ICAO

MACHINE READABLE TRAVEL DOCUMENTS TECHNICAL REPORT

ISO/IEC 39794-5 Application Profile for eMRTDs

Version – 1.00 | August 2023

ISO/IEC JTC1 SC17 WG3/TF5

FOR THE INTERNATIONAL CIVIL AVIATION ORGANIZATION

Release Control

Release	Date	Description
0.01	Jan 2023	Initial Draft for Application Profile
0.02	Feb 2023	First public release after discussions with SC17/WG3 ad-hoc group and SC37 ad-hoc group
0.03	April 2023	Comment Resolution from Haarlem meeting (April 4-5)
1.00	August 2023	Comment resolution of SC37 and Japan NB comments

Andy Hing	Singapore	Auctorizium
R Rajeshkumar	Singapore	Auctorizium

Table of contents

1. Scope

Doc 9303 uses ISO/IEC 19794:2005 to encode biometrics. In the new edition of ICAO Doc 9303, it is planned to move from using the ISO/IEC 19794 series for encoding biometric reference data to using the ISO/IEC 39794 series for this purpose.

To meet new and emerging market demands and to avoid future compatibility issues, ISO/IEC JTC 1/SC 37 developed the ISO/IEC 39794 series; the third generation of biometric data interchange formats, defining extensible biometric data interchange formats capable of including future extensions in a structured manner. 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. XML Schema Definitions form the basis for encoding biometric data in XML (Extensible Markup Language).The structure of the data format in this standard is not backward compatible with the previous generations. However, this new generation addresses, for the first time, a mechanism to maintain future extensions in a backwards and forwards compatible manner.

This technical report specifies an application profile for using [ISO/IEC 39794-1] and [ISO/IEC 39794-5] for storing Facial Images in DG2 in eMRTDs. Finger Images and Iris Images MUST be stored according to [ISO/IEC 39794-1], [ISO/IEC 39794-4] and [ISO/IEC 39794-6] in data group DG3 and DG4. This technical report does not specify an application profile for the storage of Fingerprint and Iris, and the original ISO/IEC documents [ISO/IEC 39794-1], [ISO/IEC 39794-4] and [ISO/IEC 39794-6] MUST be referenced for their storage.

1.1 Terminology

1.1.1 Technical report terminology

The key words “MUST”, “MUST NOT”, “SHALL”, “SHALL NOT”, “REQUIRED”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC 2119].

MUST
This word, or the terms “REQUIRED” or “SHALL”, means that the definition is an absolute requirement of the specification.

MUST NOT
This phrase, or the phrase “SHALL NOT”, means that the definition is an absolute prohibition of the specification.

SHOULD
This word, or the adjective “RECOMMENDED”, means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT
This phrase, or the phrase “NOT RECOMMENDED” means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY
This word, or the adjective “OPTIONAL”, means that an item is truly optional. One user may choose to include the item because a particular application requires it or because the user feels that it enhances the application while another user may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does

include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

CONDITIONAL The usage of an item is dependent on the usage of other items. It is therefore further qualified under which conditions the item is REQUIRED or RECOMMENDED. This is an additional key word used in Doc 9303 (not part of RFC 2119).

In case OPTIONAL features are implemented, they MUST be implemented as described in this Technical Report.

1.1.2 Terms and Definitions

Term	**Definition**
ASN.1	**Abstract Syntax Notation One**
DG	**Data Group**
eMRTD	**Electronic Machine Readable Travel Document**
BIT	**Biometric Information Template**
BDB	**Biometric Data Block**
DO	**Data Object**
XML	**Extensible Markup Language**

2. Data Groups for Biometric Data of eMRTD

Logical Data Structure of eMRTDs is specified in Doc 9303-10, which includes DG2 for face (mandatory), DG3 for fingerprint (optional), and DG4 for iris (optional). Each Data Group contains biometric data encoded in accordance with the international standards in order to keep international interoperability.

All of the above DGs, DG2, DG3 and DG4, MUST use the Biometric Information Template (BIT) group template with nested BITs (see Doc 9303-10). The nested BIT structure contains biometric data which can be encoded using 1 of 2 types of standards, ISO/IEC 19794 series first edition or ISO/IEC 39794 series.

Biometric data encoded in ISO/IEC 19794 series first edition is stored in the data object identified by tag ’5F2E’. Biometric data encoded in ISO/IEC 39794 series is stored in the data object identified by tag ’7F2E’.

Table 1: Tagsfor biometric data

Tag	Standard No.
5F2E	ISO/IEC 19794 series first edition
7F2E	ISO/IEC 39794 series

Biometric data encoded in the data object identified by tag ‘7F2E’ MUST use the data structure in the table below.

