INTERNATIONAL STANDARD

ISO/IEC 23092-3

Second edition
2022-10

Information technology — Genomic information representation —

Part 3: Metadata and application programming interfaces (APIs)

Technologie de l'information — Représentation des informations génomiques —

Partie 3: Métadonnées et interfaces de programmation d'application (API)

Reference number
ISO/IEC 23092-3:2022(E)

Documents
Document Sharing Hub

© ISO/IEC 2022

Full document details: ISO IEC 23092-3-2022

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

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

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 23092-3-2022

Contents

Page

Foreword ... **v**
Introduction ... **vi**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **2**
4 Abbreviated terms ... **2**
5 Conventions ... **2**
5.1 Character encoding ... 2
5.2 Bit Ordering ... 2
5.3 Syntax functions and data types ... 3
5.4 Graphic notations ... 3
6 Information metadata ... **4**
6.1 General ... 4
6.2 Dataset group metadata ... 4
6.3 Reference metadata ... 5
6.4 Dataset metadata ... 5
6.5 Metadata protection ... 7
6.6 Mechanism for extensions of the metadata set ... 7
6.6.1 General ... 7
6.6.2 Example for dataset metadata extensions ... 8
6.6.3 Example for obfuscating labels ... 8
6.6.4 Example for obfuscating sequences ... 8
6.7 Metadata profiles ... 8
6.7.1 General ... 8
6.7.2 Example of metadata profile — Run ... 8
6.7.3 Example of metadata profile — Genomic data commons ... 9
7 Protection metadata ... **10**
7.1 General ... 10
7.2 Encryption of gen_info elements and blocks ... 10
7.2.1 General ... 10
7.2.2 EncryptionParameters carried in dataset group protection ... 10
7.2.3 EncryptionParameters carried in dataset protection ... 12
7.2.4 Key retrieval ... 15
7.2.5 Decryption ... 16
7.3 Privacy rules for the use of the genomic information ... 18
7.3.1 General ... 18
7.3.2 Example of use of privacy rules ... 19
7.4 Digital signature of `gen_info` elements and blocks ... 19
7.4.1 General ... 19
7.4.2 Signatures carried in dataset group protection ... 19
7.4.3 Signatures carried in dataset protection ... 20
7.4.4 Signatures carried in access unit protection ... 21
7.4.5 Signatures carried in descriptor stream protection ... 22
8 Access unit information ... **22**
8.1 General ... 22
8.2 genAuxRecord ... 22
8.3 genAux ... 24
8.4 genTag ... 24
9 Decoding process for metadata ... **25**
9.1 General ... 25
9.2 Initialization of parameters ... 26

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

9.2.1 General ... 26
9.2.2 Properties ... 26
9.2.3 Parameters ... 27
9.2.4 Constants ... 28
9.2.5 Process ... 28
9.3 Macros ... 30
9.4 Decoding process ... 32

10 Application programming interfaces (APIs) ... 40
10.1 General ... 40
10.2 Structure of the API ... 40
10.3 Detailed specification of the API ... 40
10.3.1 Data types ... 40
10.3.2 Return codes ... 41
10.3.3 Metadata fields ... 41
10.3.4 Output structures ... 42
10.3.5 Filters ... 49
10.3.6 Genomic information ... 53
10.3.7 Metadata ... 58
10.3.8 Protection ... 59
10.3.9 Reference ... 61
10.3.10 Statistics ... 62

Annex A (normative) **XML schemas corresponding to metadata information and protection elements ... 65**

Annex B (informative) **XML schemas and XML-based data ... 66**

Annex C (informative) **SAM interoperability ... 76**

Annex D (informative) **Example of key transport ... 83**

Bibliography ... 87

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Foreword

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

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

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

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

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

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

This second edition cancels and replaces the first edition (ISO/IEC 23092-3:2020), which has been technically revised.

The main changes are as follows:

— A few upper/lower case changes in identifiers for consistency.

— Some clarifications on the notation (new subclauses 5.2 and 5.4).

— Minor changes on data structures and parameters.

— A few fixes on references to annexes, tables and subclauses.

— Adding a subclause (7.2.3.3.1.3) to describe the “URI unaligned” for consistency.

— Some corrections in Table 15 including its splitting into 2 tables.

A list of all parts in the ISO/IEC 23092 series can be found on the the ISO and IEC websites.

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

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Introduction

The advent of high-throughput sequencing (HTS) technologies has the potential to boost the adoption of genomic information in everyday practice, ranging from biological research to personalized genomic medicine in the clinic. As a consequence, the volume of generated data has increased dramatically during the last few years, and an even more pronounced growth is expected in the near future.

At the moment, genomic information is mostly exchanged through a variety of data formats, such as FASTA/FASTQ for unaligned sequencing reads and SAM/BAM/CRAM for aligned reads. With respect to such formats, the ISO/IEC 23092 series provides a new solution for the representation and compression of genome sequencing information by:

— specifying an abstract representation of the sequencing data rather than a specific format with its direct implementation;

— being designed at a time point when technologies and use cases are more mature. This permits the addressing of one limitation of the textual SAM format, for which incremental ad-hoc addition of features followed along the years, resulting in an overall redundant and suboptimal format which at the same time results not general and unnecessarily complicated;

— normatively separating free-field user-defined information with no clear semantics from the normative genomic data representation. This allows a fully interoperable and automatic exchange of information between different data producers;

— allowing multiplexing of relevant metadata information with the data since data and metadata are partitioned at different conceptual levels;

— following a strict and supervised development process which has proven successful in the last 30 years in the domain of digital media for the transport format, the file format, the compressed representation and the application program interfaces.

This document provides the enabling technology that will allow the community to create an ecosystem of novel, interoperable solutions in the field of genomic information processing. In particular, it offers:

— consistent, general and properly designed format definitions and data structures to store sequencing and alignment information. A robust framework which can be used as a foundation to implement different compression algorithms;

— speed and flexibility in the selective access to coded data, by means of newly designed data clustering and optimized storage methodologies;

— low latency in data transmission and consequent fast availability at remote locations, based on transmission protocols inspired by real-time application domains;

— built-in privacy and protection of sensitive information, thanks to a flexible framework which allows customizable secured access at all layers of the data hierarchy;

— reliability of the technology and interoperability among tools and systems, owing to the provision of a normative procedure to assess conformance to the standard on an exhaustive dataset;

— support to the implementation of a complete ecosystem of compliant devices and applications, through the availability of a normative reference implementation covering the totality of the specification.

The fundamental structure of the ISO/IEC 23092 series data representation is the *genomic record*. The genomic record is a data structure consisting of either a single sequence read, or a paired sequence read, and its associated sequencing and alignment information; it may contain detailed mapping and alignment data, a single or paired read identifier (read name) and quality values.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Without breaking traditional approaches, the genomic record introduced in the ISO/IEC 23092 series provides a more compact, simpler and manageable data structure grouping all the information related to a single DNA template, from simple sequencing data to sophisticated alignment information.

The genomic record, although it is an appropriate logic data structure for interaction and manipulation of coded information, is not a suitable atomic data structure for compression. To achieve high compression ratios, it is necessary to group genomic records into clusters and to transform the information of the same type into sets of descriptors structured into homogeneous blocks. Furthermore, when dealing with selective data access, the genomic record is a too small unit to allow effective and fast information retrieval.

For these reasons, this document introduces the concept of access unit, which is the fundamental structure for coding and access to information in the compressed domain.

The access unit is the smallest data structure that can be decoded by a decoder compliant with ISO/IEC 23092-2. An access unit is composed of one block for each descriptor used to represent the information of its genomic records; therefore, a block payload is the coded representation of all the data of the same type (i.e. a descriptor) in a cluster.

In addition to clusters of genomic records compressed into access units, reads are further classified in six data classes: five classes are defined according to the result of their alignment against one or more reference sequences; the sixth class contains either reads that could not be mapped or raw sequencing data. The classification of sequence reads into classes enables to develop powerful selective data access. In fact, access units inherit a specific data characterization (e.g. perfect matches in Class P, substitutions in Class M, indels in Class I, half-mapped reads in Class HM) from the genomic records composing them, and thus constitute a data structure capable of providing powerful filtering capability for the efficient support of many different use cases.

Access units are the fundamental, finest grain data structure in terms of content protection and in terms of metadata association. In other words, each access unit can be protected individually and independently. Figure 1 shows how access units, blocks and genomic records relate to each other in the ISO/IEC 23092 series data structure.

