INTERNATIONAL STANDARD

ISO/IEC 39794-1

First edition
2019-12

Information technology — Extensible biometric data interchange formats —

Part 1: Framework

Reference number
ISO/IEC 39794-1:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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).

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 http://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.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

A list of all parts in the ISO/IEC 39794 series can be found on the ISO website.

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

The purchase of this ISO/IEC document carries a copyright licence for the purchaser to use ISO/IEC copyright in the schemas in the annexes to this document for the purpose of developing, implementing, installing and using software based on those schemas, subject to ISO/IEC licensing conditions set out in the schemas.

Introduction

Biometric data interchange formats enable the interoperability of different biometric systems. The first generation of biometric data interchange formats was published between 2005 and 2007 in the first edition of the ISO/IEC 19794 series. From 2011 onwards, the second generation of biometric data interchange formats was published in the second edition of the established parts and the first edition of some new parts of the ISO/IEC 19794 series. In the second generation of biometric data interchange formats, new useful data elements such as those related to biometric sample quality were added, the header data structures were harmonized across all parts of the ISO/IEC 19794 series, and XML encoding was added in addition to the binary encoding.

The second generation of the biometric data interchange formats turned out to be syntactically incompatible with their first generation. The second generation, however, did not cancel and replace the first generation because the first generation has been adopted widely, e.g. for biometric data stored in machine-readable travel documents, which will be in the field for a long time. Therefore, the first editions of the ISO/IEC 19794 series are expected to be retained in the standards catalogue as long as needed alongside their second editions.

In anticipation of the need for additional data elements, and in order to avoid future compatibility issues, the ISO/IEC 39794 series provides standard biometric data interchange formats capable of being extended in a defined way. Extensible specifications in ASN.1 (Abstract Syntax Notation One) and the Distinguished Encoding Rules of ASN.1 form the basis for encoding biometric data in binary tag-length-value formats. XSDs (XML schema definitions) form the basis for encoding biometric data in XML (eXtensible Markup Language).

This document defines what is common for the extensible biometric data interchange formats considered in the specific parts of the ISO/IEC 39794 series, i.e. the common content, meaning and representation of biometric data interchange formats.

The ISO/IEC 39794 series is one of a family of international standards being developed by ISO/IEC JTC 1/SC 37 that supports interoperability and data interchange among biometric applications and systems. This family of standards specifies requirements on a wide variety of biometric recognition applications, whether such applications operate in an open systems environment or consist of a single, closed system. Open systems are built on standards-based, publicly defined data formats, interfaces and protocols to facilitate data interchange and interoperability with other systems, which may include components of different design or manufacture. A closed system can also be built on publicly defined standards, and may include components of different design or manufacture, but inherently has no requirement for data interchange and interoperability with any other system.

The ISO/IEC JTC 1/SC 37 biometric standards family includes a layered set of standards consisting of biometric data interchange formats and biometric interfaces, as well as biometric profiles that describe the use of these standards in specific application areas. Figure 1 shows the interrelation of biometrics-related areas of standardization. Biometric data complying with one of the biometric data interchange formats defined in the ISO/IEC 19794 series[7] and the ISO/IEC 39794 series represent the core component of biometric interoperability. The formats defined in the ISO/IEC 19785 series[4] may be used as a wrapper around biometric data. Since biometric data are sensitive data and subject to attack, cryptographic protection is required in interchange environments. Biometrics-related profiles, security evaluation and performance evaluation also play an important role. Biometric interfaces are essential to facilitate easy integration and usage of biometric components. The harmonized biometric vocabulary is recommended for use in describing biometric technology. The deployment of applications using biometric verification or identification takes place within the context of societal and cross-jurisdictional requirements.

The ISO/IEC 19794 series and the ISO/IEC 39794 series specify biometric data interchange formats for different types of biometric characteristics. Parties that agree on a biometric data interchange format specified in the ISO/IEC 19794 series or the ISO/IEC 39794 series should be able to decode each other's biometric data.

The biometric interface standards include the Common Biometric Exchange Formats Framework (CBEFF) series (ISO/IEC 19785[4]) and the Biometric Application Programming Interface (BioAPI) series (ISO/IEC 19784[3]). These standards support exchange of biometric data within a system or among systems. The CBEFF series specifies the basic structure of a standardized biometric information record (BIR) which includes one or more biometric data blocks (BDB) with added metadata, such as date and time when it was captured, its expiry date, whether it is encrypted, etc. The BioAPI series specifies an open system API that supports communications between software applications and underlying biometric technology services.

The biometric profile series (ISO/IEC 24713[8]) facilitates implementations of the base standards (e.g. biometric data interchange format standards and biometric interface standards and possibly non-biometric standards) for defined applications. These profiles define the functions of an application (e.g. physical access control for employees at airports) and then specify use of options in the base standards to ensure biometric interoperability.

The ISO/IEC 24779[10] series specifies a family of icons and symbols used in association with devices for biometric enrolment, verification and/or identification. The symbols and icons are intended to show the type of biometric characteristics and to advise on the appropriate preparation and behaviour required when using a biometric system. They are also intended to assist capture subjects by guiding them as they use the biometric system.

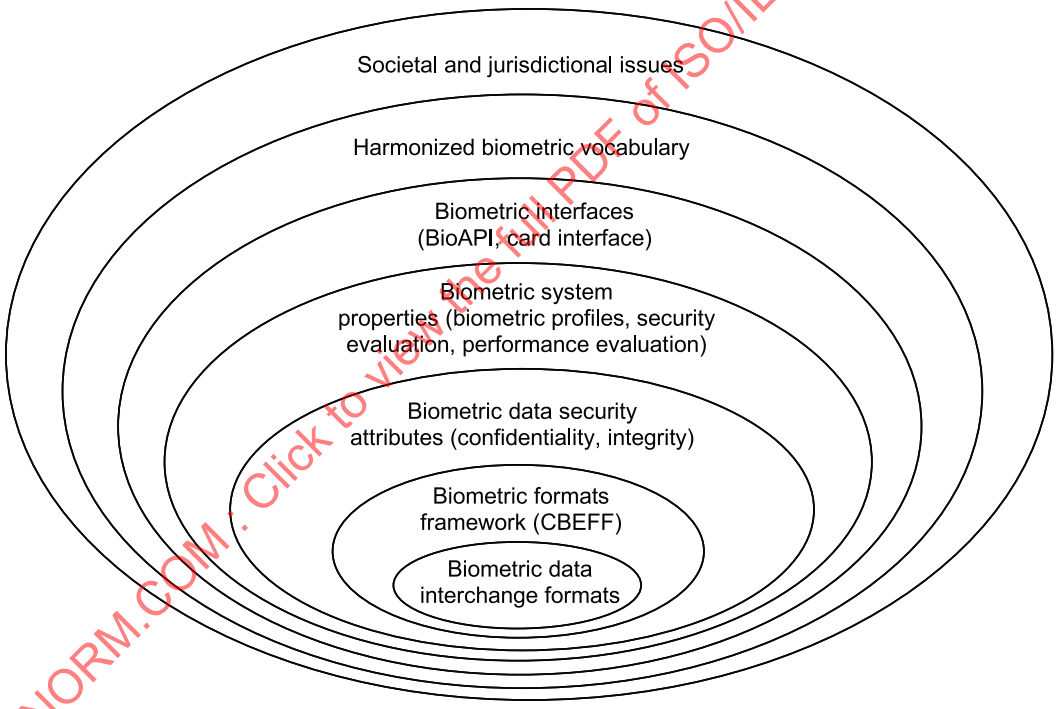

Figure 1 — General interrelation model of biometric issues

Information technology — Extensible biometric data interchange formats —

Part 1: Framework

1 Scope

This document specifies:

— rules and guidelines for defining extensible biometric data interchange formats that are extensible without invalidating previous data structures;

— the meaning of common data elements for use in extensible biometric data interchange formats;

— common data structures for tagged binary data formats based on an extensible specification in ASN.1;

— common data structures for textual data formats based on an XML schema definition; and

— conformance testing concepts and methodologies for testing the syntactic conformance of biometric data blocks.

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 2382-37, *Information technology — Vocabulary — Part 37: Biometrics*

ISO 8601 (all parts), *Date and time — Representations for information interchange*

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1*

ISO/IEC 8825-1, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) — Part 1*

ISO/IEC 19785-2,[1)] *Information technology — Common Biometric Exchange Formats Framework — Part 2: Procedures for the operation of the biometric registration authority*

ISO/IEC 29794-1, *Information technology — Biometric sample quality — Part 1: Framework*

ISO/IEC 30107-2, *Information technology — Biometric presentation attack detection — Part 2: Data formats*

IETF RFC 5141, *A Uniform Resource Name (URN) Namespace for the International Organization for Standardization (ISO)*

IETF RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*

W3C Recommendation, *XML Schema Part 1: Structures* (Second Edition), 28 October 2004, http://www.w3.org/TR/xmlschema-1/

1) Second edition under preparation. Stage at time of publication: ISO/IEC DIS 19785-2:2018.

W3C Recommendation, *XML Schema Part 2: Datatypes* (Second Edition), 28 October 2004, http://www.w3.org/TR/xmlschema-2/

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 2382-37 and the following apply.

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

— IEC Electropedia available at http://www.electropedia.org/;

— ISO Online Browsing Platform available at http://www.iso.org/obp.

3.1
biometric behavioural data
biometric data representing behavioural biometric characteristics of an individual

EXAMPLE Data resulting from writing, speaking or typing.

3.2
biometric data block
BDB
block of data conforming to a defined format

Note 1 to entry: The BDB is normally opaque to the processing of a standard biometric header (SBH) and is not required to be self-delimiting.

Note 2 to entry: This definition is aligned with ISO/IEC 19785-1.

3.3
biometric feature data unit
smallest individual unit of extracted biometric feature data

EXAMPLE Minutiae of a fingerprint.

3.4
biometric image data
biometric data that results from the presentation of biological biometric characteristics of an individual and is represented by pixels in a spatial coordinate system

EXAMPLE Fingerprint image data.

3.5
bit depth
number of bits used to represent a data element

3.6
octet
byte
contiguous sequence of 8 bits processed as a single unit of information

3.7
pixel
picture element
point in an image that is represented by an *n*-by-*m* matrix of points, where *n* is the number of horizontal rows and *m* is the number of vertical columns

4 Abbreviated terms

ABNF	Augmented Backus-Naur Form
API	application programming interface
ASN.1	Abstract Syntax Notation One
BDB	biometric data block
BIR	biometric information record
CBEFF	Common Biometric Exchange Formats Framework
DER	Distinguished Encoding Rules
HTTP	Hypertext Transfer Protocol
ICS	implementation conformance statement
IUT	implementation under test
NSS	namespace-specific string
PAD	presentation attack detection
SBH	standard biometric header
TLV	tag-length-value
URI	uniform resource identifier
URN	uniform resource name
UTC	Coordinated Universal Time
XML	eXtensible Markup Language
XSD	XML schema definition

5 Conformance

A binary biometric data interchange format conforms to this document if it satisfies the requirements specified within Clauses 7, 8, 9, and A.1.

A textual biometric data interchange format conforms to this document if it satisfies the requirements specified within Clauses 7, 8, 10, and A.2.

A general BDB embedding BDBs in formats defined elsewhere conforms to this document if it satisfies the requirements specified in Annex B.

A biometric data interchange format conformance test conforms to this document if it satisfies the requirements specified in Annex C.

6 General biometric system

6.1 Conceptual representation of general biometric system

Given the variety of applications and technologies, it can seem difficult to draw any generalizations about biometric systems. All such systems, however, have many elements in common. Captured

biometric samples are acquired from a subject by a biometric capture device. The biometric capture device output is sent to a processor that extracts the distinctive but repeatable measures of the sample (the biometric features), discarding all other components. The resulting features can be stored in the biometric enrolment database as a biometric reference. In other cases, the sample itself (without biometric feature extraction) may be stored as the reference. A subsequent query or probe biometric sample can be compared to a specific reference, to many references, or all references already in the database to determine if there is a match. A decision regarding the biometric claim is made based upon the similarities or dissimilarities between the features of the probe and those of the reference or references compared.

Figure 2 illustrates the information flow within a general biometric system, showing a general biometric system consisting of data capture, signal processing, data storage, comparison and decision subsystems. Figure 2 illustrates both enrolment and the operation of verification and identification systems. The subclauses in Clause 6 describe each of these subsystems in more detail.

