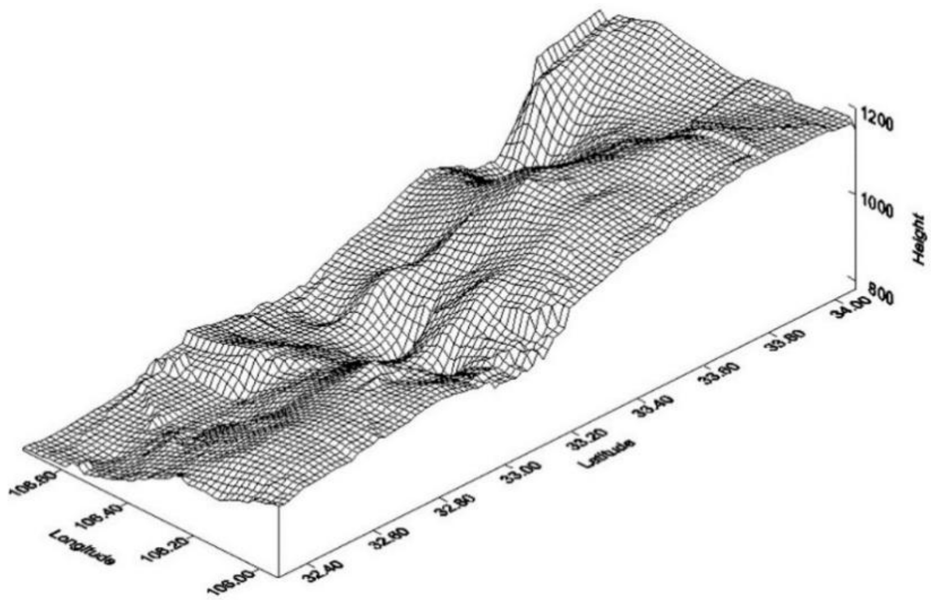


Figure 8: Writers store Single-SIMA content, such as pixel geodetics, using an image codec

Interleaving modes of the uncompressed codec allow for adjacent storage of multiple component values at a single array address. The adjacent storage supports efficient memory access when reading related component values at the same time. With the use of the uncompressed codec for encoding Single-SIMA items, flexible options are available for arranging data, including interleaving, tiling, padding, blocking, alignment, etc. Flexible options allow Single-SIMA data to potentially mirror the data access patterns of associated images.

When declaring uncompressed items carrying one or more types of Single-SIMA content, the default setting is to assign each component as a monochrome type in the uncompressed codec's ComponentDefinitionBox ('cmpd'). To label each individual component stored in a Single-SIMA item with type information, the ItemComponentContentID (see Section 6.7) for each component is matched to an ontologically defined 'type' value in a Turtle item. The ontologically structured Turtle content may include specific information about each component, such as labels for latitude, longitude, and height above ellipsoid in the geodetics example. Writers use the instructions in Section 6.4.1.2 to declare the ontology structured metadata and link the Turtle metadata item to the Single-SIMA item using a 'cdsc' item reference.

When declaring a Single-SIMA item, writers also declare a HEIF-defined Accessibility text ('altt') descriptive item property and associate the Accessibility text with the Single-SIMA item. For Single-SIMA items, writers always set the `alt_text` field of the 'altt' item property to a utf8string value of *'Supplemental Image Metadata'* and the `alt_lang` field to a utf8string value of 'en_US'.

When writers do not intend readers to display the Single-SIMA content, writers set the least significant bit of the ItemInfoEntry's flags field to 1 (e.g., flags & 0x000001 = 1). This bit value set to 1 indicates the item has the hidden attribute and reader applications should not display the item.

When there is a need to compress Single-SIMA content, in either a lossless or lossy manner, writers utilize an image codec supporting compression (see Section 6.2.1). The ability to display the Single-SIMA content depends on the codec chosen and on the capabilities of the viewing software. In certain cases, because of the bit depth, component format (floating point and complex number), or other aspects of the data, the form of the content may require format conversion, manipulation, scaling, etc. to facilitate display. Writers may construct a separate displayable image of the Single-SIMA content, encode the image (using one of the compressed or uncompressed codec options), and declare the image as a separate item. Writers then link the image, with an item reference type of ‘base’, as being from the displayable item and to the original Single-SIMA item. The type ‘base’ indicates the image is a derived form of the original Single-SIMA item. When generating the displayable version of the item, applications may optionally apply transformative item properties. Figure 9 shows the declaration for an uncompressed image item holding latitude, longitude, and height above ellipsoid arrays as Single-SIMA content. The ItemComponentContentIDProperty lists the ComponentContentID’s for the Single-SIMA item. The notional Turtle item indicates the ontological type for each component of the Single-SIMA content. This diagram does not show the TAI Timestamp and other potential parameters.

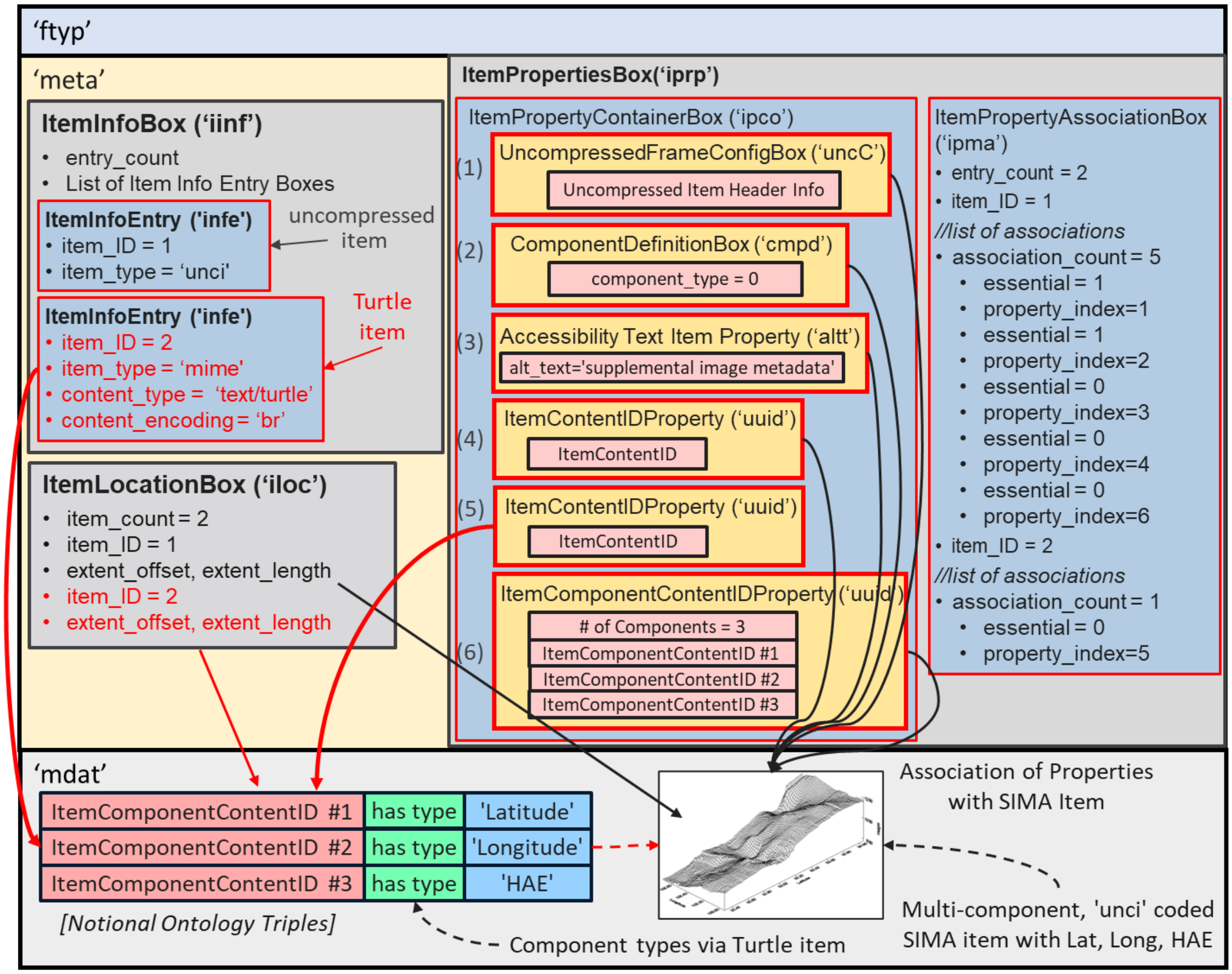


Figure 9: Declaration for a multi-component, uncompressed Single-SIMA item

Requirement(s)	
NGA.STND.0076-01_V1.0-36	Writers shall declare Single-SIMA items in the file level MetaBox.
NGA.STND.0076-01_V1.0-37	Writers shall not store Single-SIMA content in an item containing imagery content.
NGA.STND.0076-01_V1.0-38	When storing a Single-SIMA item as uncompressed content, writers shall declare the content as an ISO/IEC 23001-17 item.
NGA.STND.0076-01_V1.0-39	When storing a Single-SIMA item in a compressed form, writers shall use one of the codecs listed in Table 4.
NGA.STND.0076-01_V1.0-40	When a ComponentDefinitionBox is present in a GIMI file, it shall not contain user-defined component types.
NGA.STND.0076-01_V1.0-41	When declare a Single-SIMA item, it shall be associated with an Accessibility text ('altt') descriptive item property with the `alt_text` entry set to a utf8string value of 'Supplemental Image Metadata' and the `alt_lang` entry set to a utf8string value of 'en_US'.

Requirement(s)	
NGA.STND.0076-01_V1.0-42	Where a Single Instance Application Metadata item provides information about another item, writers shall generate a content describes ('cdsc') item reference from the Single Instance Application Metadata item to the item it describes.

6.4.2 Scoping and Prioritization

Writers declare Single Instance Application Metadata in a MetaBox, which exist at the file level, the presentation level (in the MovieBox and MovieFragmentBoxes), and the track level (in a TrackBox and TrackFragmentBoxes). Due to HEIF rules, writers may only declare Single-SIMA items in the file-level MetaBox. To limit the scope of association of a Single-SIMA item to another item, such as a specific still image, a writer includes a content describes ('cdsc') item reference.

Writers may declare Single-Article metadata items in any of a file's MetaBoxes. In cases where writers to do not include an item reference to another item, the scoping of the Single-Article item is to all the stored media at the level of the declaration. When a Single-Article metadata item resides at the file level without any item references, the metadata applies to all media in the file. When the writer associates a Single-Article metadata item to one or more other declared media items using a 'cdsc' item reference, the association restricts the scoping of the Single-Article metadata item to the linked media items.

From a containment perspective, a File may contain a MovieBox, and a MovieBox contains one or more TrackBoxes. For media inside a TrackBox, Single Instance Application Metadata at all three levels may apply to the track media. In cases of conflict between Single Instance Application Metadata at the different levels, the metadata at the lower level takes precedence over the metadata from a higher level.

In the case of files with fragmentation, metadata prioritization may also occur via placement on the timeline. Figure 10 provides a high-level diagram of a progressive MP4 file, where the presentation metadata is in the MetaBox of the MovieBox. In this case, the file carries only one instance of a presentation MetaBox. Any Single-Article metadata items in the presentation MetaBox, without references to other items, apply to the entire presentation. Similarly, Single-Article metadata items in the MetaBox of the TrackBox, without references to other items, apply to the entire track.

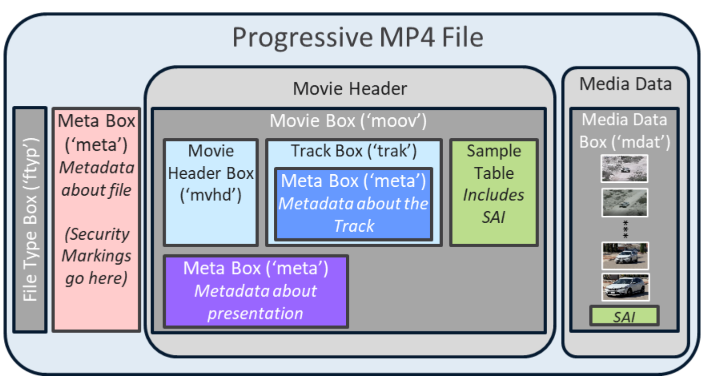


Figure 10: A progressive MP4 file.

Figure 11 provides a high-level diagram of a fragmented MP4 file, where the writer partitions the movie (or presentation) into an initial MovieBox and one or more movie fragment ('moof') boxes. Each fragment box contains a MetaBox. The metadata in the 'moov' or any of the 'moof' MetaBoxes applies to the entire presentation, including all the presentation's tracks. In addition, for any specific fragment, all metadata needed for the fragment must reside in the MetaBox of the MovieBox or the MetaBox of the specific fragment.

In fragmented files, when reading the series of fragments, readers update the Single-Article Metadata (with the same item_ID) with the copy from the latest fragment on the track's timeline. This same behavior applies to Single-Article Metadata items carried in the MetaBoxes of a track and associated track fragments. When the Single-Article Metadata is dynamic metadata, it belongs in a sequential metadata track.

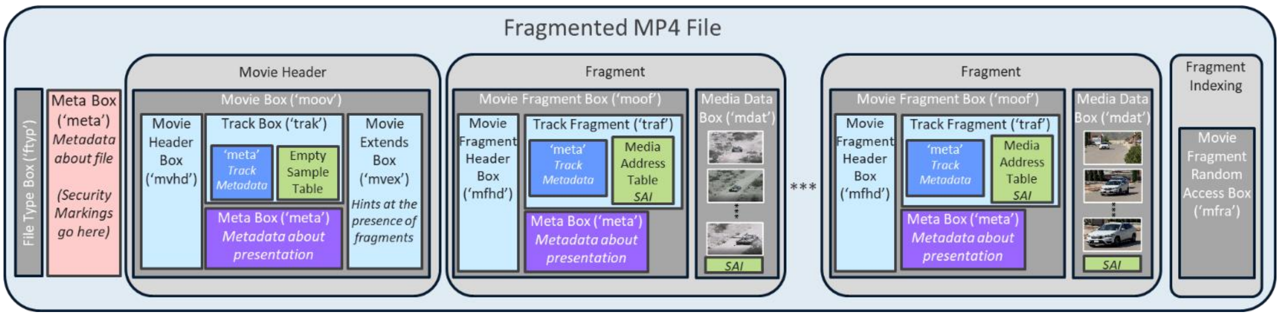


Figure 11: A fragmented MP4 file.

6.5 Sequential Application Metadata

There are two types of Sequential Application Metadata, Sequential-Packet and Sequential-SIMA. Writers encode Sequential-Packet Application Metadata as tracks using KLV. Likewise, writers encode Sequential-SIMA as tracks using image codecs approved by this standard. Writers declare both forms as a track, generate the necessary Content Identifiers for the track (see Section 6.7) and generate absolute TAI timestamps as sample auxiliary information for each sample in the track (See Section 6.6).

When carrying timestamps inside the Sequential-Packet samples, writers ensure the primary internal packet timestamp information aligns with each sample's SAI TAI timestamp information. Differences in formatting, timestamp resolution, clock metadata, and time standard epoch may occur depending on the underlying standard defining the Sequential-Packet content makeup and coding. The implementation of TAI timestamps as SAI guarantees a common baseline for the absolute timing for all tracks.

Sequential-SIMA does not have a mechanism for carrying timestamp information inside its samples. Writers generate and store TAI timestamps for Sequential-SIMA track samples as sample auxiliary information in a manner identical to other types of tracks.

6.5.1 Sequential-Packet Application Metadata Tracks

Systems collecting temporal metadata may store the application metadata as a media track. This metadata includes time varying sensor and platform metadata, such as gimbal angles, GPS location, zooming focal lengths, etc. Sequential-Packet Application Metadata tracks have the following features:

- NGA.STND.0076-01 uses MISB-defined KLV sets and packs to store Sequential-Packet Application Metadata in ISOBMFF tracks.
- Sequential-Packet Application Metadata tracks include a TrackContentID.
- Sequential-Packet Application Metadata tracks include a TAI timestamp for each sample.
- Sequential-Packet Application Metadata tracks include a SampleContentID for each sample.
- When a Sequential-Packet Application Metadata track describes another media track, writers include a content describes ('cdsc') track reference.
- Zero or more Sequential-Packet Application Metadata tracks can exist in a file.
- Tracks containing Sequential-Packet Application Metadata may be synchronous or asynchronous with respect to referenced Motion Imagery tracks.
- Sequential-Packet Application Metadata tracks may occur at rates independent of other tracks.
- Writers may include Sequential-Packet Application Metadata tracks in fragmented presentations.

Implementing a Sequential-Packet Application Metadata track follows the ISOBMFF process for implementing a video track and shares the same timeline structure. ISOBMFF rules defined for chunking and sample runs apply to Sequential-Packet Application Metadata tracks.

6.5.1.1 Sequential-Packet Key-Length-Value Metadata Encoding

A writer instantiates a Sequential-Packet KLV metadata track by including a TrackBox (inside the MovieBox) dedicated to the track. The writer sets the handler type to 'meta' inside the HandlerBox and includes a NullMediaInfoHeaderBox ('nmhd') inside the MediaInfoBox ('minf'). Including the NullMediaInfoHeaderBox indicates the track is not a Motion Imagery or audio track, but instead a track for metadata.

Writers indicate the sample entry type (i.e., KLV) by including a URI string in a URIBox('uri '), which is within the URIMetaSampleEntryBox ('urim'), which itself is inside the SampleDescriptionBox ('stsd'). Writers format the string as a uniform resource name, with a QUADBYTE formatted 16-byte MISB KLV key. Figure 12 shows the URI format.

```
urn:smpte:ul:[QUADBYTE formatted16-byte key]
```

Figure 12: Format of 'theURI' string to signal a Sequential-Packet KLV metadata track

Writers exclude the brackets from the actual URN and format the 16-byte key following the QUADBYTE notation in RFC 5119. This URN formatting is identical to the URN declaration described for a Single-Article KLV metadata item described in Section 6.4.1.1. In the context of a track, the URN identifies the encoding of a KLV set (or pack) used in a Sequential-Packet metadata track. GWG Focus Groups document and approve KLV sets and packs for use with Sequential-Packet KLV metadata tracks.