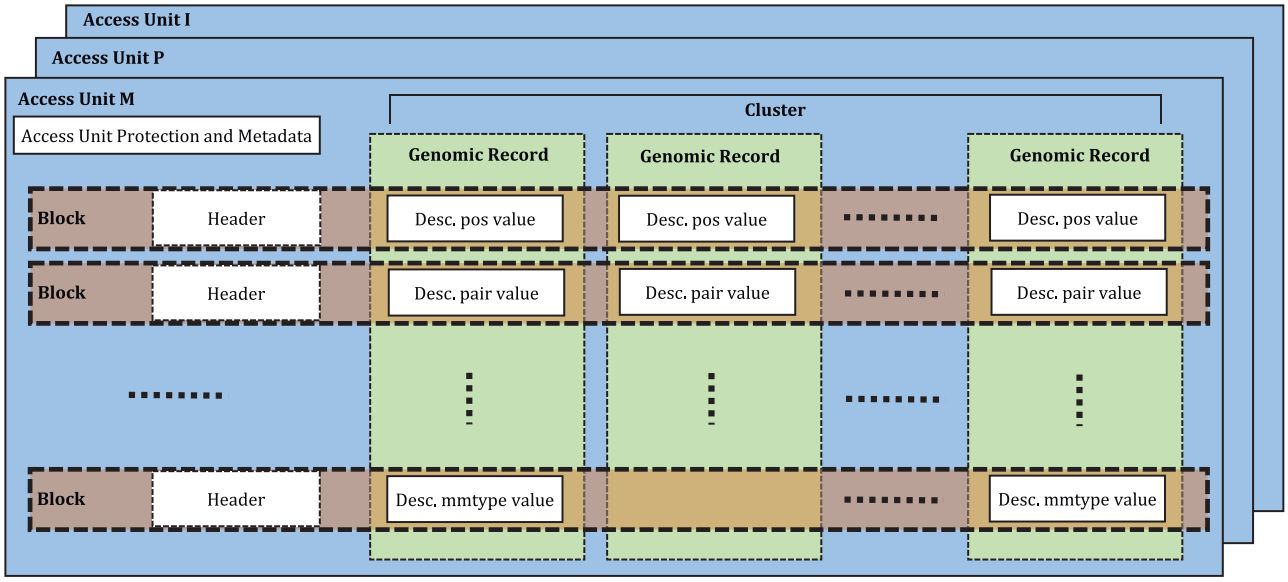

Figure 1 — Access units, blocks and genomic records

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

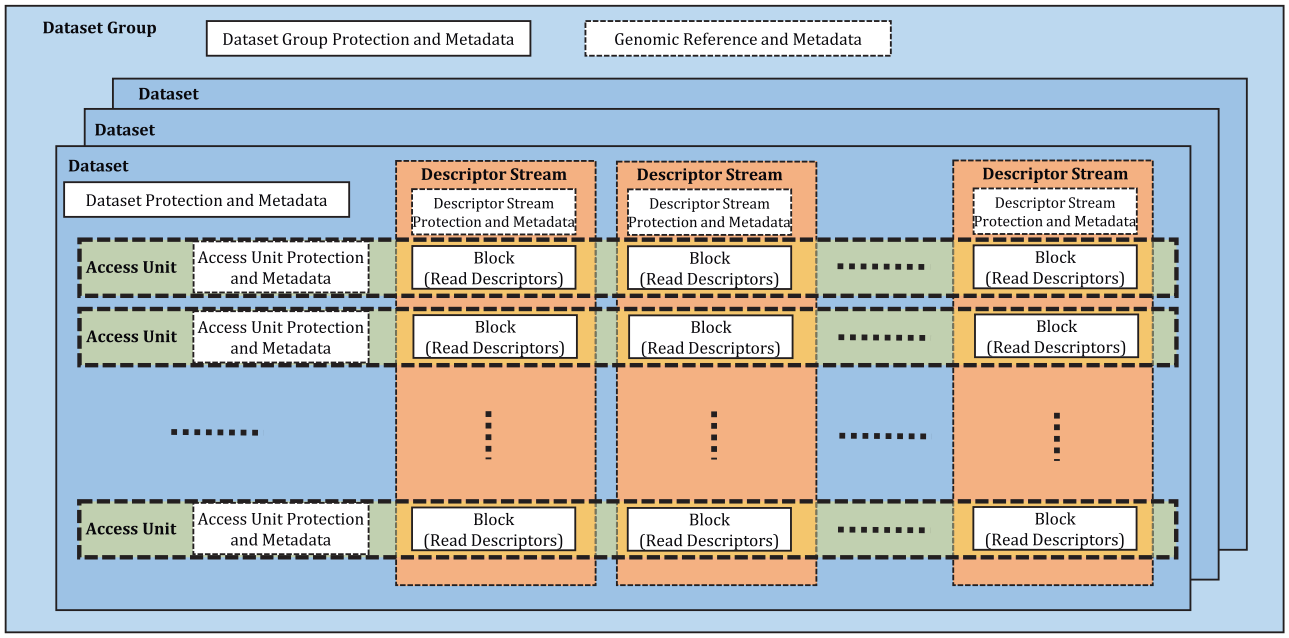

Figure 2 — High-level data structure: datasets and dataset group

A dataset is a coded data structure containing headers and one or more access units. Typical datasets could, for example, contain the complete sequencing of an individual, or a portion of it. Other datasets could contain, for example, a reference genome or a subset of its chromosomes. Datasets are grouped in dataset groups, as shown in Figure 2.

A simplified diagram of the dataset decoding process is shown in Figure 3.

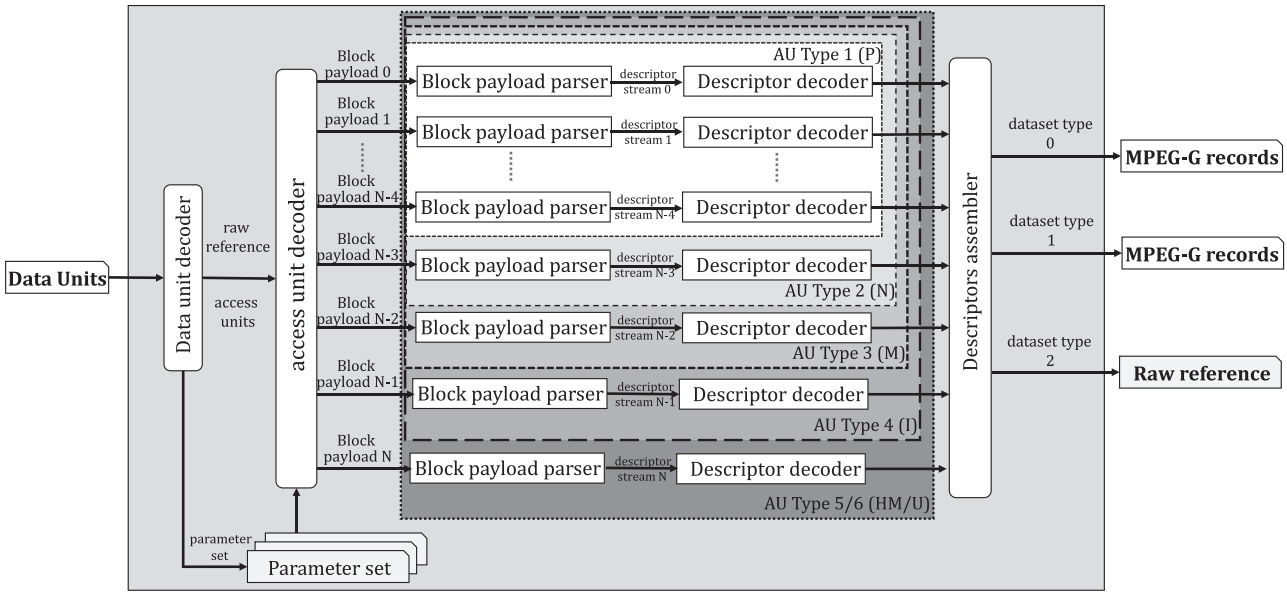

Figure 3 — Decoding process

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Documents
Document Sharing Hub

Full document details: ISO IEC 23092-3-2022

Information technology — Genomic information representation —

Part 3: Metadata and application programming interfaces (APIs)

1 Scope

This document specifies information metadata, auxiliary fields, SAM interoperability, protection metadata and programming interfaces of genomic information. It defines:

— metadata storage and interpretation for the different encapsulation levels as specified in ISO/IEC 23092-1 (in Clause 6);

— protection elements providing confidentiality, integrity and privacy rules at the different encapsulation levels specified in ISO/IEC 23092-1 (in Clause 7);

— how to associate auxiliary fields to encoded reads (in Clause 8);

— mechanisms for backward compatibility with existing SAM content, and exportation to this format (in Annex C);

— interfaces to access genomic information coded in compliance with ISO/IEC 23092-1 and ISO/IEC 23092-2 (in subclause 8.1).