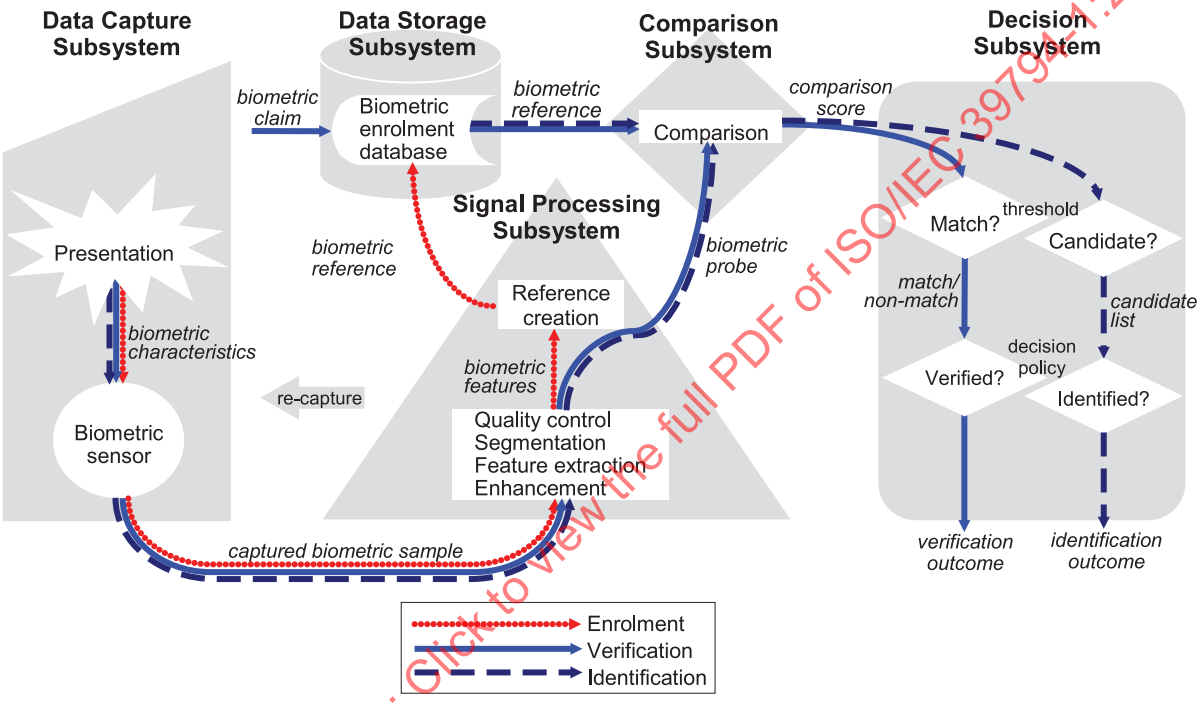

Figure 2 — Components of a general biometric system

NOTE In any implemented system, some of these conceptual components can be absent or could not have a direct correspondence with a physical or software entity.

6.2 Conceptual components of a general biometric system

6.2.1 Data capture subsystem

The data capture subsystem collects an image or signal of a subject's biometric characteristics presented to the biometric capture device (sensor) and outputs this image/signal as a captured biometric sample.

6.2.2 Transmission subsystem

The transmission subsystem (not portrayed in Figure 2; not always present or visibly present in a biometric system) will transmit samples, features, probes and references between different subsystems. The captured biometric sample may be compressed and/or encrypted before transmission and expanded and/or decrypted before use. A captured biometric sample may be altered in transmission due to noise in the transmission channel as well as losses in the compression/expansion process. Data may be transmitted using standardized biometric data interchange formats and cryptographic

techniques may be used to protect the authenticity, integrity and confidentiality of stored and transmitted biometric data.

6.2.3 Signal processing subsystem

Signal processing may include processes such as:

— enhancement, i.e. improving the quality and clarity of the captured biometric sample;

— segmentation, i.e. locating the signal of the subject's biometric characteristics within the captured biometric sample;

— biometric feature extraction, i.e. deriving the subject's repeatable and distinctive measures from the captured biometric sample; and

— quality control, i.e. assessing the suitability of samples, features, references, etc., possibly affecting other processes, such as:

  — returning control to the data capture subsystem to collect further samples or

  — modifying parameters for segmentation, biometric feature extraction, comparison.

In the case of enrolment, the signal processing subsystem creates a biometric reference. Sometimes the enrolment process requires features from several presentations of the individual's biometric characteristics. Sometimes the reference comprises just the features, in which case the reference may be called a biometric template. Sometimes the reference comprises just the sample, in which case biometric feature extraction from the reference occurs immediately before comparison.

In the case of verification and identification, the signal processing subsystem creates a biometric probe. Sequencing and iteration of the above-mentioned processes are determined by the specifics of each system.

6.2.4 Data storage subsystem

References are stored within an enrolment database held in the data storage subsystem. Each reference may be associated with some details of the enrolled subject or the enrolment process. Prior to being stored in the enrolment database, references may be reformatted into a biometric data interchange format. References may be stored within a biometric capture device, on a portable medium such as an integrated-circuit card, on a personal computer or local server, or in a central database.

6.2.5 Comparison subsystem

In the comparison subsystem, the features extracted from probes are compared against one or more references and comparison scores are passed to the decision subsystem. The comparison scores indicate the similarities or dissimilarities between the probes and references compared. For verification, a single specific claim of subject enrolment would lead to a single comparison score. For identification, many or all references may be compared with the probe, and a comparison score may be produced for each comparison.

6.2.6 Decision subsystem

The decision subsystem uses the comparison scores generated from one or more comparison attempts to provide the decision outcome for a verification or identification transaction.

In the case of verification, the probe is considered to match a compared reference when (assuming that higher scores correspond to greater similarity) the comparison score exceeds a specified threshold. A biometric claim can then be verified on the basis of the decision policy, which may allow or require multiple attempts.

In the case of identification, an enrolee is a potential candidate when (assuming that higher scores correspond to greater similarity) the comparison score exceeds a specified threshold, and/or when the comparison score is among the highest ranked values generated during comparisons across the entire database. The decision policy may allow or require multiple attempts before making an identification decision.

NOTE Conceptually, it is possible to treat multi-biometric systems in the same manner as unibiometric systems, by treating the combined captured biometric samples/references/scores as if they were a single sample/reference/score and allowing the decision subsystem to operate score fusion or decision fusion as and if appropriate. See also ISO/IEC TR 24722[9].

6.2.7 Administration subsystem

The administration subsystem (not shown in Figure 2) governs the overall policy, implementation, configuration and usage of the biometric system, in accordance with the relevant legal, jurisdictional and societal constraints and requirements. Illustrative examples include:

— interacting with the biometric capture subject including providing guidance feedback to the subject during and/or after data capture and requesting additional information from the subject;

— storing and formatting of interchanged biometric data;

— providing final arbitration on output from decision and/or scores;

— setting threshold values for decision;

— setting biometric capture parameters;

— controlling the operational environment and non-biometric data storage;

— providing appropriate safeguards for subject privacy and data security; and

— interacting with the application that utilizes the biometric system.

6.2.8 Interface

The biometric system may or may not interface to an external application or system via a web services interface, an application programming interface, a hardware interface or a protocol interface (not shown in Figure 2).

6.3 Functions of general biometric system

6.3.1 Enrolment

In enrolment, a transaction by a biometric capture subject is processed by the system in order to generate and store a biometric reference for that individual.

Enrolment typically involves:

— capturing one or more biometric samples;

— sample restoration or enhancement;

— segmentation;

— biometric feature extraction;

— quality checks (which may reject the sample/features as being unsuitable for creating a reference, and require capture of further samples);

— (where system policy requires it) comparison against the stored biometric references to ensure that the subject is not already enrolled;

— biometric reference creation (which may require features from multiple samples) and possibly conversion into a biometric data interchange format;

— storage of the biometric reference;

— test verification or identification attempts to ensure that the resulting enrolment is usable; and

— allowing for repeat enrolment attempts, should the initial enrolment be deemed unsatisfactory (dependent on the enrolment policy).

6.3.2 Verification

In access control applications, a transaction by a subject may be processed by the system in order to verify a specific positive claim about the subject's enrolment (e.g. "I am enrolled as subject X").

NOTE Some biometric systems allow a single subject to enrol more than one instance of a biometric characteristic (for example, an iris system can allow subjects to enrol both iris images, while a fingerprint system can require enrolment of additional fingers for fall-back in case a primary finger is damaged).

Verification of a specific positive biometric claim typically involves:

— capturing one or more biometric samples;

— sample restoration or enhancement;

— segmentation;

— biometric feature extraction and possibly conversion into a standardized biometric data interchange format;

— quality checks (which may reject the samples/features as being unsuitable for comparison and require capture of further samples);

— comparison of the probe features against the reference for the claimed identity producing a comparison score; and

— a verification decision based on whether the comparison score exceeds a threshold (assuming that higher scores correspond to greater similarity) in one or more attempts as dictated by the decision policy.

The decision subsystem will either accept or reject the specific positive claim. The verification decision outcome is considered to be erroneous if either a false claim is accepted (false accept) or a true claim is rejected (false reject).

EXAMPLE In a verification system allowing up to three attempts, a false reject occurs if the reference for the biometric claim actually stems from the present biometric capture subject, but on all three attempts the probe does not match this reference. A false accept occurs if the reference for the biometric claim does not stem from the present biometric capture subject, but the probe falsely matches this reference on any of the three attempts.

6.3.3 Identification

In identification, a transaction by a subject is processed by the system and the enrolment database is searched to return identifiers of similar references. Identification provides a candidate list of identifiers that will contain zero, one, or more identifiers. Identification is considered correct when the subject is enrolled, and an identifier for their enrolment is in the candidate list. The identification is considered to be erroneous if either an enrolled subject's identifier is not in the resulting candidate list (false-negative identification error) or if a transaction by a non-enrolled subject produces a non-empty candidate list (false-positive identification error).

Identification typically involves:

— capturing one or more biometric samples;

— sample restoration or enhancement;

— segmentation;

— biometric feature extraction and possibly conversion into a standardized biometric data interchange format;

— quality checks (which may reject the sample/features as being unsuitable for comparison, and require capture of further samples);

— comparison of the probe features against some or all references in the enrolment database, producing a comparison score for each comparison; and

— determination of a candidate list based on whether the comparison score exceeds a threshold and/or is among the highest ranked scores returned (assuming that higher scores correspond to greater similarity) in one or more attempts, as dictated by the decision policy.

7 Rules and guidelines

7.1 Capture date and time

Some biometric characteristics (e.g. facial image or signature dynamics) may undergo changes as a capture subject ages. Therefore, the capture date and time of biometric data should be recorded.

7.2 Degree of processing

7.2.1 Overview

The degree of processing of biometric data may be one of the following:

— captured biometric sample: the data are in raw form as delivered by the capture device;

— intermediate biometric sample: the data has been processed from the form delivered by the sensor, but is not in a form usable for comparison – such data is referred to as biometric image data or biometric behavioural data;

— processed data: the data is in a form that can be used for comparison – such data is referred to as biometric feature set.

Figure 3 illustrates the degrees of processing of biometric data. For data interchange, intermediate biometric samples (e.g. image data or behavioural data) and biometric feature sets are of special relevance. Examples of scenarios using biometric data at different degrees of processing are shown in Annex D.

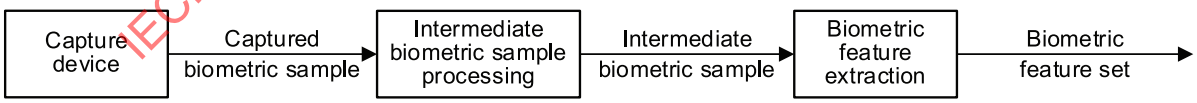

Figure 3 — Captured biometric sample, intermediate biometric sample and biometric feature set

7.2.2 Captured biometric sample

The captured biometric sample is influenced by some or all of the following:

— underlying biometric characteristic;

— presentation of the biometric characteristic to the capture device;

— intermediate biometric sample processing (as part of data acquisition) within the capture device;

— technical characteristics of the capture device;

— environmental conditions (e.g. lighting, background noise).

Captured biometric samples are usually not used for data interchange.

7.2.3 Intermediate biometric sample

7.2.3.1 Biometric image data

In many cases, the captured biometric sample of a static biometric characteristic delivered by a biometric capture device is subsampled, scaled, interpolated, compressed or otherwise processed to produce an image of the characteristic. The first important convention to be made concerns the general image file format (e.g. BMP, TIFF, GIF, JPEG, JPEG-LS, JPEG2000) and the compression level to make images readable for all systems. Further conventions are needed for certain parameters concerning the image capturing process and the hardware to be used, which have a strong impact on the resulting image, e.g. bit depth (8 bit, 16 bit, etc.), spatial sampling rate, position of the biometric characteristic to be represented, and lighting conditions during image capture process.