Table 2: Data Structure under DO’7F2E’

Tag	L	Value				
7F2E	Var.	Biometric data template defined in ISO/IEC 7816-11.				
		Tag	L	Value		
		A1	Var.	Biometric data in standardized format (Constructed)		
				Tag	L	Value
				64, 65 or 66	Var	DO defined in the ISO/IEC 39794 series. See Table 3.

Table 3: Tagsfor DOs defined in ISO/IEC 39794

Standard No.	Tag
ISO/IEC 39794-4	64
ISO/IEC 39794-5	65
ISO/IEC 39794-6	66

3. Encoding

3.1 Encoding Scheme

eMRTDs MUST encode the biometric references using the Tagged Binary Encoding Scheme which results from the application of ASN.1 Distinguished Encoding Rules (DER). eMRTDs MUST NOT use the XML Encoding Scheme.

ISO/IEC 39794 uses an extensible ASN.1 structure to encode the Biometric Data Block (BDB) and may add additional elements in the future. An inspection system based on an old version of the format SHALL be able to read BDBs based on the new version of the format and SHALL ignore unknown data elements.

This technical report makes some changes to the ASN.1 modules defined in [ISO/IEC 39794-1] and [ISO/IEC 39794-5]. The normative Annex A contains the ASN.1 modules that comply to this technical report.

3.2 Group biometric data template

[ISO/IEC 39794-5] allows for each Face Image Block to contain one or more geometric representations of a human face.

For eMRTDs, each Face Image Block MUST contain only a single geometric representation of a human face. For encoding multiple instances, the Biometric Information Template Group

Template MUST be used. This technical report specifies the requirements for only the first Facial Image stored in the Biometric Information Template Group Template.

4. Extensibility

[ISO/IEC 39794-1] defines the following types of extensibilities of the ASN.1 structure.

1. Addition of components to sequence types.
2. Addition of components to choice types.
3. Extension of an enumerated type with a new value
    a. Extensible enumerated type with a fallback value
    b. Extensible enumerated type without a fallback value

The application profile in this technical report modifies the structure of an Extensible Enumerated type with fallback value.

4.1 Extensible Enumerated Types with a Fallback Value

According to [ISO/IEC 39794-1], 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.

This TR requires that the original type is always encoded using a fallback value, even if there is no extension.

An example of extensible enumerated types with a fallback value is shown below. Extensible enumerated types with a fallback value used in biometric data structures MUST use the structure as detailed in the example below, in particular:

- The enumerated value MUST be encoded inside the fallback value (see `AnnotationReasonCode` and `fallback` in the example).
- The `fallback` MUST be encoded within the extension block (see `AnnotationReasonExtensionBlock` in the example). Hence, a extensible enumerated types with a fallback value is always encoded using a fallback value, even if there are no extensions.
- Only one value, which is the extension block, MUST be provided in the choice (see `AnnotationReason` in the example).
- If extensions are defined in the future, then for additional values an ENUMERATED type MUST be specified (see `AnnotationReasonCodeV2` and `AnnotationReasonCodeV3` in the example).
- These enumerated types MUST be added to the extension block and marked OPTIONAL (see `codeV2` and `codeV3` in the example).

Example

Note: This example is taken from ISO/IEC 39794-1, Section 9.6.4

```
AnnotationReason ::= CHOICE {
   extensionBlock [1] AnnotationReasonExtensionBlock
}
AnnotationReasonExtensionBlock ::= SEQUENCE {
   fallback [0] AnnotationReasonCode,
   ...,
   [[2025: -- Extension added in the year 2025 version
      codeV2 [1] AnnotationReasonCodeV2 OPTIONAL
   ]],
   [[2026: -- Extension added in the year 2026 version
      codeV3 [2] AnnotationReasonCodeV3 OPTIONAL
   ]]
}
AnnotationReasonCode ::= ENUMERATED {
   unknown(0),
   other(1),
   amputated(2),
   unableToPrint(3),
   bandaged(4),
   physicallyChallenged(5),
   diseased(6)
}
AnnotationReasonCodeV2 ::= ENUMERATED {
   -- Additional code defined in the 2025 version
   frostbit(7)
}
AnnotationReasonCodeV3 ::= ENUMERATED {
   -- Additional code defined in the 2026 version
   skinGraft(8)
}
```

Binary data without extension:

Hexadecimal Value: A1 05 A1 03 80 01 03
Hierarchical structure:

```
[1] A1          (choice)
  [1] A1        (extensionBlock)
    [0] 80 03   (fallback – unable to print)
```

Binary data with extension:

Hexadecimal Value: A1 08 A1 06 80 01 03 81 01 07
Hierarchical structure:

```
[1] A1                (choice)
  [1] A1              (extensionBlock)
    [0] 80 03         (fallback – unable to print)
    [1] 81 07         (codeV2 - frostbit)
```

5. Changes to Metadata Elements in Biometric Data of eMRTD

The values of metadata elements to capture and encoding of face images MUST follow ISO/IEC 39794-5 Annex D.1. except for the changes that are detailed in this section.