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 23092-1:2020, *Information technology — Genomic information representation — Part 1: Transport and storage of genomic information*

ISO/IEC 23092-2:2019, *Information technology — Genomic information representation — Part 2: Coding of genomic information*

IEEE, 754-2008, IEEE Standard for Floating-Point Arithmetic, August 2008, Available: https://ieeexplore.ieee.org/document/4610935

IETF, RFC 8017, PKCS #1: RSA Cryptography Specifications Version 2.2, November 2016, Available: https://tools.ietf.org/html/rfc8017

IETF, RFC 8018, PKCS #5: Password-Based Cryptography Specification Version 2.1, January 2017, Available: https://tools.ietf.org/html/rfc2898https://tools.ietf.org/html/rfc8018

IETF, RFC 3394, Advanced Encryption Standard (AES) Key Wrap Algorithm, September 2002, Available: https://tools.ietf.org/html/rfc3394

OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0, 2013, Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf

W3C XPath, XML Path Language, Version 1.0, 16 November 1999, Available: https://www.w3.org/TR/xpath-10/

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

3 Terms and definitions

For the purposes of this document, the terms and definitions in ISO/IEC 23092-1 and ISO/IEC 23092-2 and the following apply.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

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

3.1
BAM
compressed binary version of SAM

3.2
dataset group
collection of one or more datasets

Note 1 to entry: Which information is represented varies depending on the genomic information representation.

4 Abbreviated terms

AU	access unit
AUC	access unit contiguity
DSC	descriptor stream contiguity
EBI	European Bioinformatics Institute
EGA	European Genome Archive
ENA	European Nucleotide Archive
LSB	least significant bit
NCBI	National Center for Biotechnology Information
SAM	sequence alignment/map
SRA	sequence read archive
URN	uniform resource name

5 Conventions

5.1 Character encoding

The implementation of the specifications described in this document shall use UTF-8 character encoding.

5.2 Bit Ordering

The bit order of syntax fields in the syntax tables is specified to start with the most significant bit (MSB) and proceed to the least significant bit (LSB).

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

5.3 Syntax functions and data types

The functions presented here are used in the syntactical description. These functions are expressed in terms of the value of a bitstream pointer that indicates the position of the next bit to be read by the decoding process from the bitstream.

byte_aligned( ) is specified as follows:

— If the current position in the bitstream is on a byte boundary, i.e., the next bit in the bitstream is the first bit in a byte, the return value of byte_aligned( ) is equal to TRUE.

— Otherwise, the return value of byte_aligned( ) is equal to FALSE.

read_bits(n) reads the next n bits from the bitstream and advances the bitstream pointer by n bit positions. When n is equal to 0, read_bits(n) is specified to return a value equal to 0 and to not advance the bitstream pointer.

Size(array_name[]) returns the number of elements contained in the array named array_name.

The following data types specify the parsing process of each syntax element:

— f(n): fixed-pattern bit string using n bits written (from left to right) with the left bit first. The parsing process for this data type is specified by the return value of the function read_bits(n).

— st(v): null-terminated string encoded as universal coded character set (UCS) transmission format-8 (UTF-8) characters as specified in ISO/IEC 10646. The parsing process is specified as follows: st(v) begins at a byte-aligned position in the bitstream and reads and returns a series of bytes from the bitstream, beginning at the current position and continuing up to but not including the next byte-aligned byte that is equal to 0x00, and advances the bitstream pointer by (stringLength + 1) * 8 bit positions, where stringLength is equal to the number of bytes returned.

NOTE The st(v) syntax data type is only used in this document when the current position in the bitstream is a byte-aligned position.

— i(n): signed integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this data type is specified by the return value of the function read_bits(n) interpreted as a two's complement integer representation with most significant bit written first.

— u(n): unsigned integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this data type is specified by the return value of the function read_bits(n) interpreted as a binary representation of an unsigned integer with most significant bit written first.

— F32: 32 bit single precision floating-point as specified by IEEE 754-2008. The parsing process is specified as follows: u(1) is used for the sign value, followed by an u(8) used for the exponent value, followed by an u(23) used for the fraction value.

— F64: 64 bit double precision floating-point as specified by IEEE 754-2008. The parsing process is specified as follows: u(1) is used for the sign value, followed by an u(11) used for the exponent value, followed by an u(52) used for the fraction value.

— c(n): sequence of n ASCII characters.

5.4 Graphic notations

The notation `->` (arrow) is used in this document to indicate the access to a member of a data structure.

The notations `|` `|=` are used in this document to indicate the bitwise OR operation and assignment respectively. `a |= b` is equivalent to `a = a | b`.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

The notations `& &=` are used in this document to indicate the bitwise AND operation and assignment respectively. `a &= b` is equivalent to `a = a & b`.

The notation `return_error()` is used in this document to indicate that the decoding process has to stop due to a decoding error which cannot be handled.

The notation `continue` is used in this document within `for` and `while` statements to signal that the process shall continue to the next iteration without executing any further statement in the current iteration.

The notation `*(ptr)` is used in this document to access the data/value in the memory that the pointer `ptr` points to - the contents of the address with that numerical index. The operator * is said to *dereference* the pointer `ptr`.

The notation `strncmp(str1, str2, n)` is used in this document to indicate if first n characters of two strings `str1` and `str2` match with each other. If all `n` characters match, then it returns 0, otherwise 1.

6 Information metadata

6.1 General

This clause defines a minimum core set of metadata elements, which users and applications can then extend by including extra information elements. Metadata sets are specified for dataset groups, datasets and references, as specified in ISO/IEC 23092-1. The structure of these metadata sets and their elements is specified using XML v1.1.

Extensions to (i.e., new elements for) the metadata sets specified in this clause are represented with an identifier of the extension type in the form of a URI, a value and a pointer to a resource documenting the semantics of the extension type.

Metadata profiles are specific subsets of metadata sets specified using mechanisms provided in this document. A metadata profile specified in this document may correspond to well-known metadata sets specified or used out of the ISO/IEC 23092 series, such as those in ENA or EGA[11] and NCBI specifications,[10] as examples. This allows easy interoperability with already existing systems. A metadata profile includes a subset of (or all) core elements described in subclauses 6.2 and 6.4, and a set of new elements specified with the extension mechanism specified in subclause 6.6.

The rest of clauses specify dataset group metadata (subclause 6.2), reference metadata (subclause 6.3), dataset metadata (subclause 6.4), extensions (subclause 6.6) and profiles (subclause 6.7).

6.2 Dataset group metadata

Compressed dataset group metadata are stored within the `DG_metadata_value` element of the *DG_metadata* box (with key `dgmd`), as specified in ISO/IEC 23092-1. The decoding process of `DG_metadata_value` is specified in Clause 9. The output of the decoding process is an XML document, where the root node is `DatasetGroupMetadata`. Annex A.1 provides the XML schema for a decoded dataset group metadata.

As previously introduced in subclause 6.1, an extensions type is the combination of three elements: the value, the identifier of the extension, and a link to a resource documenting the interpretation of the extension. In the XML schema, this is translated as an element with three child elements: the `Type` element (of type URI), the `Documentation` (of type URI) and the value which is represented as the element taking the place of the element `any` in the schema. Additionally, for extensions belonging to the dataset group, the Boolean element `Inheritable` (as specified in Annex A.1) of the extension element indicates if the extension is only relevant to the dataset group, or if the dataset also inherits it. The resource documentation can be human readable, and the extensions parsing is not required.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

6.3 Reference metadata

Compressed reference metadata are stored within the reference metadata box, as specified in ISO/IEC 23092-1, in the `reference_metadata_value` field. Clause 9 specifies the decoding process of `reference_metadata_value`. The output of the decoding process is an XML document, with a root element `ReferenceMetadata`. Annex A provides the related XML schema. Table 1 specifies the semantics of the fields.