7.2.3.2 Biometric behavioural data

In contrast to image data, which is based on a static biological characteristic like a fingerprint, behavioural data is based on a dynamic action with contributions from conditioned behaviour patterns as well as biological characteristics. For behavioural biometric characteristics, common analysis approaches are time-based and frequency-based analysis. Therefore, standardization concentrates on data interchange formats for such approaches.

7.2.3.3 Other intermediate biometric samples

Not all data used for the extraction of biometric features is image data or behavioural data.

EXAMPLE DNA profiles are neither extracted from image data nor from behavioural data.

7.2.4 Biometric feature set

A biometric feature set may consist of several feature data units. A biometric feature data unit may consist of several data elements, e.g. coordinates and angles. The structure and content of a feature data unit depends on the type of biometric characteristics.

7.3 Relationship to CBEFF

7.3.1 Overview

A BDB in a format defined in the ISO/IEC 39794 series may, but need not be, embedded in a record in an appropriate CBEFF patron format (see ISO/IEC 19785-3[6]).

7.3.2 BDB format owner and format identifiers

Within records in a CBEFF patron format, the format of the BDB is identified by:

— the CBEFF biometric organization identifier of the BDB format owner, and

— a BDB format identifier that is unambiguous within the scope of the BDB format owner.

For all extensible biometric data interchange formats defined in the ISO/IEC 39794 series, the format owner is ISO/IEC JTC 1/SC 37. The CBEFF biometric organization identifier for ISO/IEC JTC 1/SC 37 is 257 (0101_{Hex}).

For each extensible biometric data interchange format defined in the ISO/IEC 39794 series, ISO/IEC JTC 1/SC 37 assigns a BDB format identifier. Each part of the ISO/IEC 39794 series defines and lists the BDB format identifier, short name and full object identifier for each extensible biometric data interchange format it contains.

7.4 Types of extensible biometric data interchange formats

The structure, content, and encoding of BDBs depend on where the biometric data is intended to be used. There may be:

— self-contained data structures providing all necessary information;

— data structures designed for embedment in a CBEFF biometric information record (BIR) not duplicating information that is present in the CBEFF standard biometric header (SBH); and

— data structures designed for on-card biometric comparison as defined in ISO/IEC 24787[11].

Depending on the available storage space and data transmission rate, biometric data may be encoded in different encoding styles including:

— tagged binary formats; and

— more verbose, human readable textual formats following XML schema definitions.

7.5 Criteria for standardizing biometric data interchange formats

The standardization of biometric data interchange formats is intended to provide interoperability. Therefore, the number of standardized formats should be kept small and manageable. The conditions listed below should be considered before a new data interchange format may enter the standardization process. The new data interchange format:

— uses an encoding style that is very different from the one of an already standardized data interchange format, such as XML encoding vs. binary encoding;

— represents the essential data required for an alternative mathematical approach of biometric feature extraction and/or comparison;

— is a prevalent alternative representation of data that is defined in the ISO/IEC 19794 series or the ISO/IEC 39794 series;

— represents data of a widely used type of biometric characteristics not yet considered in the ISO/IEC 19794 series or the ISO/IEC 39794 series;

— represents data of a different degree of processing and has become widely used for data interchange or has the potential for it;

— enables interoperability among algorithms that use individual non-standardized data interchange formats of an advanced degree of processing;

— significantly reduces the size of data of an already standardized data interchange format and is suitable for use within integrated circuit cards and other tokens;

— has the potential to be used for different types of biometric characteristics, e.g. an image format;

— combines existing data interchange formats without increasing size;

— allows an improved biometric performance.

7.6 Extensibility

Requirements for new data elements may arise in the future. The extensible biometric data interchange formats shall be capable of being extended without invalidating the already standardized formats.

If a future amendment or revision will augment an extensible biometric data interchange format defined in the ISO/IEC 39794 series, systems capable of reading data conformant to the base format shall be able to also parse data conformant to the revised format, ignoring new data elements that were not in the base format. The revised format shall at least include the mandatory data items of the base standard. Systems capable of reading data conformant to the revised format shall also be able to read and understand data in the base format.

7.7 Naming conventions for biometric data interchange formats

The name of the data interchange format should indicate the type of biometric characteristics and, where different biometric sample or feature data formats are available, the respective biometric sample or feature.

EXAMPLE Finger minutiae data, finger image data, signature/sign time series data, hand geometry silhouette data, voice data, DNA data.

7.8 Treatment of multi-biometric data

Multi-biometrics may be used to improve the performance of biometric systems in terms of error rates. Multi-biometrics can be divided into five subcategories, which are defined in ISO/IEC TR 24722[9]:

— multi-type – use of multiple types of biometric characteristics such as face and fingerprint;

— multi-algorithmic – use of two or more distinct algorithms for processing the same biometric sample;

— multi-instance – use of at least two instances of the same type of biometric characteristics, e.g. left and right iris or left and right index finger;

— multi-sensorial – use of multiple capture devices for capturing samples of one biometric instance;

— multi-presentation – use of either multiple presentation samples of one instance of a biometric characteristic or of a single presentation that results in the capture of multiple samples.

If multi-type systems are used, data structures of several parts of the ISO/IEC 39794 series may be involved in a verification or identification process. Multi-instance or multi-presentation data may be stored in several biometric representations that are contained in one BDB.

7.9 Capture conditions

Biometric samples and reference data from the same capture subject do not generally result in a perfect match when performing a comparison. The data capture subsystem may never capture exactly identical biometric samples since these depend on a lot of factors where small changes result in different data (e.g. translation, rotation and distortion of fingers). Therefore, the other parts of the ISO/IEC 39794 series not only define specific biometric data interchange formats, but also capture conditions necessary for achieving interoperability.

7.10 Capture device requirements

Capture device requirements should be defined to the extent necessary to achieve interoperability. Subjects of these definitions may include:

— capture device technology;

— spatial sampling rate;

— size;

— range of grey or colour levels;

— temporal sampling rate;

— illumination type and intensity; and

— signal-to-noise ratio.

7.11 Quality requirements for biometric data

In order to achieve good comparison results, the other parts of the ISO/IEC 39794 series specify application-specific minimum quality requirements for biometric data.

EXAMPLE Minimum number of finger minutiae for on-card comparison or portrait quality requirements for passport reference images

7.12 Biometric feature extraction algorithms

If data interchange formats are specified for biometric feature sets (e.g. finger minutiae), then the way for deriving these features shall be specified to the extent necessary to facilitate interoperability, i.e. the comparison results of different implementations shall be within the range of allowed differences.

7.13 Biometric feature comparison algorithms

If data interchange formats are specified for biometric feature sets (e.g. finger minutiae), then the means for comparing the biometric probe with a biometric reference shall be specified to the extent necessary to facilitate interoperability, i.e. the comparison results of different implementations shall be within the range of allowed differences.

7.14 Identifiers for resources related to the ISO/IEC 39794 series

As a persistent, location-independent resource identifier for resources related to a part of the ISO/IEC 39794 series, a uniform resource name (URN) shall be constructed according to the rules of IETF RFC 5141. The URN begins with the prefix "urn:iso:" and continues with a namespace-specific string (NSS). The syntax for URNs for resources related to the ISO/IEC 39794 series is defined as follows in Augmented Backus-Naur Form (ABNF) specified in IETF RFC 5234:

```
URN =            "urn:iso:" std-nss
std-nss =        "std:" docidentifier *supplement [addition]
docidentifier =  "iso-iec:39794:" partnumber ":" edition [":" docversion]
partnumber =     "-" DIGITS
edition =        "ed-" DIGITS
docversion =     "v" DIGITS
DIGITS =         DIGIT *DIGIT
DIGIT =          %x30-39 ; 0 to 9
```

The optional element <docversion> is intended to designate the version number of a document's <edition>. It may be altered by correction or amendment and is distinct from a revision, which changes the edition number. The first version published is 1. Each correction or amendment may increase the version number by 1. If no <docversion> is specified, the reference is to the highest version number available for the denoted <edition>.

```
supplement =     ":" suppltype ":" supplnumber [":" supplversion]
suppltype =      "amd" / "cor"
supplnumber =    DIGITS
supplversion =   "v" DIGITS
```


The optional element <supplement> is intended to designate a technical alteration of an international standard that does not result in a new <edition> or <docversion>. Supplements are numbered consecutively within each supplement type. <supplnumber> identifies the number of the supplement. <supplversion> designates the version of a published supplement. When a supplement is published, it is version 1. If that supplement is corrected by issuing a corrected version, the corrected version is version 2.

The optional element <addition> is intended to identify a particular resource such as an ASN.1 module or XML schema definition contained in the identified document. The syntax of <addition> is not defined in IETF RFC 5141, but by ISO/IEC JTC 1/SC 37:

```
addition =        ":" filename "." extension
filename =        filenameroot "-" edition ["-" docversion]
filenameroot =    "ISO-IEC-39794" partnumber ["-" resourcename]
resourcename =    ALPHA *(ALPHA / DIGIT / "-")
ALPHA =           %x41-5A / %x61-7A ; A-Z / a-z
extension =       "asn" / "xsd"
```

If the optional <resourcename> is not present, the filename designates the core specification of the biometric data interchange format.

<filenameroot> shall begin with an upper-case letter because the file name should correspond to the module name given at the beginning of an ASN.1 module, and this module name must begin with an upper-case letter, as per ISO/IEC 8824-1.

A URN in the ISO namespace can be transformed into an HTTP URI by replacing “std” with the domain name “standards.iso.org”, replacing all occurrences of “:” with “/”, and converting characters to lower case (see IETF RFC 5141). ISO maintains a website implementation of these URIs.

EXAMPLE urn:iso:std:iso-iec:39794:-1:ed-1 corresponds to http://standards.iso.org/iso-iec/39794/-1/ed-1.

8 Abstract data elements

8.1 General

This clause describes the contents of data elements common to all biometric data interchange formats defined in the ISO/IEC 39794 series. The description is independent of the encoding of the data elements.

Certain data elements are optional. Such data elements need not be included in a BDB. An optional data element may be omitted altogether from the encoding.

EXAMPLE 1 In an ASN.1 module, optional data elements are marked with the keyword OPTIONAL. When an optional element is not present, then the tag, length and value octets of this data element are omitted from the tagged binary encoding.

EXAMPLE 2 A data element in an XML schema definition is optional if the value of its minOccurs attribute is 0. When an optional element is not present, then the opening and closing tags as well as the value of this data element are omitted from the XML encoding.

If all child elements of a data element are optional, then this data element shall be marked optional as well.

An application profile may change optional data elements to mandatory data elements, but not the other way around.

8.2 Version block

Abstract values: Sequence of two integers

Contents: This data element shall indicate the version number of the biometric data interchange format. It shall consist of two elements:

— the generation of the biometric data interchange format; and

— the year of publication of the standard or amendment or corrigendum.

For all formats defined in the ISO/IEC 39794 series the generation value shall be 3. If a BDB contains representations encoded using different versions of an extensible biometric data interchange format, then the version block shall indicate the version number of the representation encoded using the highest version.

Figure 4 illustrates the structure of a version block.

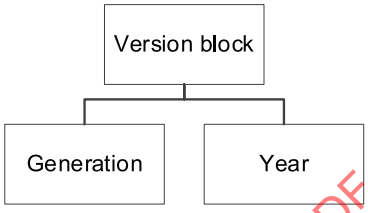

Figure 4 — Structure of a version block

8.3 Representation block

NOTE The other parts of the ISO/IEC 39794 series refine the description of representation block.

8.3.1 Capture device block

NOTE The other parts of the ISO/IEC 39794 series refine the description of capture device block.

8.3.1.1 Model identifier block

Abstract values: Sequence of two integers 1 to 65 535

Contents: This data element shall identify the product type that captured the biometric sample. It shall consist of two elements:

— the capture device vendor identifier; and

— the capture device model identifier.

The capture device vendor identifier shall be one of the biometric organization identifiers registered in accordance with ISO/IEC 19785-2. The capture device model identifier shall be one of the capture device model identifiers associated with the given capture device vendor identifier.

NOTE ISO/IEC 19785-1[5] states that registration of biometric capture device model identifiers is optional.

8.3.1.2 Capture device technology identifier

Abstract values: See other parts of the ISO/IEC 39794 series

Contents: This data element shall indicate the class of technology used in the biometric capture device. Abstract values for the capture device technology identifier and their meanings shall be defined in the other parts of the ISO/IEC 39794 series.

