DRAFT INTERNATIONAL STANDARD

ISO/IEC DIS 30108-2

ISO/IEC JTC **1**/SC **37** Secretariat: **ANSI**

Voting begins on:
2022-10-31

Voting terminates on:
2023-01-23

Information technology – Identity attributes verification services —

Part 2:
RESTful specification

ICS: 35.240.15

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 30108-2:2022(E)

COPYRIGHT PROTECTED DOCUMENT

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

Published in Switzerland

Contents

Page

Foreword ..... **v**

Introduction ..... **vi**

1 Scope **1**

2 Normative References **1**

3 Terms and Definitions **1**

4 Symbols and Abbreviated Terms **1**

5 Conformance **2**

6 System Context **2**
6.1 Data formats 2

7 Biometric Identity Assurance Services **2**
7.1 Introduction 2
7.2 Primitive Services 2
7.2.1 Introduction 2
7.2.2 Add Subject To Gallery 2
7.2.3 Check Quality 4
7.2.4 Classify Biometric Data 7
7.2.5 Create Encounter 8
7.2.6 Create Subject 10
7.2.7 Delete Biographic Data 11
7.2.8 Delete Biometric Data 13
7.2.9 Delete Document Data 15
7.2.10 Delete Encounter 17
7.2.11 Delete Subject 18
7.2.12 Delete Subject From Gallery 19
7.2.13 Get Identify Subject Results 21
7.2.14 Identify Subject 22
7.2.15 List Biographic Data 25
7.2.16 List Biometric Data 27
7.2.17 List Document Data 29
7.2.18 Perform Fusion 31
7.2.19 Query Capabilities 33
7.2.20 Retrieve Biographic Data 34
7.2.21 Retrieve Biometric Data 36
7.2.22 Retrieve Document Data 38
7.2.23 Set Biographic Data 40
7.2.24 Set Biometric Data 42
7.2.25 Set Document Data 44
7.2.26 Transform Biometric Data 47
7.2.27 Update Biographic Data 49
7.2.28 Update Biometric Data 50
7.2.29 Update Document Data 52
7.2.30 Verify Subject 54
7.3 Aggregated Services 56
7.3.1 Delete 56
7.3.2 Enrol 59
7.3.3 Get Deletion Results 61
7.3.4 Get Enrol Results 63
7.3.5 Get Identify Results 64
7.3.6 Get Update Results 66
7.3.7 Get Verify Results 67
7.3.8 Identify 69
7.3.9 Retrieve Data 71

7.3.10 Update ... 73
7.3.11 Verify ... 75

8 Data Elements and Data Types ... 79
8.1 Introduction ... 79
8.2 Biographic Data ... 79
8.2.1 Biographic Data Item Type ... 79
8.2.2 Biographic Data List Type ... 80
8.2.3 Biographic Data Set Type ... 80
8.2.4 Biographic Data Type ... 81
8.3 Biometric Data ... 81
8.3.1 Biometric Data Element Type ... 81
8.3.2 Biometric Data List Type ... 82
8.3.3 Biometric Type ... 82
8.3.4 CBEFF BIR Type ... 83
8.3.5 CBEFF BIR List Type ... 83
8.4 Candidate Lists ... 84
8.4.1 Candidate List Type ... 84
8.4.2 Candidate Type ... 84
8.5 Document Data ... 85
8.5.1 Document Data List Type ... 85
8.5.2 Document Data Type ... 85
8.6 Capabilities ... 86
8.6.1 Capability List Type ... 87
8.6.2 Capability Type ... 87
8.7 Fusion Information ... 97
8.7.1 Fusion Identity List Type ... 97
8.7.2 Fusion Information List Type ... 97
8.7.3 Fusion Information Type ... 97
8.8 Other Data Types ... 98
8.8.1 Encounter Category Type ... 98
8.8.2 Encounter List Type ... 99
8.8.3 Information Type ... 99
8.8.4 List Filter Type ... 100
8.8.5 Option Type ... 100
8.8.6 Processing Options Type ... 100
8.8.7 Token Type ... 101

9 Error Handling and Notification ... 101
9.1 Introduction ... 101
9.2 Generic HTTP responses ... 101
9.3 Error Condition Codes ... 102
9.4 YAML specification ... 105

10 Security ... 114

Annex A (normative) **OpenAPI specification ... 115**

Annex B (normative) **CBEFF Patron Format in YAML ... 117**

Annex C (informative) **Implementation example ... 122**

Bibliography ... 127

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 30108-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 37, *Biometrics*.