This section covers the changes made to the metadata elements in biometric data of eMRTD.

The changes specified in this section apply only to the first Facial Image stored in the Biometric Information Template Group Template.

5.1 Gender

The value of this element SHALL be one of the following:

- Other;
- Male;
- Female;

5.2 Image representation block

The Image representation block MUST be a 2D image representation block.

5.3 Image data format

The values allowed in this element are:

- Jpeg
- Jpeg2000 lossy
- Jpeg2000 lossless

All other values including ‘png’ SHALL NOT be used.

5.4 2D Face Image Kind

The value in this element MUST be ‘MRTD’. Other values are not allowed.

5.5 3D shape representation block

This element MUST NOT be used.

6. Reference documentation

The following documentation served as reference for this Technical Report:

[ISO/IEC 39794-1]	Information technology — Extensible biometric data interchange formats — Part 1: Framework (2019)
[ISO/IEC 39794-5]	Information technology — Extensible biometric data interchange formats — Part 5: Face image data (2019)
[ISO/IEC 19794]	Information technology — Biometric data interchange formats
[Doc 9303-10]	ICAO Doc 9303, 8th Edition, “Machine Readable Travel Documents”

Annex A ASN.1 for Application Profile (Normative)

A.1 MRTD application profile of the common data types ASN.1 module from ISO/IEC 39794-1

The ASN.1 module ID-ICAO-ISO-IEC-39794-1-ed-1-v1 is available at:

https://raw.githubusercontent.com/ICAO-TRIP-ISO-WG3/39794-5-AP/main/ID-ICAO-ISO-IEC-39794-1-ed-1-v1.asn

```
ID-ICAO-ISO-IEC-39794-1-ed-1-v1 { joint-iso-itu-t(2) international-
organizations(23) icao(136) mrtd(1) iso-iec-39794(39794) part-1(1) ed-1(1)
v1(1) iso-iec-39794-1(0) }

-- This ASN.1 module is derived from ISO-IEC-39794-1-ed-1-v1

-- 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 {
        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 {
        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 {
        extensionBlock                [1] PADCaptureContextExtensionBlock
}

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

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

PADSupervisionLevel ::= CHOICE {
        extensionBlock                [1] PADSupervisionLevelExtensionBlock
}

PADRiskLevel ::= Score

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

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

PADCriteriaCategory ::= CHOICE {
        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 MRTD application profile of face image data ASN.1 module from ISO/IEC 39794-5

The ASN.1 module ID-ICAO-ISO-IEC-39794-5-ed-1-v1 is available at:
https://raw.githubusercontent.com/ICAO-TRIP-ISO-WG3/39794-5-AP/main/id-icao-iso-iec-39794-5-ed-1-v1.asn

```
ID-ICAO-ISO-IEC-39794-5-ed-1-v1 { joint-iso-itu-t(2) international-
organizations(23) icao(136) mrtd(1) iso-iec-39794(39794) part-5(5) ed-1(1)
v1(1) iso-iec-39794-5(0) }

-- This ASN.1 module is derived from ISO-IEC-39794-5-ed-1-v1

-- 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 39794-5
--   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

        IMPORTS
                VersionBlock,
                CaptureDateTimeBlock,
                QualityBlocks,
                PADDataBlock,
                CoordinateCartesian2DUnsignedShortBlock,
                CoordinateCartesian3DUnsignedShortBlock,
                RegistryIdBlock,
                CertificationIdBlocks
        FROM ID-ICAO-ISO-IEC-39794-1-ed-1-v1;

        FaceImageDataBlock ::= [APPLICATION 5] SEQUENCE {
```

```
        versionBlock [0] VersionBlock,
        representationBlocks [1] RepresentationBlocks,
        ...
}

RepresentationBlocks ::= SEQUENCE SIZE (1) OF RepresentationBlock

RepresentationBlock ::= SEQUENCE {
        representationId [0] INTEGER (0..MAX),
        imageRepresentation [1] ImageRepresentation,
        captureDateTimeBlock [2] CaptureDateTimeBlock OPTIONAL,
        qualityBlocks [3] QualityBlocks OPTIONAL,
        padDataBlock [4] PADDataBlock OPTIONAL,
        sessionId [5] INTEGER (0..MAX) OPTIONAL,
        derivedFrom [6] INTEGER (0..MAX) OPTIONAL,
        captureDeviceBlock [7] CaptureDeviceBlock OPTIONAL,
        identityMetadataBlock [8] IdentityMetadataBlock OPTIONAL,
        landmarkBlocks [9] LandmarkBlocks OPTIONAL,
        ...
}