The declaration of the 16-byte set (or pack) key identifies the allowed metadata content within the track. For independent metadata tracks (i.e., not referenced to another track) the implementation is the same, but without the 'cdsc' track reference to another track. This enables writers the ability to build GIMI files with only Sequential-Packet Application Metadata tracks.

Within this standard, the formatting of a KLV track follows the implementation of the box structure shown in Figure 13 with items of note highlighted with dashed ovals. When storing a Sequential-Packet KLV metadata track, writers store the key, lengths, and values into three different locations. Writers store the key in 'theURI' declaration string of the URIBox. Writers store the length of each KLV packet in the SampleSizeBox or CompactSampleSizeBox (not shown). Writers finally store the KLV values in a chosen location, typically a MediaDataBox.

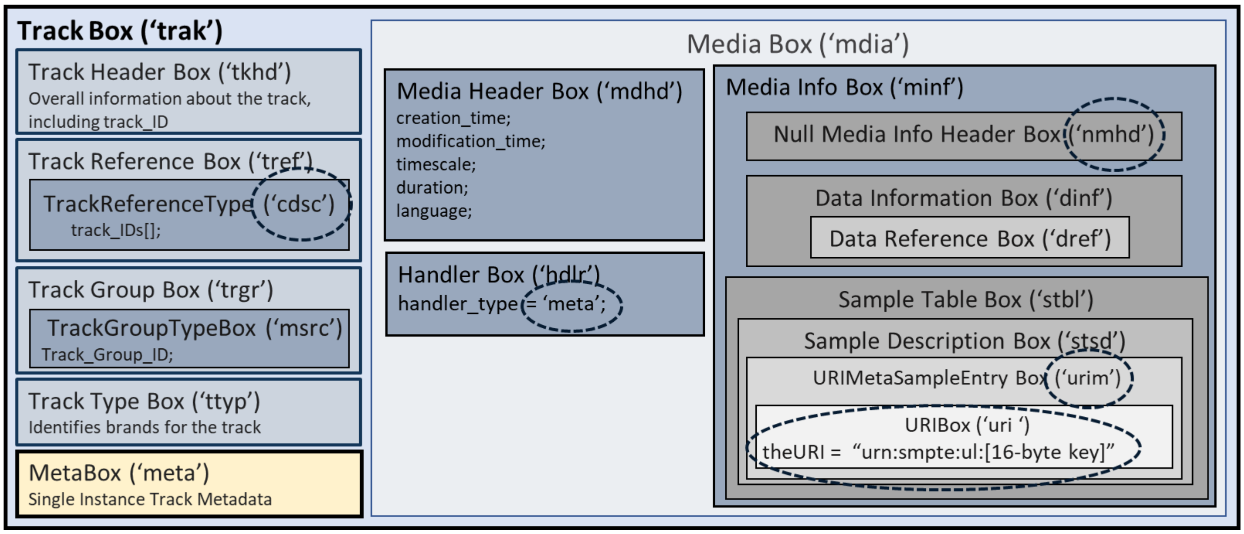


Figure 13: A TrackBox configuration for a Sequential-Packet KLV metadata track

Figure 14 shows a Motion Imagery track with file-scoped security metadata, a TrackContentID applied to the track, TAI timestamps and SampleContentIDs applied to each image in the track, and a separate, parallel track of KLV-encoded Sequential-Packet metadata adding additional context for the Motion Imagery. In this case, the figure shows storing individual sample value portions of the KLV metadata in the Media Data box, along with their associated TAI timestamps and SampleContentIDs carried as sample auxiliary information.

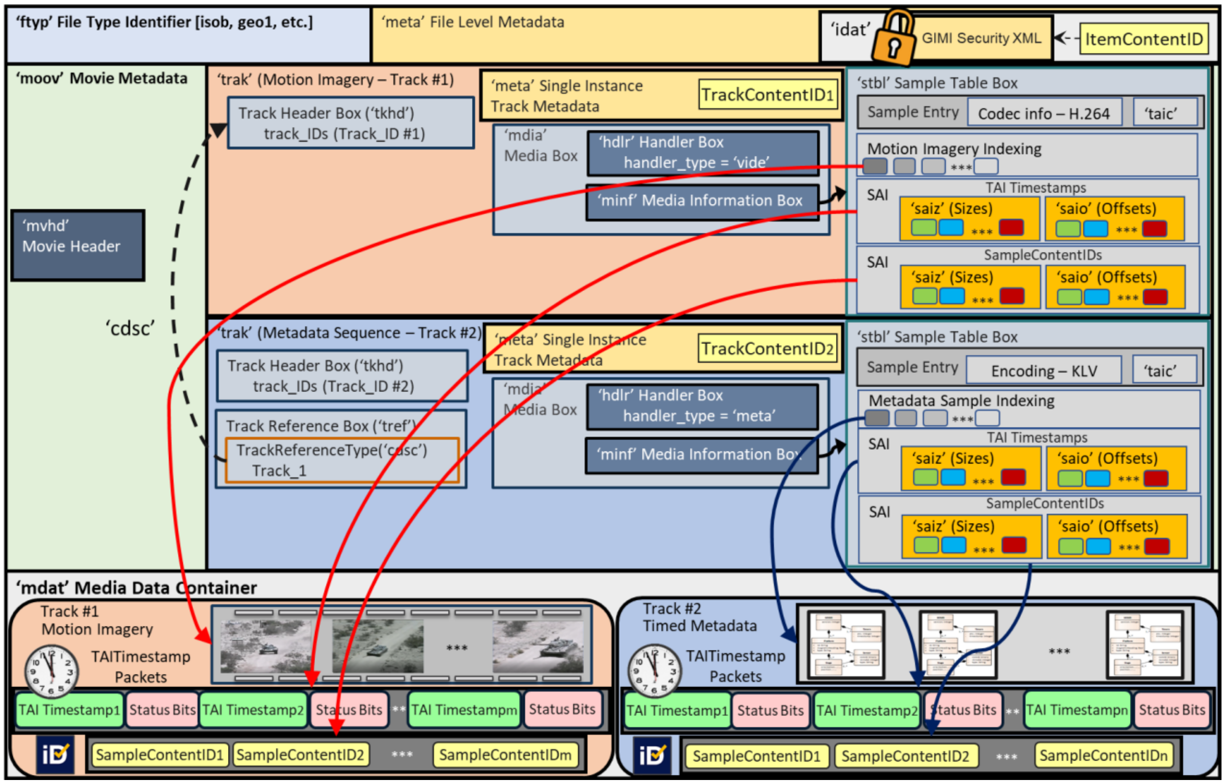


Figure 14: A Motion Imagery file with Sequential-Packet KLV metadata

When writing Sequential-Packet metadata from multiple KLV sets or packs, the metadata from each unique set (or pack) is carried in its own Sequential-Packet metadata track. Writers configure each metadata set in the normal manner for a KLV encoded Sequential-Packet metadata track. Each individual track carries a unique track ID, a unique TrackContentID, a URN defined by the 16-byte key of its set (or pack), and two sets of sample auxiliary information with TAI timestamps and SampleContentIDs.

In situations where data samples from multiple sensors are asynchronously captured at different rates, it is good practice for writers to place the data from each individual sensor into their own respective tracks. This supports consistent sample timing within each track.

When a KLV Sequential-Packet metadata track references one or more other tracks (primarily Motion Imagery tracks), a track reference indicates the relationship. A track reference type of 'cdsc' (content describes) denotes the general relationship of how metadata provides additional information and context for another media track. In certain cases, one metadata track may apply to multiple Motion Imagery tracks. For example, a system with two cameras on a single gimbal or ball turret system generates two tracks of Motion Imagery, but the metadata generated from the gimbal and the platform is common to both. In this case, the single track of platform metadata references both Motion Imagery tracks.

Requirement(s)	
NGA.STND.0076-01_V1.0-43	Sequential-Packet metadata tracks using Key-Length-Value encoding shall only use GWG approved KLV sets and packs.
NGA.STND.0076-01_V1.0-44	KLV encoded Sequential-Packet metadata tracks with content describing another declared media track shall have a content describes ('cdsc') track reference linking the two items together.

6.5.2 Sequential Supplemental Image Metadata Array Track Encoding

When generating Sequential-SIMA Application Metadata tracks, this standard allows for the use of both image and video codecs. Table 4 lists the allowed image codecs. Table 6 and Table 7 list the allowed video codecs.

The declaration of a track uses a VisualSampleEntry with the codingname set for the chosen codec, such as 'uncv' for the uncompressed codec. Writers encode one or more layers of Sequential-SIMA content to create single or multicomponent samples. The number of components a codec supports determines the number of encodable Sequential-SIMA layers. The arrangement of all components in each sample is configurable using the options available for the chosen codec, which may include tiling, interleaving, padding, blocking, endianness, and alignment. Traditional image and video codecs, such as AVC and HEVC, support one or three components. JPEG 2000 and the uncompressed codec support large numbers of components.

When using the uncompressed codec, components of Sequential-SIMA content have a default MPEG component type setting of monochrome for their component format in the ComponentDefinitionBox ('cmpd'). Applications may choose other options to meet specific needs, such as assigning three components as red, green, and blue to support a color rendering of the three components. The intended use for the MPEG component type is to guide display behavior for the component(s). This standard does not allow the use of user-defined MPEG component types from the ComponentDefinitionBox.

To label various aspects of a Sequential-SIMA track, including the entire track, individual track portions, individual components, and individual samples with descriptive information, writers include the appropriate Content IDs (see Section 6.7) for the track in Turtle RDF triples along with the ontologically defined labels. The global uniqueness of the Content IDs binds the information together, regardless of where the writer stores the Turtle content. To explicitly indicate the link to the Turtle content within the GIMI file, a writer can declare a Turtle metadata item in a MetaBox with an internal or external data reference indicating where the Turtle content is located.

When declaring a Sequential-SIMA track, the `handler_type` for the track is set to 'pict' or 'vide' in the track's HandlerBox, depending on the requirement for the Sample Entry coding method. To indicate the track is a Sequential-SIMA track, and not a regular image content track, the `name` variable in the HandlerBox is set to the utf8string *'Supplemental Image Metadata'*.

When a writer intends for players to display Sequential-SIMA track content, the `track_in_movie` flag in the TrackHeaderBox is set to 1. When a writer intends for players to not display Sequential-SIMA track content, the `track_in_movie` flag in the TrackHeaderBox is set to 0. Players may choose to override this setting.

Requirement(s)	
NGA.STND.0076-01_V1.0-45	Where a writer stores Sequential-SIMA content as uncompressed content, the track shall be conformant to ISO/IEC 23001-17.
NGA.STND.0076-01_V1.0-46	Where a writer stores Sequential-SIMA content as compressed content, the writer shall implement the track using one of the codec standards listed in Table 4.
NGA.STND.0076-01_V1.0-47	Where a writer stores Sequential-SIMA content in the form of Class 1 Motion Imagery, the writer shall implement the track using one of the codec standards listed in Table 6.
NGA.STND.0076-01_V1.0-48	Where a writer stores Sequential-SIMA content in the form of Class 2 Motion Imagery, the writer shall implement the track using one of the codec standards listed in Table 7.
NGA.STND.0076-01_V1.0-49	Where a writer stores a Sequential-SIMA track with content which describes another track, the writer shall link the two tracks together using the content describes ('cdsc') track reference.
NGA.STND.0076-01_V1.0-50	Writers shall place Sequential-SIMA content and imagery content in separate tracks.
NGA.STND.0076-01_V1.0-51	When generating a Sequential-SIMA track, writers shall set the name variable in the track's HandlerBox to the utf8string *'Supplemental Image Metadata'*.

6.6 GIMI-Essential: Timing Information

Editor's Note: TAI Timing for ISOBMFF/ HEIF is in the emerging amendment 1 to ISO/IEC 23001-17. Appendix E includes the final draft text for the amendment which contains normative requirements for TAI timestamp implementation in NGA.STND.0076-01. When ISO publishes the amendment, the authors will remove the Appendix and ISO/IEC 23001-17 will become the normative source for TAI timestamp implementation in NGA.STND.0076-01. The target publication date of the amendment 1 to ISO/IEC 23001-17 is the summer of 2025.

To provide necessary timing information for GEOINT applications, NGA.STND.0076-01 supplies tools to support the labeling of collected media items and track samples with absolute TAI timestamps. Appendix E describes a method for including TAI timestamps and associated timing metadata for media items and media track samples in ISOBMFF and HEIF files. This description includes definitions for required data structures and rules for implementing those structures.

Uncompressed video and image collection use cases requiring very high-fidelity data drive the need for TAI timestamps. Additionally, the TAI timestamp capability finds use with other forms of measured sample data and items, including compressed imagery. In the case of both Single-Article and Sequential-Packet metadata, KLV and Turtle encoded metadata carry timestamps internally. Both writer and reader applications need to account for different forms of timestamps and time base epochs in use between different metadata encoding standards, network streaming protocol standards, legacy image formats, and this standard, which uses nanosecond TAI timestamps.

Within ISOBMFF, multiple forms of timing information exist within a file. The ISOBMFF standard defines internal mechanisms for labeling absolute time, based on UTC, and for creating timelines for sequential media, which supports the playout of the media. NGA.STND.0076-01 utilizes the ISOBMFF timeline functionality but does not rely on ISOBMFF's UTC timing information. The ISOBMFF timeline facilitates presentation playout of one or more tracks. The ISOBMFF timeline is not, however, designed to facilitate high precision, nanosecond sensor measurements, analysis, and exploitation. With reliance on 32-bit integers, the longest presentation possible using nanosecond timing is less than five seconds. By implementing TAI timestamps independent of the ISOBMFF built-in timeline, NGA.STND.0076-01 achieves necessary high precision measurement timing as well as being able to leverage the existing timeline tools for playout with standard consumer commercial video players.

There are multiple time systems in use across the GEOINT Enterprise, including International Atomic Time (TAI), Universal Coordinated Time (UTC), Global Positioning System time (GPS), and Motion Imagery Standards Profile (MISP) time. The MISP Handbook [24] and Appendix E provide details on the relationships between these time systems.

The TAI time system is synchronous with the GPS time system (with a fixed offset of 19 seconds) and avoids problems associated with leap seconds, which occasionally occur in the UTC time system. GIMI GEOINT timestamp measurements and calculations are not based on UTC time to avoid issues with leap seconds. Software applications with requirements for UTC time, such as for human display, convert TAI timestamps to UTC time with the proper leap second corrections and adjustments for time zones, etc. Applications need to be mindful of properly conveying to users the type and source of time when displayed.

To demonstrate the inclusion of TAI timestamps in a Still Image application, Figure 15 shows the declaration of a JPEG 2000 compressed image ('j2k1') in the ItemInfoBox ('iinf').

The ItemLocationBox ('iloc') defines the storage location (via black arrow) of the image item within the MediaDataBox ('mdat'). The ItemPropertiesBox ('iprp') declares the properties for the image item and links the properties to the image using an association. The property associations include the JPEG 2000 header, the TAI clock information, and the TAI timestamp packet. When image items are copied to separate files, writers ensure the timing information for the image item persists by copying the associated item properties with the image item.

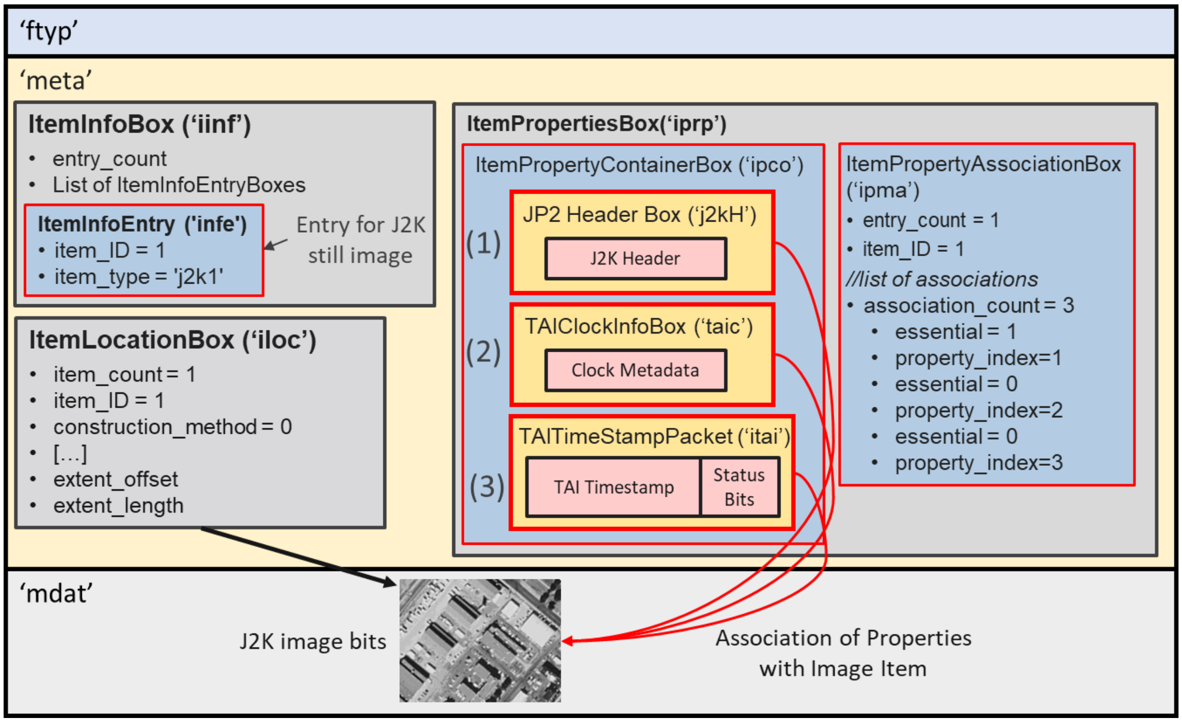


Figure 15: Implementation of a TAI timestamp for a J2K image item

For track samples, writers use sample auxiliary information to attach TAI timestamp information to the track samples. This capability enables labeling sequential media, including video, image sequences, Sequential-Packet and Sequential-SIMA Application Metadata with absolute, nanosecond resolution TAI timestamps. Figure 16 shows the TAI timestamp implementation for a sequential media track, with a TAI clock information box ('taic') located in the Sample Entry, and the TAI timestamp packets carried as sample auxiliary information.