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revised.

ISO/IEC 30108 consists of the following parts, under the general title *Information Technology — Identity Attributes Verification Services*:

— *Part 1: IAVS Services*

— *Part 2: RESTful specification*

Introduction

This part of ISO/IEC 30108 defines the architecture, operations, data elements, and basic requirements for identity attributes verification services – a framework for the implementation of generic, identity services within a service-oriented environment. An identity in the context of IAVS comprises a subject, biographic data, and biometric data. Other parts are intended to define specific IAVS implementations (or bindings) within specific environments – for example, SOAP Web services.

IAVS services are generic in nature, being modality neutral and not targeted at any particular business application. These services include those related to identity data management, transformation, and biometric comparison. Services are invoked by a IAVS requester and implemented by a IAVS service provider (responder). It does not prescribe the architecture or business logic of either the requester or service provider.

Two categories of identity services are defined – primitive and aggregate. Primitive services are more atomic and well-defined, whereas the aggregate services tend to be higher level and enable more flexibility on the part of the IAVS service provider.

Two identity models are also defined – person-centric and encounter-based. Person-centric systems maintain a single up-to-date record (set of data) for a given subject, whereas an encounter-based system retains data related to each interaction the subject has with the system.

This international standard represents a version of IAVS defined in ISO/IEC 30108-1, but using a RESTful approach.

Information technology – Identity attributes verification services —

Part 2:
RESTful specification

1 Scope

This part of ISO/IEC 30108 implements the specification provided in part 1 using Representational State Transfer (REST). Therefore, the scope in part 1 applies to this part of ISO/IEC 30108.

2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below, in addition to those stated in ISO/IEC 30108-1.

ECMA Standard ECMA-404 (2017 2nd Edition) The JSON Data Interchange Format ISO\IEC 21778

Hyper Text Transfer Protocol, (HTTP/1.1) RFC 7230 [https://httpwg.org/specs/rfc7230.html]

Hypertext Transfer Protocol, (HTTP/1.1): Semantics and Content RFC 7231 [1)]

Schema JSON, A Media Type for Describing JSON Documents [http://json-schema.org/draft/2019-09/json-schema-core.html]

Ain't Markup Language YAML, (YAML™) Version 1.2: 3rd Edition, Patched at 2009-10-01 [https://yaml.org/spec/1.2/spec.html]

ISO/IEC 19785-3, *Information technology — Common Biometric Exchange Formats Framework — Part 3: Patron format specifications*

RFC 3339 - *Date and Time on the Internet: Timestamps*:[2)]

3 Terms and Definitions

For the purposes of this document, the terms and definitions in ISO/IEC 30108-1 and ISO/IEC 2382-37 apply.

4 Symbols and Abbreviated Terms

For the purposes of this document, the symbols and abbreviated terms in ISO/IEC 30108-1 and the following apply.

JSON JavaScript Object Notation

REST Representational State Transfer

1) https://tools.ietf.org/html/rfc7231#section-6

2) https://www.rfc-editor.org/rfc/rfc3339

YAML Yet Another Markup Language[3]

5 Conformance

ISO/IEC 30108-1 Annex A specifies the conformance requirements for systems/components claiming conformance to this standard.

6 System Context

This clause provides an overview of Representational State Transfer (REST), in the scope of IAVS.

6.1 Data formats

The specification included in this International Standard, has been written in OpenAPI, using YAML as the specification language. OpenAPI is an extended way to specify RESTful services, allowing a variety of tools to develop client and server applications, as well as using data in any of the data formats typically used in RESTful services, e.g., JSON or XML.

A goal for this IAVS Standard is to be as open as possible. To that end, this specification is also available in an electronic OpenAPI file, available at https://standards.iso.org/iso-iec/30108/-2/ed-1 .

7 Biometric Identity Assurance Services

7.1 Introduction

This clause defines two categories of IAVS services, primitive and aggregate. Primitive services are lower-level operations that are used to request a specific capability. Aggregate services operate at a higher-level, performing a sequence of primitive operations in a single request. (An example of such a sequence would be a negative search where a 1:N identification which results in no matches being found is immediately followed by the addition of the biometric sample into that search population.) Implementers are not restricted to one or the other but may provide (and requesters use) a mixture of both primitive and aggregate types.

