INTERNATIONAL STANDARD

ISO/IEC 29141

First edition
2009-12-01

Information technology — Biometrics — Tenprint capture using biometric application programming interface (BioAPI)

Technologies de l'information — Biométrie — Saisie de «Tenprint» à l'aide de l'interface de programmation d'application biométrique (BioAPI)

Reference number
ISO/IEC 29141:2009(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web 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. 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 29141 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

Introduction

This International Standard specifies how a BioAPI application can interact with a BioAPI framework to support a tenprint capture (capture of the fingerprints of all ten fingers). It specifies and supports the deployment of large-scale identity management and credentialing systems (which often require a tenprint capture as part of the identity vetting and background-checking process).

This International Standard provides additional standardization of fields (that are left undefined in BioAPI), in order to standardize the support for a tenprint capture, but is in all other respects a profiling of BioAPI to support the tenprint capture.

This International Standard defines a biometric data block format for carrying the data obtained from a tenprint capture. It specifies all the parameters, function calls, and other information needed by an application for the use of BioAPI in support of tenprint capture.

It also specifies the conformance requirements on a biometric service provider that supports tenprint capture using this International Standard.

Information technology — Biometrics — Tenprint capture using biometric application programming interface (BioAPI)

1 Scope

This International Standard specifies requirements for the use of ISO/IEC 19784-1, as amended by ISO/IEC 19784-1/Amd. 1 (BioAPI) for the purpose of performing a tenprint capture operation.

It specifies a BDB format that is used to interact with a BioAPI framework (and hence with BSPs) to support an application wishing to perform a tenprint capture.

It specifies a capture control block and a capture output block that conforming BSPs are required to support if they conform to this International Standard.

2 Conformance

Biometric Service Providers (BSPs) that claim conformance to this International Standard shall satisfy requirements specified in Clauses 6 through 10.

3 Normative references

The following referenced documents are indispensable for the application 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.

ANSI/NIST-ITL 1-2007, *Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information*

ISO/IEC 19784-1, *Information technology — Biometric application programming interface — Part 1: BioAPI specification*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19784-1, ANSI/NIST-ITL 1-2007, and the following apply.

4.1
capture control block
CCB
parameter passed to a BSP that determines what it should return

4.2
capture output block
COB
formatted data structure containing data objects returned by the BSP and also empty fields for those data objects that could not be returned by the BSP

4.3
livescan device
hardware scanner specifically designed and used to capture digital fingerprint images

NOTE The term “livescan” is used to differentiate from “card scan”.

4.4
plain impression
fingerprint image captured by pressing a single finger or multiple fingers on the sensor without rolling

4.5
rolled prints
image of a single finger, or set of single fingers, obtained by rolling each finger horizontally from the extreme sides of the finger

NOTE The purpose of this operation is to obtain as much information from the fingertip as possible (i.e. “nail-to-nail”).

4.6
slap prints
image of multiple fingerprints from multiple fingers captured simultaneously by placing all four left fingers or all four right fingers on the capture surface (e.g. platen of a fingerprint scanner) and used to derive single-finger plain impressions

4.7
segment position calculation
identification of the locations of each individual fingerprint image in a multi-finger slap print image

4.8
tenprint capture
electronic capture of ten human fingerprints

4.9
Type-2 record
biometric data record included in an ANSI/NIST-ITL 1-2007 file that contains metadata

4.10
Type-4 record
biometric data record included in an ANSI/NIST-ITL 1-2007 file that contains 500 ppi fingerprint images and is typically used for criminal submissions

4.11
Type-10 record
biometric data record included in an ANSI/NIST-ITL 1-2007 file that contains one or more mugshots

4.12
Type-14 record
biometric data record included in an ANSI/NIST-ITL 1-2007 file that contains fingerprint images and enhanced metadata, which enables transport of newer image formats such as 1000 ppi images and slap prints, and is often used for civil submissions

4.13
vertical rolls
fingerprint images that have been captured by rolling a finger vertically from the slap position over the finger tip to the nail

NOTE This is in contrast to horizontal rolls, which are captured by horizontal rolling from the nail over the slap position to the other side of the nail as described in the definition for **rolled prints** (4.5).