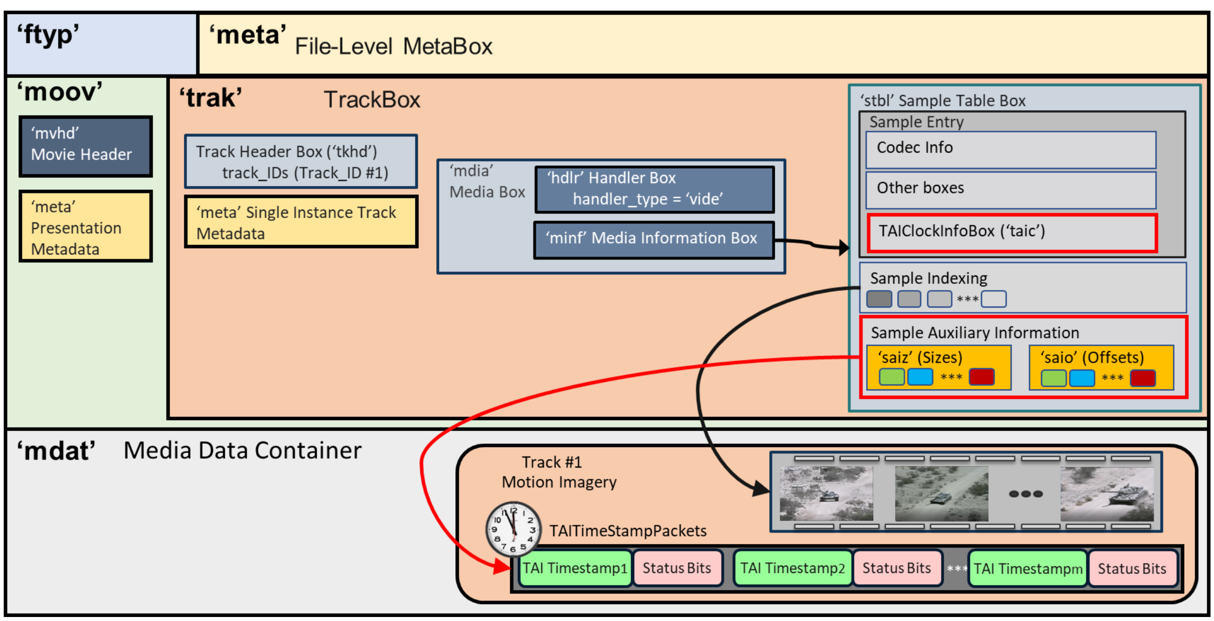


Figure 16: Implementation of TAI timestamps for a video track

Requirement(s)	
NGA.STND.0076-01_V1.0-52	Writers shall associate imagery samples in a track with TAI timing information in a manner conformant to ISO/IEC 23001-17 Amd1.
NGA.STND.0076-01_V1.0-53	Writers shall associate Sequential-Packet samples with TAI timing information in a manner conformant to ISO/IEC 23001-17 Amd1.
NGA.STND.0076-01_V1.0-54	Writers shall associate Sequential-SIMA samples with TAI timing information in a manner conformant to ISO/IEC 23001-17 Amd1.
NGA.STND.0076-01_V1.0-55	Writers shall associate sensor derived image items with TAI timing information in a manner conformant to ISO/IEC 23001-17 Amd1.
NGA.STND.0076-01_V1.0-56	Writers shall associate dynamic Single-SIMA items with TAI timing information in a manner conformant to ISO/IEC 23001-17 Amd1.

6.7 GIMI-Essential: Content Identification Labeling

Many applications require the use of universal and persistent identifiers for managing information across the GEOINT Enterprise. ISOBMFF defines an internal entity ID system to link media together; however, the ID system does not support global uniqueness.

To facilitate the tracking and management of media content throughout a workflow process, over time, and across the GEOINT Enterprise, NGA.STND.0076-01 defines and mandates the use of Content IDs, which take one of two forms: Intelligence Community Identifier or URN Identifier. Applications with a requirement for implementing identifiers in the form of an Intelligence Community identifier, implement Content IDs as an IC-ID [25], a.k.a. GUIDE, with constraints and formatting defined below. Applications without this requirement implement URN Content IDs in the form of a standard URN string formatted UUID (i.e., 'urn:uuid:<uuid>') [26].

Both forms of Content ID are valid IRIs and are interoperable in terms of their use in GIMI files and Turtle sidecar files. The IC identifier is an IRI using an unregistered scheme of "guide", a one to 16 digit assigned numerical prefix, and an application defined suffix. Within NGA.STND.0076-01, an IC-ID suffix is always implemented as a standard canonical form string formatted UUID. The ODNI controls and assigns IC_ID prefixes. The GIMI mandated UUID suffix for an IC-ID is conformant to ISO/IEC 9834-8 and follows the guidance in NGA.RP.0001 [27] for the generation and use of UUIDs in the NSG. Table 9 shows the format of IC-ID and URN string formatted Content IDs.

Table 9: Syntax Options for Content IDs

Scope of Use	Defining Standard	Format of Content ID
Applications requiring the use of IC-IDs	ODNI XML Data Encoding Specification for Intelligence Community Identifier	IC-ID Content ID format: 'guide://<prefix>/<uuid>' Ex: 'guide://999999/fba91974-fdf8-43af-ae52-e3783f4985e8'
Others	ISO/IEC 9834-8	URN Content ID format: 'urn:uuid:<uuid>' Ex: 'urn:uuid:47df9863-6731-4115-bec2-2073a8da3420'

Writers assign a Content ID to every declared item (image, metadata, and region), track, track portion, track sample, item component, and sample component in a file. The exception to this is Content IDs declared as metadata items, i.e., applications do not assign Content IDs to declared Content IDs.

The Content IDs provide a means to identify content, aide in the process of locating content, attach security markings to content, and link to related ontology structured data. Across the GEOINT Enterprise, Content IDs uniquely identify a specific piece of content.

A Content ID identifies a specific sequence of bits, which represent a piece of content, such as an image item, a metadata item, or a track. Content IDs are immutable meaning if the bits change in the content in any way, including the removal of bits or the addition of more bits, systems must generate a new Content ID for the modified content. In this way, a provenance tracking system can maintain information showing the media with Content ID #2 derived from the media with Content ID #1 and how/why it differs. If an old form of content no longer exists, either through modification or deletion from the GIMI file, applications remove the associated Content ID from the GIMI file. Should both the old and new forms of content remain, both Content IDs remain and apply to their respective content. Systems copying content, without modification, to a new location maintain the association of the content with its original Content ID.

Writers declare and store Content IDs for each unique piece of content using built-in capabilities of ISOBMFF and HEIF. Table 10 lists the media type, their Content ID types, and their declaration method.

Table 10: Declaration methods for Content IDs

Media Type	Content ID Type	Declaration Method
Items: still images, Single Instance Application Metadata, and still image regions	ItemContentID	'uuid' extended type item property
Tracks: video, image sequences, sequential application metadata, and audio	TrackContentID	URI-defined metadata item in the track's MetaBox
Track Portions: video, image sequences, sequential application metadata, and audio	TrackPortionContentID	URI-defined metadata item in the track's MetaBox
Samples: sequential images and metadata	SampleContentID	Sample auxiliary information
Image Item Components: still images, Single-SIMA	ItemComponentContentID	'uuid' extended type item property
Track Imagery Components: video, image sequences, Sequential-SIMA	TrackComponentContentID	URI-defined metadata item in the track's MetaBox

NGA.STND.0076-01 uses different methods of declaring Content IDs depending on the Content ID type (see column 2 of Table 10); sections 6.7.1 through 6.7.5 below provide details for each type. Table 11 shows a summary of the Content ID types, and their declaration parameters. Appendix D describes the process for generating the fixed value UUIDs.

Table 11: Content ID types and their declaration parameters

Content ID	Declaration Parameter	Form
ItemContentID	0x261ef3741d975bbaacbd9d2c8ea73522	uint(8)[16]
TrackContentID	'urn:uuid:15beb8e4-944d-5fc6-a3dd-cb5a7e655c73'	'urn:uuid:<uuid>'
TrackPortionContentID	'urn:uuid:7e26ca23-c8f9-5e55-8776-cd342f81e16d'	'urn:uuid:<uuid>'
SampleContentID	'suid'	4CC
ItemComponentContentID	0x9db9dd6e373c5a4e811021fc83a911fd	uint(8)[16]
TrackComponentContentID	urn:uuid:fef58f02-43a6-5aaf-a891-099b1953d1f6	'urn:uuid:<uuid>'

Requirement(s)	
NGA.STND.0076-01_V1.0-57	Content IDs shall use one of the formats listed in Table 9.
NGA.STND.0076-01_V1.0-58	Where writers construct an Intelligence Community identifier (IC-ID) Content ID, the Content ID shall conform to the ODNI Intelligence Community Technical Specification - XML Data Encoding Specifications for Intelligence Community Identifier.
NGA.STND.0076-01_V1.0-59	Where writers construct an Intelligence Community identifier (IC_ID) Content ID, writers shall use the canonical form UUID conforming to ISO/IEC 9834-8 and NGA.RP.0001 for the suffix of the IC_ID.
NGA.STND.0076-01_V1.0-60	Where writers construct a URN Content IDs, a writer shall use the canonical form URN UUID ('urn:uuid:<uuid>') conforming to ISO/IEC 9834-8 and NGA.RP.0001.
NGA.STND.0076-01_V1.0-61	Where media content has a Content ID, when modifying the media content, writers shall replace the media content's Content ID with a new Content ID
NGA.STND.0076-01_V1.0-62	All Content IDs in an NGA.STND.0076-01 file shall have associated media content.

6.7.1 Item Content IDs

When declaring a media item (other than a Content ID declared as a metadata item) in any of the MetaBoxes in a file, writers generate an ItemContentID and associate it with the media item using a 'uuid' extended type item property. The item property's *extended type* value is always set to 0x261ef3741d975bb4a24aacbd9d2c8ea73522. The ItemContentID value for a specific item is either an IC-ID or a URN Content ID. The ItemContentID value becomes the payload for an ItemContentID property. The writer associates the property with the appropriate item using an item property association.

Figure 17 shows a block diagram of the property, including the extended type value for the item property box.

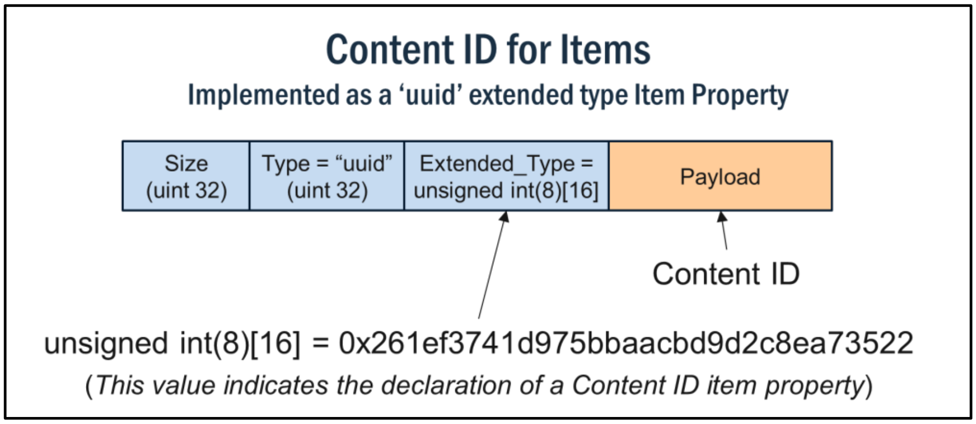


Figure 17: The extended type ItemContentID item property

The following is the formal definition, including the SDL, of the ItemContentID.

ItemContentID item property

Definition

Box type: '`uuid`'
Extended type: `0x261ef3741d975bbaacbd9d2c8ea73522`
Property type: Descriptive extended type item property
Container: ItemPropertyContainerBox
Mandatory (per item): Yes
Quantity (per item): One

The `ItemContentIDProperty` provides universally unique identification for content declared as an item. This includes image items, metadata items, and region items declared in the file-scoped MetaBox, as well as metadata items declared inside the MetaBox of a MovieBox or a TrackBox. Properties are carried in the ItemPropertyContainerBox ('ipco') so the ItemContentIDs for all items are stored in their respective MetaBox.

`Essential` is set equal to 0 for an ItemContentID item property.

Syntax

```
aligned(8) class ItemContentIDProperty
extends ItemProperty('uuid',0x261ef3741d975bbaacbd9d2c8ea73522){
  utf8string ItemContentID;
}
```

Semantics

`ItemContentID` A universally unique Content ID for a declared item, where `ItemContentID` is an IC-ID of the form 'guide://<prefix>/<uuid>', or a UUID of the form 'urn:uuid:<uuid>'. In both cases, writers generate the UUID in accordance with the guidance provided in ISO/IEC 9834-8 and NGA.RP.0001.

Figure 18 shows an example of a file with an HEVC coded image and an ItemContentID item property (shown with an IC-ID implementation).

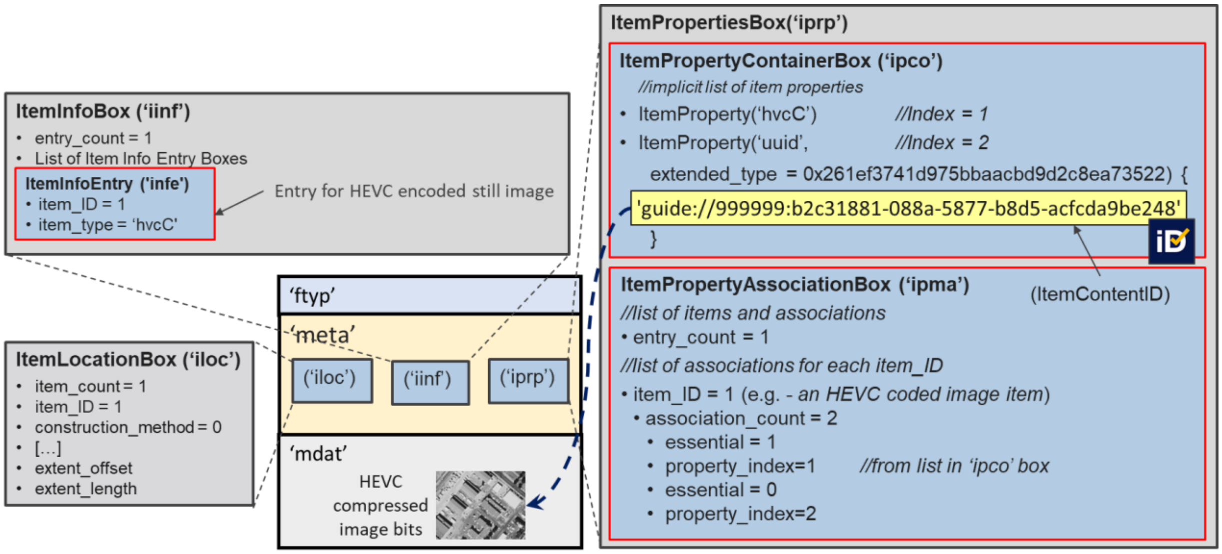


Figure 18: Declaration of an ItemContentID for an HEVC encoded image item

In the figure, the ItemPropertyAssociationBox ('ipma') shows the association of both the 'hvcC' property and the ItemContentID property with the HEVC image item. The hvcC property is labeled as essential as the property is required for decoding the HEVC image. While the ItemContentID is a requirement of this standard, writers do not label the property as essential so viewers not supporting the ItemContentID item property can still display the image.

Requirement(s)	
NGA.STND.0076-01_V1.0-63	Image items shall associate with an ItemContentID property.
NGA.STND.0076-01_V1.0-64	When a metadata item is not a Content ID itself, the metadata item shall associate with an ItemContentID property.
NGA.STND.0076-01_V1.0-65	When a region item is declared within an image, the region item shall associate with an ItemContentID property.

6.7.2 Track Content IDs

Definition

When writers generate a media track, the writer creates a Content ID for the track as a metadata item in the track's MetaBox. To facilitate the declaration of the Content ID, this standard defines a `TrackContentIDMetadataItem` class. The class includes a single parameter providing a Content ID for video, image sequence, sequential metadata, and audio track media content. The payload for a `TrackContentIDMetadataItem` is a string formatted `TrackContentID`.

Syntax

```
aligned(8) class TrackContentIDMetadataItem {
     utf8string TrackContentID;
}
```

Semantics

`TrackContentID` is the payload for a `TrackContentIDMetadataItem` and a Content ID for a declared track, where `TrackContentID` is either an IC-ID or a URN Content ID.

Declaration of a TrackContentIDMetadataItem follows these steps:

1. Writers declare the TrackContentIDMetadataItem in the track's MetaBox by constructing an ItemInfoEntry and adding it to the MetaBox's ItemInfoBox list.
2. The ItemInfoEntry's item_type is set to 'uri '.
3. For TrackContentIDMetadataItems, the ItemInfoEntry's item_uri_type is a URN with a constant value of 'urn:uuid:15beb8e4-944d-5fc6-a3dd-cb5a7e655c73', per instructions in Appendix D using the text 'TrackContentID'. Table 12 summarizes the uri and item_uri_type values of the ItemInfoEntry.

Table 12: ItemInfoEntry settings for declaring a Content ID for a track

ItemInfoEntry Box Parameter	ItemInfoEntry Box Value Setting
item_type	'uri '
item_uri_type	'urn:uuid:15beb8e4-944d-5fc6-a3dd-cb5a7e655c73'

4. The payload of the TrackContentIDMetadataItem is a TrackContentID, which writers always store in the local ItemDataBox inside the track's MetaBox. This requires a construction_method set to 1 in the ItemLocationBox when declaring the TrackContentIDMetadataItem.

The TrackContentIDMetadataItem does not require a reference as it associates with the track through encapsulation. The TrackContentID applies to, and uniquely identifies the entire track. When using track fragments, writers store a single TrackContentID in the TrackBox of the track, and the TrackContentID applies to the entire track. The MetaBoxes in track fragments do not carry a TrackContentID.