Table 1 — Semantics of reference sequence's fields

Tag name	Description
length	Length in base pairs[a] of the sequence
alternative_locus_location	The sequence is an alternative locus from an unknown region. A child element `chromosome_name` identifies on which chromosome the sequence has an alternative locus. If present, a child element `position` indicates the start and end position of the alternative locus.
alternative_sequence_name	List of alternative names
genome_assembly_identifier	Genome assembly identifier
description	Human readable textual description
species	Name of the species
URI	URI of the sequence
[a] In this document, "base" or "base pair" is used as a synonym for "nucleotide".	

6.4 Dataset metadata

Compressed dataset metadata are stored within the `DT_metadata_value` field of the `DT_metadata` box (marked as `dtmd`), as specified in ISO/IEC 23092-1. Clause 9 specifies the decoding process of `DT_metadata_value`. The output of the decoding process is an XML document with an element `DatasetMetadata` as root. Annex A provides the XML schema for dataset metadata. A dataset metadata element overwrites the corresponding element whose values differ from the one indicated at the dataset group level (i.e., the new value in the dataset is a specialization of the value at the dataset group level).

Table 2 defines the process to obtain the dataset metadata with inherited elements. In this table, the following notations are used:

— .has(): the function returns true if the element has a child element with an unqualified name equal to the parameter given, and false otherwise

— .get(): the function returns the content of the child element with an unqualified name equal to the parameter given, as an array of characters

— .getElement(): the function returns the content of the child element with an unqualified name equal to the parameter given

— .getByIndex(): the function returns the content of the ith child element with an unqualified name equal to the first parameter given and i equal to the second parameter given, as an array of characters

— .getEncoding(): the function returns the content of the element as an array of characters

— .set(): the function sets the content of the child element with an unqualified name equal to the first parameter given, to the array of characters given as the second parameter

— .add(): the function creates a new child element with an unqualified name equal to the first parameter given, and a content equal to the second parameter. The created element is appended to the content of the current element.

— .getNumber(): the function returns the number of child elements with an unqualified name equal to the parameter given.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Table 2 — Decoding process of dataset metadata

```
datasetMetadataWithInheritance = datasetMetadata
if (!datasetMetadata.has("Type")){
  datasetMetadataWithInheritance.set(
    "Type",
    datasetGroupMetadata.get("Type")
  )
}
if (!datasetMetadata.has("Abstract")){
  datasetMetadataWithInheritance.set(
    "Abstract",
    datasetGroupMetadata.get("Abstract")
  )
}
if (!datasetMetadata.has("ProjectCentre")){
  datasetMetadataWithInheritance.set(
    "ProjectCentre",
    datasetGroupMetadata.get("projectCentre")
  )
}
if (!datasetMetadata.has("Description")){
  datasetMetadataWithInheritance.set(
    "Description",
    datasetGroupMetadata.get("Description")
  )
}
if(!datasetMetadata.has("Samples")){
  datasetMetadataWithInheritance.set(
    "Samples",
    datasetGroupMetadata.get("Samples")
  )
}
extensions = datasetGroupMetadata.getElement("Extensions")
extensionsDataset = datasetMetadata.getElement("Extensions")
for(i=0; i < extensions.getNumber("Extension"); i++){
  extension = extensions.getByIndex("Extension", i)
  typeExtension = extension.get("Type")
  if (extension.get("Inheritable") == "true"){
    continue
  }
  found = false
  for(j=0; j< extensionsDataset.getNumber("Extension"); j++){
    extensionDataset = extensionsDataset.getByIndex("Extension", j)
    typeExtensionDataset = extensionDataset.get("Type")
    if( typeExtension == typeExtensionDataset){
      found = true
      break
    }
  }
  if(!found){
    extensionsDataset.add("Extension", extension.getEncoding())
```


© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Table 2 *(continued)*

```
    }
}
```

After executing the inheritance process, the extensions are ordered in the alphabetical order of their Type element.

For example, one can have datasets for patients A, B and C; therefore the dataset group metadata includes a list of samples representing A, B and C. The datasets then provide only one sample description (respectively of A, B or C). The base set of elements in the dataset metadata is the same as for the dataset group, but the elements are not mandatory (so there is no need to repeat them), since by default their values are considered equal to the values indicated in the dataset group. This is always the case for the values belonging to the core set, or by default for the extensions except for those cases that have the inheritance parameter (element named `Inheritable` in the definition of the extension type in Annex A) set to false.

As in the case of the dataset group metadata, Annex A provides the resulting schema.

Also as in the case of the dataset group, the extension mechanism is available to include new attributes where necessary. See subclause 6.6.2 on extensions for an example in the case of dataset metadata.

6.5 Metadata protection

The schemas defined in Annex A include choices to either provide certain values as plaintext or encrypted content. The encrypted content shall be such that after decryption, a metadata element is obtained which is valid according to the schemas, but which does not contain any encrypted information. The mechanism to transmit the knowledge of the keys shall be established through another channel.

No call to functions specified in 10.3.7 may return encrypted content.

The schemas defined in Annex A allow for the signature of the metadata or parts thereof. No call to functions specified in 10.3.7 may return content for which the signature could not be verified.

6.6 Mechanism for extensions of the metadata set

6.6.1 General

This subclause provides a mechanism for adding new elements to the different core metadata sets (dataset group and dataset levels).

An extended element consists of:

— an information type identifier (provided in the form of a URI),

— a value,

— documentation (provided in the form of a URI).

In the case of extensions at the dataset group level, a third value, the inheritance flag of type Boolean, is optionally present. If not present, the default value is True. If the flag is set to True, the value of the extension is inherited by the datasets belonging to the dataset group. If the flag is set to False, the value only applies to the dataset group.

Annex A defines the extension schema. Using extensions, the core metadata sets can be adapted to multiple use cases. This document defines profiles (see subclause 6.7), which rely on well-known extensions, defined in this document and for which the URI pointer is known. To be compliant with a profile specified in this document, a tool shall implement the list of extensions included in the profile.

At the end of the decoding process, the extensions have to be ordered in growing lexicographic order based on the information type identifier.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

This subclause presents examples of datasets using the concept of label specified in ISO/IEC 23092-1.

6.6.2 Example for dataset metadata extensions

ISO/IEC 23092-1 introduces the concept of dataset `label`, which supports the identification of one or more genomic intervals and the related data classes with a single identifier. Labels are not conceived to provide detailed documentation of the identified genomic regions. Table 3 shows how to associate an ontology term to a label.

Table 3 — `Sample` metadata extended with a `label` linked to an ontology

Field name	Field type	Mandatory
Label name	st(v)	Yes
Ontology term	URN	Yes

6.6.3 Example for obfuscating labels

In case the label name is considered sensitive information to be protected, this document specifies ways to obfuscate label names. One solution is to provide an extension to the metadata, which provides a translation between the obfuscated name and the real name. Using XML encryption, the access to the sensitive information can be restricted. Annex B.5 provides a schema for the extension supporting multiple labels obfuscation where a transmitted label name is translated to a non-obfuscated one.

6.6.4 Example for obfuscating sequences

In order to maintain privacy, it can be necessary to hide which genomic region an access unit maps to. In order to hide this piece of information, it is necessary to obfuscate the sequences. One solution is then to provide an extension to the metadata, which provides a translation between the obfuscated sequence name and the real name. Access to the sensitive information can then be restricted using XML encryption. Annex B.5 provides a schema for the extension supporting multiple sequence obfuscation where a transmitted sequence name is translated to a non-obfuscated one, and the real sequence start position.

6.7 Metadata profiles

6.7.1 General

Profiles are specific metadata extension sets. Subclause 6.6 provides the mechanism to specify them. Subclause 6.7.2 provides a formalized profile.

A profile corresponds to a well-known metadata set. Profiles ares specified in this document, such as the one defined to support the Run sets of the SRA (Sequence Read Archive) schema version 1,5.[9] Additional profiles may be specified following the mechanisms defined in this document.

A profile includes a subset of (or all) the core elements described in this document, and a set of new elements specified with the extension mechanism (see subclause 6.6).

The metadata schemas in Annex A define the attribute `profile` in the dataset group and dataset metadata XML element, to define the profile being used. The profile is identified with a URI. In case no profile is active, the attribute is not used.