CaptureDeviceBlock ::= SEQUENCE {
        modelIdBlock [0] RegistryIdBlock OPTIONAL,
        certificationIdBlocks [1] CertificationIdBlocks OPTIONAL,
        ...
}

IdentityMetadataBlock ::= SEQUENCE {
        gender [0] Gender OPTIONAL,
        eyeColour [1] EyeColour OPTIONAL,
        hairColour [2] HairColour OPTIONAL,
        subjectHeight [3] SubjectHeight OPTIONAL,
        propertiesBlock [4] PropertiesBlock OPTIONAL,
        expressionBlock [5] ExpressionBlock OPTIONAL,
        poseAngleBlock [6] PoseAngleBlock OPTIONAL,
        ...
}

GenderCode ::= ENUMERATED {
        other (1),
        male (2),
        female (3)
}

GenderExtensionBlock ::= SEQUENCE {
        fallback [0] GenderCode,
        ...
}

Gender ::= CHOICE {
        extensionBlock [1] GenderExtensionBlock
}

EyeColourCode ::= ENUMERATED {
        unknown (0),
        other (1),
        black (2),
        blue (3),
        brown (4),
        grey (5),
        green (6),
        hazel (7),
        multi-coloured (8),
        pink (9)
}

EyeColourExtensionBlock ::= SEQUENCE {
        fallback [0] EyeColourCode,
        ...
}

EyeColour ::= CHOICE {
        extensionBlock [1] EyeColourExtensionBlock
}

HairColourCode ::= ENUMERATED {
        unknown (0),
        other (1),
        bald (2),
        black (3),
        blonde (4),
        brown (5),
        grey (6),
        white (7),
        red (8),
        knownColoured (9)
}

HairColourExtensionBlock ::= SEQUENCE {
        fallback [0] HairColourCode,
        ...
}

HairColour ::= CHOICE {
        extensionBlock [1] HairColourExtensionBlock
}

SubjectHeight ::= INTEGER (1..65535)

PropertiesBlock ::= SEQUENCE {
        glasses [0] BOOLEAN OPTIONAL,
        moustache [1] BOOLEAN OPTIONAL,
        beard [2] BOOLEAN OPTIONAL,
        teethVisible [3] BOOLEAN OPTIONAL,
        pupilOrIrisNotVisible [4] BOOLEAN OPTIONAL,
        mouthOpen [5] BOOLEAN OPTIONAL,
        leftEyePatch [6] BOOLEAN OPTIONAL,
        rightEyePatch [7] BOOLEAN OPTIONAL,
        darkGlasses [8] BOOLEAN OPTIONAL,
        biometricAbsent [9] BOOLEAN OPTIONAL,
        headCoveringsPresent [10] BOOLEAN OPTIONAL,
        ...
}

ExpressionBlock ::= SEQUENCE {
        neutral [0] BOOLEAN OPTIONAL,
        smile [1] BOOLEAN OPTIONAL,
        raisedEyebrows [2] BOOLEAN OPTIONAL,
        eyesLookingAwayFromTheCamera [3] BOOLEAN OPTIONAL,
        squinting [4] BOOLEAN OPTIONAL,
        frowning [5] BOOLEAN OPTIONAL,
        ...
}

PoseAngleBlock ::= SEQUENCE {
        yawAngleBlock [0] AngleDataBlock OPTIONAL,
        pitchAngleBlock [1] AngleDataBlock OPTIONAL,
        rollAngleBlock [2] AngleDataBlock OPTIONAL
}

AngleDataBlock ::= SEQUENCE {
        angleValue [0] AngleValue,
        angleUncertainty [1] AngleUncertainty OPTIONAL,
        ...
}

AngleValue ::= INTEGER (-180..180)

AngleUncertainty ::= INTEGER (0..180)

LandmarkBlocks ::= SEQUENCE OF LandmarkBlock

LandmarkBlock ::= SEQUENCE {
        landmarkKind [0] LandmarkKind,
        landmarkCoordinates [1] LandmarkCoordinates OPTIONAL,
        ...
}

LandmarkKind ::= CHOICE {
        base [0] LandmarkKindBase,
        extensionBlock [1] LandmarkKindExtensionBlock
}

LandmarkKindBase ::= CHOICE {
        mpeg4FeaturePoint [0] MPEG4FeaturePoint,
        anthropometricLandmark [1] AnthropometricLandmark
}