8.3.1.3 Certification identifier block

Abstract values: Sequence of two integers 1 to 65 535

Contents: This data element shall identify a certification scheme according to which a biometric capture device was certified. It shall consist of two elements:

— the certification authority identifier; and

— the certification scheme identifier.

The certification authority identifier shall be registered in accordance with ISO/IEC 19785-2. Those parts of the ISO/IEC 39794 series that specify certification schemes according to which biometric capture devices may define lists of allowed certification scheme identifiers.

8.3.2 Capture date/time block

Abstract values: 0000-01-01T00:00:00,000Z to 9999-12-31T23:59:59,999Z

NOTE The abstract values are given in the extended date-time format of ISO 8601 (all parts). The character "Z" is the designator for UTC (Coordinated Universal Time).

Contents: This data element shall indicate the date and time when the capture of the representation completed in UTC (in accordance with ISO 8601 (all parts)) to a precision of up to one millisecond.

Figure 5 illustrates the structure of a capture date/time block.

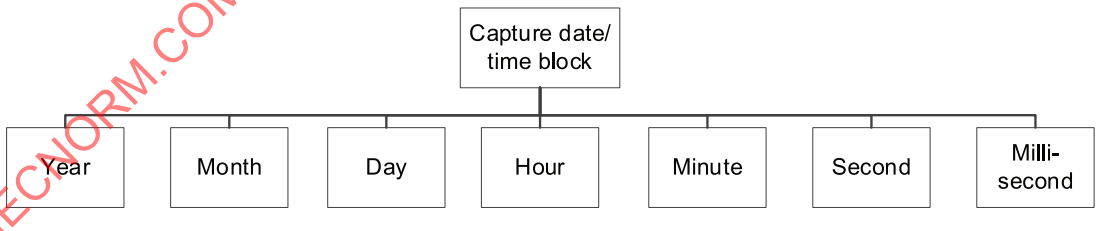

Figure 5 — Structure of a capture date/time block

8.3.3 Quality blocks

NOTE A sequence of quality blocks corresponds to a quality record as per ISO/IEC 29794-1.

8.3.3.1 Quality block

Figure 6 illustrates the structure of a quality block.

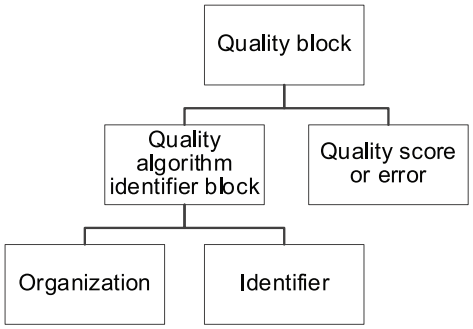

Figure 6 — Structure of a quality block

8.3.3.1.1 Quality algorithm identifier block

Abstract values: Sequence of two integers 1 to 65 535

Contents: This data element shall identify a quality algorithm. It shall consist of two elements:

— the quality algorithm vendor identifier; and

— the quality algorithm identifier.

The quality algorithm vendor identifier shall be one of the biometric organization identifiers registered in accordance with ISO/IEC 19785-2. The quality algorithm identifier shall be one of the quality algorithm identifiers associated with the given quality algorithm vendor identifier.

NOTE ISO/IEC 19785-1[5] states that registration of quality algorithm identifiers is optional.

8.3.3.1.2 Quality score or error

8.3.3.1.2.1 Quality score

Abstract values: Integer 0 to 100

Contents: A quality score shall be a quantitative expression of the predicted verification performance of the biometric sample, as defined in ISO/IEC 29794-1.

8.3.3.1.2.2 Quality scoring error

8.3.3.1.2.2.1 Quality scoring error code

Abstract values: failureToAssess

Contents: The abstract value failureToAssess shall indicate that the quality assessment process has failed.

8.3.3.1.2.2.2 Quality scoring error extension block

If a quality block contains a quality scoring error code from a later version, then it shall also contain a fallback quality scoring error code from the first version (see 8.3.3.1.2.2.1). If a parser does not know the later version, it shall revert to the fallback value.

8.3.4 PAD data block

Figure 7 illustrates the structure of a PAD data block.

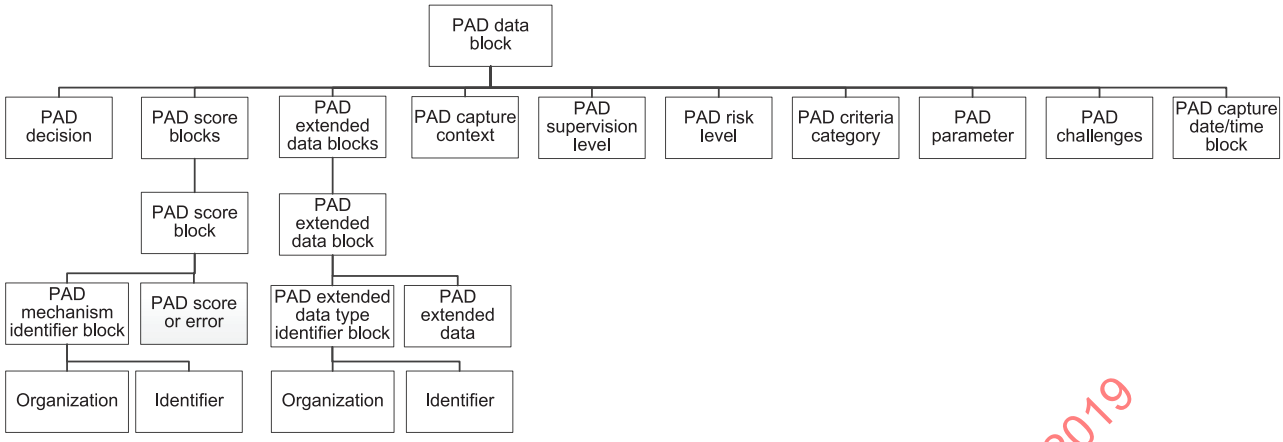

Figure 7 — Structure of a PAD data block

8.3.4.1 PAD decision

8.3.4.1.1 PAD decision code

Abstract values: see Table 1

Contents: This data element shall indicate whether a presentation attack attempt has been detected by the PAD subsystem. Table 1 lists the abstract values of this data element and defines what these values shall mean.

Table 1 — Abstract values for PAD decision code

Abstract value	Meaning
attack	A presentation attack attempt has been detected by the PAD subsystem.
noAttack	No presentation attack attempt has been detected by the PAD subsystem.
failureToAssess	The PAD decision process has failed.

8.3.4.1.2 PAD decision extension block

If a PAD data block contains a PAD decision code from a later version, then it shall also contain a fallback PAD decision code from the first version (see 8.3.4.1.1). If a parser does not know the later version, it shall revert to the fallback value.

8.3.4.2 PAD score blocks

8.3.4.2.1 PAD score block

8.3.4.2.1.1 PAD mechanism identifier block

Abstract values: Sequence of two integers 1 to 65 535

Contents: This data element shall identify a PAD mechanism (referred to as PAD technique in ISO/IEC 19785-1[5]). It shall consist of two elements:

— the PAD mechanism vendor identifier; and

— the PAD mechanism identifier.

The PAD mechanism vendor identifier shall be one of the biometric organization identifiers registered in accordance with ISO/IEC 19785-2. The PAD mechanism identifier shall be one of the PAD mechanism identifiers associated with the given PAD mechanism vendor identifier.

NOTE ISO/IEC 19785-1[5] states that registration of PAD mechanism identifiers is optional.

Table 2 lists PAD mechanism identifiers for PAD approaches not connected with a particular vendor. For the PAD mechanism identifiers listed in Table 2, the biometric organization identifier of ISO/IEC JTC 1/SC 37, which is 257 (0101_{Hex}), shall be used as PAD mechanism vendor identifier. These identifiers have been registered in accordance with ISO/IEC 19785-2.

Table 2 — PAD mechanism identifiers for PAD approaches not connected with a particular vendor

PAD mechanism vendor identifier	PAD mechanism identifier	Description
257 (0101_{Hex})	1 (0001_{Hex})	Challenge/involuntary response
257 (0101_{Hex})	2 (0002_{Hex})	Challenge/voluntary response
257 (0101_{Hex})	3 (0003_{Hex})	Challenge/response as a combination of what you are and know
257 (0101_{Hex})	4 (0004_{Hex})	Non-stimulated observation of liveness

8.3.4.2.1.2 PAD score or error

8.3.4.2.1.2.1 PAD score

Abstract values: Integers 0 to 100

Contents: This data element shall indicate the PAD result as a score between 0 and 100 as defined in ISO/IEC 30107-2. Bona-fide presentations shall tend to generate lower scores. Presentation attacks shall tend to generate higher scores.

8.3.4.2.1.2.2 PAD scoring error

8.3.4.2.1.2.2.1 PAD scoring error code

Abstract values: failureToAssess

Contents: The abstract value failureToAssess shall indicate that the computation of the PAD score has failed.

If the PAD score value is failureToAssess, then, if present, the PAD decision value shall also be failureToAssess.

8.3.4.2.1.2.2.2 PAD scoring error extension block

If a PAD data block contains a PAD scoring error code from a later version, then it shall also contain a fallback PAD scoring error code from the first version (see 8.3.4.2.1.2.2.1). If a parser does not know the later version, it shall revert to the fallback value.

8.3.4.3 PAD extended data blocks

8.3.4.3.1 PAD extended data block

8.3.4.3.1.1 PAD extended data type identifier block

Abstract values: Sequence of two integers 1 to 65535

Contents: This data element shall identify the type of data in a PAD extended data block. It shall consist of two elements:

— a biometric organization identifier; and

— the data type identifier.

The biometric organization identifier shall be one of the biometric organization identifiers registered in accordance with ISO/IEC 19785-2. The data type identifier shall be assigned by the identified biometric organization.

8.3.4.3.1.2 PAD extended data

Abstract values: Any octet string

Contents: This data element shall include additional PAD-related information that cannot be held by the other data elements specified in 8.3.4. The structure of this data is defined by the identified biometric organization.

8.3.4.4 PAD capture context

8.3.4.4.1 PAD capture context code

Abstract values: See Table 3

Contents: This data element shall indicate the context of capture. Table 3 lists the abstract values of this data element and defines what these values shall mean.

Table 3 — Abstract values for PAD capture context code

Abstract value	Meaning
enrolment	The context of capture is enrolment.
verification	The context of capture is biometric verification.
identification	The context of capture is biometric identification.

8.3.4.4.2 PAD capture context extension block

If a PAD data block contains a PAD capture context code from a later version, then it shall also contain a fallback PAD capture context code from the first version (see 8.3.4.4.1). If a parser does not know the later version, it shall revert to the fallback value.

8.3.4.5 PAD supervision level

8.3.4.5.1 PAD supervision level code

Abstract values: See Table 4

Contents: This data element shall indicate the level of supervision/surveillance during the capture process. Table 4 lists the abstract values of this data element and defines what these values shall mean.

Table 4 — Abstract values for PAD supervision level code

Abstract value	Meaning
unknown	The supervision level was not captured or was lost.
controlled	Operator physically controls the biometric capture subject to acquire biometric samples.
assisted	Person available to provide assistance to the biometric capture subject presenting the biometric characteristics.
observed	Person present to observe operation of the device but provides no assistance.[a]
unattended	No one present to observe or provide assistance.

[a] This category includes observing user interaction with the biometric capture system through remote sensing, e.g. video surveillance, also known as telepresence.

8.3.4.5.2 PAD supervision level extension block

If a PAD data block contains a PAD supervision level code from a later version, then it shall also contain a fallback PAD supervision level code from the first version (see 8.3.4.5.1). If a parser does not know the later version, it shall revert to the fallback value.

8.3.4.6 PAD risk level

Abstract values: Integers 0 to 100

Contents: This data element shall indicate the risk level as a score between 0 and 100, with lower scores being indicative of a lower risk and higher scores being indicative of higher risk. If the risk level is unknown, then this data element shall not be present.

This field has been left vaguely defined so that system developers may devise their own qualitative or quantitative risk assessment methodologies.

8.3.4.7 PAD criteria category

8.3.4.7.1 PAD criteria category code

Abstract values: See Table 5

Contents: This data element shall be used to distinguish between PAD decision criteria specific to the individual biometric capture subject and PAD decision criteria common to all biometric capture subjects. Table 5 lists the abstract values of this data element and defines what these values shall mean.

Table 5 — Abstract values for PAD criteria category code

Abstract value	Meaning
unknown	The information whether the PAD criteria were specific to the individual biometric capture subject or common for all biometric capture subjects was not captured or was lost.
individual	Criteria are specific to the individual biometric capture subject.
common	Criteria are global, i.e. common for all biometric capture subjects.