Figure 19 shows an example of adding a TrackContentID to a video track. The writer declares the TrackContentIDMetadataItem as a metadata item using the parameters of Table 12. The construction method in the ItemLocationBox is set to 1 because the TrackContentID has been stored as an IC-ID in the ItemDataBox of the Track's MetaBox.

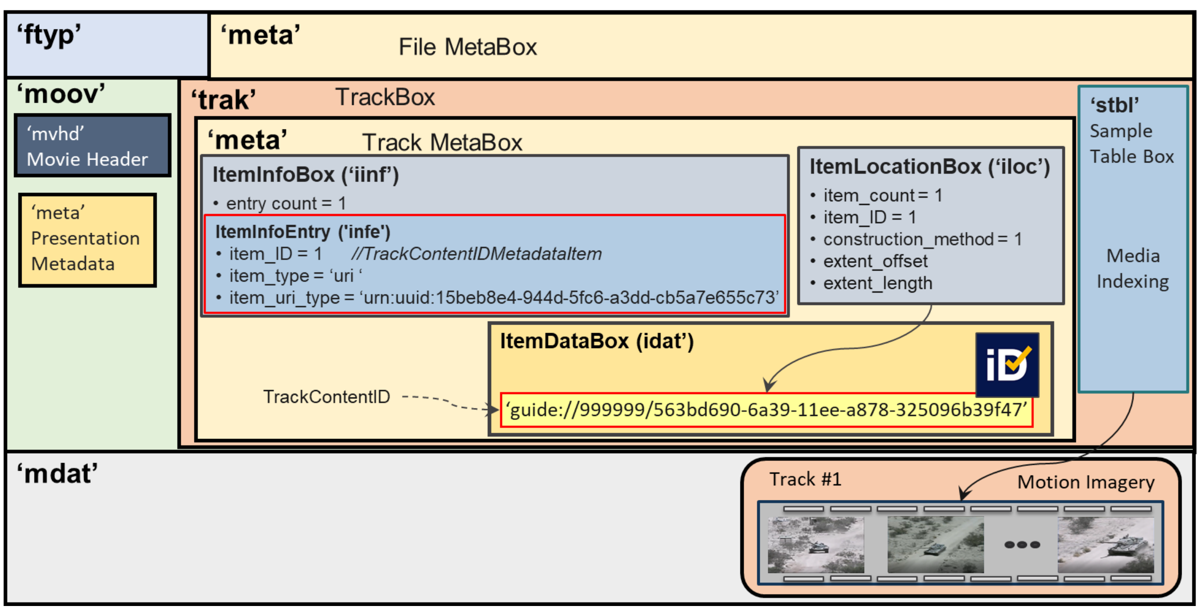


Figure 19: Generating a Content ID for a media track

Requirement(s)	
NGA.STND.0076-01_V1.0-66	Writers shall store exactly one unique TrackContentID in the ItemDataBox of every track's MetaBox.

6.7.3 Track Portion Content IDs

Definition

Writers use track portions to identify specific temporal regions of a track. NGA.STND.0076-01 defines track portions as a contiguous set of track samples in a track. A track either does not contain any track portions or the track contains two or more track portions; a track will never contain just one track portion. When a track does not contain any track portions there is no need for track portion content IDs. Track portions can be as small as a single sample of a track. When present, track portions do not overlap and all samples in a track are in one, and only one, track portion. In the case where a track does not contain any track portions, a writer can treat the track as having a single virtual track portion, with an ID equal to the TrackContentID.

For each track portion a writer defines a TrackPortionContentID, which is the Content ID for the track portion. When a track does not use track portions, no TrackPortionContentIDs are present, and the TrackContentID identifies all samples in the track.

The TrackPortionContentID serves two roles. First, it provides support data with a mechanism for describing and attaching information to user defined portions (sub-sections) of a track. Second, it provides an ability to apply unique security markings to different portions of a track. For instance, if a track is ten minutes long and an event occurs over ten contiguous seconds of the track, writers can label the ten second portion with different access restrictions than the remainder of the track. In cases where a user needs to label a single sample with a different classification than its neighboring samples, the writer typically defines three track portions: a track portion for the samples preceding the single sample, a track portion for the single sample, and a track portion for the samples after the single sample. In the case where the single sample is the first or last sample, the writer only needs two portions. When writers define track portions, the writer labels each track portion with its own classification. Track portions are the only method for assigning security to individual or groups of track samples; NGA.STND.0076-01 does not use Sample Content IDs (SampleContentID) to label samples with security markings.

A track portion begins with a sync sample, where the first sample in a track portion is an independently decodable media access unit. Samples in a track portion are decodable using only the samples contained within the same track portion. This ensures readers can decode, extract, and examine all samples in a track portion without any required interaction with a neighboring track portion. Related to this, when selecting a start sample and an end sample for a track portion with more restrictive classification labeling, writers must expand the track portion forwards and/or backwards as necessary to ensure the first sample of the track portion is a sync sample and the last sample is decodable using only sample information inside the given track portion. The first sample of the next track portion is also independently decodable. An additional use case for consideration is when a contiguous section of a track covers a single GOP structure, but two events, with potentially conflicting markings, each more restrictive than the other in some way, occur within the contiguous section of the GOP structure. In this case, the end user must resolve the conflict, potentially by re-encoding to add an additional I-frame between the two events to generate two track portions with appropriate labeling for each portion. In a similar case, where using an open GOP structure and references from predictive frames occur over an I-frame boundary, writers may need to re-encode leading and trailing GOPs into closed GOPs to meet the labeling requirements.

When using track fragments, the TrackBox in the MovieBox and the additional TrackFragmentBoxes in one or more MovieFragmentBoxes contain samples for an entire track. When generating a set of track portions for a fragmented track, the portions address all samples in the track, including the samples in the fragments. Even though each TrackFragmentBox contains its own MetaBox, writers always store the TrackPortionList in the MetaBox of the TrackBox associated with the track.

Writers store track portions compactly in a TrackPortionList inside the MetaBox of a track. Figure 20 illustrates an example of track portions. Each item of the TrackPortionList contains two items, the TrackPortionContentID and the sample number of the first sample in the track portion. The sample numbers in ISOBMFF are the ordinal index number of a given sample where the first sample of a track has sample number 1. Providing the sample number of the first sample in the track portion enables locating the beginning of the track portion. The end of the track portion is the sample immediately preceding the first sample of the next track portion (or the last sample in the track in the case of the last track portion).

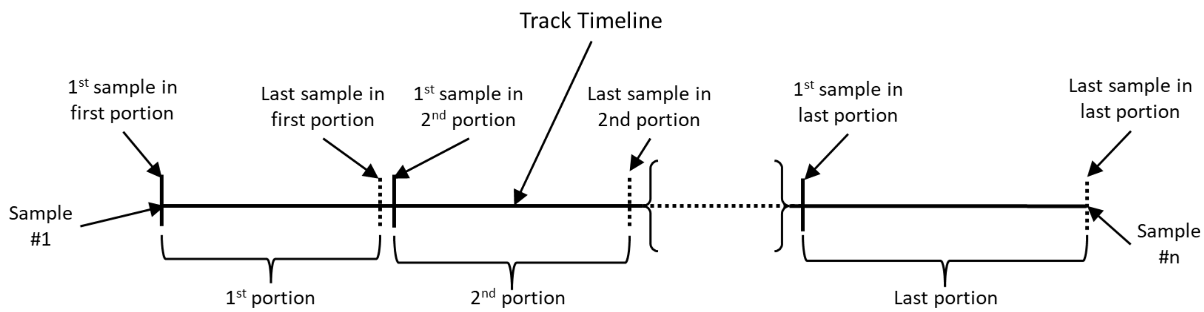


Figure 20: Segmenting a track's timeline into track portions

A TrackPortionList carries information necessary to describe a complete set of track portions for a given track.

Syntax

```
aligned(8) class TrackPortionList {
      unsigned int(32) number_of_portions;
      for(i=0;i<number_of_portions; i++){
            utf8string TrackPortionContentID;
            unsigned int(32) first_sample_in_portion;
      }
```

Semantics

`number_of_portions` indicates the number of user defined portions in the track. The number of portions is two or more.

`TrackPortionContentID` is a Content ID for a specific track portion and is an IC-ID or URN Content ID.
`first_sample_in_portion` is the sample number of the first sample in the portion, provided to locate the first sample of the portion.

Declaration of a TrackPortionList metadata item follows these steps:

1. Writers declare the TrackPortionList metadata item in the track's MetaBox by constructing an ItemInfoEntry and adding it to the MetaBox's ItemInfoBox list.
2. The ItemInfoEntry's item_type is set to 'uri '.
3. For a TrackPortionList, the ItemInfoEntry's item_uri_type is a URN with the constant value of 'urn:uuid:7e26ca23-c8f9-5e55-8776-cd342f81e16d', per instructions in Appendix D using the text "TrackPortionContentID". Table 13 shows the UUID in its URN form. Writers always use this URN value when declaring a list of TrackPortionContentIDs.

Table 13: ItemInfoEntry settings for declaring a list of TrackPortionContentIDs

ItemInfoEntry Box Parameter	ItemInfoEntry Box Value Setting
item_type	'uri '
item_uri_type	'urn:uuid:7e26ca23-c8f9-5e55-8776-cd342f81e16d'

4. The payload of the TrackPortionList includes the number of portions, and a list of paired TrackPortionContentIDs and sample numbers for the first sample in the track portion.
5. The track portion list is always in the local ItemDataBox inside the track's MetaBox. This requires a construction_method set to 1 in the ItemLocationBox when declaring the TrackContentID.

The track portion list metadata item does not require a reference as it is associated with the track through encapsulation and applies to the entire track. When using track fragments, a writer includes a single track portion list in the TrackBox and the list describes all track portions, including those contained in track fragments.

Requirement(s)	
NGA.STND.0076-01_V1.0-67	Where writers generate two or more track portions for a track, writers shall store a TrackPortionList for the track in the ItemDataBox of the track's MetaBox.
NGA.STND.0076-01_V1.0-68	Writers shall create exactly one unique TrackPortionContentID for every Track Portion.
NGA.STND.0076-01_V1.0-69	All track portions shall begin on a sync sample.
NGA.STND.0076-01_V1.0-70	Samples in a track portion shall be decodable using only the samples within the track portion.

6.7.4 Track Sample Content IDs

This standard requires a writer to create a unique Content ID, i.e., SampleContentID, for every track sample. Writers associate a SampleContentID with a sample using Sample Auxiliary

Information boxes. To support the creation and association of SampleContentIDs, this standard defines a SampleContentIDPacket class and a Sample Auxiliary Type.

6.7.4.1 SampleContentIDPacket

Definition

The `SampleContentIDPacket` class contains a `SampleContentID` value for a sample in a track. The `SampleContentID` provides a mechanism for support data to associate information with each specific sample in a track, such as each image frame in a video track. To attach `SampleContentID` values to track samples, writers include `SampleContentIDPackets` as sample auxiliary information.

Syntax

```
aligned(8) class SampleContentIDPacket{
     utf8string SampleContentID;
}
```

Semantics

`SampleContentID` is a Content ID for a track sample and is sample auxiliary information for a given track. Each `SampleContentID` is either an IC-ID or URN Content ID (see Section 6.7).

6.7.4.2 SampleContentIDs as Sample Auxiliary Information

This standard defines the 'suid' sample auxiliary information type to enable assigning ContentIDs to samples.

Definition

Aux Info Type: `'suid'`
Container: Sample auxiliary information
Mandatory: Yes
Quantity: one per sample

A `SampleContentIDPacket` is a sample auxiliary information payload for each sample in a track.

Syntax

```
SampleContentIDPacket SampleContentID_packet;
```

Semantics

`SampleContentID_packet` is an instance of the `SampleContentIDPacket` class, which Section 6.7.4.1 describes.

`aux_info_type` (an `'saiz'` and `'saio'` box parameter) is set to `'suid'`, which indicates `SampleContentIDPacket` values.

`aux_info_type_parameter` (an `'saiz'` and `'saio'` box parameter) is unused for the `aux_info_type` of `'suid'`. For the 'suid' auxiliary type, the `aux_info_type_parameter` is a reserved value, and all 32-bits of the unsigned word are set to 0.

Writers set the remaining 'saiz' and 'saio' box parameters as per ISO/IEC 14496-12 box specification.

Requirement(s)	
NGA.STND.0076-01_V1.0-71	For every sample in an imagery track, a writer shall include a SampleContentID as SampleAuxilaryInformation of aux_info_type equal to 'suid'.
NGA.STND.0076-01_V1.0-72	For every sample in a sequential metadata track, a writer shall include a SampleContentID as SampleAuxilaryInformation of aux_info_type equal to 'suid'.

6.7.5 Component Content IDs

Component Content IDs allow support data to describe features of individual components of media declared with an image codec, including still imagery, motion imagery, and Single-SIMA and Sequential-SIMA metadata. Figure 21 illustrates the labeling of each media component with a Content ID.

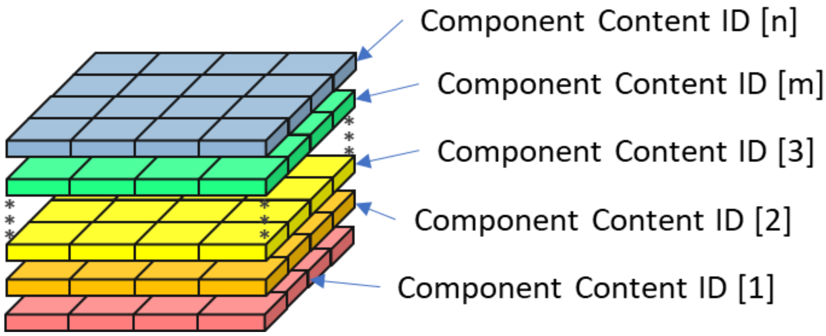


Figure 21: Labeling of image bands with component Content IDs

Writers can link application metadata such as image band specifics (e.g., center wavelength, targeting information in a specific band, etc.) using component Content IDs. For still image and Single-SIMA items, writers implement component Content IDs as a list of ItemComponentContentIDs in an item property. For video, image sequence, and Sequential-SIMA tracks, component Content IDs apply to all samples in the track and writers implement the Content IDs as a metadata item in the track's MetaBox in the form of a list of TrackComponentContentIDs. The ordering of the TrackComponentContentIDs in the list follows the ordering of components as defined by the codec in use.

Component Content IDs apply to content declared as items and tracks using approved imagery codecs. When Single-SIMA items or Sequential-SIMA tracks are using the ISO/IEC 23001-17 uncompressed codec, writers generate component Content IDs for each component. This facilitates the labeling of individual components within SIMA content, such as image quality metrics and pixel level geodetics. The MPEG component type selection for SIMA content is chosen to guide display behavior. To further describe these layers of SIMA content, ontology structured metadata refer to the component Content ID of a specific component to assign detailed meaning to the component.

A unique, but common form of uncompressed imagery component is the filter array component, which carries Bayer and other types of patterned imagery. While a Bayer image carries a pattern of one red, one blue, and two green pixels in a two-by-two configuration, the overall component has a single component Content ID. The description of the pattern is in ISO/IEC 23001-17's ComponentPatternDefinitionBox, defined in ISO/IEC 23001-17.

With the use of approved compression codecs, each declared component receives a component Content ID. For common implementations, such as RGB and YUV formatted imagery, there are three component Content IDs. When implementing multi or hyperspectral images, writers will create tens to thousands of component Content IDs to fully define the media in each item or track.

6.7.5.1 Item Component Content IDs

ItemComponentContentID item property

Definition

Box type: 'uuid'
Extended type: 0x9db9dd6e373c5a4e811021fc83a911fd
Property type: Descriptive extended type item property
Container: ItemPropertyContainerBox
Mandatory (per item): Yes
Quantity (per item): One

The ItemComponentContentIDProperty provides universally unique identification for a specific component in a declared image item. Writers assign an ItemComponentContentIDProperty with a unique ItemComponentContentID to each image component.

Essential is set equal to 0 for an ItemComponentContentIDProperty item property.

Syntax

```
aligned(8) class ItemComponentContentIDProperty
extends ItemProperty('uuid',0x9db9dd6e373c5a4e811021fc83a911fd){
  unsigned int(32) number_of_components;
  for(i=0;i< number_of_components;i++)
    utf8string ItemComponentContentID;
}
```

Semantics

number_of_components is the number of components in the item associated with the property.

ItemComponentContentID is an IC_ID or URN Content ID (see Section 6.7). There is one ItemComponentContentID for each component of an item declared with an image codec. The ordering of the list's ItemComponentContentIDs matches the ordering of the image codec's components; thereby each image component matches its unique ItemComponentContentID correctly.

Requirement(s)	
NGA.STND.0076-01_V1.0-73	Writers shall create exactly one unique `ItemComponentContentID` for every image item's components.
NGA.STND.0076-01_V1.0-74	Writers shall associate every image item component with its `ItemComponentContentIDs` using the `ItemComponentContentIDProperty` in the file-level MetaBox.

6.7.5.2 Track Component Content IDs, TrackComponentContentIDList

Definition

This standard requires storing component Content IDs of a video, image sequence, or Sequential-SIMA track as a list in the MetaBox of the track. The `TrackComponentContentIDList` class defines the list of Component Content IDs which every sample uses in the track.

In ISOBMFF each sample refers to a SampleEntry (in the SampleDescriptionBox) which defines characteristics of the sample. Tracks may define only one SampleEntry, in which case all samples refer to the same SampleEntry. Tracks may use multiple SampleEntrys, with samples referring to one of the SampleEntrys. With multiple SampleEntrys, the component makeup of the samples in the track are not necessarily consistent, i.e., a sample's component list might change depending on which SampleEntry the sample refers to. NGA.STND.0076-01_V1.0 restricts the number of sample entries for a track to one; however, to facilitate future extensibility and allow for multiple SampleEntrys, the following class structure includes the ability to map SampleEntry indexes to a unique list of TrackComponentContentIDs.

Syntax

```
class TrackComponentContentIDList{
     unsigned int(32) number_of_sample_entries;
     for(i=1;i<=number_of_sample_entries;i++){
          unsigned int(32) sample_entry_index;
          unsigned int(32) number_of_components;
          for(j=0;j<number_of_components; j++){
               utf8string TrackComponentContentID;
          }
     }
}
```

Semantics

`number_of_sample_entries` indicates the number of `SampleEntrys` in the track. This value equals the number of `SampleEntrys` in the track. In this version of NGA.STND.0076-01, this value is set to 1.