LandmarkKindExtensionBlock ::= SEQUENCE {
        ...
}

MPEG4FeaturePointCode ::= ENUMERATED {
        mpeg4PointCode-02-01 (0),
        mpeg4PointCode-02-02 (1),
        mpeg4PointCode-02-03 (2),
        mpeg4PointCode-02-04 (3),
        mpeg4PointCode-02-05 (4),
        mpeg4PointCode-02-06 (5),
        mpeg4PointCode-02-07 (6),
        mpeg4PointCode-02-08 (7),
        mpeg4PointCode-02-09 (8),
        mpeg4PointCode-02-10 (9),
        mpeg4PointCode-02-11 (10),
        mpeg4PointCode-02-12 (11),
        mpeg4PointCode-02-13 (12),
        mpeg4PointCode-02-14 (13),
        mpeg4PointCode-03-01 (14),
        mpeg4PointCode-03-02 (15),
        mpeg4PointCode-03-03 (16),
        mpeg4PointCode-03-04 (17),
        mpeg4PointCode-03-05 (18),
        mpeg4PointCode-03-06 (19),
        mpeg4PointCode-03-07 (20),
        mpeg4PointCode-03-08 (21),
        mpeg4PointCode-03-09 (22),
        mpeg4PointCode-03-10 (23),
        mpeg4PointCode-03-11 (24),
        mpeg4PointCode-03-12 (25),
               mpeg4PointCode-03-13 (26),
               mpeg4PointCode-03-14 (27),
               mpeg4PointCode-04-01 (28),
               mpeg4PointCode-04-02 (29),
               mpeg4PointCode-04-03 (30),
               mpeg4PointCode-04-04 (31),
               mpeg4PointCode-04-05 (32),
               mpeg4PointCode-04-06 (33),
               mpeg4PointCode-05-01 (34),
               mpeg4PointCode-05-02 (35),
               mpeg4PointCode-05-03 (36),
               mpeg4PointCode-05-04 (37),
               mpeg4PointCode-06-01 (38),
               mpeg4PointCode-06-02 (39),
               mpeg4PointCode-06-03 (40),
               mpeg4PointCode-06-04 (41),
               mpeg4PointCode-07-01 (42),
               mpeg4PointCode-08-01 (43),
               mpeg4PointCode-08-02 (44),
               mpeg4PointCode-08-03 (45),
               mpeg4PointCode-08-04 (46),
               mpeg4PointCode-08-05 (47),
               mpeg4PointCode-08-06 (48),
               mpeg4PointCode-08-07 (49),
               mpeg4PointCode-08-08 (50),
               mpeg4PointCode-08-09 (51),
               mpeg4PointCode-08-10 (52),
               mpeg4PointCode-09-01 (53),
               mpeg4PointCode-09-02 (54),
               mpeg4PointCode-09-03 (55),
               mpeg4PointCode-09-04 (56),
               mpeg4PointCode-09-05 (57),
               mpeg4PointCode-09-06 (58),
               mpeg4PointCode-09-07 (59),
               mpeg4PointCode-09-08 (60),
               mpeg4PointCode-09-09 (61),
               mpeg4PointCode-09-10 (62),
               mpeg4PointCode-09-11 (63),
               mpeg4PointCode-09-12 (64),
               mpeg4PointCode-09-13 (65),
               mpeg4PointCode-09-14 (66),
               mpeg4PointCode-09-15 (67),
               mpeg4PointCode-10-01 (68),
               mpeg4PointCode-10-02 (69),
               mpeg4PointCode-10-03 (70),
               mpeg4PointCode-10-04 (71),
               mpeg4PointCode-10-05 (72),
               mpeg4PointCode-10-06 (73),
               mpeg4PointCode-10-07 (74),
               mpeg4PointCode-10-08 (75),
               mpeg4PointCode-10-09 (76),
               mpeg4PointCode-10-10 (77),
               mpeg4PointCode-11-01 (78),
               mpeg4PointCode-11-02 (79),
               mpeg4PointCode-11-03 (80),
               mpeg4PointCode-11-04 (81),
               mpeg4PointCode-11-05 (82),
               mpeg4PointCode-11-06 (83),
               mpeg4PointCode-12-01 (84),
               mpeg4PointCode-12-02 (85),
               mpeg4PointCode-12-03 (86),
               mpeg4PointCode-12-04 (87)
          }
```

```
        MPEG4FeaturePointExtensionBlock ::= SEQUENCE {
                fallback [0] MPEG4FeaturePointCode,
                ...
        }

        MPEG4FeaturePoint ::= CHOICE {
                extensionBlock [1] MPEG4FeaturePointExtensionBlock
        }

        AnthropometricLandmark ::= CHOICE {
                base [0] AnthropometricLandmarkBase,
                extensionBlock [1] AnthropometricLandmarkExtensionBlock
        }