8.3.4.7.2 PAD criteria category extension block

If a PAD data block contains a PAD criteria category code from a later version, then it shall also contain a fallback PAD criteria category code from the first version (see 8.3.4.7.1). If a parser does not know the later version, it shall revert to the fallback value.

8.3.4.8 PAD parameter

Abstract values: Any octet string

Contents: This data element shall indicate any external parameter (e.g. threshold) employed to make the PAD decision.

8.3.4.9 PAD challenges

8.3.4.9.1 PAD challenge

Abstract values: Octet string

Contents: This data element shall indicate any challenge that was given to the data capture subject.

8.3.4.10 PAD capture date/time block

Abstract values: 0000-01-01T00:00:00,000Z to 9999-12-31T23:59:59,999Z

NOTE The abstract values are given in the extended date-time format of ISO 8601 (all parts). The character "Z" is the designator for UTC (Coordinated Universal Time).

Contents: This data element shall indicate the date and time when the capture of the PAD data started in UTC (in accordance with ISO 8601 (all parts)) to a precision of one millisecond.

8.3.5 Extended data blocks

8.3.5.1 Extended data block

Figure 8 illustrates the structure of an extended data block.

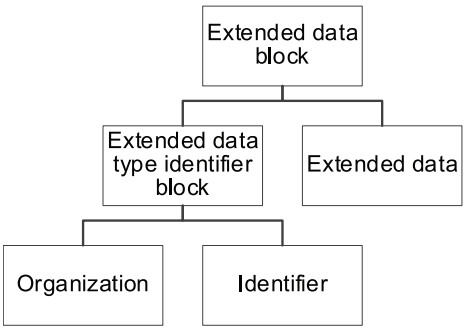

Figure 8 — Structure of an extended data block

8.3.5.1.1 Extended data type identifier block

Abstract values: Sequence of two integers 1 to 65 535

Contents: This data element shall identify the type of data in an extended data block. It shall consist of two elements:

— a biometric organization identifier; and

— the data type identifier.

The biometric organization identifier shall be one of the biometric organization identifiers registered in accordance with ISO/IEC 19785-2. The data type identifier shall be assigned by the identified biometric organization.

8.3.5.1.2 Extended data

Abstract values: Any octet string

Contents: This data element shall include additional information that cannot be held by the other elements of the data object where this data element is embedded. The structure of this data is defined by the identified biometric organization.

9 Tagged binary encoding scheme

9.1 General

Each of the other parts of the ISO/IEC 39794 series may specify the abstract syntax of a tagged binary data interchange format in ASN.1, which is defined in ISO/IEC 8824-1. The tagged binary format shall be obtained by application of the ASN.1 Distinguished Encoding Rules (DER), which are defined in ISO/IEC 8825-1, to the given ASN.1 module. The DER encoding of each data object has three parts: tag octets that identify the data object, length octets that give the number of subsequent value octets, and the value octets.

9.2 Naming conventions for ASN.1 modules in the ISO/IEC 39794 series

9.2.1 ASN.1 module names

According to ISO/IEC 8824-1, each ASN.1 module begins with a module name that identifies the module for human beings. Each module name begins with an upper-case letter. For each ASN.1 module defined in the ISO/IEC 39794 series, the module name shall be a <filename> constructed according to the rules given in 7.14.

EXAMPLE iso-iec-39794-1-ed-1-v1.asn is the filename of the ASN.1 module named ISO-IEC-39794-1-ed-1-v1.

9.2.2 Object identifier for ASN.1 modules

Each ASN.1 module defined in the ISO/IEC 39794 series should be assigned an object identifier value for unambiguous identification. An object identifier value consists of a sequence of non-negative integers, corresponding to the path from the root to a node of the international object identifier tree defined in ISO/IEC 9834-1[2]. According to ISO/IEC 8824-1, each of the integers may be accompanied by a non-integer secondary identifier that does not begin with an upper-case letter.

For each ASN.1 module defined in the ISO/IEC 39794 series, the object identifier value shall consist of:

— the object identifier value of the international standard that contains the ASN.1 module;

— a component that identifies the edition of that standard;

— a component that identifies the version of that edition; and

— a component that identifies the ASN.1 module within that edition.

The object identifier value of the international standard is assigned in accordance with ISO/IEC 9834-1[2]. The other components are defined in the ISO/IEC 39794 series as follows:

— The relative object identifier value for the first edition of a part of the ISO/IEC 39794 series shall be {ed-1(1)}, and so forth.

— The relative object identifier value for the first version of an edition shall be {v1(1)}, and so forth.

— The ASN.1 modules within an edition of a part of the ISO/IEC 39794 series shall be assigned non-negative integers starting with 0. The secondary identifier shall be the <filenameroot> defined as per 7.14 with characters converted to lower case.

EXAMPLE 1 {iso(1) standard(0) iso-iec-39794(39794) part-1(1) ed-1(1) v1(1) iso-iec-39794-1(0)} is the object identifier for the ASN.1 module ISO-IEC-39794-1-ed-1-v1.

EXAMPLE 2 {iso(1) standard(0) iso-iec-39794(39794) part-1(1) ed-1(1) v1(1) iso-iec-39794-1-generalbdb(1)} is the object identifier for the ASN.1 module ISO-IEC-39794-1-GeneralBDB-ed-1-v1.

9.2.3 Type and component names

ASN.1 type names shall be in upper camel-case notation derived from subclause titles in the abstract data elements clause. For data types that apply to more than one abstract data element, the greatest common name part is used as type name.

EXAMPLE 1 The ASN.1 type name corresponding to “Capture date/time block” and “PAD data capture date/time block” is “CaptureDateTimeBlock”.

Names of components of a structured type shall be in lower camel-case notation derived from subclause titles in the abstract data elements clause. Name parts that replicate parts of the name of the containing structured type are omitted. Names of components of a structured type shall be unique within the scope of this type.

EXAMPLE 2 The component name corresponding to “PAD decision” within a “PADDataBlock” is “decision”.

If a subclause title started with a number (e.g. “2D”), then this name part should be moved towards the end.

EXAMPLE 3 The component name corresponding to “2D image representation block” within an “image representation block” is “block2D”.

The name part “Identifier” should be replaced with “Id”.

EXAMPLE 4 The component name corresponding to “PAD mechanism identifier block” within a “PAD score block” is “mechanismIdBlock”.

9.3 Prototypes

The ASN.1 types defined in the other parts of the ISO/IEC 39794 series are structurally similar to the prototypes specified below. The root type “XyzDataBlock” should be named after the title of the part of the ISO/IEC 39794 series, e.g., “FingerImageDataBlock” or “FaceImageDataBlock”. “XyzDataBlock” is to be assigned an application-class tag “[APPLICATION x]”. The tag number “x” should be the same as the number of the part of the ISO/IEC 39794 series that defines “XyzDataBlock”.

NOTE An ellipsis ... is an extension marker indicating an insertion point where future amendments or revisions can insert extension additions.

```
CaptureDeviceBlock ::= SEQUENCE {

   modelIdBlock             [0] RegistryIdBlock             OPTIONAL,
   technologyId             [1] CaptureDeviceTechnologyId   OPTIONAL,
   certificationIdBlocks    [2] CertificationIdBlocks       OPTIONAL,
   ...
}

RepresentationBlock ::= SEQUENCE {
   captureDeviceBlock     [0] CaptureDeviceBlock     OPTIONAL,
   captureDateTimeBlock   [1] CaptureDateTimeBlock   OPTIONAL,
   qualityBlocks          [2] QualityBlocks          OPTIONAL,
   pADDataBlock           [3] PADDataBlock           OPTIONAL,
   extendedDataBlocks     [4] ExtendedDataBlocks     OPTIONAL,
   ...
}

RepresentationBlocks ::= SEQUENCE OF RepresentationBlock

XyzDataBlock ::= [APPLICATION x] SEQUENCE {
   versionBlock           [0] VersionBlock,
   representationBlocks   [1] RepresentationBlocks,
   ...
}
```

9.4 Abstract syntax of common data types for the ISO/IEC 39794 series, in ASN.1

A.1 contains the ASN.1 module ISO-IEC-39794-1-ed-1-v1 that specifies common data types to be imported into the ASN.1 modules in the other parts of the ISO/IEC 39794 series.

9.5 Abstract syntax of general BDB, in ASN.1

The TLV-encoded patron format for use with smartcards or other tokens defined in ISO/IEC 19785-3[6] assigns the application-class tag $7F2E_{Hex}$ (“[APPLICATION 46]”) to a general BDB, no matter which particular format. Annex B contains the ASN.1 module ISO-IEC-39794-1-GeneralBDB-ed-1-v1 that specifies how to embed BDBs in formats defined in the other parts of the ISO/IEC 39794 series into a general BDB with tag $7F2E_{Hex}$ (“[APPLICATION 46]”).

9.6 Definition extension in ASN.1

9.6.1 General

The ASN.1 modules defined in the ISO/IEC 39794 series allow the extension of definitions in a backward and forward compatible way as specified in ISO/IEC 8824-1.

In new editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, new additional data elements and values may be added, and the use of previously defined optional data elements and values may be deprecated. Previously defined mandatory data elements must not be removed, and new mandatory data elements must not be added.

A parser for BDBs based on an old version of the format shall be able to read BDBs based on the new version of the format. Unknown data elements shall be ignored.

9.6.2 Addition of components to sequence types

Let a sequence type definition in the first version contain an extension marker "..." at the end of the list of component types.

EXAMPLE 1

```
RepresentationBlock ::= SEQUENCE {
   captureDeviceBlock       [0] CaptureDeviceBlock      OPTIONAL,
   captureDateTimeBlock     [1] CaptureDateTimeBlock    OPTIONAL,
   qualityBlocks            [2] QualityBlocks           OPTIONAL,
   pADDataBlock             [3] PADDataBlock            OPTIONAL,
   extendedDataBlocks       [4] ExtendedDataBlocks      OPTIONAL,
   imageRepresentation      [5] ImageRepresentation     OPTIONAL,
   ...
}
```

In later editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, additional components must be added after the extension marker. Additional components shall be grouped in extension addition groups within double square brackets. The opening double square bracket "[[" shall be followed by a version number for the extension addition group.

EXAMPLE 2

```
ColourSpaceCorrection ::= ....-- Type defined in the year 2025 version

RepresentationBlock ::= SEQUENCE {
   captureDeviceBlock        [0] CaptureDeviceBlock        OPTIONAL,
   captureDateTimeBlock      [1] CaptureDateTimeBlock      OPTIONAL,
   qualityBlocks             [2] QualityBlocks             OPTIONAL,
   pADDataBlock              [3] PADDataBlock              OPTIONAL,
   extendedDataBlocks        [4] ExtendedDataBlocks        OPTIONAL,
   imageRepresentation       [5] ImageRepresentation       OPTIONAL,
   ...
   [[2025:  -- Extension added in the year 2025 version
      colourSpaceCorrection   [6] ColourSpaceCorrection     OPTIONAL
   ]]
}
```

Further components may be added after previously added components.

EXAMPLE 3

```
ColourSpaceCorrection ::= ....-- Type defined in the year 2025 version

RepresentationBlock ::= SEQUENCE {
   captureDeviceBlock       [0] CaptureDeviceBlk          OPTIONAL,
   captureDateTimeBlock     [1] CaptureDateTimeBlock      OPTIONAL,
   qualityBlocks            [2] QualityBlocks             OPTIONAL,
   pADDataBlock             [3] PADDataBlock              OPTIONAL,
   extendedDataBlocks       [4] ExtendedDataBlocks        OPTIONAL,
```

```
   imageRepresentation     [5]ImageRepresentation       OPTIONAL,
   ...,
   [[2025:  -- Extension added in the year 2025 version
      colourSpaceCorrection [6]  ColourSpaceCorrection  OPTIONAL
   ]],
   [[2030:  -- Extensions added in the year 2030 version
      fNumber               [7]  REAL                   OPTIONAL,
      shutterSpeed          [8]  REAL                   OPTIONAL
   ]]
}
```

9.6.3 Addition of components to choice types

An extensible choice type definition shall be emulated by type definitions of the following form.

EXAMPLE

```
ImageRepresentationBase ::= CHOICE {
   block2D                 [0]   ImageRepresentation2DBlock,
   block3D                 [1]   ShapeRepresentation3DBlock
}

ImageRepresentationExtensionBlock ::= SEQUENCE {
   ...
}

ImageRepresentationBlock ::= CHOICE {
   base                   [0]  ImageRepresentationBase,
   extensionBlock         [1]  ImageRepresentationExtensionBlock
}
```

In later editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, additional components must be added after the extension marker.