6.7.2 Example of metadata profile — Run

The ISO/IEC 23092 series dataset metadata shares characteristics with both the concepts of run and analysis as used by EGA.[3] This series profile provides interoperability with the existing metadata schemas, such that an ISO/IEC 23092 series dataset is interoperable with an EGA run element.

This profile is identified with the URI: "`urn:mpeg:mpeg-g:metadata:profile:ega:run`".

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

To be compliant with the profile, the following extension shall be present at the dataset group metadata level

— One extension with an element `StudyExtension` with a type URI "`urn:mpeg:mpeg-g:metadata:extension:ega:Study`".

To be compliant with the profile, the following extensions shall be present at either the dataset group metadata level (shall be marked as inheritable), or the dataset metadata level:

— One extension with an element `ExperimentExtension` with a type URI "`urn:mpeg:mpeg-g:metadata:extension:ega:Experiment`".

— One extension with an element `RunExtension` with a type URI "`urn:mpeg:mpeg-g:metadata:extension:ega:Run`".

Each Sample metadata shall have one extension with an element SampleExtension with a type URI "`urn:mpeg:mpeg-g:metadata:extension:ega:Sample`"

Furthermore, to be compliant, the following restrictions have to be observed:

— description element in the Dataset Group metadata schema is mandatory;

— the TaxonID shall be present and equal to 9096 (human).

The contents of `SRA.common.xsd` and `SRA.run.xsd` can be found in Reference [9].

This profile extends the MPEG-G Dataset core metadata to match the Run schema used by EGA. The file element from SRA's run metadata schema is not needed in MPEG-G because the metadata is placed within the element it refers to.

6.7.3 Example of metadata profile — Genomic data commons

The genomic data commons (GDC) classifies the information in projects, which contain the information of multiple cases, represented as a collection of files of different types.

The ISO/IEC 23092 series dataset group shares characteristics with the GDC projects. From all types of information stored in the GDC repository, the ISO/IEC 23092 series only supports raw sequencing analysis. Therefore, the metadata profile designed to allow interoperability between GDC and the ISO/IEC 23092 series only supports that data element from the core metadata.

If the GDC metadata profile is active, the dataset group is considered equivalent to the GDC's project.

This profile is identified with the URI: "`urn:mpeg:mpeg-g:metadata:profile:gdc`".

When this profile is active, the following extensions are required:

— One extension of type ProjectsExtensionType as specified in Annex A, identified with a type URI "`urn:mpeg:mpeg-g:metadata:profile:gdc:ProjectsExtensionType`", listed in the dataset group metadata extensions specified in ISO/IEC 23092-1.

— One extension of type CaseExtensionType as specified in Annex A, identified with a type URI "`urn:mpeg:mpeg-g:metadata:profile:gdc:CaseExtensionType`", listed in the dataset metadata extensions specified in ISO/IEC 23092-1. Alternatively, the extension can be provided as an inheritable extension in the dataset group metadata extensions list.

— One extension of type ClinicalExtensionType as specified in Annex A, identified with a type URI "`urn:mpeg:mpeg-g:metadata:profile:gdc:ClinicalExtensionType`", listed in the dataset metadata extensions specified in ISO/IEC 23092-1. Alternatively, the extension can be provided as an inheritable extension in the dataset group metadata extensions list.

To be compliant with profile "`urn:mpeg:mpeg-g:metadata:profile:gdc`" all UUID elements shall be present, either in plaintext or encrypted form.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

7 Protection metadata

7.1 General

ISO/IEC 23092-1 defines `gen_info` structures supporting the protection of the genomic information at different layers of the hierarchy, namely the elements `dgpr`, `dtpr`, `aupr`, and `dspr`, protecting the dataset group, dataset, descriptor stream and access unit respectively as shown in Table 4. These `gen_info` structures provide information to implement data protection in terms of confidentiality (encryption), integrity verification (digital signature) and privacy rules enforcement. Such information is coded in the form of XML documents as specified in Table 4.

This clause is divided into the three main aspects of protection: encryption, privacy rules and integrity. The protection of specific elements in the metadata is specified in subclause 6.5. If the contents (`gen_info` boxes or blocks) are protected (either signed and/or encrypted), they shall be processed in the following order: verify signature (if signed), perform decryption (if encrypted) and decompress (if `gen_info` boxes).

Table 4 — Content of the protection boxes specified in ISO/IEC 23092-1

Content of the `gen_info` structure specified in ISO/IEC 23092-1	Key	Value
`DG_protection_value`	dgpr	coded representation of the XML document specified in A.3. The decoding process to retrieve the XML document from the coded representation is specified in Clause 9.
`DT_protection_value`	dtpr	coded representation of the XML document specified in A.4. The decoding process to retrieve the XML document from the coded representation is specified in Clause 9.
`AU_protection_value`	aupr	coded representation of the XML document specified in A.5. The decoding process to retrieve the XML document from the coded representation is specified in Clause 9.
`DS_protection_value`	dspr	coded representation of the XML document specified in A.6. The decoding process to retrieve the XML document from the coded representation is specified in Clause 9.

7.2 Encryption of gen_info elements and blocks

7.2.1 General

This subclause specifies how the encryption parameters convey the protection information of some of the `gen_info` elements specified in ISO/IEC 23092-1. A protection `gen_info` structure supporting coded data protection conveys the information on encryption applied to some of its sibling boxes and other `gen_info` structures supporting the protection of a layer immediately below. This information is represented with a list of `EncryptionParameters` elements (of type `EncryptionParametersType`) as specified in Annex A (for dataset group and dataset).

7.2.2 EncryptionParameters carried in dataset group protection

Any number of `EncryptionParameters` elements can be present in the dataset group protection box and shall use one of the following URIs for the `encryptedLocations` attribute, where each URI refers to a specific `gen_info` box as specified below (text within curly brackets, including the curly brackets themselves, shall be replaced by some alphanumeric sequence compliant with the description of each entry).

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

7.2.2.1 URIs for `gen_info` boxes

7.2.2.1.1 URIs structure for `gen_info` boxes

7.2.2.1.1.1 General

The following URIs shall correspond to specific `gen_info` boxes:

```
refmeta/{id}
metadata
dataset/{id}/protection
```

7.2.2.1.1.2 URI `refmeta/{id}`

This URI shall correspond to all but the two first bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the `gen_info` structure with key "`rfmd`" (i.e. reference metadata defined in ISO/IEC 23092-1:2020, subclause 6.5.2.4, the two bytes corresponding to values `dataset_group_ID` and `reference_ID`), identified by the field `reference_ID`, and which is stored in the same dataset group as the current dataset group protection. Table 7 describes the type and semantics of the parameter of the URI.

Table 5 — Type and semantics of parameters of the URI `refmeta/{id}`

Parameter	Type	Semantics
`id`	unsigned integer	ID of the reference metadata to which the URI correspond. It corresponds to `reference_ID` defined in ISO/IEC 23092-1:2020, Table 14

7.2.2.1.1.3 URI `metadata`

This URI shall correspond to the bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info box with key "`dgmd`" (i.e. dataset group metadata defined in ISO/IEC 23092-1:2020, subclause 6.5.2.6), stored in the same dataset group as the current dataset group protection.

7.2.2.1.1.4 URI `dataset/{id}/protection`

This URI shall correspond to the bytes of the the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.5.4.5) of the gen_info box with key "`dtpr`" (i.e. dataset protection defined in ISO/IEC 23092-1:2020, subclause 6.5.3.4), stored in the dataset identified by `dataset_ID`, and which belongs to the current dataset group. Table 10 describes the type and semantics of the parameter of the URI.

Table 6 — Type and semantics of parameters of the URI `dataset/{id}/protection`

Parameter	Type	Semantics
`id`	unsigned integer	Identifier of the dataset to which the dataset protection to be retrieved belongs. It corresponds to the field `dataset_ID` defined in ISO/IEC 23092-1:2020, Table 20.

7.2.2.1.2 URIs processing

The `EncryptionParameters` elements with either one of the URIs listed above shall adhere to the following rules:

— the `IV` element shall be present;

— the `TAG` element shall be present if the cipher is used in GCM mode (as described in Table 14);

— the `configurationID` attribute shall not be present.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

The ciphertext of the gen_info boxes identified by the URIs specified in this subclause shall be decrypted using the process specified in 7.2.5.1.