5 Abbreviated terms

ANSI – American National Standards Institute

BDB – biometric data block

BIR – biometric information record

BSP – biometric service provider

CBEFF – common biometric exchange file format

IDC – image designation character

NFIQ – NIST fingerprint image quality

NIST – National Institute of Standards and Technology

TOT – type of transaction

UUID – universally unique identifier

WSQ – Wavelet Scalar Quantization

6 Requirements

This International Standard specifies the requirements for a BioAPI-compliant BSP to support tenprint capture using livescan devices.

Selection of the finger images captured and the processing done on the images shall be done through use of a capture control block (CCB) based upon the ANSI/NIST ITL 1-2007 standard. This CCB contains incomplete data fields for each desired image. Each CCB shall have empty fields for information that the application requires the BSP to generate and output as a capture output block (COB). This COB shall be based upon the ANSI/NIST ITL 1-2007 standard and may contain multiple Type-4 or Type-14 fingerprint image records.

Functionality associated with amputated or bandaged fingers is supported by the ANSI/NIST ITL 1-2007 standard for Type 14 records. Where Type 14 records are used, information about bandaged or amputated fingers shall be included in the Type 14 record. Where Type 4 records are used, this information shall be conveyed in the Type-2 record. See Clause 9.4, Amputated or bandaged fingers, which describes how this International Standard addresses the need for this functionality.

Two BioAPI interface functions shall be used by an application to perform the tenprint data acquisition:

1. BioAPI_ControlUnit()

2. BioAPI_Capture()

Optionally, a third function, BioAPI_QueryUnits() may be used as part of the discovery process.

See Clause 7: BioAPI function calls.

When performing a tenprint capture, the application shall first send the CCB to the BSP using the InputData parameter of the BioAPI_ControlUnit() function.

The application shall then call the BioAPI_Capture() function to trigger the capture and processing of the requested images. The BioAPI_Capture() function shall return a COB containing the processed image data.

Note that Type 14 records are recommended because use of Type 4 records is being deprecated by some portion of the user community.

Note that this specification is not intended for use to perform segmentation, and such use can yield unanticipated results.

7 BioAPI function calls

This clause specifies how certain BioAPI function calls are to be used when performing tenprint capture operations.

7.1 BioAPI_ControlUnit()

7.1.1 Usage

This function is used to pass in a CCB, which shall contain incomplete data fields. The incomplete fields shall be used by a BSP to control how BioAPI_Capture() function calls are processed. The CCB shall continue to be used for any subsequent BioAPI_Capture() function calls until a new CCB is set for the BSP using another BioAPI_ControlUnit() call.

7.1.2 Function definition

```
BioAPI_RETURN BioAPI BioAPI_ControlUnit

    (BioAPI_HANDLE   BSPHandle,

    BioAPI_UNIT_ID   UnitID,

    uint32_t   ControlCode,

    const BioAPI_DATA   *InputData,

    BioAPI_Data   *OutputData);
```

7.1.3 Parameters

7.1.3.1 BSPHandle (input)

The handle of the attached BSP.

7.1.3.2 UnitID (input)

ID of the BioAPI Unit.

7.1.3.3 ControlCode (input)

Indicates the purpose of this function to differentiate it from other BioAPI_ControlUnit() calls. The value of “1” shall be used to indicate that a CCB is being set.

7.1.3.4 InputData (input)

Pointer to a BioAPI_Data structure. This structure shall contain the address and length of the CCB.

7.1.3.5 OutputData (output)

No output is needed. The buffer address will be NULL and the length set to “0” within the BioAPI_DATA structure. Usage of the OutputData parameter to return diagnostic data about the CCB may be desirable.

7.1.3.6 Additional error return codes

Error codes should be used to indicate errors detected in the CCB passed in the InputData parameter. See Clause 11.

7.2 BioAPI_Capture()

7.2.1 Usage

This function shall trigger a BSP to perform fingerprint image captures and to create a new CCB. The fingers captured and the processing done on the fingers shall be controlled by inclusion of empty fields in the current CCB, which was specified in the BioAPI_ControlUnit() call.