9.6.4 Extension of an enumerated type with a new value

There are three categories of enumerated types:

— extensible enumerated types with a fallback value;

— extensible enumerated types without a fallback value; and

— non-extensible enumerated types.

For extensible enumerated types with a fallback value, decoders can use a fallback value that is defined in the original type instead of an unknown value. For this category of enumerations, the extension strategy incorporates a mandatory fallback value that has a type of the first version. In that way, if an unknown value is received by a version 1 decoder, that decoder is able to revert to the fallback value that it can decode.

EXAMPLE 1

```
AnnotationReasonCode ::= ENUMERATED {
   unknown(0),
   other(1),
   amputated(2),
   unableToPrint(3),
   bandaged(4),
   physicallyChallenged(5),
   diseased(6)
}

AnnotationReasonExtensionBlock ::= SEQUENCE {
   fallback          [0]   AnnotationReasonCode,
   ...
}
```

```
AnnotationReason ::= CHOICE {
   code                 [0]   AnnotationReasonCode,
   extensionBlock       [1]   AnnotationReasonExtensionBlock
}
```

In later editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, additional components must be added after the extension marker after the fallback data element.

EXAMPLE 2

```
AnnotationReasonCodeV2 ::= ENUMERATED { -- Additional code defined in the 2025 version
   frostbit(7)
}

AnnotationReasonExtensionBlock ::= SEQUENCE {
   fallback          [0]   AnnotationReasonCode,
   ...,
   [[2025: -- Extension added in the year 2025 version
      codeV2         [1]   AnnotationReasonCodeV2  OPTIONAL
   ]]
}
```

A second category of enumerated types is also extensible, but for them no fallback mechanism is possible. For this category, if a version 1 decoder receives an unknown value, it will not be able to interpret the data; but it could not anyway as it does not have the facility for dealing with the new information. In example 3, the decoder would fail to decode an image compressed using a codec it does not have.

EXAMPLE 3

```
ImageDataFormatCode ::= ENUMERATED {
   jpeg(0),
   jpeg2000(1),
   jpeg2000Lossy(2),
   jpeg2000Lossless(3),
   png(4)
}

ImageDataFormatExtensionBlock ::= SEQUENCE {
   ...
}

ImageDataFormat ::= CHOICE {
   code                 [0]   ImageDataFormatCode,
   extensionBlock       [1]   ImageDataFormatExtensionBlock
}
```

In later editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, additional components must be added after the extension marker.

EXAMPLE 4

```
ImageDataFormatCodeV2 ::= ENUMERATED { -- Additional code defined in the 2025 version
   jpegXR(5)
}

ImageDataFormatExtensionBlock ::= SEQUENCE {
   ...,
   [[2025: -- Extension added in the year 2025 version
      codeV2         [1]   ImageDataFormatCodeV2OPTIONAL
   ]]
}
```

Lastly, there are some enumerations for which extensibility is not desired. In this case, there is only a base enumeration.

EXAMPLE 5

```
UnitDimensionCode ::= ENUMERATED {
   inch(0),
   cm(1)
}
```

10 XML encoding scheme

10.1 General

Each of the other parts of the ISO/IEC 39794 series may specify an XML schema. The syntax of XML documents encoding biometric data shall be based on the XML schema definition in the appropriate part of the ISO/IEC 39794 series, not on an ASN.1 module and the ASN.1 XML encoding rules (XER).

10.2 Structure of XML schema definitions

The XML data type and element names shall correspond to data types and elements in the tagged binary format specified in the same part of the ISO/IEC 39794 series, if any. The syntax and semantics of the XML schemas describing the data interchange formats shall conform to W3C Recommendations, *XML Schema Parts 1 and 2*.

All parts of the ISO/IEC 39794 series shall identify optional and mandatory elements. An element is optional if the value of the minOccurs attribute is 0. An element is required to appear if the value of the minOccurs attribute is 1 or more. The default value for the minOccurs attribute is 1 (see W3C Recommendation, *XML Schema Parts 1*).

Anonymous inline type definitions shall not be used.

When a type and/or element has multiple child elements with mixed bounds, elements with the possibility of multiple occurrences shall be encapsulated inside a wrapper element. The name of the wrapper element shall be the plural form of the name of the encapsulated element.

EXAMPLE "RepresentationBlocksType" wraps multiple occurrences of "representationBlock". An element of "RepresentationBlocksType" is named "representationBlocks" and used within "XyzDataBlockType" (see 10.4).

```
<xs:complexType name="RepresentationBlocksType">
   <xs:sequence>
      <xs:element name="representationBlock" type="RepresentationBlockType"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

All white space directives shall be omitted in XML schema files through all parts of the ISO/IEC 39794 series.

10.3 Naming conventions for XML schema definitions in the ISO/IEC 39794 series

10.3.1 XML namespace names

For each XML schema defined in the ISO/IEC 39794 series, the namespace name shall be an HTTP URI derived from a <docidentifier> as described in 7.14.

10.3.2 Type and element names

Type names shall be in upper camel-case notation derived from subclause titles in the abstract data elements clause. Type names shall end with the word "Type". For data types that apply to more than one abstract data element, the greatest common name part is used as type name.

EXAMPLE 1 The type name corresponding to "Capture date/time block" and "PAD data capture date/time block" is "CaptureDateTimeBlockType".

Element names within a complex type shall be in lower camel-case notation derived from subclause titles in the abstract data elements clause. Name parts that replicate parts of the name of the containing structured type are omitted. Names of elements of a complex type shall be unique within the scope of this type.

EXAMPLE 2 The element name corresponding to "PAD decision" within "PADDataBlockType" is "decision".

If a subclause title started with a number (e.g. "2D"), then this name part should be moved towards the end.

EXAMPLE 3 The element name corresponding to "2D image representation block" within an "image representation block" is "block2D".

The name part "Identifier" should be replaced with "Id".

EXAMPLE 4 The element name corresponding to "PAD mechanism identifier block" within a "PAD score block" is "mechanismIdBlock".

10.4 Prototypes

The XML datatypes defined in the other parts of the ISO/IEC 39794 series are structurally similar to the prototypes specified in this subclause. The root element should be named after the title of the part, e.g. "fingerImageData" or "faceImageData".

NOTE "xyzData" is used exceptionally instead of "xyzDataBlock" to reflect the title of the corresponding part of the ISO/IEC 39794 series in the textual encoding for "xyz" data.

```
<xs:complexType name="CaptureDeviceBlockType">
   <xs:sequence>
      <xs:element name="modelIdBlock" type="cmn:RegistryIdBlockType" minOccurs="0"/>
      <xs:element name="technologyId" type="CaptureDeviceTechnologyIdType"
         minOccurs="0"/>
      <xs:element name="certificationIdBlocks" type="cmn:CertificationIdBlocksType"
         minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="RepresentationBlockType">
   <xs:sequence>
      <xs:element name="captureDeviceBlock" type="CaptureDeviceBlockType"
         minOccurs="0"/>
      <xs:element name="captureDateTimeBlock" type="cmn:CaptureDateTimeBlockType"
         minOccurs="0"/>
      <xs:element name="qualityBlocks" type="cmn:QualityBlocksType" minOccurs="0"/>
      <xs:element name="pADDataBlock" type="cmn:PADDataBlockType" minOccurs="0"/>
      <xs:element name="extendedDataBlocks" type="cmn:ExtendedDataBlocksType"
         minOccurs="0"/>
      <xs:any namespace="##other" processContents= "lax" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="RepresentationBlocksType">
   <xs:sequence>
      <xs:element name="representationBlock" type="RepresentationBlockType"
         maxOccurs="unbounded"/>
```

```
   </xs:sequence>
</xs:complexType>”

<xs:complexType name="XyzDataBlockType">
   <xs:sequence>
      <xs:element name="versionBlock" type="cmn:VersionBlockType"/>
      <xs:element name="representationBlocks" type="RepresentationBlocksType">
      <xs:any namespace="##other" processContents= "lax" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:element name="xyzData" type="XyzDataBlockType">
```

10.5 XML schema definition of common data types for the ISO/IEC 39794 series

A.2 contains an XML schema definition that specifies common data types to be imported into the XML schema definitions in the other parts of the ISO/IEC 39794 series.

10.6 Definition extension in XML

10.6.1 General

The XML schema definitions defined in the ISO/IEC 39794 series can be extended in a backward and forward compatible way.

In new editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, new data elements and values may be added, and the use of previously defined optional data elements and values may be deprecated. Previously defined mandatory data elements must not be removed, and new mandatory data elements must not be added. A new edition of, or amendment to, a part of the ISO/IEC 39794 series will maintain the XSDs specified in the previous editions as they stand in order to be able to import them.

10.6.2 Extending XML simple types

XML simple types do not support extension. If a type requires a future extension, the type needs to be implemented as XML complex type.

10.6.3 Extending XML sequence types

Extensible sequence types shall contain an <any> element to hold an extension. The <any> element shall be implemented as <xs:any namespace="##other" processContents="lax" minOccurs="0"/>. The <any> element shall use the namespace “##other” to avoid ambiguity with the current namespace. The processing of the extension content shall be lax, which means that the content shall be validated against the schema whenever possible. The <any> element is optional and shall not occur more than once as the default attribute value maxOccurs="1" applies.

EXAMPLE 1

```
<xs:schema
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
   xmlns="http://standards.iso.org/iso-iec/39794/-18"
   targetNamespace="http://standards.iso.org/iso-iec/39794/-18"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified"
   vc:minVersion="1.0">

   <xs:import namespace="http://standards.iso.org/iso-iec/39794/-1"
      schemaLocation="iso-iec-39794-1-ed-1-v1.xsd"/>

   <xs:complexType name="RepresentationBlockType">
      <xs:sequence>
         <xs:element name="captureDeviceBlock" type="CaptureDeviceBlockType"
            minOccurs="0"/>
```

```
            <xs:element name="captureDateTimeBlock" type="CaptureDateTimeBlockType"
               minOccurs="0"/>
            <xs:element name="qualityBlocks" type="QualityBlocksType" minOccurs="0"/>
            <xs:element name="pADDataBlock" type="PADDataBlockType" minOccurs="0"/>
            <xs:element name="extendedDataBlocks" type="ExtendedDataBlocksType"
               minOccurs="0"/>
            <xs:element name="imageRepresentation" type="ImageRepresentationType"
               minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>

</xs:schema>
```

To enable extension with multiple new data elements, the extension content shall also be of a sequence type. The extension shall carry the same name as the sequence type being extended. The name collision shall be resolved by using the namespaces of the base and extended schemas. An extension shall contain also the <any> element for further extension. An extended XSD shall import previous XSDs (via <xs:import namespace="{namespace}" schemaLocation="{XSD file path}"/>) to preserve their original namespaces.

EXAMPLE 2

```
<xs:schema
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
   xmlns="http://standards.iso.org/iso-iec/39794/-18/ed-1/v2"
   targetNamespace="http://standards.iso.org/iso-iec/39794/-18/ed-1/v2"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified"
   vc:minVersion="1.0">

   <xs:import namespace="http://standards.iso.org/iso-iec/39794/-18"
      schemaLocation="iso-iec-39794-18-ed-1-v1.xsd"/>

   <!-- ColourSpaceCorrectionType to be defined in edition 1, version 2 -->

   <xs:complexType name="RepresentationBlockType">
      <xs:sequence>
         <xs:element name="colourSpaceCorrection" type="ColourSpaceCorrectionType"
            minOccurs="0"/>
         <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>

</xs:schema>
```

10.6.4 Extending XML choice types

A choice type definition shall not contain an <any> element to avoid violation of the unique particle Attribution rule. An extensible choice type definition shall be emulated by type definitions of the following form.

EXAMPLE

```
<xs:complexType name="ImageRepresentationBaseType">
   <xs:choice>
      <xs:element name="block2D" type="ImageRepresentation2DBlockType"/>
      <xs:element name="block3D" type="ShapeRepresentation3DBlockType"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="ImageRepresentationExtensionBlockType">
   <xs:sequence>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="ImageRepresentationBlockType">
   <xs:choice>
      <xs:element name="base" type="ImageRepresentationBaseType"/>
      <xs:element name="extensionBlock" type="ImageRepresentationExtensionBlockType"/>
   </xs:choice>
</xs:complexType>
```

10.6.5 Extending XML enumerations

Extending an XML enumeration requires a well-defined policy to solve the forward compatibility. The use of the standard XML enumeration based on the simple string type is not supported. The enumeration values shall be implemented as choice of fixed integer values.

For extensible enumerated types with a fallback value, decoders can use the fallback value instead of an unknown value. For this category of enumerations, the extension strategy incorporates a mandatory fallback value that has a type of the base version. In that way, if an unknown value is received by a version 1 decoder, that decoder is able to revert to the fallback value that it can decode.

EXAMPLE 1