Aggregate services do not have to utilize primitive services.

Each service is identified by an *<interface>* tag and must include a *name* attribute. Service parameters are identified by a *<parameter>* tag and must include a *name*, *type*, and *direction* attribute. The *direction* attribute specifies whether the parameter is an input parameter (*in*), an output parameter (*out*), or an input/output parameter (*inout*). Parameters may also include a *use* attribute to indicate if the parameter is required, optional, or conditional. If the parameter is conditional, the service description must identify the conditions.

7.2 Primitive Services

7.2.1 Introduction

IAVS specifies the following set of primitive services.

7.2.2 Add Subject To Gallery

7.2.2.1 Description

The ***Add Subject To Gallery*** service shall register a subject to a given gallery or population group. As an optional parameter, the value of the claim to identity by which the subject is known to the gallery may be specified. This claim to identity shall be unique across the gallery. If no claim to identity is

specified, the subject ID (assigned with the ***Create Subject*** service) shall be used as the claim to identity. Additionally, in the encounter-centric model, the encounter ID associated with the subject's biometrics that will be added to the gallery shall be specified.

NOTE In the IAVS model, the creation and management of galleries are the responsibility of the service provider implementation. Services are not exposed to the requester for this purpose.

7.2.2.2 REST method

POST

7.2.2.3 Parameters

— *Gallery ID* – the identifier of the gallery or population group to which the subject will be added

— *Subject ID* – the identifier of the subject

— *Identity Claim (optional)* – the identifier by which the subject is known to the gallery

— *Encounter ID (conditional)* – the identifier of the encounter, required for encounter-centric models

7.2.2.4 Responses

In addition to the generic HTTP responses defined in clause 9.2, the execution of this service shall provide one of the following responses (where the meaning of each of the Error Condition Codes can be found in clause 9.3):

Table 1 — Particular responses for *Add Subject To Gallery* service

HTTP Status Code	HTTP Status Code Meaning	Error Condition Code	Description
200	OK	SUCCESS	
400	Bad Request	INVALID_IDENTITY_CLAIM	
		INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_ID	
404	Not Found	UNKNOWN_GALLERY	
		UNKNOWN_SUBJECT	
		UNKNOWN_IDENTITY_CLAIM	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	CANNOT_STORE_DATA	
		INTERNAL_DATABASE_ERROR	

7.2.2.5 YAML specification

To be placed in section `#/paths`:

```
/addSubjectToGallery:
    post:
      summary: Add Subject To Gallery
      parameters:
        - name: galleryID
          in: query
          required: true
          schema:
            type: string
        - name: subjectID
          in: query
          required: true
```

```
      schema:
        type: string
    - name: identityClaim
      in: query
      required: false
      schema:
        type: string
    - name: encounterID
      in: query
      required: false #conditional
      schema:
        type: string
  responses:
    200:
      $ref: '#/components/responses/success'
    400:
      description: Bad Request
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/responses/invalidIdentityClaim'
              - $ref: '#/components/responses/invalidSubjectId'
              - $ref: '#/components/responses/invalidEncounterId'
    404:
      description: Not Found
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/responses/unknownGallery'
              - $ref: '#/components/responses/unknownSubject'
              - $ref: '#/components/responses/unknownIdentityClaim'
              - $ref: '#/components/responses/unknownEncounter'
    500:
      description: Internal Server Error
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/responses/cannotStoreData'
              - $ref: '#/components/responses/internalDatabaseError'
              - $ref: '#/components/responses/unknownError'
    503:
      $ref: '#/components/responses/x503'
```


7.2.3 Check Quality

7.2.3.1 Description

The ***Check Quality*** service shall return a quality score for a given biometric or a specified subject. Either a biometric sample or a subject ID shall be provided. The biometric input is provided in a CBEFF basic structure or CBEFF record, which in this standard is called a CBEFF-BIR. The algorithm vendor and algorithm vendor product ID may be optionally provided in order to request a particular algorithm's use in calculating the biometric quality. If an algorithm vendor is provided, then the algorithm vendor product ID is required. If no algorithm vendor is provided, the implementing system shall provide the algorithm vendor and algorithm vendor product ID that were used to calculate the biometric quality as output parameters.