        AnthropometricLandmarkBase ::= CHOICE {
                anthropometricLandmarkName [0] AnthropometricLandmarkName,
                anthropometricLandmarkPointName [1]
AnthropometricLandmarkPointName,
                anthropometricLandmarkPointId [2]
AnthropometricLandmarkPointId
        }

        AnthropometricLandmarkExtensionBlock ::= SEQUENCE {
                ...
        }

        AnthropometricLandmarkNameCode ::= ENUMERATED {
                vertex                      (0),
                glabella                    (1),
                opisthocranion              (2),
                eurionLeft                  (3),
                eurionRight                 (4),
                frontotemporaleLeft         (5),
                frontotemporaleRight        (6),
                trichion                    (7),
                zygionLeft                  (8),
                zygionRight                 (9),
                gonionLeft                  (10),
                gonionRight                 (11),
                sublabiale                  (12),
                pogonion                    (13),
                menton                      (14),
                condylionLateraleLeft       (15),
                condylionLateraleRight      (16),
                endocanthionLeft            (17),
                endocanthionRight           (18),
                exocanthionLeft             (19),
                exocanthionRight            (20),
                centerPointOfPupilLeft      (21),
                centerPointOfPupilRight     (22),
                orbitaleLeft                (23),
                orbitaleRight               (24),
                palpebraleSuperiusLeft      (25),
                palpebraleSuperiusRight     (26),
                palpebraleInferiusLeft      (27),
                palpebraleInferiusRight     (28),
                orbitaleSuperiusLeft        (29),
                orbitaleSuperiusRight       (30),
                superciliareLeft            (31),
                superciliareRight           (32),
                nasion                      (33),
                sellion                     (34),
        alareLeft                     (35),
        alareRight                    (36),
        pronasale                     (37),
        subnasale                     (38),
        subalare                      (39),
        alarCurvatureLeft             (40),
        alarCurvatureRight            (41),
        maxillofrontale               (42),
        christaPhiltraLandmarkLeft    (43),
        christaPhiltraLandmarkRight   (44),
        labialeSuperius               (45),
        labialeInferius               (46),
        cheilionLeft                  (47),
        cheilionRight                 (48),
        stomion                       (49),
        superauraleLeft               (50),
        superauraleRight              (51),
        subauraleLeft                 (52),
        subauraleRight                (53),
        preaurale                     (54),
        postaurale                    (55),
        otobasionSuperiusLeft         (56),
        otobasionSuperiusRight        (57),
        otobasionInferius             (58),
        porion                        (59),
        tragion                       (60)
}

AnthropometricLandmarkNameExtensionBlock ::= SEQUENCE {
        fallback [0] AnthropometricLandmarkNameCode,
        ...
}

AnthropometricLandmarkName ::= CHOICE {
        extensionBlock [1] AnthropometricLandmarkNameExtensionBlock
}

AnthropometricLandmarkPointNameCode ::= ENUMERATED {
        pointCode-01-01 (0),
        pointCode-01-02 (1),
        pointCode-01-05 (2),
        pointCode-01-06 (3),
        pointCode-01-07 (4),
        pointCode-01-08 (5),
        pointCode-01-09 (6),
        pointCode-02-01 (7),
        pointCode-02-02 (8),
        pointCode-02-03 (9),
        pointCode-02-04 (10),
        pointCode-02-05 (11),
        pointCode-02-06 (12),
        pointCode-02-07 (13),
        pointCode-02-09 (14),
        pointCode-02-10 (15),
        pointCode-03-01 (16),
        pointCode-03-02 (17),
        pointCode-03-03 (18),
        pointCode-03-04 (19),
        pointCode-03-05 (20),
        pointCode-03-06 (21),
        pointCode-03-07 (22),
        pointCode-03-08 (23),
        pointCode-03-09 (24),
                pointCode-03-10 (25),
                pointCode-03-11 (26),
                pointCode-03-12 (27),
                pointCode-04-01 (28),
                pointCode-04-02 (29),
                pointCode-04-03 (30),
                pointCode-04-04 (31),
                pointCode-05-01 (32),
                pointCode-05-02 (33),
                pointCode-05-03 (34),
                pointCode-05-04 (35),
                pointCode-05-06 (36)
        }

        AnthropometricLandmarkPointNameExtensionBlock ::= SEQUENCE {
                fallback [0] AnthropometricLandmarkPointNameCode,
                ...
        }

        AnthropometricLandmarkPointName ::= CHOICE {
                extensionBlock [1]
AnthropometricLandmarkPointNameExtensionBlock
        }