```
<xs:complexType name="AnnotationReasonCodeType">
   <xs:choice>
      <xs:element name="unknown" type="xs:int" fixed="0"/>
      <xs:element name="other" type="xs:int" fixed="1"/>
      <xs:element name="amputated" type="xs:int" fixed="2"/>
      <xs:element name="unableToPrint" type="xs:int" fixed="3"/>
      <xs:element name="bandaged" type="xs:int" fixed="4"/>
      <xs:element name="physicallyChallenged" type="xs:int" fixed="5"/>
      <xs:element name="diseased" type="xs:int" fixed="6"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="AnnotationReasonExtensionBlockType">
   <xs:sequence>
      <xs:element name="fallback" type="AnnotationReasonCodeType"/>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="AnnotationReasonType">
   <xs:choice>
      <xs:element name="code" type="AnnotationReasonCodeType"/>
      <xs:element name="extensionBlock" type="AnnotationReasonExtensionBlockType"/>
   </xs:choice>
</xs:complexType>
```

The <any> child element shall hold exactly one value defined in an extended XSD. The extended XSD shall be identified by the namespace.

In later editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, additional enumeration items may be defined.

EXAMPLE 2

```
<xs:complexType name="AnnotationReasonCodeType">
   <xs:choice>
      <xs:element name="frostbit" type="xs:int" fixed="7"/>
   </xs:choice>
</xs:complexType>
```

For extensible enumerated types without a fallback value, if a version 1 decoder receives a value of a new type, it will not be able to interpret the data; but it could not anyway as it does not have the facility for dealing with the new information. In EXAMPLE 3, the decoder would fail to decode an image compressed using a codec it does not have.

EXAMPLE 3

```
<xs:complexType name="ImageEncodingAlgorithmCodeType">
   <xs:choice>
      <xs:element name="jpeg" type="xs:int" fixed="0"/>
      <xs:element name="jpeg2000" type="xs:int" fixed="1"/>
      <xs:element name="jpeg2000Lossy " type="xs:int" fixed="2"/>
      <xs:element name="jpeg2000Lossless " type="xs:int" fixed="3"/>
      <xs:element name="png" type="xs:int" fixed="4"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="ImageEncodingAlgorithmExtensionBlockType">
   <xs:sequence>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="ImageEncodingAlgorithmType">
   <xs:choice>
      <xs:element name="code" type="ImageEncodingAlgorithmCodeType"/>
      <xs:element name="extensionBlock"
         type="ImageEncodingAlgorithmExtensionBlockType"/>
   </xs:choice>
</xs:complexType>
```

The <any> child element shall hold exactly one value defined in an extended XSD. The extended XSD shall be identified by the namespace.

In later editions of, or amendments to, a part of the ISO/IEC 39794 series, if needed, additional enumeration items may be defined.

EXAMPLE 4

```
<xs:complexType name="ImageEncodingAlgorithmCodeType">
   <xs:choice>
      <xs:element name="jpegXR" type="xs:int" fixed="5"/>
   </xs:choice>
</xs:complexType>
```

For non-extensible enumerated types, there is only a base enumeration.

EXAMPLE 5

```
<xs:complexType name="UnitDimensionCodeType">
   <xs:choice>
      <xs:element name="inch" type="xs:int" fixed="0"/>
      <xs:element name="cm" type="xs:int" fixed="1"/>
   </xs:choice>
</xs:complexType>
```

Annex A
(normative)

Formal specifications of common data types for the ISO/IEC 39794 series

A.1 Abstract syntax of common data types for the ISO/IEC 39794 series in ASN.1

This subclause contains the ASN.1 module ISO-IEC-39794-1-ed-1-v1 that specifies common data types to be imported into the ASN.1 modules in the other parts of the ISO/IEC 39794 series. Following the rules in 9.2.2, the ASN.1 module is assigned the unique object identifier {iso(1) standard(0) iso-iec-39794(39794) part-1(1) ed-1(1) v1(1) iso-iec-39794-1(0)}. The ASN.1 module ISO-IEC-39794-1-ed-1-v1 is available at http://standards.iso.org/iso-iec/39794/-1/ed-1/en.

NOTE The predefined date-time types are deliberately not used to allow optional date-time fields.

```
ISO-IEC-39794-1-ed-1-v1
   {iso(1) standard(0) iso-iec-39794(39794) part-1(1) ed-1(1) v1(1) iso-iec-39794-1(0)}

-- Use of ISO/IEC copyright in this Schema is licensed for the purpose of
-- developing, implementing, and using software based on this Schema, subject
-- to the following conditions:
--
-- * Software developed from this Schema must retain the Copyright Notice,
--   this list of conditions and the disclaimer below ("Disclaimer").
--
-- * Neither the name or logo of ISO or of IEC, nor the names of specific
--   contributors, may be used to endorse or promote software derived from
--   this Schema without specific prior written permission.
--
-- * The software developer shall attribute the Schema to ISO/IEC and
--   identify the ISO/IEC standard from which it is taken. Such attribution
--   (e.g., "This software makes use of the Schema from ISO/IEC 39794-1
--   within modifications permitted in the relevant ISO/IEC standard.
--   Please reproduce this note if possible."), may be placed in the
--   software itself or any other reasonable location.
--
-- The Disclaimer is:
-- THE SCHEMA ON WHICH THIS SOFTWARE IS BASED IS PROVIDED BY THE COPYRIGHT
-- HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
-- INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
-- AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
-- THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
-- INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
-- NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
-- DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
-- THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
-- (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
-- THE CODE COMPONENTS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- ================================================================
-- Common type definitions to be used in other parts of ISO/IEC 39794
-- ================================================================
VersionGeneration ::= INTEGER (3..65535)

VersionYear ::= INTEGER (2019..9999)

VersionBlock ::= SEQUENCE {
   generation           [0] VersionGeneration,
   year                 [1] VersionYear,
```

```
   ...
}

RegistryId ::= INTEGER (1..65535)

RegistryIdBlock ::= SEQUENCE {
   organization            [0] RegistryId,
   id                      [1] RegistryId
}

CertificationIdBlock ::= RegistryIdBlock

CertificationIdBlocks ::= SEQUENCE OF CertificationIdBlock

Year ::= INTEGER (0..9999)

Month ::= INTEGER (1..12)

Day ::= INTEGER (1..31)

Hour ::= INTEGER (0..23)

Minute ::= INTEGER (0..59)

Second ::= INTEGER (0..59)

Millisecond ::= INTEGER (0..999)

DateTimeBlock ::= SEQUENCE {
   year              [0] Year,
   month             [1] Month         OPTIONAL,
   day               [2] Day           OPTIONAL,
   hour              [3] Hour          OPTIONAL,
   minute            [4] Minute        OPTIONAL,
   second            [5] Second        OPTIONAL,
   millisecond       [6] Millisecond   OPTIONAL
}

CaptureDateTimeBlock ::= DateTimeBlock

Score ::= INTEGER (0..100)

ScoringErrorCode ::= ENUMERATED {
   failureToAssess(0)
}

ScoringErrorExtensionBlock ::= SEQUENCE {
   fallback          [0] ScoringErrorCode,
   ...
}

ScoringError ::= CHOICE {
   code              [0] ScoringErrorCode,
   extensionBlock    [1] ScoringErrorExtensionBlock
}

ScoreOrError ::= CHOICE {
   score             [0] Score,
   error             [1] ScoringError
}

QualityBlock ::= SEQUENCE {
   algorithmIdBlock  [0] RegistryIdBlock,
   scoreOrError      [1] ScoreOrError,
   ...
}

QualityBlocks ::= SEQUENCE OF QualityBlock

PADDecisionCode ::= ENUMERATED {
   noAttack(0),
```

```
   attack(1),
   failureToAssess(2)
}

PADDecisionExtensionBlock ::= SEQUENCE {
   fallback            [0] PADDecisionCode,
   ...
}

PADDecision ::= CHOICE {
   code                [0] PADDecisionCode,
   extensionBlock      [1] PADDecisionExtensionBlock
}

PADScoreBlock ::= SEQUENCE {
   mechanismIdBlock    [0] RegistryIdBlock,
   scoreOrError        [1] ScoreOrError,
   ...
}

PADScoreBlocks ::= SEQUENCE OF PADScoreBlock

ExtendedDataBlock ::= SEQUENCE {
   dataTypeIdBlock     [0] RegistryIdBlock,
   data                [1] OCTET STRING
}

ExtendedDataBlocks ::= SEQUENCE OF ExtendedDataBlock

PADExtendedDataBlocks ::= ExtendedDataBlocks

PADCaptureContextCode ::= ENUMERATED {
   enrolment(0),
   verification(1),
   identification(2)
}

PADCaptureContextExtensionBlock ::= SEQUENCE {
   fallback            [0] PADCaptureContextCode,
   ...
}

PADCaptureContext ::= CHOICE {
   code                [0] PADCaptureContextCode,
   extensionBlock      [1] PADCaptureContextExtensionBlock
}

PADSupervisionLevelCode ::= ENUMERATED {
   unknown(0),
   controlled(1),
   assisted(2),
   observed(3),
   unattended(4)
}

PADSupervisionLevelExtensionBlock ::= SEQUENCE {
   fallback            [0] PADSupervisionLevelCode,
   ...
}

PADSupervisionLevel ::= CHOICE {
   code                [0] PADSupervisionLevelCode,
   extensionBlock      [1] PADSupervisionLevelExtensionBlock
}

PADRiskLevel ::= Score

PADCriteriaCategoryCode ::= ENUMERATED {
   unknown(0),
   individual(1),
   common(2)
```

```
}

PADCriteriaCategoryExtensionBlock ::= SEQUENCE {
   fallback             [0] PADCriteriaCategoryCode,
   ...
}

PADCriteriaCategory ::= CHOICE {
   code                 [0] PADCriteriaCategoryCode,
   extensionBlock       [1] PADCriteriaCategoryExtensionBlock
}

PADChallenge ::= OCTET STRING

PADChallenges ::= SEQUENCE OF PADChallenge

PADDataBlock ::= SEQUENCE {
   decision             [0] PADDecision             OPTIONAL,
   scoreBlocks          [1] PADScoreBlocks          OPTIONAL,
   extendedDataBlocks   [2] PADExtendedDataBlocks   OPTIONAL,
   captureContext       [3] PADCaptureContext       OPTIONAL,
   supervisionLevel     [4] PADSupervisionLevel     OPTIONAL,
   riskLevel            [5] PADRiskLevel            OPTIONAL,
   criteriaCategory     [6] PADCriteriaCategory     OPTIONAL,
   parameter            [7] OCTET STRING            OPTIONAL,
   challenges           [8] PADChallenges           OPTIONAL,
   captureDateTimeBlock [9] CaptureDateTimeBlock    OPTIONAL,
   ...
}

CoordinateCartesian2DUnsignedShortBlock ::= SEQUENCE {
   x                [0] INTEGER (0..65535),
   y                [1] INTEGER (0..65535)
}

CoordinateCartesian3DUnsignedShortBlock ::= SEQUENCE {
   x                [0] INTEGER (0..65535),
   y                [1] INTEGER (0..65535),
   z                [2] INTEGER (0..65535)
}

CoordinateCartesian2DIntBlock ::= SEQUENCE {
   x                [0] INTEGER,
   y                [1] INTEGER
}

CoordinateCartesian3DIntBlock ::= SEQUENCE {
   x                [0] INTEGER,
   y                [1] INTEGER,
   z                [2] INTEGER
}

CoordinateCartesian2DDoubleBlock ::= SEQUENCE {
   x                [0] REAL,
   y                [1] REAL
}

CoordinateCartesian3DDoubleBlock ::= SEQUENCE {
   x                [0] REAL,
   y                [1] REAL,
   z                [2] REAL
}

CoordinatePolarIntBlock ::= SEQUENCE {
   radius           [0] INTEGER,
   azimuth          [1] INTEGER
}

CoordinatePolarDoubleBlock ::= SEQUENCE {
   radius           [0] REAL,
   azimuth          [1] REAL
```

```
}

CoordinateSphericalIntBlock ::= SEQUENCE {
   radius            [0] INTEGER,
   inclination       [1] INTEGER,
   azimuth           [2] INTEGER
}

CoordinateSphericalDoubleBlock ::= SEQUENCE {
   radius            [0] REAL,
   inclination       [1] REAL,
   azimuth           [2] REAL
}

CoordinateCylindricalIntBlock ::= SEQUENCE {
   radius            [0] INTEGER,
   azimuth           [1] INTEGER,
   height            [2] INTEGER
}

CoordinateCylindricalDoubleBlock ::= SEQUENCE {
   radius            [0] REAL,
   azimuth           [1] REAL,
   height            [2] REAL
}

END
```

A.2 XML schema definition of common data types for the ISO/IEC 39794 series

This subclause contains an XML schema definition that specifies common data types to be imported into the XML schema definitions in the other parts of the ISO/IEC 39794 series. The XML schema definition is available at http://standards.iso.org/iso-iec/39794/-1/ed-1/en.

NOTE The predefined date-time types are deliberately not used to allow optional date-time fields.