`sample_entry_index` is an integer identifying the `SampleEntry's` index in the track.

`number_of_components` indicates the number of components defined in the `SampleEntry` defined by the current `sample_entry_index`.

`TrackComponentContentID` is a Content ID for a component within a video, image sequence, or Sequential-SIMA track. There is one `TrackComponentContentID` for each sample component of a given `SampleEntry`. The ordering of the `TrackComponentContentIDs` in the `TrackComponentContentIDList` matches the component ordering of the sample as defined by the codec for the given sample entry.

Instructions

Declaration of a TrackComponentContentIDList metadata item follows these steps:

1. Writers declare a TrackComponentContentIDList in the track's MetaBox by constructing an ItemInfoEntry and adding it to the MetaBox's ItemInfoBox list.
2. The ItemInfoEntry's item_type is set to 'uri '.
3. The associated item_uri_type is an absolute URI and is encoded as a URN in the form 'urn:uuid:<uuid>'.
4. For a TrackComponentContentIDList, the ItemInfoEntry's item_uri_type is a URN with the constant value of 'urn:uuid:fef58f02-43a6-5aaf-a891-099b1953d1f6', per instructions in Appendix D using the text "TrackComponentContentID". Table 14 summarizes the item_type and the item_uri_type values of the ItemInfoEntry. Writers always use this URN value when declaring a list of TrackComponentContentIDs.

Table 14: ItemInfoEntry settings for declaring a TrackComponentContentIDList

ItemInfoEntry Box Parameter	ItemInfoEntry Box Value Setting
item_type	'uri '
item_uri_type	'urn:uuid:fef58f02-43a6-5aaf-a891-099b1953d1f6'

5. The payload of the TrackComponentContentIDList includes the number of sample entries in the track (set to 1), and for each sample entry, a list of the number of components in a sample, and a list of TrackComponentContentIDs for each component is a sample for the given sample entry.
6. Writers store the TrackComponentContentIDList in the ItemDataBox inside the track's MetaBox. This requires a construction_method set to 1 in the ItemLocationBox when declaring the TrackComponentContentIDList.

The TrackComponentContentIDList metadata item does not require a reference as it is associated with the track through encapsulation and applies to the entire track. When using track fragments, writers store a single TrackComponentContentIDList in the TrackBox of the track, and the TrackComponentContentIDList describes all components in the track, including those contained in track fragments. The MetaBoxes in track fragments do not carry a TrackComponentContentIDList.

Requirement(s)	
NGA.STND.0076-01_V1.0-75	Writers shall create a TrackComponentContentID for every component in the samples of a track.
NGA.STND.0076-01_V1.0-76	Writers shall store a TrackComponentContentID for every component in a track using the TrackComponentContentIDList in the ItemDataBox of the track's MetaBox.
NGA.STND.0076-01_V1.0-77	A track shall contain only one sample entry.

6.8 Security Information Labeling

This section defines security marking requirements specifically for NSG (including DoD and IC) producers and consumers of GIMI files. For GIMI implementations outside the NSG, the GIMI security marking requirements are dependent on end-user requirements. GIMI files leverage the ISOBMFF compatible branding mechanism to signal specific security marking implementations. Within the NSG, applications use the brand for GIMI Security XML (see Section 6.8.1). Outside the NSG, applications creating GIMI files for NATO may implement a brand specifying the carriage of NATO specified security markings. This standard does not require purely commercial GIMI files to carry security markings.

6.8.1 Branding for GIMI Security XML in NSG Applications

To facilitate the quick confirmation of the presence of security markings in a file, NGA.STND.0076-01 uses branding in the FileTypeBox to indicate the presence of GIMI Security XML in a file. When readers initially access the FileTypeBox in a file, the presence of a security brand indicates there are security markings within the file. Implementations within the NSG include the 'sm01' brand to indicate conformance to the requirements of Section 6.8.6.

For implementations outside the scope of the NSG, applications do not include the GIMI Security XML brand ('sm01') in the FileTypeBox nor the GIMI Security XML in the file. Applications outside the scope of the NSG may carry other forms of security markings and an associated brand, depending on end-user requirements for security markings. Applications outside the NSG with no requirements for security markings do not carry security markings and do not contain a security marking brand in the FileTypeBox.

6.8.2 GIMI Security XML

The GIMI Security XML is an XML Document of elements and attributes conforming to the GIMI Security XML Schema (see Section 6.8.6). The GIMI Security XML schema uses the ODNI XML Data Encoding Specification for Information Security Markings (ISM.XML) for all NSG derived security markings within a GIMI file. To add a security marking to the overall GIMI file and to individual primary-media-content (see Section 6.8.3), an NSG writer constructs GIMI Security XML and adds the GIMI Security XML as an XML data object in the ItemDataBox ('idat') of the file-scoped MetaBox. ISOBMFF defines a mechanism for the declaration and carriage of general XML content via a MetaBox. In the ItemInfoBox a writer includes an ItemInfoEntry box to implement the declaration. The process involves assigning the following parameters in the ItemInfoEntry box:

1. an item ID
2. setting the 'item_type' to 'mime'
3. setting the content type equal to 'application/nga-gimi-ism+xml'

This standard defines the content type for a GIMI Security XML document as 'application/nga-gimi-ism+xml', which is not registered with the Internet Assigned Numbers Authority (IANA).

The writer assigns a storage location for the GIMI Security XML content using the ItemLocationBox. This standard requires storing the GIMI Security XML content in the file-scoped MetaBox's ItemDataBox. Writers generate an ItemContentID as an item property (see Section 6.7.1) and associates the property with the GIMI Security XML metadata item.

Figure 22 shows the instantiation of information security marking metadata. The setting of the construction_method to 1 in the ItemLocationBox assigns the storage location to the ItemDataBox of the file-scoped MetaBox.

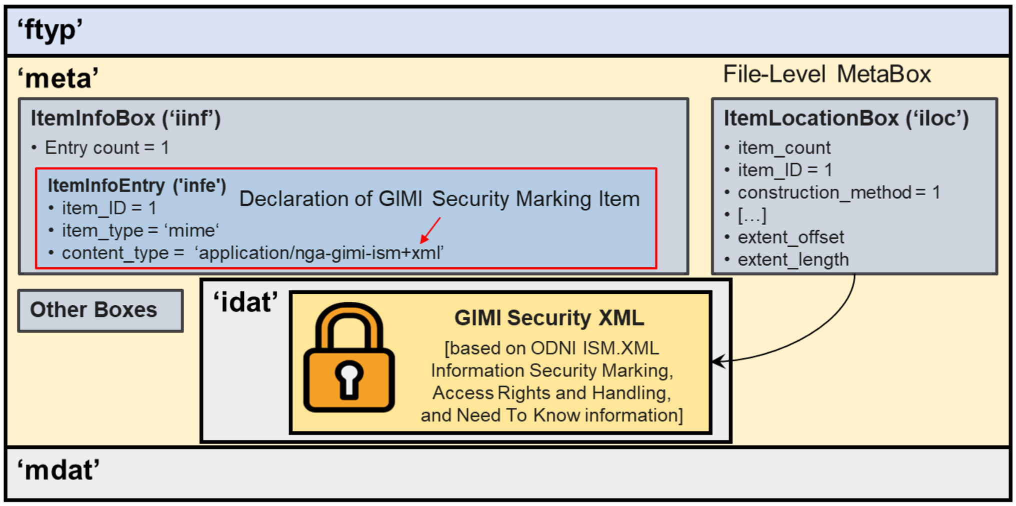


Figure 22: Declaration of GIMI Security XML Information Security Markings

Figure 23 shows the declaration of an ItemContentID for the security marking metadata and the association of the ItemContentID item property to the GIMI Security XML. For NSG applications, this standard requires the GIMI Security XML and the associated ItemContentID.

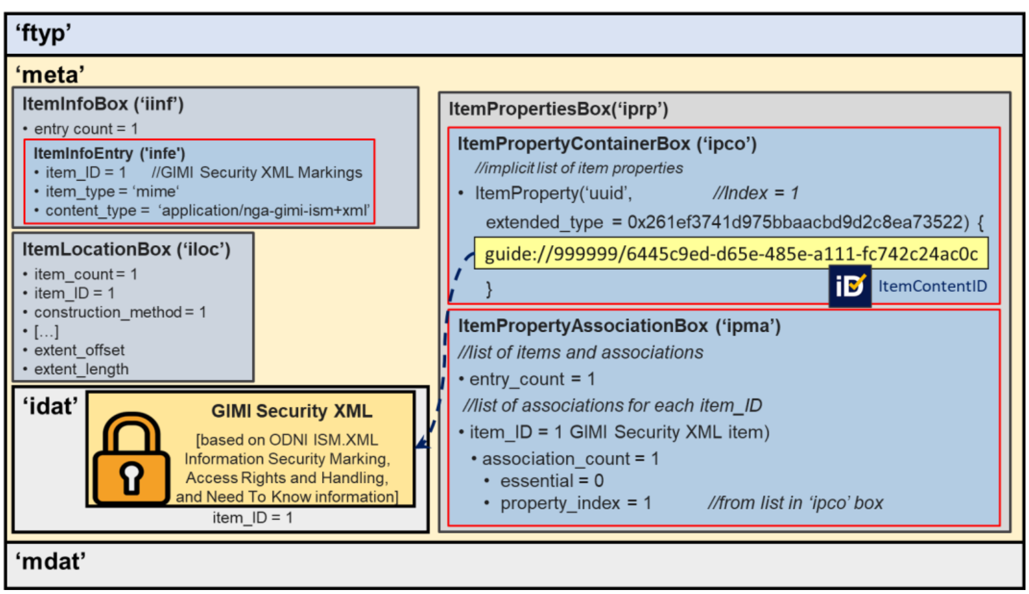


Figure 23: Adding an ItemContentID to the GIMI Security XML metadata

Requirement(s)	
NGA.STND.0076-01_V1.0-78	Where writers include security markings in an NGA.STND.0076-01 file, the writer shall store the security markings in the ItemDataBox of the file-scoped MetaBox of the GIMI file.

6.8.3 Content Marking of a GIMI file and Primary-Media-Content

The GIMI Security XML schema defines a template for assigning security markings to the entire GIMI file and the primary-media-content elements. Primary-media-content are specific elements of GIMI media, with content identifiers, which encompass a logical collection of data for direct security marking. The primary-media-content elements are image and metadata items (declared in a file-level, movie, or movie fragment MetaBox), tracks, and track portions. GIMI files also contain non-primary-media-content, referred to as residual-media-content, which contribute to a primary-media-content element's security marking.

The GIMI Security XML is the method for marking elements of media content contained inside a GIMI file. ISOBMFF and this standard allows for "adjunct-content", where applications declare content in a file, but store the actual content outside of the file, e.g., in a separate file. Applications do not apply security markings to adjunct-content data or their GIMI references; the security marking of data outside a GIMI file is out of scope for NGA.STND.0076-01.

6.8.4 Content Without Specific Security markings

The focus of the GIMI security methodology is the marking of primary-media-content; however, there is residual-media-content throughout a GIMI file. Applications do not assign security markings directly to residual-media-content but consider the impact of this content when marking specific pieces of primary-media-content and performing roll ups, e.g., if the residual-media-content is within a track, applications account for this residual-media-content when determining the marking for the track. Implementation of the wide range of ISOBMFF features and capabilities varies depending on media type and choice of codec. For example, optional boxes in a sample entry are available to carry calibration parameters. This information is residual-media-content and may cause a track to have a higher security marking than when the residual-media-content is not present.

In addition to the impact of residual-media-content, applications do not assign security markings directly to SampleContentIDs, ItemComponentContentIDs, and TrackComponentContentIDs. The purpose of these specific ContentIDs is to link ontology structured data to individual samples or components of an item or track, respectively. TrackPortionContentIDs are available to assign security markings to individual samples or multiple sequential lists (or “runs”) of samples in a more efficient manner than individually marking each sample.

Items and track portions (down to individual samples if required) are the most granular level of security marking for primary-media-content. When marking an image item, the marking applies to all components of the image. Uniquely labeling a single component in an image item (or track) with security markings requires moving the component into its own declared item (or single component track). When still images or Single-SIMI items contain components, the component inherits the markings of its parent item. When tracks contain images or Sequential-SIMA, each component inherits the markings of the track or track portion, depending on the situation. When declaring region items in a specific image item, the region items are residual-media-content of the image. The region item(s) and any associated properties or metadata contribute to the marking of their referenced image item.

6.8.5 Security Marking Process

The security marking process for a GIMI file involves determining and applying a security marking to each piece of primary-media-content in a file and then performing a roll up to determine a marking for the file. The file's security marking must be at or above the most restrictive of the markings for the individual marked elements within the file. An analogy is assigning security to a document. Paragraphs are elements of document sections, so paragraph classifications roll up to form a section's security marking. Sections are elements of an overall document, so the section security markings roll up to form the overall document security marking.

Applications perform the security marking process in steps due to the hierarchy of information in a file. For instance, the labeling of track portions, when present, occurs before the labeling of a track because the track is composed of the track portions. The initial step is to determine the marking levels for individual elements at the lowest points in the hierarchy. These markings depend on the security policy pertaining to a specific GIMI file. After marking the lower level elements, collections of elements are rolled up to determine the marking for the next higher level in the hierarchy. This continues until reaching the top-most file level.

The roll up process relies on the SecMax(a, b, …) operation, as well as any rules regarding the aggregation of content. The SecMax(a, b, …) operation produces the most restrictive security marking from the a, b, … elements, along with any aggregation considerations. For a simple example, if a=Unclassified and b=Secret, and there are no aggregation considerations, then SecMax(a,b) would be Secret. The SecMax operation is dependent on the security policy the file adheres to. The security policy defines the guidelines determining what classification, caveats, releasability, etc. are more restrictive than another classification, caveats, releasability.

With a GIMI file, the marking process begins by defining the hierarchy of primary-media-content elements receiving a marking in the GIMI Security XML. These include:

- GIMI File
    - Items (those declared in the file-level, movie, and fragmented movie MetaBoxes)
    - Tracks
        - Track Portions

The determination of marking levels begins at the lower levels of the hierarchy and proceeds to the higher levels of the hierarchy, eventually ending at the marking for the file level. When tracks are present in a file, the first determination is whether there is a need for track portions. When they are, markings for the track portions are determined. These then contribute to the markings for the containing track. The markings for tracks and items are determined uniquely for each individual track and item element. At this point, a final roll up occurs to determine the overall marking for the file. The following sections describe the steps for the individual element types in more detail.

6.8.5.1 Marking Track Portions

When there is a need to mark samples in a track at different levels, applications use track portions to sequentially group samples requiring the same marking level. An example is situations where a lengthy track contains multiple short events, with each event needing unique marking requirements. Track portions allow each event's samples, and the remaining non-event samples to be marked at an appropriate level.

When marking track portions, applications only consider the marking of the track sample's content and not the impact of other information in the track, such as content in the track's MetaBox, sample auxiliary information, and the track's residual-media-content. Applications use this other information when marking the entire track (see Section 6.8.5.2). Applications define TrackPortionContentIDs to identify each track portion. Applications use the TrackPortionContentID to assign a track portion’s security markings in the GIMI Security XML by writing both the security marking and TrackPortionContentID into the XML.

6.8.5.2 Marking Tracks

There are two separate cases when marking a track, with and without track portion markings.

Case 1: When track portions are present, the markings for a track consider:

- The markings for all the track portions in the track
- The track's metadata items declared in the track's MetaBox
- The track's sample auxiliary information
- The track's residual-media-content

Case 2: When track portions are not present, the marking for a track considers:

- All the media samples in the track
- The track's metadata items declared in the track's MetaBox
- The track's sample auxiliary information
- The track's residual-media-content

The track MetaBox items include those in any fragmented track boxes ('traf'). The sample auxiliary information includes timestamps and potentially other metadata for each sample in the track. Residual-media-content for a track may include extra boxes in the sample entry, such as image calibration data, and information in the user data box for the track. Applications define TrackContentIDs to identify each track. Applications use the TrackContentID to assign a track’s security marking in the GIMI Security XML by writing both the security marking and TrackContentID into the XML.

6.8.5.3 Marking Items

The marking of items applies to both image and metadata items.

Items declared in the file-level, movie, and movie fragment MetaBoxes form a list of zero or more individual item elements. Items declared in a TrackBox's MetaBox are not primary-media-content and do not receive a unique marking; however, items declared in a TrackBox's MetaBox are part of the Track's marking roll up process (see Section 6.8.5.2).

When marking items, applications consider all information attached to the item, such as metadata and region items referenced to an image item, and item properties. Item elements are marked using the ItemContentID for each primary-media-content item. Applications define ItemContentIDs to assign an item's security marking in the GIMI Security XML by writing both the security marking and ItemContentID into the XML.

6.8.5.4 GIMI File Marking Roll Up

When determining the overall file level security marking for a GIMI file, the marking addresses the roll up contributions and the aggregation of the items, tracks, and file-level residual-media-content. Track portions are not directly included in the file marking roll up step as their impact has already been included in their track's roll-up process. To determine the file level security marking, applications apply the SecMax operation on the list of items, the list of tracks, and the residual-media-content at the file level. The resulting file level security marking is written into the GIMI Security XML File level arh:Security element.

6.8.5.5 Security Marking Summary

Table 15 lists the various steps in the GIMI security marking process, including the preparation steps involving security marking determination for both primary-media-content and residual-media-content, and the steps involving rollups within the hierarchy. The process concludes with the file level roll up process, the generation of the GIMI Security XML document, and the insertion of the GIMI Security XML into the ItemDataBox of the file level MetaBox.