7.2.2 Function definition

```
BioAPI_RETURN BioAPI BioAPI_Capture

    (BioAPI_HANDLE   BSPHandle,

    BioAPI_BIR_PURPOSE   Purpose,

    BioAPI_BIR_SUBTYPE   Subtype,

    BioAPI_BIR_BIOMETRIC_DATA_FORMAT   *OutputFormat,

    BioAPI_BIR_HANDLE   *CapturedBIR,

    Int32_t   Timeout,

    BioAPI_BIR_HANDLE   *AuditData);
```

7.2.3 Parameters

7.2.3.1 BSPHandle (input)

The handle of the attached BSP.

7.2.3.2 Purpose (input)

A value indicating the purpose of the biometric data capture. For the purposes of tenprint capture, any purpose value may be used.

7.2.3.3 Subtype (input/optional)

Specifies which subtype to capture. This parameter does not have the flexibility to describe the range of possible finger image sets that might be captured. A value of BioAPI_NO_SUBTYPE_AVAILABLE (0x00) shall be used to indicate that the BSP is to select the subtypes using the current CCB. No other value may be used.

7.2.3.4 OutputFormat (input/optional)

Specifies which BDB (biometric data block) format to use for the returned CapturedBIR. The CBEFF Format Owner and CBEFF Format Type shall be set to 0x001B and 0x8013 respectively to designate that the BDB in the CBEFF BIR is a COB.

7.2.3.5 CapturedBIR (output)

A handle to a BIR whose BDB is a COB containing the captured data. See Clause 10 for a specification of this BIR.

7.2.3.6 Timeout (input)

Specifies the timeout value (in milliseconds) for the entire capture operation requested by the application. If this timeout is reached, the function returns an error, and no results. This value can be any positive number. Avalue of -1 means the BSP's default timeout value will be used.

7.2.3.7 AuditData (output/optional)

A handle value of BioAPI_UNSUPPORTED_BIR_HANDLE shall be used to indicate that AuditData is not available.

7.2.4 Additional error return codes

BioAPI includes codes such as BioAPIERR_UNABLE_TO_CAPTURE to indicate that the BSP could not collect the data that it is meant to. Additional error codes specific to tenprint capture are specified in Clause 11.

7.3 BioAPI_QueryUnits()

7.3.1 Usage

Applications will use this function to determine if the device attached to the BSP conforms to this International Standard and if so, to discover what specific capabilities are supported by the attached tenprint capture device (unit).

7.3.2 Function definition

```
BioAPI_RETURN BioAPI BioAPI_QueryUnits

    (const BioAPI_UUID   *BSPUuid,

    BioAPI_UNIT_SCHEMA   **UnitSchemaArray,

    uint32_t   *NumberOfElements);
```

7.3.3 Parameters

7.3.3.1 BSPUuid (input)

The unique identifier for the (tenprint) BSP to which the unit (tenprint scanning device) is attached and for which the unit information is to be returned.

7.3.3.2 UnitSchemaArray (output)

A pointer to the address of the array of elements of type BioAPI_UNIT_SCHEMA containing the unit schema information. Specific elements of interest are defined in 6.3.4, below.

7.3.3.3 NumberOfElements (output)

A pointer to the number of elements in the array.

7.3.4 BioAPI_UNIT_SCHEMA

This schema is defined in Clause 7.55 of the base standard (BioAPI specification). There are three elements of this structure that are of particular interest for the purpose of tenprint capture (the UnitProperties, UnitPropertyID, and UnitProperty elements), though other elements of the schema may also be of general use (e.g., vendor information, firmware version, etc.).

If more than one type of BioAPI Unit is attached to the BSP, more than one Unit schema may be returned. Tenprint scanner devices shall set UnitCategory to BioAPI_CATEGORY_SENSOR (0x00000008).

7.3.4.1 UnitProperties

UUID indicating a set of properties of the BioAPI Unit. The indicated set can either be specified by each vendor or follow a related standard. The UUID “ba38567f-66be-468c-9ada-0b1eeb0cc7fb” shall indicate that the unit supports the profile/standard for “Tenprint Capture Using BioAPI”.

7.3.4.2 UnitPropertyID

UUID of the format of the following Unit property structure. The UUID “10568e83-c7f8-418d-9913-4e80f01aecfb” shall be used for the format ID for Tenprint Scanner Properties.