7.2.3 EncryptionParameters carried in dataset protection

7.2.3.1 General

Any number of `EncryptionParameters` elements can be present in the dataset protection box and shall use one of the URIs defined in 7.2.3.2 or in 7.2.3.3 for the `encryptedLocations` attribute, where each URI refers either to a specific `gen_info` box or to a collection of access units as specified below. These URIs can contain some placeholders identified by a text enclosed in curly brackets. These placeholders (including curly brackets) shall be replaced by an alphanumeric value as specified below.

7.2.3.2 URIs for `gen_info` boxes

7.2.3.2.1 URIs structure for `gen_info` boxes

7.2.3.2.1.1 General

The following URIs shall correspond to specific `gen_info` boxes:

```
pars/{id}
metadata
descstream/{classId}/{descriptorId}/protection
au/{auType}/{sequenceName}/{accessUnitId}/protection
au/unaligned/{accessUnitId}/protection
```

7.2.3.2.1.2 URI `pars/{id}`

This URI shall correspond to the bytes of the element "`encoding_parameters()`" (defined in ISO/IEC 23092-1:2020, Table 24) in the gen_info with key "`pars`" (i.e. dataset parameter set defined in ISO/IEC 23092-1:2020, subclause 6.5.3.5), stored in the same dataset as the current Dataset Protection, identified by `descriptorId`. Table 7 describes the type and semantics of the parameter of the URI.

Table 7 — Type and semantics of parameters of the URI `pars/{id}`

Parameter	Type	Semantics
`parameterId`	unsigned integer	ID of the `encoding_parameters` to which the URI correspond. It corresponds to `parameter_set_ID` defined in ISO/IEC 23092-1:2020, Table 24.

7.2.3.2.1.3 URI `metadata`

This URI shall correspond to the bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info box with key "`dtmd`" (i.e. dataset metadata defined in ISO/IEC 23092-1:2020, subclause 6.5.3.3), stored in the same dataset as the current dataset protection.

7.2.3.2.1.4 URI `descstream/{classId}/{descriptorId}/protection`

This URI shall correspond to the bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info box with key "`dspr`" (i.e. descriptor stream protection defined in ISO/IEC 23092-1:2020, subclause 6.6.3.3), stored in the same dataset as the current Dataset Protection, identified by `classId` and `descriptorId`. Table 8 describes the type and semantics of the parameters of the URI.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Table 8 — Type and semantics of parameters of the URI `descstream/{classId}/{descriptorId}/protection`

Parameter	Type	Semantics
`classId`	unsigned integer	ID of the class to which the URI correspond. It corresponds to the field `class_ID` defined in ISO/IEC 23092-1:2020, Table 33.
`descriptorId`	unsigned integer	ID of the descriptor to which the URI correspond. It corresponds to the field `descriptor_ID` defined in ISO/IEC 23092-1:2020, Table 33.

7.2.3.2.1.5 URI `au/{auType}/{sequenceName}/{accessUnitId}/protection`

This URI shall correspond to the bytes of the the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.5.4.5) of the gen_info box with key "`aupr`" (i.e. access unit protection) stored in the access unit, stored in the same dataset as the current Dataset Protection, identified by `auType`, `sequenceName` and `accessUnitId`. Table 13 describes the type and semantics of the parameters of the URI.

Table 9 — Type and semantics of parameters of the URI `au/{auType}/{sequenceName}/{accessUnitId}/protection`

Parameter	Type	Semantics
`auType`	unsigned integer	Identifier of the class to which the URI correspond. It corresponds to the field `class_ID` defined in ISO/IEC 23092-1:2020, Table 26. The value shall be greater or equal than 1 and strictly less than 6.
`sequenceName`	st(v)	Textual identifier of the reference sequence the access unit to which the access unit protection to be retrieved is mapped to. It corresponds to `sequence_name` specified in ISO/IEC 23092-1:2020, Table 11.
`accessUnitId`	unsigned integer	Identifier of the access unit to which the access unit protection to be retrieved belongs. It corresponds to the field `access_unit_ID` defined in ISO/IEC 23092-1:2020, Table 26.

7.2.3.2.1.6 URI `au/unaligned/{accessUnitId}/protection`

This URI shall correspond to the bytes of the the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.5.4.5) of the gen_info box with key "`aupr`" (i.e. access unit protection) stored in the access unit, stored in the same dataset as the current Dataset Protection, for which the associated `AU_type` defined in ISO/IEC 23092-1:2020, Table 26 is equal to the parameter `accessUnitId` provided. Table 10 describes the type and semantics of the parameter of the URI.

Table 10 — Type and semantics of parameters of the URI `au/unaligned/{accessUnitId}/protection`

Parameter	Type	Semantics
`accessUnitId`	unsigned integer	Identifier of the access unit to which the access unit protection to be retrieved belongs. It corresponds to the field `access_unit_ID` defined in ISO/IEC 23092-1:2020, Table 26.

7.2.3.2.2 URIs processing

The `EncryptionParameters` elements with either one of the URIs listed above shall adhere to the following rules:

— the `IV` element shall be present;

— the `TAG` element shall be present if the cipher is used in GCM mode (as described in Table 14);

— the `configurationID` attribute shall not be present.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

The ciphertext of the gen_info boxes identified by the URIs specified in this subclause shall be decrypted using the process specified in 7.2.5.1.

7.2.3.3 URIs for collection of access units

7.2.3.3.1 URIs structure for access units

7.2.3.3.1.1 General

The following URIs shall correspond to collections of access units:

```
genomic_region/{auType}/{sequenceName}/{start}/{end}
unaligned
unaligned/{signature}
```

7.2.3.3.1.2 URI genomic_region/{auType}/{sequenceName}/{start}/{end}

This URI shall correspond to access units containing at least one read mapping between `start` and `end` (included) on the reference sequence identified by `sequenceName`. Table 11 describes the type and semantics of the parameters of the URI.

Table 11 — Type and semantics of parameters of the URI `genomic_region/{auType}/{sequenceName}/{start}/{end}`

Parameter	Type	Semantics
`auType`	unsigned integer equal or greater than 0 and less than 6	Type of access units to be retrieved. When set to 0, all access units having `AU_type` (as specified in ISO/IEC 23092-1:2020, Table 26) less than 6 shall be retrieved.
`sequenceName`	st(v)	Textual identifier of the reference sequence the access units to be retrieved are mapped to. It corresponds to `sequence_name` specified in ISO/IEC 23092-1:2020, Table 11.
`start`	unsigned integer	Left-most position of the interval to be considered for access units retrieval. It shall be less than or equal to `end`.
`end`	unsigned integer	Right-most position of the interval to be considered for access units retrieval. It shall be greater than or equal to `start`.

7.2.3.3.1.3 URI unaligned

This URI shall correspond to access units having AU_type (as specified in ISO/IEC 23092-1:2020, Table 26) equal to 6.

7.2.3.3.1.4 URI unaligned/{signature}

This URI shall correspond to access units having `AU_type` (as specified in ISO/IEC 23092-1:2020, Table 26) equal to 6 and at least one of the `U_cluster_signature` fields (specified both in ISO/IEC 23092-1:2020, Table 26 and in Table 31) matching the value of `signature`. Table 12 describes the type and semantics of the parameter of the URI.

Table 12 — Type and semantics of parameters of the URI `unaligned/{signature}`

Parameter	Type	Semantics
`signature`	st(v)	At least one of the `U_cluster_signature` fields (specified both in ISO/IEC 23092-1:2020, Table 26 and in Table 31) of the retrieved access units shall match this value.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

7.2.3.3.1.5 URI label_selection/{labelId}

This URI shall correspond to access units containing at least one read mapping in at least one of the genomic regions listed in the label identified by `labelId` as specified in ISO/IEC 23092-1:2020, subclause 6.5.2.5.4. Table 13 describes the type and semantics of the parameter of the URI.

Table 13 — Type and semantics of parameters of the URI `label_selection/{labelId}`

Parameter	Type	Semantics
`labelId`	unsigned integer	Identifier of the label to be considered for retrieval. It corresponds to `label_ID` specified in ISO/IEC 23092-1:2020, subclause 6.5.2.5.4.

7.2.3.3.2 URIs processing

For all access units belonging to the collection resolved by any of the previous URIs, the bytes that shall be decrypted are:

— If the access unit contains a gen_info with key "auin", the bytes corresponding to the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the `gen_info` with key "`auin`" (the access unit information defined in ISO/IEC 23092-1:2020, subclause 6.5.4.4)

— In AUC mode:

  — the bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the access unit, discarding all bytes prior to and including the last byte of the access unit protection

  In DSC mode:

  — all bytes corresponding to the associated `blocks` structures (as specified in ISO/IEC 23092-1). concatenated, in growing order of `descriptor_ID` defined in ISO/IEC 23092-1:2020, Table 20.