Table 15: GIMI Security Marking Implementation Steps

Step	Security Marking Preparation Steps
Track portion marking	When present, determine the security markings for all track portions in each track.
Track marking (when using track portions)	When one or more tracks are present with track portions, determine the security markings for each track taking the following into consideration: $Track_i$ security marking is determined by the contributions of the $track_i$ track portions, the $track_i$ MetaBox content, the $track_i$ sample auxiliary information, and the $track_i$ residual-media-content.
Track marking (when not using track portions)	When one or more tracks are present without using track portions, determine the security markings for each track taking the following into consideration: $Track_i$ security marking is determined by the contributions of the $track_i$ media samples, the $track_i$ MetaBox content, the $track_i$ sample auxiliary information, and the $track_i$ residual-media-content.
Item marking	When one or more items are present, determine the security markings for each item by taking the following into consideration: $Item_i$ security marking is determined by the contributions of the $item_i$ media content, the associated item properties, and association with other media content.
File marking	Determine the file scoped security marking level by performing the roll up process on the list of marked primary-media-content and considering the impact of the file residual-media-content and the impact of content aggregation in the file.
Step	**GIMI Security XML Generation**
GIMI Security XML implementation	Generate the GIMI XML Security document by including the file level security marking and the primary-media-content using the Content IDs and associated marking levels determined in the preparation steps.

The rules and processes for labeling content and performing rollups, etc. are determined by the appropriate security policies. For U.S. DoD applications, markings conform to DoD Manual 5200.01: DoD Information Security Program Volume 2: Marking of Information [28]. For U.S. IC applications, markings conform to the Intelligence Community Directive 710, Classification Management and Control Markings System [29].

6.8.6 GIMI Security XML Schema

The GIMI Security XML Schema defines the security declaration rules for linking the security markings to the GIMI file and the primary-media-content's content identifiers, i.e., item, track, and track portion content identifiers. Figure 23 shows a high level diagram for the GIMI Security XML schema.

Per the GIMI Security XML schema, the GIMISecurity is the root element containing a list of File elements and each file has a list of Content elements. This standard only supports a single File's security; therefore, writers only use one File element in the XML. The GIMI File Security element includes attributes for a file's overall security marking (i.e., ism:classification plus other security attributes as needed) which is the result of the security marking process from Section 6.8.5, and an attribute which indicates the security marking is for the entire GIMI file (i.e., ism:resourceElement="true").

The Content elements address the security marking of each element of primary-media-content, i.e., items, tracks, and track portions. A Content element consists of an identifier attribute containing the primary-media-content's Content ID and security marking attributes (e.g., ism:classification plus other security attributes as needed). The Security XML does not store information about the content type, the XML only stores the pairings of a security marking to a Content ID in a simple flat list. To determine the marking for a specific piece of content with a given Content ID, a reader searches the list of Content elements until a match occurs with the Content ID. This identifies the media content of interest, along with its security marking.

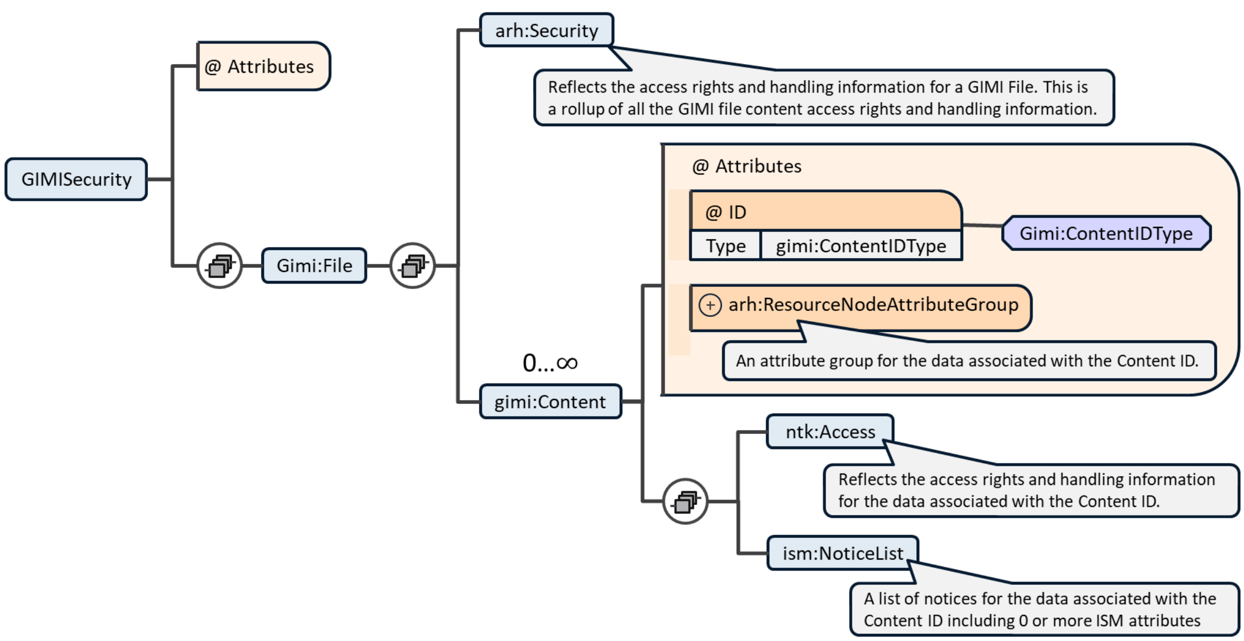


Figure 24: GIMI Security XML Schema diagram

Every element of primary-media-content in a GIMI file must have a security marking in the GIMI Security XML document. When creating GIMI Security XML, the resulting XML must comply with the version 1 GIMI Security XML schema in Table 16. The GIMI Security XML schema relies on the ISM.XML schemas, so for any security marking questions please refer to the ISM documentation [6] at https://w3id.org/ic/standards/public/ism.

Table 16: GIMI Security XML Schema File

NGA.STND.0076-01_Security.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources [
  <!ENTITY canonical_uuid_string "[a-f0-9]{8}-([a-f0-9]{4}-){3}[a-f0-9]{12}">
]>
<?xml-model href="../ISM/Schematron/ISM/ISM_XML.sch"
            type="application/xml"
            schematypens="http://purl.oclc.org/dsdl/schematron"?>
<?xml-model href="../ISM/Schematron/ISMCAT/ISMCAT_XML.sch"
            type="application/xml"
            schematypens="http://purl.oclc.org/dsdl/schematron"?>
<!-- Notices - distEditionBlockReplace -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
   vc:minVersion="1.0" targetNamespace="urn:us:mil:nga:stnd:0076:ism"
   xmlns:ism="urn:us:gov:ic:ism" xmlns:arh="urn:us:gov:ic:arh"
   xmlns:ntk="urn:us:gov:ic:ntk" xmlns:gimi="urn:us:mil:nga:stnd:0076:ism"
   ism:resourceElement="true"
   xmlns:xhtml="http://www.w3.org/1999/xhtml-StopBrowserRendering"
   ism:compliesWith="USGov" ism:createDate="2011-12-01"
   ism:ISMCATCESVersion="202405" ism:DESVersion="202405"
   ism:classification="U" ism:ownerProducer="USA" version="202405"
   elementFormDefault="qualified">
  <xs:import namespace="urn:us:gov:ic:ism"
     schemaLocation="../ISM/Schema/ISM/IC-ISM.xsd" />
   <xs:import namespace="urn:us:gov:ic:arh"
     schemaLocation="../ISM/Schema/ISM/IC-ARH.xsd" />
   <xs:import namespace="urn:us:gov:ic:ntk"
     schemaLocation="../ISM/Schema/ISM/IC-NTK.xsd" />
  <xs:element name="GIMISecurity">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gimi:File" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="ism:ISMRootNodeAttributeGroup"/>
      <xs:attributeGroup ref="ism:ISMResourceAttributeOptionGroup"/>
      <xs:attribute name="GIMISecVer" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="File">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="arh:Security" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="gimi:Content" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
```

NGA.STND.0076-01_Security.xsd

```
  </xs:element>
  <xs:element name="Content">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">
            Lifted from ISM to avoid the questionable rule 511 about
            security elements being required to be resource node true.
          </xhtml:p>
        </xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element ref="ntk:Access" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="ism:NoticeList" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="id" type="gimi:ContentIDType" use="required"/>
      <xs:attributeGroup ref="arh:ResourceNodeAttributeGroup"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="ContentIDType">
    <xs:restriction base="xs:string">
      <!-- Leading digit of guide group is not zero, max 15 digits -->
      <xs:pattern value="urn:uuid:&canonical_uuid_string;|guide://[1-9][0-
      9]{0,14}/&canonical_uuid_string;"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

The following requirements apply to applications within the NSG and define the scope of the 'sm01' security marking brand:

Requirement(s)	
NGA.STND.0076-01_V1.0-79	Where a DoD application creates an NGA.STND.0076-01 file, the security markings shall be conformant to DoD Manual 5200.01 Volume 2.
NGA.STND.0076-01_V1.0-80	Where an IC application creates an NGA.STND.0076-01 file, the security markings shall be conformant to the Intelligence Community Directive 710, Classification Management and Control Markings System.
NGA.STND.0076-01_V1.0-81	For applications in the NSG, writers shall perform the file marking steps in Table 15 and include the file level security marking in the root GIMISecurity File Security element.
NGA.STND.0076-01_V1.0-82	NSG applications writing NGA.STND.0076-01 files shall generate GIMI Security XML for the content in the file.
NGA.STND.0076-01_V1.0-83	NSG applications writing an NGA.STND.0076-01 file shall store GIMI Security XML in the ItemDataBox of the file-level MetaBox.
NGA.STND.0076-01_V1.0-84	GIMI Security XML shall conform to the GIMI Security XML schema in Table 16.

Requirement(s)	
NGA.STND.0076-01_V1.0-85	Security markings in the GIMI Security XML shall conform to the ODNI XML Data Encoding Specification for Information Security Markings (ISM.XML).
NGA.STND.0076-01_V1.0-86	The GIMI Security XML “<File>” element declaration shall include the ism:resourceElement="true" statement.
NGA.STND.0076-01_V1.0-87	Where an NSG application generates GIMI Security XML, the GIMI Security XML shall include the Content ID for all primary-media-content in the file.
NGA.STND.0076-01_V1.0-88	Where an NSG application generates GIMI Security XML, all Content IDs in the GIMI Security XML shall associate with an ISM.XML security marking.
NGA.STND.0076-01_V1.0-89	When a track includes track portions, the track's security marking shall be at or above the highest marking of all the track portions in the track.
NGA.STND.0076-01_V1.0-90	When a still image item references the coded image content of a track's sample, the image’s ItemContentID and sample’s marking via its associated track portion marking or track marking (when no track portions are present) shall have the same security marking in the GIMI Security XML.
NGA.STND.0076-01_V1.0-91	The security marking for a GIMI file shall be at or above the highest marking for all the primary-media-content security markings in the file.

6.9 Audio Tracks

Applications may require (or desire) the inclusion of audio content. This standard does not require but does allow for encoding and including audio content within a GIMI file. To support interoperability, NGA.STND.0076-01 requires the use of profiles defined in the Advanced Audio Coding (AAC) standard, which is defined in ISO/IEC 13818-7:2006 (MPEG2-Part 7: Advanced Audio Coding) [30] and ISO/IEC 14496-3 (MPEG4 Part 3: Audio) [31]. Of note, MPEG2-Part 7 is a MISB approved audio codec, via ST 1101.1 [32]. ISO/IEC 14496-3 and 14496-12 define the formatting and carriage capabilities for audio tracks. When audio tracks are present, they are treated as primary-media-content for the purposes of security markings.

Table 17: ISO/IEC standards associated with audio coding

Encoding Method	Codec Standard	Carriage in ISOBMFF
MPEG-2 Part 7: Advanced Audio Coding (AAC)	ISO/IEC 13818-7:2006	ISO/IEC 14496-12
MPEG-4 Coding of audio-visual objects Part 3: Audio	ISO/IEC 14496-3	ISO/IEC 14496-12

Requirement(s)	
NGA.STND.0076-01_V1.0-92	Where storing audio in an NGA.STND.0076-01 file, the writer shall code the audio with one of the codec standards from Table 17
NGA.STND.0076-01_V1.0-93	The carriage of audio tracks in an NGA.STND.0076-01 file shall conform to the carriage standard for the audio codec in Table 17.

6.10 File Generation and Editing - Privacy and Security Considerations

When generating and editing files, files must address privacy and security considerations. As NGA.STND.0076-01 can store information not normally displayed by consumer commercial software applications, GIMI software applications will require processes and tools to ensure GIMI files only contain authorized content. The classification of files must address all content within a file, not just items which are viewable. Hidden images, for example, may raise the classification level, but may not be viewable using readily available software applications intended for consumer commercial or personal use.

Removal of items and tracks from a file requires specific actions to delete the content from a MediaDataBox or ItemDataBox. When removing content from a file, applications must take care to edit or remove any references, groups, entities, associated metadata, properties, sample auxiliary information, etc. associated with the removed content. Deletion of content, in the context of this standard, involves either rewriting the file without the content, or overwriting the content, conformant with required regulations, to ensure the deleted content is not retrievable.

Annex N of ISO/IEC 23008-12 provides an overview of file implementation and editing topics with suggested best practices related to privacy and security of image and metadata content.

6.11 File Complexity

NGA.STND.0076-01 leverages ISOBMFF branding to provide an upfront indicator of interoperability information. The branding allows an application to easily probe a file for content type and makeup. Files may contain a single type and encoding of content, or a combination of independently usable content of varying types with the same or different encoding methods. Even though a file may be large and quite complex, there may be individual pieces of content accessible by modest software tools. Readers can discover compatible images, metadata items, and tracks and then the application can access, open, and manage as capabilities allow. All other content the reader is not capable of reading and handling is avoided. While not a requirement, applications may, where appropriate, use the ability to include alternative versions of images encoded in uncommon forms of encoding. As an example, re-encoding an uncompressed or JPEG 2000 encoded image using HEVC or AVC, and possibly in a lower resolution, to provide a visual of the image in a more highly interoperable form.

Appendix A - Retired Requirements

N/A

Appendix B - 4CC Information for GIMI Codecs (Informative)

Individual codecs utilize 4CCs to signal their presence and configuration information for specific content within a file. In some cases, branding indicates codec specific implementation within a certain format standard, such as HEIF. For video tracks, codec signaling is through the Sample Entry 4CC. The use of brand, item type, and sample entry codes indicates to a reader the requirements for decoding and interpreting encoded content. Table 18 lists a summary of 4CCs for implementation of image items using GIMI approved codecs. Table 19 lists a summary of 4CCs for image sequences, and Table 20 lists a summary of 4CCs for video tracks. These lists are not exhaustive but address common usage in GIMI. Additional derivative brands, outlined in referenced documentation for each codec, are available for addressing specific features such as higher-level profiles, etc.

For scientific, engineering, and other metric imaging applications, the uncompressed and JPEG 2000 codecs provide for maximum fidelity in coded image sets. For use cases requiring interoperability with consumer commercial media applications, tools, and devices, the HEVC and AVC family of profiles provides for very broad interoperability and are good options for use with disadvantaged users, disaster recovery situations, etc.

Table 18: Common 4CC codes for Still Imagery

Image Items				
Codec	**Brand**	**Image Item Type**	**Codec Configuration**	**Description**
Uncompressed	N/A	'unci'	'uncC'.	Image item coding and carriage in HEIF is per ISO/IEC 23001-17. There is no brand for uncompressed images.
JPEG 2000, HTJ2K	'j2ki'	'j2k1'	'j2kH'	Image item coding is per JPEG 2000 or HTJ2K. The JP2 Header box is included as an essential item property ('j2kH') for the image item. J2K and HTJ2K codestreams are carried identically in HEIF. A codestream using HTJ2K is indicated in the CAP marker segment in the codestream header. HEIF metadata alone does not indicate a distinction. Carriage in HEIF is per ISO/IEC 15444-16.
HEVC	'heic'	'hvc1'	'hvcC'	Image item coding is per ISO/IEC 23008-2. Carriage in HEIF is per ISO/IEC 23008-12.
HEVC	'heix'	'hvc1'	'hvcC'	Image item coding is per ISO/IEC 23008-2. The content of the item conforms to the Main 10 profile or any of the format range extensions profiles of HEVC. Carriage in HEIF is per ISO/IEC 23008-12.
AVC	'avci'	'avc1'	'avcC'	Image item coding is per ISO/IEC 14496-10. Carriage in HEIF is per ISO/IEC 23008-12.

Table 19: Common 4CC codes for image sequences

Image Sequence Tracks				
Codec	**Brand**	**Sample Entry Type**	**Codec Configuration**	**Description**
Uncompressed	N/A	'uncv'	'uncC'	Implementation is per ISO/IEC 23001-17, and there is no brand for uncompressed image sequences.
JPEG 2000, HTJ2K	'j2is'	'j2ki'	'j2kH'	An image sequence of JPEG 2000 or HTJ2K codestreams. The J2KSampleEntry is defined in ISO/IEC 15444-16.
HEVC	'hevc'	'hvc1'	'hvcC'	An HEVC image sequence, defined in ISO/IEC 23008-12, with bitstream conforming to the Main profile of HEVC. The HEVCSampleEntry is used as specified in ISO/IEC 14496-15.
HEVC	'hevx'	'hvc1'	'hvcC'	An HEVC image sequence, defined in ISO/IEC 23008-12, with bitstream conforming to the Main 10 profile or any of the format range extensions profiles of HEVC. The HEVCSampleEntry is used as specified in ISO/IEC 14496-15.
AVC	'avcs'	'avc1'	'avcC'	An AVC image sequence, defined in ISO/IEC 23008-12. The AVCSampleEntry is used as specified in ISO/IEC 14496-15.

Table 20: Common 4CC codes for video tracks

Video Tracks				
Codec	**Brand**	**Sample Entry Type**	**Codec Configuration**	**Description**
Uncompressed	N/A	'uncv'	'uncC'	Implementation is per ISO/IEC 23001-17, and there is no brand for uncompressed image sequences.
HEVC	varies	'hvc1'	'hvcC'	An HEVC video track, defined in ISO/IEC 14496-15, with bitstream conforming to the Main profile of HEVC. The HEVCSampleEntry is used as specified in ISO/IEC 14496-15.
AVC	varies	'avc1'	'avcC'	An AVC image sequence, defined in ISO/IEC 23008-12. The AVCSampleEntry is used as specified in ISO/IEC 14496-15.

Note: The implementation of JPEG 2000 video tracks in ISOBMFF is expected to become available in Q1 FY26, when ISO approves Edition 3 of ISO/IEC 15444-16.