```
<?xml version="1.0" encoding="utf-8"?>
<!--Use of ISO/IEC copyright in this Schema is licensed for the purpose of developing,
implementing, and using software based on this Schema, subject to the following
conditions:

* Software developed from this Schema must retain the Copyright Notice, this list of
conditions and the disclaimer below ("Disclaimer").

* Neither the name or logo of ISO or of IEC, nor the names of specific contributors, may
be used to endorse or promote software derived from this Schema without specific prior
written permission.

* The software developer shall attribute the Schema to ISO/IEC and identify the ISO/IEC
standard from which it is taken. Such attribution (e.g., "This software makes use of the
Schema from ISO/IEC 39794-1 within modifications permitted in the relevant ISO/IEC
standard. Please reproduce this note if possible."), may be placed in the software itself
or any other reasonable location.

The Disclaimer is:
THE SCHEMA ON WHICH THIS SOFTWARE IS BASED IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY
WAY OUT OF THE USE OF THE CODE COMPONENTS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.-->

<xs:schema
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
```



```
xmlns="http://standards.iso.org/iso-iec/39794/-1"
targetNamespace="http://standards.iso.org/iso-iec/39794/-1"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
vc:minVersion="1.0">

<xs:simpleType name="VersionGenerationType">
   <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="3"/>
      <xs:maxInclusive value="65535"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="VersionYearType">
   <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="2019"/>
      <xs:maxInclusive value="9999"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="VersionBlockType">
   <xs:sequence>
      <xs:element name="generation" type="VersionGenerationType"/>
      <xs:element name="year" type="VersionYearType"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:simpleType name="RegistryIdType">
   <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="65535"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="RegistryIdBlockType">
   <xs:sequence>
      <xs:element name="organization" type="RegistryIdType"/>
      <xs:element name="id" type="RegistryIdType"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CertificationIdBlockType">
   <xs:complexContent>
      <xs:extension base="RegistryIdBlockType"/>
   </xs:complexContent>
</xs:complexType>

<xs:complexType name="CertificationIdBlocksType">
   <xs:sequence>
      <xs:element name="certificationIdBlock" type="CertificationIdBlockType"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:simpleType name="YearType">
   <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="9999"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="MonthType">
   <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="12"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="DayType">
   <xs:restriction base="xs:unsignedInt">
```

```
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="31"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="HourType">
   <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="23"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="MinuteType">
   <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="59"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SecondType">
   <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="59"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="MillisecondType">
   <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="999"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="DateTimeBlockType">
   <xs:sequence>
      <xs:element name="year" type="YearType"/>
      <xs:element name="month" type="MonthType" minOccurs="0"/>
      <xs:element name="day" type="DayType" minOccurs="0"/>
      <xs:element name="hour" type="HourType" minOccurs="0"/>
      <xs:element name="minute" type="MinuteType" minOccurs="0"/>
      <xs:element name="second" type="SecondType" minOccurs="0"/>
      <xs:element name="millisecond" type="MillisecondType" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CaptureDateTimeBlockType">
   <xs:complexContent>
      <xs:extension base="DateTimeBlockType"/>
   </xs:complexContent>
</xs:complexType>

<xs:simpleType name="ScoreType">
   <xs:restriction base="xs:unsignedShort">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="ScoringErrorCodeType">
   <xs:choice>
      <xs:element name="failureToAssess" type="xs:int" fixed="0"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="ScoringErrorExtensionBlockType">
   <xs:sequence>
      <xs:element name="fallback" type="ScoringErrorCodeType"/>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="ScoringErrorType">
   <xs:choice>
      <xs:element name="code" type="ScoringErrorCodeType"/>
      <xs:element name="extensionBlock" type="ScoringErrorExtensionBlockType"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="ScoreOrErrorType">
   <xs:choice>
      <xs:element name="score" type="ScoreType"/>
      <xs:element name="error" type="ScoringErrorType"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="QualityBlockType">
   <xs:sequence>
      <xs:element name="algorithmIdBlock" type="RegistryIdBlockType"/>
      <xs:element name="scoreOrError" type="ScoreOrErrorType"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="QualityBlocksType">
   <xs:sequence>
      <xs:element name="qualityBlock" type="QualityBlockType" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADDecisionCodeType">
   <xs:choice>
      <xs:element name="noAttack" type="xs:int" fixed="0"/>
      <xs:element name="attack" type="xs:int" fixed="1"/>
      <xs:element name="failureToAssess" type="xs:int" fixed="2"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="PADDecisionExtensionBlockType">
   <xs:sequence>
      <xs:element name="fallback" type="PADDecisionCodeType"/>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADDecisionType">
   <xs:choice>
      <xs:element name="code" type="PADDecisionCodeType"/>
      <xs:element name="extensionBlock" type="PADDecisionExtensionBlockType"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="PADScoreBlockType">
   <xs:sequence>
      <xs:element name="mechanismIdBlock" type="RegistryIdBlockType"/>
      <xs:element name="scoreOrError" type="ScoreOrErrorType"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADScoreBlocksType">
   <xs:sequence >
      <xs:element name="scoreBlock" type="PADScoreBlockType" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="ExtendedDataBlockType">
   <xs:sequence>
      <xs:element name="dataTypeIdBlock" type="RegistryIdBlockType"/>
      <xs:element name="data" type="xs:base64Binary"/>
   </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="ExtendedDataBlocksType">
   <xs:sequence >
      <xs:element name="extendedDataBlock" type="ExtendedDataBlockType"
         maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADExtendedDataBlocksType">
   <xs:complexContent>
      <xs:extension base="ExtendedDataBlocksType"/>
   </xs:complexContent>
</xs:complexType>

<xs:complexType name="PADCaptureContextCodeType">
   <xs:choice>
      <xs:element name="enrolment" type="xs:int" fixed="0"/>
      <xs:element name="verification" type="xs:int" fixed="1"/>
      <xs:element name="identification" type="xs:int" fixed="2"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="PADCaptureContextExtensionBlockType">
   <xs:sequence>
      <xs:element name="fallback" type="PADCaptureContextCodeType"/>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADCaptureContextType">
   <xs:choice>
      <xs:element name="code" type="PADCaptureContextCodeType"/>
      <xs:element name="extensionBlock" type="PADCaptureContextExtensionBlockType"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="PADSupervisionLevelCodeType">
   <xs:choice>
      <xs:element name="unknown" type="xs:int" fixed="0"/>
      <xs:element name="controlled" type="xs:int" fixed="1"/>
      <xs:element name="assisted" type="xs:int" fixed="2"/>
      <xs:element name="observed" type="xs:int" fixed="3"/>
      <xs:element name="unattended" type="xs:int" fixed="4"/>
   </xs:choice>
</xs:complexType>

<xs:complexType name="PADSupervisionLevelExtensionBlockType">
   <xs:sequence>
      <xs:element name="fallback" type="PADSupervisionLevelCodeType"/>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADSupervisionLevelType">
   <xs:choice>
      <xs:element name="code" type="PADSupervisionLevelCodeType"/>
      <xs:element name="extensionBlock" type="PADSupervisionLevelExtensionBlockType"/>
   </xs:choice>
</xs:complexType>

<xs:simpleType name="PADRiskLevelType">
   <xs:restriction base="ScoreType"/>
</xs:simpleType>

<xs:complexType name="PADCriteriaCategoryCodeType">
   <xs:choice>
      <xs:element name="unknown" type="xs:int" fixed="0"/>
      <xs:element name="individual" type="xs:int" fixed="1"/>
      <xs:element name="common" type="xs:int" fixed="2"/>
   </xs:choice>
</xs:complexType>
```

```
<xs:complexType name="PADCriteriaCategoryExtensionBlockType">
   <xs:sequence>
      <xs:element name="fallback" type="PADCriteriaCategoryCodeType"/>
      <xs:any namespace="##other" processContents="lax"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADCriteriaCategoryType">
   <xs:choice>
      <xs:element name="code" type="PADCriteriaCategoryCodeType"/>
      <xs:element name="extensionBlock" type="PADCriteriaCategoryExtensionBlockType"/>
   </xs:choice>
</xs:complexType>

<xs:simpleType name="PADChallengeType">
   <xs:restriction base="xs:base64Binary"/>
</xs:simpleType>

<xs:complexType name="PADChallengesType">
   <xs:sequence>
      <xs:element name="challenge" type="PADChallengeType" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="PADDataBlockType">
   <xs:sequence>
      <xs:element name="decision" type="PADDecisionType" minOccurs="0"/>
      <xs:element name="scoreBlocks" type="PADScoreBlocksType"
         minOccurs="0"/>
      <xs:element name="extendedDataBlocks" type="PADExtendedDataBlocksType"
         minOccurs="0"/>
      <xs:element name="captureContext" type="PADCaptureContextType" minOccurs="0"/>
      <xs:element name="supervisionLevel" type="PADSupervisionLevelType"
         minOccurs="0"/>
      <xs:element name="riskLevel" type="PADRiskLevelType" minOccurs="0"/>
      <xs:element name="criteriaCategory" type="PADCriteriaCategoryType"
         minOccurs="0"/>
      <xs:element name="parameter" type="xs:base64Binary" minOccurs="0"/>
      <xs:element name="challenges" type="PADChallengesType" minOccurs="0"/>
      <xs:element name="captureDateTimeBlock" type="CaptureDateTimeBlockType"
         minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CoordinateCartesian2DUnsignedShortBlockType">
   <xs:sequence>
      <xs:element name="x" type="xs:unsignedShort"/>
      <xs:element name="y" type="xs:unsignedShort"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CoordinateCartesian3DUnsignedShortBlockType">
   <xs:sequence>
      <xs:element name="x" type="xs:unsignedShort"/>
      <xs:element name="y" type="xs:unsignedShort"/>
      <xs:element name="z" type="xs:unsignedShort"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CoordinateCartesian2DIntBlockType">
   <xs:sequence>
      <xs:element name="x" type="xs:int"/>
      <xs:element name="y" type="xs:int"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="CoordinateCartesian3DIntBlockType">
   <xs:sequence>
      <xs:element name="x" type="xs:int"/>
      <xs:element name="y" type="xs:int"/>
```

```
            <xs:element name="z" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinateCartesian2DDoubleBlockType">
        <xs:sequence>
            <xs:element name="x" type="xs:double"/>
            <xs:element name="y" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinateCartesian3DDoubleBlockType">
        <xs:sequence>
            <xs:element name="x" type="xs:double"/>
            <xs:element name="y" type="xs:double"/>
            <xs:element name="z" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinatePolarIntBlockType">
        <xs:sequence>
            <xs:element name="radius" type="xs:int"/>
            <xs:element name="azimuth" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinatePolarDoubleBlockType">
        <xs:sequence>
            <xs:element name="radius" type="xs:double"/>
            <xs:element name="azimuth" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinateSphericalIntBlockType">
        <xs:sequence>
            <xs:element name="radius" type="xs:int"/>
            <xs:element name="inclination" type="xs:int"/>
            <xs:element name="azimuth" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinateSphericalDoubleBlockType">
        <xs:sequence>
            <xs:element name="radius" type="xs:double"/>
            <xs:element name="inclination" type="xs:double"/>
            <xs:element name="azimuth" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinateCylindricalIntBlockType">
        <xs:sequence>
            <xs:element name="radius" type="xs:int"/>
            <xs:element name="azimuth" type="xs:int"/>
            <xs:element name="height" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="CoordinateCylindricalDoubleBlockType">
        <xs:sequence>
            <xs:element name="radius" type="xs:double"/>
            <xs:element name="azimuth" type="xs:double"/>
            <xs:element name="height" type="xs:double"/>
        </xs:sequence>
    </xs:complexType>

</xs:schema>
```