Both decryptions shall be performed independently, using their own IVs and TAGs (present only with GCM mode) signaled in the `AccessUnitEncryptionParameters` element present in the associated AU protection box (specified in A.5).

The `EncryptionParameters` elements with either of the URIs listed above shall adhere to the following rules:

— the `IV` element shall not be present;

— the `TAG` element shall not be present;

— the `configurationID` attribute shall be present. If an access unit belongs to one of the collection resolved by the URIs listed above, the `AccessUnitEncryptionParameters` element of the access unit protection shall contain one or more `wrappedKey` elements each referring to the `configurationID` of one of the `EncryptionParameters` element resolving to the access unit. The key associated to each of these `EncryptionParameters` shall allow to unwrap the corresponding `wrappedKey`.

7.2.4 Key retrieval

7.2.4.1 General

Across all DatasetGroupProtection, DatasetProtection, AccessUnitProtection or DescriptorStreamProtection elements present in one file compliant to the ISO/IEC 23092 series, multiple `KeyTransportation` elements defining a key with the same name may be present, and all of them shall encapsulate the same value, albeit with diferent parameters. The name of the key is encapsulated in the `key_name` element of the `KeyTransportation` element. A `KeyTransportation` element can transport the key in three diferent ways, as described in the three following subclauses. Additionally, a key can be referenced by a name not appearing in the `key_name` element of any `KeyTransportation` element; in that case, the key shall be transported through another channel. Some examples are provided in Annex D.

© ISO/IEC 2022 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

7.2.4.2 Key derivation

The information is provided in a `KeyDerivation` element as defined in Annex A (for dataset group and dataset). The value of the key shall be obtained by executing the PBKDF2 algorithm as defined in RFC 8018. The elements defined in the `KeyDerivation` tag indicate the parameters to be used as input for the PBKDF2 algorithm. The value of the `password` used in the execution of the PBKDF2 algorithm shall be retrieved as the value of the key having the name equal to the value of the `passwordName` element, and be retrieved as specified in subclause 7.2.4.

7.2.4.3 Symmetric encrypted key

The information is provided in a `KeySymmetricWrap` element as defined in Annex A (for dataset group and dataset). The key in wrapped form is stored in the `wrappedKey` element of the `KeySymmetricWrap`. The value of key to be used in the unwrapping procedure is the value of the key having the name equal to the value of the `kek` element, and is to be retrieved as if it was a key, as explained in subclause 7.2.4. The key is to be unwrapped according to the RFC 3394 algorithm, the method to be used is determined by the length of the key referenced by the value of the kek element, and the expected key length (derived from the selected cipher as specified in Table 14).

7.2.4.4 Asymmetric encrypted key

The information is provided in a `KeyAsymmetricWrap` element as defined in Annex A (for dataset group and dataset). The key shall be obtained by applying the decrypting process defined in RFC 8017 for `RSAES-OAEP`. The ciphertext to be decrypted is the content of the element `wrappedKey`, the key to be used is the private key belonging to the asymmetric key pair identified by the value of the element `publicKeyName`, and the hash method to be used is indicated by the `hashFunction` element. The `Label parameter` used in the decryption process of RFC 8017 shall be provided as an empty string. The MGF function to be used shall be the MGF1 as defined in RFC 8017, and the hash function to be used in the MGF function is identified by the `maskGenerationHashFunction` element.

7.2.5 Decryption

7.2.5.1 Decryption of gen_info elements

This clause specifies the decryption process when the value of the `encryptedLocations` element of the XML `EncryptionParameters` element matches one of the URI syntaxes defined in subclauses 7.2.2 and 7.2.3.2.

In order to decrypt the associated content, the key with name equal to the value of the `key` element in the `EncryptionParameter` shall be retrieved as specified in 7.2.4. The cipher is identified by the `cipher` element value of the `EncryptionParameter`. The key shall have exactly the size required by the cipher.

Table 14 — List of supported ciphers and key sizes

Cipher	Size required
`urn:mpeg:mpeg-g:protection:aes128-ctr`	128 bits
`urn:mpeg:mpeg-g:protection:aes192-ctr`	192 bits
`urn:mpeg:mpeg-g:protection:aes256-ctr`	256 bits
`urn:mpeg:mpeg-g:protection:aes128-gcm`	128 bits
`urn:mpeg:mpeg-g:protection:aes192-gcm`	192 bits
`urn:mpeg:mpeg-g:protection:aes256-gcm`	256 bits

The cipher used is AES as defined in FIPS 197, the length of the key used is either 128, 192 or 256 as inidicated in the URI. The "`CTR`" in the cipher name refers to the Counter mode of operations as defined in 800-38A. The "`GCM`" refers to the Gallois Counter Mode mode of operations as defined in 800-38D. If the GCM mode is used, then the Encryption Parameters shall have an element `TAG`.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

Upon decryption, the ciphertext bytes shall be replaced in-place with the decrypted bytes.

7.2.5.2 Decryption of blocks and access unit information

This clause specifies the decryption process when the value of the `encryptedLocations` element of the XML `EncryptionParameters` element matches one of URI defined in 7.2.3.3.

The ciphers used are the same as in 7.2.5.1.

When an access unit belongs to a collection which an `EncryptionParameters` resolved to, then it is to be decrypted as below:

(1) Retrieve from the `EncryptionParamters` element:

a. the `key`;

b. the `configurationID`.

(2) Retrieve from the `AccessUnitEncryptionParameters` element present in the associated AU protection box (specified in A.5):

a. the cipher (possible values are listed Table 14);

b. the `wrappedKey` matching the `configurationID` (retrieved in step 1);

c. `auinIV`, if the access unit contains an access unit information box;

d. `auinTAG`, if the access unit contains an access unit information box and the cipher uses GCM mode;

e `aublockIV`;

f. `aublockTAG` if the cipher uses GCM mode.

(3) Retrieve the decryption key for the current access unit by unwrapping the contents of `wrappedKey` by applying method specified in RFC 3394 using the key (the actual method to use is derived from the length of the unwrapping key identified by the `EncryptionParameters'` `key` and the expected key length derived from the selected cipher as specified in Table 14).

(4) If the access unit contains an `access unit information box`, decrypt using the cipher and the `key` (obtained in the previous step), `auinIV` and `auinTAG` (if present), the `Value` field of the auin gen_info box of the access unit and replace the ciphertext with the plaintext in place.

(5) Decrypt using the cipher and the `key` (obtained in the previous step), `aublockIV` and `aublock_TAG` (if present),

a. if in AUC mode:

Decrypt using the cipher and the `key` (obtained in the previous step), `aublockIV` and `aublockTAG` (if present), all bytes after the last byte of the access unit protection (defined in ISO/IEC 23092-1:2020, Table 25), and replace the ciphertext bytes with the plaintext bytes in place.

b. Else if in DSC mode:

Decrypt using the cipher and the `key` (obtained in the previous step), `aublock_IV` and `aublock_TAG` (if present), all bytes corresponding to the associated `blocks` payloads (resolved as specified in ISO/IEC 23092-1), concatenated, in growing order of `descriptor_ID` defined in ISO/IEC 23092-1:2020, Table 20, and replace the ciphertext bytes with the plaintext bytes in place, replacing the content of one `block` associated to the access unit (as specified in ISO/IEC 23092-1) at a time, without modifying the byte orderting (the number of bytes

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

to replace for a specific block is the one derived from the Master Index Table as specified in ISO/IEC 23092-1).

7.3 Privacy rules for the use of the genomic information

7.3.1 General

The `privacy_rules` tag is defined in the schema for protection metadata at the dataset group and dataset levels of the hierarchy. The `privacy_rules` tag shall be a valid policy element specified according to the OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0 specification. Exporting this tag as the root element of a new document, a valid policy document is obtained.

The `privacy_rules` specify who can execute a given action and under which conditions. The information is conveyed according to the XACML specification.