7.3.4.3 UnitProperty

UnitProperty is of type BioAPI_DATA, which is defined in 7.19 of the BioAPI specification (ISO/IEC 19784-1). It includes the address and length of a memory buffer containing the Unit. The format and content of the Unit property can either be specified by the vendor or be specified in a related standard. For tenprint scanners conforming to this International Standard, the Unit property structure shall be defined as shown in Table 1. Use sequential, bit-oriented fields. Values for all fields shall be present. Big-endian (network) byte ordering shall be used.

Table 1 - UnitProperty structure

Field Name	Type	Size	Values	Notes
Rolls	Boolean	1B	0x00, 0x01	‘01’ indicates device can capture rolled fingerprints
Slaps	Boolean	1B	0x00, 0x01	‘01’ indicates device can capture slap prints
Segment position calculation	Boolean	1B	0x00, 0x01	‘01’ indicates device can calculate bounding boxes
Number of quality algorithms supported	Integer	1B	0-255	Number of quality algorithm identifiers contained in the next field
Quality algorithms supported	Integer (array)	Var.	Each algorithm is described by 4-bytes (see Note 1)	Array of unique identifiers for each quality algorithm supported by the device
Number of compression algorithms supported	Integer	1B	0-255	Number of compression algorithm identifiers contained in the next field
Compression algorithms supported	Integer (array)	Var.	Each algorithm is described by a single byte derived from binary values found in ANIS/NIST ITL-1 2007 Table 1	Array of unique identifiers for each compression algorithm supported by the device
Number of segment position calculation algorithms supported	Integer	1B	0-255	Number of segment position calculation algorithm identifiers contained in the next field
Segment position calculation algorithms supported	Integer (array)	Var.	Each algorithm is described by 4-bytes (see Note 1)	Array of unique identifiers for each segment position calculation algorithm supported by the device
Number of image resolutions supported	Integer	1B	1-255	Number of resolution values contained in the next field
Image resolutions supported	Integer (array)	Var.	Each resolution is described by 2-bytes	Array of values of each resolution (in ppi) supported by the device (e.g. 500)
Number of sequencing algorithms supported	Integer	1B	1-255	Number of resolution values contained in the next field
Sequencing algorithms supported	Integer (array)	Var.	Each algorithm is described by 4-bytes (see Note 1)	Array of unique identifiers for each sequencing algorithm supported by the device
Missing finger detection	Boolean	1B	0x00, 0x01	‘01’ indicates device can automatically identify missing fingers
Type-4 records	Boolean	1B	0x00, 0x01	‘01’ indicates device can return Type-4 records
Type-14 records	Boolean	1B	0x00, 0x01	‘01’ indicates device can return Type-14 records
Vertical rolls	Boolean	1B	0x00, 0x01	"01" indicates device can capture vertical enrolled prints

Note 2 Algorithm identifiers are structured as follows (analogous to the representation of data format identifiers):

> *<algorithm owner> - 2-byte integer (hex) value representing the vendor/developer of the algorithm (e.g., NIST, Vendor A) as registered with the IBIA as a product owner.*
>
> *<algorithm type> - 2-byte integer (hex) value representing the specific algorithm (i.e., product code) as assigned by the <algorithm owner>.*

8 Capture control blocks (CCB)

A BSP shall be able to parse the CCB. It shall identify data items that control the capture process and missing items that the capture process will be filling in during the subsequent capture operation. Documentation associated with a tenprint capture BSP should include indication description of the specification used for interpretation of the CCB.

A BSP shall never remove data contained in a CCB, but rather only correct missing or non-compliant data. This allows CCBs to contain some data that is not related to the capture process. For example the CCB could include a Type-2 record that already had all the demographic data filled in. The BSP shall then include this filled-in Type-2 record in its output BDB.

The BSP shall pass through non-compliant records whose correction it does not support. This will allow use of multiple BSP units each capable of filling in different types of CCBs. For example an empty Type-10 record indicating a mugshot is required could be included in the CCB. A tenprint capture BSP might not be capable of acquiring the mugshot. It should still pass back the empty Type-10 record when it generates its COB. This way a mugshot BSP that is part of the system could be passed the COB containing fingerprints and be able to generate the final, complete COB.

Documentation for a BSP should identify the CCBs and fields within CCBs that it is capable of making use of during a capture event.