NOTE Algorithm Vendors are registered with the ISO Biometric Registration Authority and assigned unique identifiers as defined in ISO/IEC 19785-2. Algorithm Product IDs are assigned by the registered algorithm vendor.

7.2.3.2 REST method

GET

7.2.3.3 Parameters

— *BIR (conditional)* – data structure containing a single biometric sample for which a quality score is to be determined; required if no Subject ID is provided

— *Subject ID (conditional)* – the identifier of the subject; required if no BIR is provided

— *Algorithm Vendor (optional)* – the identifier of the vendor of the quality algorithm used to determine the quality

— *Algorithm Vendor Product ID (conditional)* – the vendor assigned ID for the algorithm used to determine the quality; required as input if algorithm vendor is provided

7.2.3.4 Responses

In addition to the generic HTTP responses defined in clause 9.2, the execution of this service shall provide one of the following responses (where the meaning of each of the Error Condition Codes can be found in clause 9.3:

Table 2 — Particular responses for *Check Quality* service

HTTP Status Code	HTTP Status Code Meaning	Error Condition Code	Description
200	OK	SUCCESS	Adding also — *Quality Score* — *Algorithm Version* as seen below
400	Bad Request	INVALID_BIR	
		INVALID_SUBJECT_ID	
		INVALID_INPUT	
		BIOMETRIC_TYPE_NOT_SUPPORTED	
		UNKNOWN_FORMAT	
403	Forbidden	BIR_DECRYPTION_FAILURE	
		BIR_QUALITY_ERROR	
		BIR_SIGNATURE_FAILURE	
404	Not Found	UNKNOWN_SUBJECT	
500	Internal Server Error	CANNOT_CHECK_QUALITY	

A successful execution of this service will provide:

— *Quality Score* – the quality of the biometric, as defined by the Quality type in one of the JSON-coded patron formats ISO/IEC 19785-3

— *Algorithm Version* – the version of the algorithm used to determine the quality

7.2.3.5 YAML specification

To be placed in section `#/paths`:

```
/checkQuality:
  get:
    summary: Check Quality
    parameters:
      - name: bir
        in: query
```

```
      required: false #conditional
      schema:
        $ref: '#/components/schemas/CBEFF_BIR_Type'
    - name: subjectID
      in: query
      required: false #conditional
      schema:
        type: string
    - name: algorithmVendor
      in: query #inout
      required: false
      schema:
        type: string
    - name: algorithmVendorProductID
      in: query #inout
      required: false #conditional
      schema:
        type: string
  responses:
    200:
      description: Success. Returns quality score and algorithm version
      content:
        application/json:
          schema:
            type: object
            properties:
              qualityScore:
                type: string #RAUL: CBEFF-3 JSON
                description: >
                  The quality of the biometric, as defined by the Quality
                  type in one of the JSON-coded patron formats ISO/IEC
                  19785-3
              algorithmVersion:
                type: string
                description: >
                  The version of the algorithm used to determine the
                  quality
    400:
      description: Bad Request
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/responses/invalidBir'
              - $ref: '#/components/responses/invalidSubjectId'
              - $ref: '#/components/responses/invalidInput'
              - $ref: '#/components/responses/biometricTypeNotSupported'
              - $ref: '#/components/responses/unknownFormat'
    403:
      description: Forbidden
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/responses/birDecryptionFailure'
              - $ref: '#/components/responses/birQualityError'
              - $ref: '#/components/responses/birSignatureFailure'
    404:
      $ref: '#/components/responses/unknownSubject'
    500:
      description: Internal Server Error
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/responses/cannotCheckQuality'
              - $ref: '#/components/responses/unknownError'
    503:
$ref: '#/components/responses/x503'
```

7.2.4 Classify Biometric Data

7.2.4.1 Description

The ***Classify Biometric Data*** service shall attempt to classify a biometric sample. For example, a fingerprint biometric sample may be classified as a whorl, loop, or arch (or other classification classes and sub-classes). The types of classification algorithms and classes are not specified here, rather they are left for the implementing system to define. If no classification algorithm is input, then the IAVS service provider will make the selection.