Appendix C - ISM Security Marking Examples (Informative)

This Appendix contains two examples. The first example uses fake security markings and walks through the security marking roll-up process from Section 6.8.5; this example will not validate against the XML schema because the security markings are not real ISM.XML markings. The second example is the same example but without the fake security markings, therefore the file will validate against the GIMI Security XML and ISM.XML schemas.

Figure 25 illustrates an example application of the GIMI security marking roll up steps, producing a GIMI Security XML document. The illustration shows a GIMI hierarchy of both media data, roll up steps, and on the right side of the illustration are the resulting GIMI Security XML snippets.

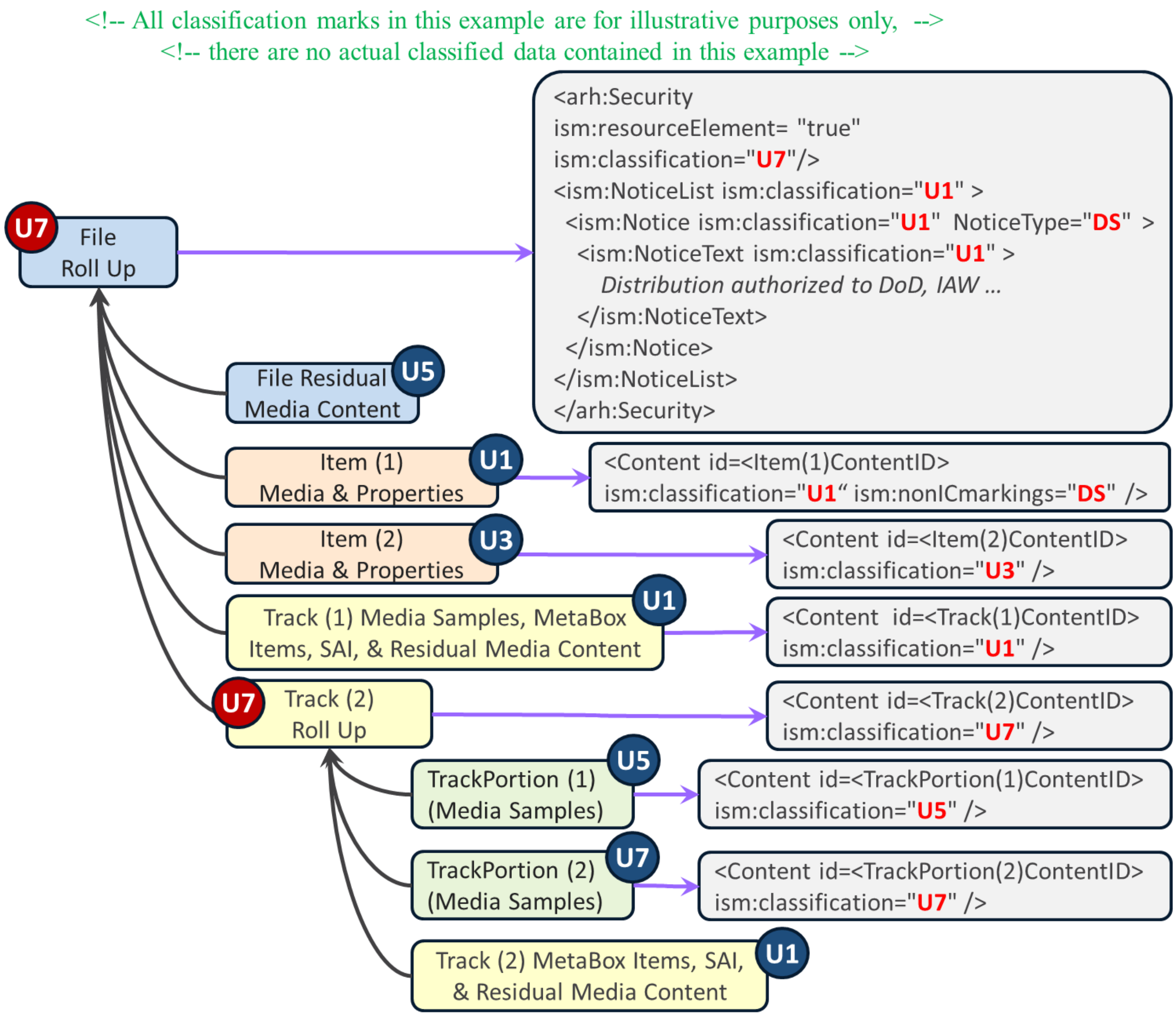


Figure 25: Illustration of Security Marking Roll-Up

This example illustrates a GIMI file with file-scoped residual-media-content (blue), two media items (with associated properties) (orange), and two tracks (with supporting media samples, metadata, sample auxiliary information, and residual-media-content) (yellow). The first track does not include track portions but does include residual-media-content. The second track includes two track portions (green) as well as MetaBox items, SAI, and residual-media-content. The first item includes a DoD Distribution Statement ("DS") notice.

In this example, initial content marking determinations result in security levels indicated in blue circles. Roll up procedures for Track(2) and the File then result in security levels indicated in red circles. An application's security policy identifies the rules for determining appropriate levels for each piece of content. Each example marking is a "U" and a numeric value, where a higher numeric value indicates a higher security marking. These markings are not real (they are fake) but are intended to show the roll up process for content with varying marked levels. The "U1" level represents the lowest classification level (equivalent to unclassified, in the example). The "U7" level represents the highest level of classification for the example. The red circles indicate a security marking roll up, using the SecMax function.

When DoD Distribution Statements are included as NoticeText in the GIMI Security XML, one or more content elements may contain an embedded Distribution Statement, but only one, with appropriate attributes, applies to the entire file and is included at the File level.

An application follows the steps in Table 15 to determine the file-scoped security markings:

Preparation steps:

- Starting with the Track Portions (Green) in Track(2) (Yellow), the application marks each track portion with the given security marking (Blue circle). These result in the GIMI Security XML snippets on the right (Gray), with the labels of U5 for Track Portion(1) and U7 for Track Portion(2).
- Track(2)'s MetaBox items, sample auxiliary information, and residual-media-content are determined to be a U1. No GIMI Security XML snippet is created since this content, by itself, is not primary-media-content. Instead, this information contributes to the roll up marking for Track(2).
- The security marking for Track(2) is generated via the roll up of the markings from the two track portions and Track(2)'s MetaBox Items, SAI, and residual-media-content, resulting in the red security marking of "U7" (Red Circle) for Track(2), which is included in the GIMI Security XML snippet on the right (Gray). The roll up for Track(2) is the result of SecMax(U5, U7, U1).
- For Track(1) (Yellow), the entire track, including its media samples, the MetaBox items, the sample auxiliary information, and the residual-media-content are given a marking level of U1, which is included in Track(1)'s GIMI Security XML snippet on the right (Gray).
- For Item(1) and Item(2) (Orange), the media item and its associated properties are evaluated and determined to be a U1 for Item(1) and a U3 for Item(2). Item(1) is marked with a DoD Distribution Statement, indicated by the ism:nonICmarkings="DS" attribute. These values are included in the GIMI Security XML snippets for each item on the right(Gray).
- The File roll up begins with an evaluation of the file level residual-media-content, which is determined to be a U5 (blue). The File scoped security marking value is then defined by rolling up the marking values for the file residual-media-content, Item(1), Item(2), Track(1), and Track(2). This is determined to be U7 = SecMax(U5, U2, U3, U1, U7). Since Item(1) is marked with a DoD Distribution Statement, the File roll up includes a Notice with NoticeText with a file level DoD Distribution Statement. A GIMI Security XML snippet for

the File is generated on the right(Gray). In this case, the File marking is included in the Security element along with the resourceElement = “true” statement.

GIMI Security XML Generation:

- The GIMI Security XML document is generated using the Security XML snippets from the preparation steps and written into the ItemDataBox of the file level MetaBox.

Table 21 lists example Content Identifiers, using IC-ID syntax, for the primary-media-content.

Table 21: Example Content Identifiers for Figure 25 Example

Content ID Name	Content ID
<Item(1)ContentID>	guide://999999/ee4ea7a9-27a1-522f-9e6d-182618b2609e
<Item(2)ContentID>	guide://999999/e03d1743-5664-586e-916f-9fa731d0315f
<Track(1)ContentID>	guide://999999/60f00d81-52c3-5c3c-9258-5f0e07a63076
<Track(2)ContentID>	guide://999999/8b40ebf8-c278-5d7a-88c6-95aa89b61deb
<TrackPortion(1)ContentID>	guide://999999/d83b7576-4fc0-59e3-9303-9d1a9830c0ed
<TrackPortion(2)ContentID)>	guide://999999/2245b213-9d44-5c8f-8124-645df7a65110

Figure 26 is the GIMI Security XML for the example in Figure 25. Figure 27 is the GIMI Security XML for the same data example, except all the markings are “U” so the XML will validate against both the GIMI and ISM.XML schemas.

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-model href="../ISM/Schematron/ISMCAT/ISMCAT_XML.sch" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<?xml-model href="../ISM/Schematron/ISM/ISM_XML.sch" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<GIMISecurity
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="urn:us:mil:nga:stnd:0076:ism" xmlns:ism="urn:us:gov:ic:ism"
   xmlns:gimi="urn:us:mil:nga:stnd:0076:ism" xmlns:arh="urn:us:gov:ic:arh"
   xsi:schemaLocation="urn:us:mil:nga:stnd:0076:ism NGA.STND.0076_Security-V1.xsd"
   ism:DESVersion="202405" ism:ISMCATCESVersion="202405" GIMISecVer="1">
   <File>
     <arh:Security ism:compliesWith="USGov USIC"
         ism:resourceElement="true" ism:createDate="2006-05-04"
         ism:classification="U7" ism:ownerProducer="USA" />
       <ism:NoticeList ism:classification="U1" ism:ownerProducer="USA">
         <ism:Notice ism:classification="U1" ism:ownerProducer="USA"
           ism:NoticeType="DS">
           <ism:NoticeText ism:classification="U1" ism:ownerProducer="USA">
           Distribution authorized to DoD, IAW 10 U.S.C. §§130 &amp; 455. Release
           authorized to U.S. DoD contractors IAW 48 C.F.R. §252.245-7000. Refer
           other requests to: Headquarters, NGA, ATTN: Disclosure and Release
           Office, Mail Stop S81-IA, 7500 GEOINT Drive, Springfield, VA 22150 or by
           email at NDRO@nga.mil. Destroy IAW DoDD 5200.01 Vol 4. Removal of this
           caveat is prohibited; the caveat must be retained regardless of
           classification. Sharing LIMDIS material beyond USG-authorized users
           requires NGA-originator approval.
           </ism:NoticeText>
         </ism:Notice>
       </ism:NoticeList>
     </arh:Security>
     <Content id="guide://999999/ee4ea7a9-27a1-522f-9e6d-182618b2609e"
         ism:classification="U1" ism:ownerProducer="USA" ism:nonICmarkings="DS">
     <Content id=" guide://999999/e03d1743-5664-586e-916f-9fa731d0315f"
         ism:classification="U3" ism:ownerProducer="USA" />
     <Content id=" guide://999999/60f00d81-52c3-5c3c-9258-5f0e07a63076"
         ism:classification="U1" ism:ownerProducer="USA" />
     <Content id=" guide://999999/8b40ebf8-c278-5d7a-88c6-95aa89b61deb"
         ism:classification="U7" ism:ownerProducer="USA" />
     <Content id=" guide://999999/d83b7576-4fc0-59e3-9303-9d1a9830c0ed"
         ism:classification="U5" ism:ownerProducer="USA" />
     <Content id=" guide://999999/2245b213-9d44-5c8f-8124-645df7a65110"
         ism:classification="U7" ism:ownerProducer="USA" />
   </File>
</GIMISecurity>
<!--All classification marks in this example are for illustrative purposes only,-->
<!--there are no actual classified data contained in this example-->
```

Figure 26: GIMI Security XML for Figure 25 (Will not validated due to fake markings)

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-model href="../ISM/Schematron/ISMCAT/ISMCAT_XML.sch" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<?xml-model href="../ISM/Schematron/ISM/ISM_XML.sch" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<GIMISecurity
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="urn:us:mil:nga:stnd:0076:ism" xmlns:ism="urn:us:gov:ic:ism"
   xmlns:gimi="urn:us:mil:nga:stnd:0076:ism" xmlns:arh="urn:us:gov:ic:arh"
   xsi:schemaLocation="urn:us:mil:nga:stnd:0076:ism NGA.STND.0076_Security-V1.xsd"
   ism:DESVersion="202405" ism:ISMCATCESVersion="202405" GIMISecVer="1">
    <File>
      <arh:Security ism:compliesWith="USGov USIC"
          ism:resourceElement="true" ism:createDate="2006-05-04"
          ism:classification="U" ism:ownerProducer="USA" />
        <ism:NoticeList ism:classification="U" ism:ownerProducer="USA">
          <ism:Notice ism:classification="U" ism:ownerProducer="USA"
            ism:noticeType="DS">
            <ism:NoticeText ism:classification="U" ism:ownerProducer="USA">
            Distribution authorized to DoD, IAW 10 U.S.C. §§130 &amp; 455. Release
            authorized to U.S. DoD contractors IAW 48 C.F.R. §252.245-7000. Refer
            other requests to: Headquarters, NGA, ATTN: Disclosure and Release
            Office, Mail Stop S81-IA, 7500 GEOINT Drive, Springfield, VA 22150 or by
            email at NDRO@nga.mil. Destroy IAW DoDD 5200.01 Vol 4. Removal of this
            caveat is prohibited; the caveat must be retained regardless of
            classification. Sharing LIMDIS material beyond USG-authorized users
            requires NGA-originator approval.
            </ism:NoticeText>
          </ism:Notice>
        </ism:NoticeList>
      </arh:Security>
      <Content id="guide://999999/ee4ea7a9-27a1-522f-9e6d-182618b2609e"
          ism:classification="U" ism:ownerProducer="USA" ism:nonICmarkings="DS">
      <Content id=" guide://999999/e03d1743-5664-586e-916f-9fa731d0315f"
          ism:classification="U" ism:ownerProducer="USA" />
      <Content id=" guide://999999/60f00d81-52c3-5c3c-9258-5f0e07a63076"
          ism:classification="U" ism:ownerProducer="USA" />
      <Content id=" guide://999999/8b40ebf8-c278-5d7a-88c6-95aa89b61deb"
          ism:classification="U" ism:ownerProducer="USA" />
      <Content id=" guide://999999/d83b7576-4fc0-59e3-9303-9d1a9830c0ed"
          ism:classification="U" ism:ownerProducer="USA" />
      <Content id=" guide://999999/2245b213-9d44-5c8f-8124-645df7a65110"
          ism:classification="U" ism:ownerProducer="USA" />
    </File>
</GIMISecurity>
<!--All classification marks in this example are for illustrative purposes only,-->
<!--there are no actual classified data contained in this example-->
```

Figure 27: GIMI Security XML with real Security Markings

Appendix D - Declaration Parameter UUID Generation Methodology (Informative)

To facilitate the generation of consistent, fixed UUIDs for the ISOBMFF declaration parameters, NGA.STND.0076-01 uses a known and repeatable process to create the declaration UUIDs. For example, a 'uuid' extended type item property needs a constant UUID value to signal Item Content IDs.

To create consistent UUIDs, the MISB maintains a Namespace UUID as the basis for creating Version 5 UUIDs MISB uses in their documents and other standards. Appendix C of the MISP supplies the Namespace UUID. When declaring constant UUID parameters for NGA.STND.0076-01, the process uses the MISB Namespace UUID along with the text for each Content ID type. As an example, to generate the 'uuid' extended type value for declaring ItemContentIDs, the NGA.STND.0076-01 authors ran the Version 5 UUID secure hash algorithm (SHA-1) using the MISB Namespace UUID and the text "ItemContentID" as input into the hash algorithm. The resulting UUID is the 'uuid' extended type value for Item Content IDs assigned to declared items. Figure 28 illustrates the process for creating the Item Content ID extended type value.

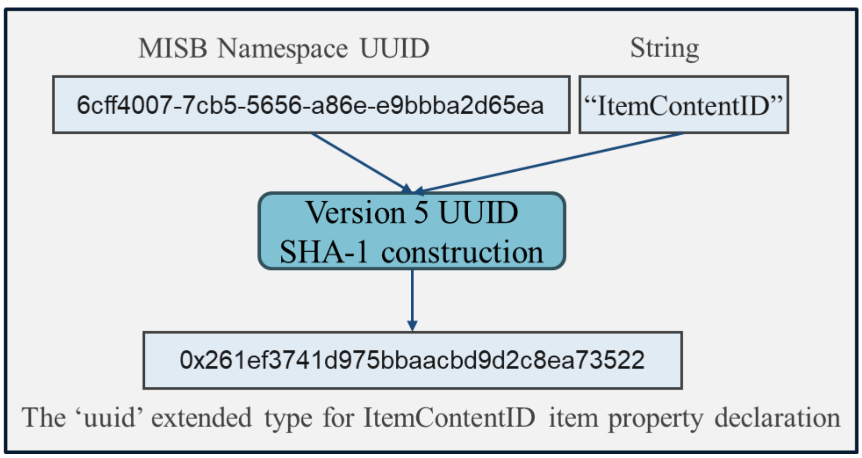


Figure 28: Generating a 'uuid' extended type for declaring ItemContentID item properties

NGA.STND.0076-01 repeats this approach for generating unique UUID values for the declaration of items, tracks, track portions, and components. The text for generating these UUID values, in combination with the MISB Namespace UUID, include 'ItemContentID', 'TrackContentID', TrackPortionContentID', 'ItemComponentContentID' and 'TrackComponentContentID'. Writers store SampleContentIDs as sample auxiliary information, which requires a 4CC for declaration, as opposed to a UUID declaration method.

Appendix E - TAI Timestamps (Normative)

The following text is from an MPEG draft amendment (Amendment 1) on high precision time tagging for ISO/IEC 23001-17. When completed and ratified by MPEG, it will form a normative part of this NGA.STND.0076-01 specification. The current draft amendment text is provided here to facilitate NTB and MISB community review and comment of this NGA.STND.0076-01 specification. This ISO/IEC 23001-17 Amd1 draft text also supports initial implementation and testing until the ISO amendment to 23001-17 is approved.