Note that this approach places within the domain of the application the identification of discrepancies between what is requested and what is received from the BSP.

9 Record types and fields

This clause indicates which record types may be present in the input CCB, which fields may be allowed for each record type, and if the field should contain information or is to be filled in during the capture process.

9.1 Control and required fields

The following table lists the fields in a CCB that will control the behavior of the BSP. It also includes fields that must be present in order to have a compliant CCB. The condition code for individual fields indicates if the field must be present if the record is present in the CCB. An “M” indicates a mandatory field, an “O” indicates an optional field.

Table 2 - Control and required fields

Record/Field	Ident	Cond Code	Purpose
Type-1 Record		M	Type-1 record always required to support NIST syntax
1.001	LEN	M	Required to support NIST syntax (record length)
1.002	VER	M	Required to support NIST syntax (NIST version). This value shall be 400.
1.003	CNT	M	Required to support NIST syntax (record types present)
1.004	TOT	O	Use to indicate the output is for a specific ANSI/NIST ITL 1-2007 TOT
Type-2 Record		O	Type-2 record is not required
2.001	LEN	M	Required to support NIST syntax (record length)
2.002	IDC	M	Required to support NIST syntax (record IDC)
2.038	DPR	O	Indicates date of fingerprinting
2.067	IMA	O	Indicates image capture equipment make, model, serial number
2.084	AMP	O	Indicates fingers which cannot be printed, see Clause 9.4
Type-4 Record		O	Used for capture of 500 ppi fingerprints
4.001	LEN	M	Required to support NIST syntax (record length)
4.002	IDC	M	Required to support NIST syntax (record IDC)
4.003	IMP	M	Type of impression, see Clause 9.7
4.004	FGP	M	Finger/image to capture, see Clause 9.7
4.005	ISR	M	Image scanning resolution, see Clause 9.5
4.006	HLL	M	Image width, see Clause 9.2
4.007	VLL	M	Image height, see Clause 9.2
4.008	GCA	M	Compression type (eg. none, WSQ), see Clause 9.6
4.009	DAT	M	Image data, see Clause 9.3
Type-14 Record		O	Used for capture of 500 or 1000 ppi fingerprints
14.001	LEN	M	Required to support NIST syntax (record length)
14.002	IDC	M	Required to support NIST syntax (record IDC)
14.003	IMP	M	Type of impression, see Clause 9.7
14.006	HLL	M	Image width, see Clause 9.2
14.007	VLL	M	Image height, see Clause 9.2
14.008	SLC	M	Sampling units for capture resolution, see Clause 9.5
14.009	HPS	M	Horizontal capture resolution, see Clause 9.5
14.010	VPS	M	Vertical capture resolution, see Clause 9.5
14.011	CGA	M	Compression type, see Clause 9.6
14.012	BPX	M	Bits per pixel for capture, see Clause 9.11
14.013	FGP	M	Finger/image to capture, see Clause 9.7
14.018	AMP	O	Amputated or bandaged fingers. See Clause 8.4.2
14.021	SEG	O	Finger segment position calculation positions, see Clause 9.8
14.022	NQM	O	NFIQ finger image quality, see Clause 9.10
14.023	SQM	O	Finger segment position calculation quality (algorithm indicated), see Clause 9.9
14.024	FQM	O	Finger image quality (algorithm indicated), see Clause 9.10
14.999	DAT	M	Image data, see Clause 9.3

9.2 Image height and width

For the Type-4 and the Type-14 records, the HLL and VLL fields indicate the dimensions of the image. For Type-4 or Type-14 records occurring in the CCB these fields are mandatory. Values of “0” in these fields shall indicate that the BSP should select the size of the captured image in accordance with ANSI/NIST ITL-1 2007. Non-zero values may be used by BSPs that are capable of sizing images, to specify the desired size of the captured image. When non-zero values are used in the CCB, BSP’s shall return an error if they are unable to satisfy the specified image size.

A COB from a BSP shall have the HLL and VLL fields in Type-4 and Type-14 records filled in with the sizes of the captured images in the records.

9.3 Image data

The Type-4 and Type-14 DAT field shall be used for holding image data. No image data should be placed in these fields for CCBs. For CCBs using the binary Type-4 record, this field shall therefore have length “0”. For Type-14 records the DAT field shall be present but should be empty.