7.2.4.2 REST method

GET

7.2.4.3 Parameters

— *BIR* – data structure containing a single biometric sample for which the classification is to be determined

— *Classification Algorithm Type (optional / output)* – identifies the type of classification algorithm to be used (input) and that was used (output) to perform the classification (e.g., for fingerprints, Henry classification)

7.2.4.4 Responses

In addition to the generic HTTP responses defined in clause 9.2, the execution of this service shall provide one of the following responses (where the meaning of each of the Error Condition Codes can be found in clause 9.3:

Table 3 — Particular responses for *Classify Biometric Data* service

HTTP Status Code	HTTP Status Code Meaning	Error Condition Code	Description
200	OK	SUCCESS	Adding also *Classification*, as seen below
400	Bad Request	INVALID_BIR	
		INVALID_INPUT	
		BIOMETRIC_TYPE_NOT_SUPPORTED	
		UNKNOWN_FORMAT	
403	Forbidden	BIR_DECRYPTION_FAILURE	
		BIR_QUALITY_ERROR	
		BIR_SIGNATURE_FAILURE	
500	Internal Server Error	CANNOT_PROCESS_DATA	

A successful execution of this service will provide:

— *Classification* – the result of the classification

7.2.4.5 YAML specification

To be placed in section `#/paths`:

```
/classifyBiometricData:
  get:
    summary: Classify Biometric Data
```

```
    parameters:
      - name: bir
        in: query
        required: true
        schema:
          $ref: '#/components/schemas/CBEFF_BIR_Type'
      - name: classificationAlgorithmType
        in: query
        required: false
        schema:
          type: string
    responses:
      200:
        description: Success. Returns classification
        content:
          application/json:
            schema:
              type: object
              properties:
                classification:
                  type: string
                  description: The result of the classification
      400:
        description: Bad Request
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/responses/invalidBir'
                - $ref: '#/components/responses/invalidInput'
                - $ref: '#/components/responses/biometricTypeNotSupported'
                - $ref: '#/components/responses/unknownFormat'
      403:
        description: Forbidden
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/responses/birDecryptionFailure'
                - $ref: '#/components/responses/birQualityError'
                - $ref: '#/components/responses/birSignatureFailure'
      500:
        description: Internal Server Error
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/responses/cannotProcessData'
                - $ref: '#/components/responses/unknownError'
      503:
    $ref: '#/components/responses/x503'
```


7.2.5 Create Encounter

7.2.5.1 Description

The ***Create Encounter*** service shall, for the specified subject, create a new encounter record and associate an encounter ID to that record. If not provided by the requester, the ***Create Encounter*** service shall generate an encounter ID that uniquely identifies the encounter within the subject record in the system.

NOTE Typically, the IAVS service provider will assign the encounter ID. In the event that the requester assigns the encounter ID, it shall be used unless it duplicates an existing encounter ID in which case an error shall be returned.

The ***Create Encounter*** service is performed prior to a ***Set Biographic Data***, ***Set Biometric Data***, or ***Set Document Data*** operation.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The IAVS service provider will create the encounter and will set the encounter type to "recognition".

7.2.5.2 REST method

POST

7.2.5.3 Parameters

— *Subject ID* – the identifier of the subject

— *Encounter Type* – the category of encounter

— *Encounter ID (optional)* – the identifier of the encounter

7.2.5.4 Responses

In addition to the generic HTTP responses defined in clause 9.2, the execution of this service shall provide one of the following responses (where the meaning of each of the Error Condition Codes can be found in clause 9.3:

Table 4 — Particular responses for *Create Encounter* service

HTTP Status Code	HTTP Status Code Meaning	Error Condition Code	Description
200	OK	SUCCESS	Adding also *Encounter ID*, as seen below
400	Bad Request	INVALID_SUBJECT_ID	
		INVALID_ENCOUNTER_TYPE	
		INVALID_ENCOUNTER_ID	
404	Not Found	UNKNOWN_SUBJECT	
		UNKNOWN_ENCOUNTER	
500	Internal Server Error	DUPLICATE_ENCOUNTER_ID	
		INTERNAL_DATABASE_ERROR	

A successful execution of this service will provide:

— *Encounter ID* – the identifier of the encounter

7.2.5.5 YAML specification

To be placed in section `#/paths`:

```
/createEncounter:
  post:
    summary: Create Encounter
    parameters:
      - name: subjectID
        in: query
        required: true
        schema:
          type: string
      - name: encounterType
        in: query
        required: true
        schema:
```