        AnthropometricLandmarkPointIdCode ::= ENUMERATED {
                v         (0),
                g         (1),
                op        (2),
                eu-left   (3),
                eu-right  (4),
                ft-left   (5),
                ft-right  (6),
                tr        (7),
                zy-left   (8),
                zy-right  (9),
                go-left   (10),
                go-right  (11),
                sl        (12),
                pg        (13),
                gn        (14),
                cdl-left  (15),
                cdl-right (16),
                en-left   (17),
                en-right  (18),
                ex-left   (19),
                ex-right  (20),
                p-left    (21),
                p-right   (22),
                or-left   (23),
                or-right  (24),
                ps-left   (25),
                ps-right  (26),
                pi-left   (27),
                pi-right  (28),
                os-left   (29),
                os-right  (30),
                sci-left  (31),
                sci-right (32),
                n         (33),
                se        (34),
                al-left   (35),
                al-right  (36),
                prn       (37),
                sn        (38),
                sbal      (39),
                ac-left   (40),
                ac-right  (41),
                mf-left   (42),
                mf-right  (43),
                cph-left  (44),
                cph-right (45),
                ls        (46),
                li        (47),
                ch-left   (48),
                ch-right  (49),
                sto       (50),
                sa-left   (51),
                sa-right  (52),
                sba-left  (53),
                sba-right (54),
                pra-left  (55),
                pra-right (56),
                pa        (57),
                obs-left  (58),
                obs-right (59),
                obi       (60),
                po        (61),
                t         (62)
        }

        AnthropometricLandmarkPointIdExtensionBlock ::= SEQUENCE {
                fallback [0] AnthropometricLandmarkPointIdCode,
                ...
        }

        AnthropometricLandmarkPointId ::= CHOICE {
                extensionBlock [1]
AnthropometricLandmarkPointIdExtensionBlock
        }

        LandmarkCoordinates ::= CHOICE {
                base [0] LandmarkCoordinatesBase,
                extensionBlock [1] LandmarkCoordinatesExtensionBlock
        }

        LandmarkCoordinatesBase ::= CHOICE {
                coordinateCartesian2DBlock [0]
CoordinateCartesian2DUnsignedShortBlock,
                coordinateTextureImageBlock [1] CoordinateTextureImageBlock,
                coordinateCartesian3DBlock [2]
CoordinateCartesian3DUnsignedShortBlock
        }

        LandmarkCoordinatesExtensionBlock ::= SEQUENCE {
                ...
        }

        CoordinateTextureImageBlock ::= SEQUENCE {
                uInPixel [0] INTEGER (0..MAX),
                vInPixel [1] INTEGER (0..MAX)
        }

        ImageRepresentation ::= CHOICE {
                base [0] ImageRepresentationBase,
                extensionBlock [1] ImageRepresentationExtensionBlock
        }
        ImageRepresentationBase ::= CHOICE {
                imageRepresentation2DBlock [0] ImageRepresentation2DBlock
        }

        ImageRepresentationExtensionBlock ::= SEQUENCE {
                ...
        }

        ImageRepresentation2DBlock ::= SEQUENCE {
                representationData2D [0] OCTET STRING,
                imageInformation2DBlock [1] ImageInformation2DBlock,
                captureDevice2DBlock [2] CaptureDevice2DBlock OPTIONAL,
                ...
        }

        CaptureDevice2DBlock ::= SEQUENCE {
                captureDeviceSpectral2DBlock [0]
CaptureDeviceSpectral2DBlock OPTIONAL,
                captureDeviceTechnologyId2D [1] CaptureDeviceTechnologyId2D
OPTIONAL,
                ...
        }

        CaptureDeviceSpectral2DBlock ::= SEQUENCE {
                whiteLight [0] BOOLEAN OPTIONAL,
                nearInfrared [1] BOOLEAN OPTIONAL,
                thermal [2] BOOLEAN OPTIONAL,
                ...
        }

        CaptureDeviceTechnologyId2DCode ::= ENUMERATED {
                unknown (0),
                staticPhotographFromUnknownSource (1),
                staticPhotographFromDigitalStillImageCamera (2),
                staticPhotographFromScanner (3),
                videoFrameFromUnknownSource (4),
                videoFrameFromAnalogueVideoCamera (5),
                videoFrameFromDigitalVideoCamera (6)
        }

        CaptureDeviceTechnologyId2DExtensionBlock ::= SEQUENCE {
                fallback [0] CaptureDeviceTechnologyId2DCode,
                ...
        }

        CaptureDeviceTechnologyId2D ::= CHOICE {
                extensionBlock [1] CaptureDeviceTechnologyId2DExtensionBlock
        }