In the privacy policies, the XACML `actions` are referred to by using the names defined in the different subclauses of subclause 8.1. Before executing an API action, the generated request shall be checked against the datasetgroup policy and possibly the dataset policy. Each action indicates, in its Authorization paragraph, whether the dataset policy should be checked. If there is no Authorization paragraph, the dataset policy shall not be checked. If it is the case, before submitting the request to the dataset policy, one new attribute is added to the list of attributes of the request: the new attribute shall have an `AttributeId` equal to `urn:mpeg:mpeg-g:protection:granted_by_dataset_group`, a DataType equal to https://www.w3.org/2001/XMLSchema#boolean, and a value of `true` if the request yields a `Permit` result at the datasetgroup level, otherwise `false`. If no `privacy_rules` element is provided in the protection box, then all actions are granted.

Actions are referred in privacy policies by using the name defined in the different subclauses of subclause 8.1. Each XACML `action` is described as an XACML `attributeValue` with category "`urn:oasis:names:tc:xacml:3.0:attribute-category:action`" and attributeId "`urn:oasis:names:tc:xacml:1.0:action:action-id`".

The rule applies to a `resource` (as defined in XACML) determined from the definition of the function in subclause 8.1, and where the rule is stored (i.e. Dataset Group, Dataset).

Rules apply to `subject` or `role`.

`subject` is described as an XACML `attributeValue` with category "`urn:oasis:names:tc:xacml:1.0:subject-category:access-subject`" and `attributeId` "`urn:oasis:names:tc:xacml:1.0:subject:subject-id`".

`role` is described as an XACML `attributeValue` with category "`urn:oasis:names:tc:xacml:1.0:subject-category:access-subject`" and `attributeId` "`urn:oasis:names:tc:xacml:3.0:example:attribute:role`".

If the entity providing the file requires the use of further attributes, the corresponding names, types and semantics shall be provided through another channel.

The API methods, described in subclause 8.1, have a set of possible parameters, each with a specific range of valid values. The privacy rules can further limit the range of accepted input parameters using conditions. The conditions shall refer to the action input parameters by referring to attributes whose `AttributeId` is equal to the name of the parameter and whose type corresponds to the type of the parameter. The request generated to test the authorization to perform a function shall include one attribute per input parameter with the `AttributeId` and type, as specified before.

In the case of getData interfaces (subclauses 10.3.6.3, 10.3.6.4, 10.3.6.5, 10.3.6.6), the privacy rules concerning access units are derived as follows:

— In the case of an access unit of unaligned data, the `getDataBySignature` privacy rule is tested for each of the access unit signatures. If the action is permitted for each signature, then the `getDataBySignature` is permitted for that access unit, otherwise it is denied.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

— In the case of an access unit of aligned data, the getDataByFilter privacy rule is tested, populating the filter with the type, start and end position of the access unit.

Annex B.5 contains examples of privacy rules and authorization requests. Subclause 7.3.2 presents an example of a use case where privacy rules and authorization requests presented in Annex B.5 are used.

7.3.2 Example of use of privacy rules

This subclause provides an example use case to illustrate when and how privacy rules can be used to protect genomic information.

Researchers have already identified some regions of the genome indicating that there is a predisposition to suffer Alzheimer's disease. When trying to access that information, one should consider where these regions are, taking into account their positions inside human genome. Regions identifying Alzheimer's disease predisposition may be protected from unauthorized access by using encryption techniques combined with privacy rules. Therefore, only users authorized by the rules can perform operations over protected regions.

In order to give permission for accessing these protected regions (or some of them), an organization may define privacy policies indicating which person or role can view the genomic regions and the conditions that have to be accomplished.

The same policy may contain different rules to give different permissions to different subjects. When access to one of these regions is requested, the rule or rules applying are evaluated, accepting or denying the corresponding permission.

As an example, a policy may contain several rules. The first rule in the policy may provide permission to perform any operation to researchers. The second one may provide permission to view part of a dataset to practitioners under an emergency situation. To do so, the API operation GetDataBySimpleFilter is defined as the action permitted by the rule, and several conditions related to Alzheimer's disease and the quality of the information stored inside the file are involved. With this purpose, the user defining the rule may specify a region number including start and end positions, class of the read, thus identifying. A final rule denies any access to the genomic file. It is the default rule, which is evaluated in case none of the above applies.

7.4 Digital signature of `gen_info` elements and blocks

7.4.1 General

At each level, the protection box may include integrity information in the form of one or more digital signatures. Each signature is provided as an XML Signature element. Detached, Enveloped and Enveloping signatures are supported. If decryption is required, signature verification shall be performed before decryption.

7.4.2 Signatures carried in dataset group protection

Any number of XML Signature elements can be present in the dataset group protection box and shall use one of the following URIs for the `URI` attribute of the `Reference` element when referring to a gen_info box. These URIs can contain some placeholders identified by a text enclosed in curly brackets. These placeholders (including curly brackets) shall be replaced by an alphanumeric value as specified below.

— "`refmeta/{id}`" resolves to the bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "`rfmd`" (i.e. reference metadata), for which the field `reference_ID` defined in ISO/IEC 23092-1:2020, Table 14 is equal to the value which is replacing "`{id}`", and which is stored in the same dataset group as the current Dataset Group Protection.

— "`metadata`" resolves to the bytes of the `Value` field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "`dgmd`" (i.e. dataset group metadata), stored in the same dataset group as the current Dataset Group Protection.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022

— "dataset/{id}/protection" resolves to bytes of the Value field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "dtpr" (i.e. dataset protection), stored in the dataset for which the field dataset_ID defined in ISO/IEC 23092-1:2020, Table 20 is equal to the value which is replacing "{id}", and which belongs to the current Dataset Group.

7.4.3 Signatures carried in dataset protection

7.4.3.1 General

Any number of XML Signature elements can be present in the dataset protection box and shall use one of the following URIs for the URI attribute of the Reference element when referring to a gen_info box or the content of an access unit. These URIs can contain some placeholders identified by the text enclosed in curly brackets. These placeholders shall be replaced by an alphanumeric value as specified below. The bytes are signed as transmitted, regardless of encryption.

7.4.3.2 URIs for gen_info boxes

— "pars/{id}" resolves to the bytes of the Value field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "pars" (i.e. dataset parameter set), for which the field parameter_set_ID defined in ISO/IEC 23092-1:2020, Table 23 is equal to the value which is replacing "{id}", and which is stored in the same dataset as the current Dataset Protection.

— "metadata" resolves to bytes of the Value field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "dtmd" (i.e. dataset metadata), stored in the same dataset as the current Dataset Protection.

— "descstream/{classId}/{descriptorId}/protection" resolves to the Value field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "dspr" (i.e. stream protection) stored in the stream for which the field class_ID defined in ISO/IEC 23092-1:2020, Table 19 is equal to the value which is replacing "{classId}", the field descriptor_ID defined in ISO/IEC 23092-1:2020, Table 19 is equal to the value which is replacing "{descriptorId}", and which belongs to the current Dataset.

— "au/{auType}/{sequenceName}/{accessUnitId}/protection" resolves to the Value field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "aupr" (i.e. access unit protection) stored in the access unit for which the associated AU_type defined in ISO/IEC 23092-1:2020, Table 25 is equal to the value which is replacing "{auType}", the associated sequence_name defined in ISO/IEC 23092-1:2020, Table 10 is equal to the value replacing "{sequenceName}" and the field access_unit_ID defined in ISO/IEC 23092-1:2020, Table 25 is equal to the value replacing "{accessUnitId}". The value of auType shall be equal or greater than 1 and strictly less than 6.

— "au/unaligned/{accessUnitId}/protection" resolves to the Value field (defined in ISO/IEC 23092-1:2020, subclause 6.3) of the gen_info with key "aupr" (i.e. access unit protection) stored in the access unit for which the associated auType defined in either ISO/IEC 23092-1:2020, Table 25 or Table 19 (under the name clid) is equal to 6, and the field access_unit_ID defined in ISO/IEC 23092-1:2020, Table 25 is equal to the value replacing "{accessUnitId}".

7.4.3.3 URIs for collection of access units

Any number of signatutes stored in the dataset protection shall also use any of the following URIs to refer to the content corresponding to a collection of access units

— "genomic_region/{auType}/{sequenceName}/{start}/{end}" resolves to one collection of access units to sign. The values replacing "{start}" and "{end}" shall be numbers such that the first one is strictly smaller than the second one. For all access units such that

  — the AU_type defined in either ISO/IEC 23092-1:2020, Table 25 is equal to the value which is replacing "{auType}", or the value replacing "{auType}" is equal to "0" and the AU_type defined in ISO/IEC 23092-1:2020, Table 25 is strictly smaller than 6.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 23092-3-2022