Information technology — MPEG systems technologies — Part 17: Carriage of uncompressed video and images in ISO base media file format — Amendment 1: High precision timing tagging

Terms and definitions

Add the following terms and definitions:

3.12
TAI
International Atomic Time

high-precision continuous scale of time derived from hundreds of precise atomic clocks from around the world and maintained as closely as possible to the International System (SI) second [33].

Note 1 to entry: Current practice achieves a maximum deviation of approximately one second every 100 million years.

Note 2 to entry: The abbreviated term comes from the French "Temps Atomique International".

3.13
TAI clock
clock capable of synchronizing to a source of TAI time and generating TAI timestamps.

3.14 Receptor clock
A clock located where measurements are made (e.g., local to a sensor) and capable of synchronizing to a source of time from a remote clock.

3.15 Remote clock
A clock capable of transmitting time over significant distances, and usually highly accurate (e.g., GPS system or PTP Grand Master).

3.16 Coordinated Universal Time (UTC)
An international standard for regulating clocks and time, forming a basis for civil time. UTC is based

on TAI but includes irregularly inserted leap second additions or subtractions to account for variation in the earth's rotation.

3.17 Global Positioning System (GPS)
A satellite system providing global positioning, navigation, and timing services. Timing services are based on TAI time.

3.18 Precision Time Protocol (PTP)
The IEEE 1588-2008 standard defines the Precision Time Protocol for synchronizing clocks to a source of TAI time across computer networks.

Note 1 to entry: PTP systems can achieve measurement uncertainties below a microsecond.

3.19 Network Time Protocol (NTP)
RFC 5905 defines the Network Time Protocol for synchronizing clocks to UTC time across computer networks.

Note 1 to entry: Systems using NTP typically achieve measurement uncertainties in the range of milliseconds.

3.20 SI Seconds
The International System of Units (SI) base unit for measuring time.

Clause 8

8 Labeling of Samples and Items

8.1 High Precision Time Tagging

To support applications requiring high resolution and high accuracy time labeling of media items and track samples, this clause provides a labeling method based on International Atomic Time (TAI). A TAI timestamp is a measurement of a TAI clock represented as a discrete integer number of nanoseconds since the TAI epoch of 1958-01-01T00:00:00.0. Additionally, this clause defines TAI timestamp quality status metadata included with each timestamp. Ideally, to be useful and support a broad range of use cases, an ideal timestamp labeling utility benefits from the following attributes:

1. Total ordering: timestamps provide total ordering in which events occur.
2. Relative differencing: timestamps support the ability to compute the difference in time between two events, in SI seconds.
3. Absolute time: timestamps are in a known, universal absolute time reference, enabling correlation of information from sensors and events in different locations.

TAI timestamps are measurements from a TAI clock. TAI clocks report timestamp values relative to the TAI epoch. They provide the ability to search and discover media content based on universal, real-world time. In situations where an adjustment or correction to a timestamp is necessary to improve the accuracy, such as when synchronization to a remote clock is not available during data collection, the ability to modify values using adjustment information post collection is provided, along with a flag to indicate the inclusion of the adjustment.

TAI clocks are “receptor clocks”, typically receiving synchronization data with one or more remote clocks (e.g., GPS system and PTP Grand Master clocks) which are sources of TAI time. Different types of receptor clocks have different levels of quality when synchronizing with remote clocks; therefore, this clause defines metadata for describing the TAI clock, its capabilities, and its state when sampling a timestamp.

Each TAI timestamp is associated with the beginning of a physical measurement, such as the start of exposure for an imaging sensor. For sensors with variable timing, such as rolling shutters, the TAI timestamp is associated with the first pixel(s) initiating an exposure for a frame. The timing associated with remaining pixel exposures is dependent on the sensor architecture and is outside the scope of the metadata described in this section. When extracting data from an image or sample, such as the location of an object in an image, the location data inherits its time from the image frame.

The creation and recording of synthetic data, such as a simulation, is not a measurement of the physical world. The implementation of time labeling, including alignment of synthetic data to real-world data is application dependent and depends on use case needs. In these cases, labeling of the media as synthetic is outside the scope of metadata described in this section.

For media captured in real-time, valid TAI timestamps for the media monotonically increase; however, there are situations where the monotonic nature does not always hold, such as when losing remote to receptor clock synchronization and a resulting discontinuity upon resync occurs. Synchronization status metadata is available to indicate when a receptor clock is not in a synchronous state with the remote clock.

TAI is different from Coordinated Universal Time (UTC) because UTC includes irregularly inserted discontinuities called leap seconds. When converting TAI time to UTC, applications convert the TAI timestamp to date text form and subtract the correct leap seconds value for the date text:

UTC date-text = TAI date-text - leap seconds (based on value at time of measurement)

When converting from timestamps derived from UTC, such as Network Time Protocol (NTP), reverse the computation. To perform a conversion from UTC time to a TAI timestamp, transform the UTC time to a UTC nanosecond timestamp relative to the TAI epoch, then adjust for leap seconds:

TAI timestamp = UTC timestamp + leap seconds (based on value at time of measurement)

A caveat of using UTC-based systems for capturing timestamps is the leap-second adjustment methods employed by various implementations. In practice these methods include stopping the remote clock, slewing the clock at a slow rate, etc. which may introduce errors and compromise the ability to generate accurate time differences on occasions when leap seconds occur. For these reasons, this standard recommends sourcing time from a clock which does not adjust for leap seconds, i.e., TAI based clocks.

Table 6 provides equations to convert times from common time sources (i.e., GPS, PTP, and NTP) to TAI time. The first column of Table 6 is the name of the time source; the second column is the epoch of the time source (for informational use only); the third column provides the equation to convert the time from the time source's epoch offset to TAI's epoch offset. The first step, before applying the equations in the table, is to convert the source time to nanoseconds. The GPS and PTP time standards measure time without using leap seconds, therefore their conversions are straight forward offset additions (based on their respective epoch). Vendors may implement receptor clocks with an epoch other than the base standard; as a result, writers must account for use of the vendor's implemented epoch when performing conversions between time systems.

The NTP standard measures time in UTC, with leap second offsets, so the NTP equation removes leap seconds, where the leap second value comes from a leap second table lookup. The GPS epoch is based on a specific UTC time, but GPS time does not implement leap seconds and tracks with TAI. The POSIX Time standard uses an epoch of Jan 1, 1970 (UTC), which is offset from Jan 1, 1970 (TAI) by 8.000082 seconds. The 82 microseconds are due to non-integer leap second computations used before 1972. As a result, the conversion factor for a clock using the POSIX epoch and tracking with UTC time is 378,691,208,000,082,000 nanoseconds plus the current UTC leap seconds*1,000,000,000. The conversion factor for a clock using the POSIX epoch and tracking with TAI time is 378,691,208,000,082,000 nanoseconds.

Table 6 – Timestamp conversion factors

Time Standard	**Epoch**	**Conversion to TAI timestamp** (nanoseconds)
International Atomic Time (TAI)	1958-01-01T00:00:00.0 (TAI)	N/A
Global Positioning System (GPS)	1980-01-06T00:00:00.0Z (UTC)	TAI timestamp = GPS timestamp + 694,656,019,000,000,000
Precision Time Protocol (PTP)	1970-01-01T00:00:00.0 (TAI)	TAI timestamp = PTP timestamp + 378,691,200,000,000,000
Network Time Protocol (NTP)[1]	1900-01-01T00:00:00.0 (UTC)	TAI timestamp = NTP timestamp – 1,830,297,600,000,000,000 + UTC leap seconds*1,000,000,000 *UTC leap seconds comes from table lookup[2] using the NTP timestamp*

[1]Use of clock sources with time derived from UTC may introduce errors due to implementation of leap second insertion.

[2]The current value for leap seconds can be found on the National Institute of Standards and Technology web site at https://www.nist.gov/pml/time-and-frequency-division/time-realization/leap-seconds

When writing a TAI timestamp to annotate an individual piece of content, the timestamp provides a record of the measurement time of the sample or item and is intended to remain with the sample or item. Subsequent media processing, like editing to change the order of the media samples, may alter the overall monotonic nature of the TAI timestamps within the structure of a track.

Two data structures support TAI timestamps:

1) `TAIClockInfoBox`: describes the TAI clock generating timestamps.

2) `TAITimestampPacket`: contains a timestamp and quality status for the timestamp.

NOTE: The tools defined in Clause 8 can apply to both uncompressed and compressed video samples and image items, as well as items and timed metadata samples. Derived specifications can allow usage of these boxes.

8.1.1 TAI clock information box

8.1.1.1 Definition

Box Type: `'taic'`

Container: `SampleEntry`, `ItemPropertyContainerBox`

Mandatory: Yes, when `TAI_timestamp` values are associated with items or samples

Quantity: One for a given item when associated with a `TAI_timestamp`; one in a sample entry of a track when carrying `TAI_timestamp` values as sample auxiliary information.

NOTE: Writers can share and associate a `TAIClockInfoBox` with multiple items when the source clock for the `TAI_timestamp` values is the same.

The `TAIClockInfoBox` provides metadata about a specific TAI clock referenced by one or more `TAITimestampPackets`. This metadata includes information about the type, quality, and status of the TAI clock. This enables readers to accurately correlate information from multiple cameras or other sensors, for example temporally aligning frames between two disparate cameras with different clock sources. For systems and situations where a TAI clock does not synchronize with a remote clock, the `TAIClockInfoBox` includes a clock type value to indicate this status condition. When a receptor clock is in synchronization with the remote clock the state of the receptor clock is "synchronized". When a remote clock is unavailable the state of the receptor clock is "unsynchronized".

When a `TAI_timestamp` value applies to a declared media item, the `TAIClockInfoBox` is a descriptive item property. When the `TAI_timestamp` values apply to track samples, the writer carries the `TAI_timestamp` values as sample auxiliary information and includes a `TAIClockInfoBox` in the `SampleEntry`. The `TAIClockInfoBox` shall be present in the sample entry when samples are associated with a `TAI_timestamp`, carried as sample auxiliary information, and shall be present as an item property when a `TAI_timestamp` is associated with an item.

The syntax below for the `TAIClockInfoBox` is for a sample entry container and extends FullBox. When used in an ItemPropertyContainerBox, the same syntax applies but the defined box extends ItemFullProperty and is a descriptive property associated with items having a corresponding `TAI_timestamp`.

8.1.1.2 Syntax

```
aligned(8) class TAIClockInfoBox extends FullBox('taic', 0, 0) {{
   unsigned int(64) time_uncertainty;
   unsigned int(32) clock_resolution;
   signed int(32) clock_drift_rate;
```

```
    unsigned int(2) clock_type;
    unsigned int(6) reserved = 0;
}
```

8.1.1.3 Semantics

time_uncertainty is the standard deviation measurement uncertainty for the timestamp generation process, expressed as an unsigned integer number of nanoseconds. Calibration testing of a device determines the time_uncertainty value relative to the true TAI time of a measuring event. When the time_uncertainty is unknown, writers shall set it to the maximum value (0xFFFF FFFF FFFF FFFF).

clock_resolution specifies the resolution of the receptor clock in nanoseconds. For example, a microsecond clock has a clock_resolution of 1000, and a ten-millisecond resolution clock has a clock_resolution of 10,000,000. Receptor clocks with resolution finer than 1 nanosecond set their clock_resolution to zero. While the clock_resolution can differ, TAI_timestamp values are always in the 64-bit nanosecond form.

NOTE: Both the time_uncertainty and clock_resolution affect the statistical accuracy of the TAI_timestamp.

clock_drift_rate represents the rate of divergence of a receptor clock and its source of time when synchronization stops. For example, GPS receptor clock losing communication with remote satellite clock. When a system loses synchronization, the clock_drift_rate provides status information to help determine the level of increasing uncertainty for the TAI_timestamp measurements over time. The clock_drift_rate is in picoseconds per second and comes from either a clock's specification sheet or laboratory testing and measurement. A positive value describes a clock drifting faster than the true time and a negative value describes a clock drifting slower than the true time. When the clock_drift_rate is unknown, writers shall set it to the maximum value (0x7FFF FFFF).

clock_type is an enumeration value indicating the synchronization capability of the clock generating the timestamps. The enumeration values are:

0	Clock type is unknown
1	The clock does not synchronize to an atomic source of absolute TAI time (e.g., unsynchronized CPU clock)
2	The clock can synchronize to an atomic source of absolute TAI time (e.g., synchronized GPS timing card)
3	Reserved – DO NOT USE

NOTE: When writer's generate TAI Timestamps for simulated media, clock information parameters are set to desired values based on simulation needs.

8.1.2 TAI timestamp packet

8.1.2.1 Definition

The TAITimestampPacket supports the carriage of a TAI_timestamp value along with status information. The TAI_timestamp provides temporal context and attaches to samples and items within a file.

When writing the TAI timestamp packet, the nominal state is synchronization of the receptor and remote clocks. However, there are two other states the TAI timestamp packet can indicate during timestamp measurement: synchronization failure and timestamp generation failure.

Synchronization failure means the receptor clock is operating correctly but has lost synchronization with the remote clock, therefore the receptor clock may be measuring timestamps which are drifting away from the actual TAI time. The TAIClockInfoBox provides drift rate estimates to aid in adjusting the time by the reader (if desired).

The timestamp generation failure means the receptor clock is not reporting a measurement, the time value is unknown, and the TAITimestampPacket is written with a "best" estimate of the true TAI timestamp. At a minimum, to achieve correct ordering of successive values, writers shall produce a TAI_timestamp which is one nanosecond greater than the last TAITimestampPacket's TAI_timestamp.

To attach TAI_timestamp values to track samples, writers include TAITimestampPackets as sample auxiliary information. To attach TAI_timestamp values to items, writers implement them as a descriptive item property and associate with the item.

8.1.2.2 Syntax

```
aligned(8) class TAITimestampPacket {
   unsigned int(64) TAI_timestamp;
   unsigned int(1) synchronization_state;
   unsigned int(1) timestamp_generation_failure;
   unsigned int(1) timestamp_is_modified;
   unsigned int(5) reserved = 0;
}
```

8.1.2.3 Semantics

TAI_timestamp is a 64-bit unsigned integer representing the number of nanoseconds since the TAI epoch of 1958-01-01T00:00:00.0. Each nanosecond is one billionth of a Standard International (SI) Second. The TAIClockInfoBox includes the uncertainty and resolution of the clock generating each TAI_timestamp.

synchronization_state indicates the remote and receptor TAI clocks are in synchronization when creating the TAI timestamp. When measuring the TAI_timestamp, if the receptor clock and a remote TAI clock are not in a state of synchronization, the synchronization_state shall be set to 0. When measuring the TAI_timestamp, if the receptor clock and a remote TAI clock are in a state of synchronization, the synchronization_state shall be set to 1. When the companion TAIClockInfoBox's clock_type value equals 0 or 1, the synchronization_state shall equal 0.

timestamp_generation_failure When a TAI receptor clock does not report a TAI_timestamp, the writer sets the timestamp_generation_failure bit to 1. On each successive data sample requiring the creation of a TAI_timestamp estimate, the writer shall generate a value of at least one nanosecond greater than the last prior TAI_timestamp written, providing for consistent total ordering of measured values associated with the given clock. When the timestamp generation is nominal (even if the clock is unsynchronized), the timestamp_generation_failure shall be set to 0.

timestamp_is_modified When modifying an existing TAI_timestamp (e.g., modification to better align its value to the actual measurement time), the timestamp_is_modified parameter signals this condition. When a writer modifies the TAI_timestamp (e.g., correction applied after initial recording) the timestamp_is_modified shall be set to 1. When the TAI_timestamp is the original recorded clock time, even if estimated when the timestamp_generation_failure is set to 1, the timestamp_is_modified shall be set to 0.

8.1.3 Sample TAI timestamps

8.1.3.1 Definition

Aux Info Type: 'stai'

Container: Sample auxiliary information

Mandatory: No

Quantity: Zero, or one per sample

A timestamp_packet is a sample auxiliary information payload for each sample in a track. A writer stores the companion TAIClockInfoBox in the sample entry associated with the sample timestamps.

8.1.3.2 Syntax

```
TAITimestampPacket timestamp_packet;
```

8.1.3.3 Semantics

timestamp_packet is an instance of the TAITimestampPacket class (see Section 8.1.2).

aux_info_type ('saiz' and 'saio' box parameter) This parameter is set to 'stai', which indicates sample TAI_timestamp values.

aux_info_type_parameter ('saiz' and 'saio' box parameter) This parameter is currently unused for the aux_info_type of 'stai'. The aux_info_type_parameter is a reserved value, and all 32-bits of the unsigned word shall be set to 0.

Writers set the remaining 'saiz' and 'saio' box parameters as per ISO/IEC 14496-12 box specification.

8.1.4 Item TAI timestamps

8.1.4.1 Definition

Box Type: `'itai'`

Property Type: Descriptive item property

Container: `ItemPropertyContainerBox`

Mandatory: No

Quantity: Zero, or one per item

The `TAITimestampBox` enables associating a `TAI_timestamp` to a declared item. This box may be present when deriving the item contents from a measured process, such as an image item or a metadata item containing measured sensor data. This box has a required companion `TAIClockInfoBox` property containing TAI clock information. A `TAI_timestamp` and `TAIClockInfoBox` are two separate properties so writers can generate clock information (`TAIClockInfoBox`) once and associated to multiple items, with each having their own `TAI_timestamp`, when the clock information is the same.

Derived image items are comprised of one or more image items, which are inputs to the derivation. In cases where all input items share the same clock source and measured `TAI_timestamp`, the derived item may be associated with the common 'taic' and 'itai' properties. In cases where the input items contain different measurement times or different clock sources, the derived item shall not be associated with a `TAITimestampBox` or `TAIClockInfoBox` and the TAI timing information, if present, shall be associated with the individual input items of the derivation.

8.1.4.2 Syntax

```
aligned(8) class TAITimestampBox extends ItemFullProperty('itai', 0,
0) {
   TAITimestampPacket timestamp_packet;
}
```

8.1.4.3 Semantics

`timestamp_packet` is an instance of the `TAITimestampPacket` class, which is described in Section 8.1.2.