        ImageInformation2DBlock ::= SEQUENCE {
                imageDataFormat [0] ImageDataFormat,
                faceImageKind2D [1] FaceImageKind2D OPTIONAL,
                postAcquisitionProcessingBlock [2]
PostAcquisitionProcessingBlock OPTIONAL,
                lossyTransformationAttempts [3] LossyTransformationAttempts
OPTIONAL,
                cameraToSubjectDistance [4] CameraToSubjectDistance
OPTIONAL,
                sensorDiagonal [5] SensorDiagonal OPTIONAL,
                lensFocalLength [6] LensFocalLength OPTIONAL,
                imageSizeBlock [7] ImageSizeBlock OPTIONAL,
                imageFaceMeasurementsBlock [8] ImageFaceMeasurementsBlock
OPTIONAL,
                imageColourSpace [9] ImageColourSpace OPTIONAL,
                referenceColourMappingBlock [10] ReferenceColourMappingBlock
OPTIONAL,
                ...
        }

        FaceImageKind2DCode ::= ENUMERATED {
                mrtd (0)
        }

        FaceImageKind2DExtensionBlock ::= SEQUENCE {
                fallback [0] FaceImageKind2DCode,
                ...
        }

        FaceImageKind2D ::= CHOICE {
                extensionBlock [1] FaceImageKind2DExtensionBlock
        }

        PostAcquisitionProcessingBlock ::= SEQUENCE {
                rotated [0] BOOLEAN OPTIONAL,
                cropped [1] BOOLEAN OPTIONAL,
                downSampled [2] BOOLEAN OPTIONAL,
                whiteBalanceAdjusted [3] BOOLEAN OPTIONAL,
                multiplyCompressed [4] BOOLEAN OPTIONAL,
                interpolated [5] BOOLEAN OPTIONAL,
                contrastStretched [6] BOOLEAN OPTIONAL,
                poseCorrected [7] BOOLEAN OPTIONAL,
                multiViewImage [8] BOOLEAN OPTIONAL,
                ageProgressed [9] BOOLEAN OPTIONAL,
                superResolutionProcessed [10] BOOLEAN OPTIONAL,
                normalised [11] BOOLEAN OPTIONAL,
                ...
        }

        LossyTransformationAttemptsCode ::= ENUMERATED {
                unknown (0),
                zero (1),
                one (2),
                moreThanOne (3)
        }

        LossyTransformationAttemptsExtensionBlock ::= SEQUENCE {
                fallback [0] LossyTransformationAttemptsCode,
                ...
        }

        LossyTransformationAttempts ::= CHOICE {
                extensionBlock [1] LossyTransformationAttemptsExtensionBlock
        }

        ImageDataFormatCode ::= ENUMERATED {

                jpeg (2),
                jpeg2000Lossy (3),
                jpeg2000Lossless (4)
        }

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

        CameraToSubjectDistance ::= INTEGER (0..50000)

        SensorDiagonal ::= INTEGER (0..2000)

        LensFocalLength ::= INTEGER (0..2000)

        ImageSizeBlock ::= SEQUENCE {
                width [0] ImageSize,
                height [1] ImageSize
        }

        ImageSize ::= INTEGER (0..65535)

        ImageFaceMeasurementsBlock ::= SEQUENCE {
                imageHeadWidth [0] INTEGER (0..MAX) OPTIONAL,
                imageInterEyeDistance [1] INTEGER (0..MAX) OPTIONAL,
                imageEyeToMouthDistance [2] INTEGER (0..MAX) OPTIONAL,
                imageHeadLength [3] INTEGER (0..MAX) OPTIONAL,
                ...
        }

        ImageColourSpaceCode ::= ENUMERATED {
                unknown (0),
                other (1),
                rgb24Bit (2),
                rgb48Bit (3),
                yuv422 (4),
                greyscale8Bit (5),
                greyscale16Bit (6)
        }

        ImageColourSpaceExtensionBlock ::= SEQUENCE {
                fallback [0] ImageColourSpaceCode,
                ...
        }

        ImageColourSpace ::= CHOICE {
                extensionBlock [1] ImageColourSpaceExtensionBlock
        }

        ReferenceColourMappingBlock ::= SEQUENCE {
                referenceColourSchema [0] OCTET STRING OPTIONAL,
                referenceColourDefinitionAndValueBlocks [1]
ReferenceColourDefinitionAndValueBlocks OPTIONAL,
                ...
        }

        ReferenceColourDefinitionAndValueBlocks ::= SEQUENCE OF
ReferenceColourDefinitionAndValueBlock

        ReferenceColourDefinitionAndValueBlock ::= SEQUENCE {
                referenceColourDefinition [0] OCTET STRING OPTIONAL,
                referenceColourValue [1] OCTET STRING OPTIONAL,
                ...
        }
```

```
END
```