A COB from a BSP shall have compliant image data in the DAT fields of any Type-4 or Type-14 records.

9.4 Amputated or bandaged fingers

9.4.1 Type-4 records

The ANSI/NIST ITL 1-2007 standard allows user defined fields in the Type-2 record. For the case when Type 4 records are exchanged, the Type-2 record field 2.084 shall be used to indicate fingers not available for image capture, e.g. due to amputation or bandaging.

When finger availability information exists prior to the capture, this shall be indicated in the CCB through use of field 2.084. A subfield shall be added for each finger that cannot be imaged. Each subfield consists of a two-character finger position code item and an amputated or bandaged code item. For this second item, the code “XX” indicates amputated, “UP” indicates unable to print. If all fingers are known to be available, this shall be indicated through use of an empty 2.084 field. Having this information available prior to capture may be helpful for managing workflow within the BSP and in performing segment position calculation within the BSP.

Another allowed option is to have the BSP make the determination of which fingers are present during the capture process. In this case, the Type-2.084 field shall not be present at all. All desired Type 4 and Type 14 fingerprint images shall be requested, but when fingers are not present, Type 4 and Type 14 images for them will not be returned.

For both of the above cases, if a Type-2 record is requested by the CCB to be provided in the COB, then field 2.084 in the COB shall exist when fingers are not available. The convention followed shall be similar to that followed for the first case for the CCB. A subfield shall be added for each finger that cannot be imaged. Each subfield shall consist of a two-character finger position code item and an amputated or bandaged code item. For this second item, the code “XX” shall indicate “amputated”, and “UP” shall indicate “unable to print”.

9.4.2 Type-14 records

The ANSI/NIST-ITL 1-2007 standard accommodates data about missing or bandaged fingers within the Type 14 record. When a Type 14 record is exchanged, Field 14.018 shall be used to indicate an amputated or bandaged finger.

9.5 Capture resolution

An application may use the Type-14 SLC, HPS, and VPS fields to indicate the desired capture resolution. It shall be defined as either 500 or 1000 ppi. If the BSP is not capable of capturing at the requested resolution, it shall return an error. See Clause 11.

All requested Type-4 records shall be captured at 500 ppi. For CCBs, the value of the ISR field in Type-4 records shall be ignored. COBs containing Type-4 records shall have this field set to an appropriate value.

Note: ANSI/NIST ITL-1 2007 specifies resolution tolerance ranges.

9.6 Image compression

The Type-14 CGA or the Type-4 GCA field shall be used to select the type of compression to perform on the images. Values for compression method shall be taken from Table 1 of the ANSI/NIST-ITL 1-2007 standard. For example, using a value of “NONE” in the Type-14 CGA field indicates that the image data should be returned in raw format.

9.7 Selecting images to be captured

The Type-14 IMP and the FGP fields may be used together to select the type of impression (rolled or plain) and the finger(s) to be captured. The captured image shall be stored in the Type-14 DAT field. Similarly the Type-4 IMP and FGP fields shall be used for selection for Type-4 records.

The Order of the IDC codes of the input CCB shall be the order of the task in the workflow.

Table 3 – Supported IMP/FGP combinations

FGP \ IMP	0x00 (plain)	0x01 (rolled)	(vertical rolled)*
1 (right thumb)	supported	Supported	supported
2 (right index)	supported	Supported	supported
3 (right middle)	supported	Supported	supported
4 (right ring)	supported	Supported	supported
5 (right little)	supported	Supported	supported
6 (left thumb)	supported	Supported	supported
7 (left index)	supported	Supported	supported
8 (left middle)	supported	Supported	supported
9 (left ring)	supported	Supported	supported
10 (left little)	supported	Supported	supported

11 (plain right thumb)	supported	Not supported	supported
12 (plain left thumb)	supported	Not supported	supported
13 (right 4 fingers)	supported	Not supported	Not supported
14 (left 4 fingers)	supported	Not supported	Not supported
15 (2 thumbs)	supported	Not supported	Not supported

9.8 Calculation of finger segment position

Fingerprint segment position calculation is possible through use of Type-14 records for fingerprint images. The Type-14 SEG field may be used to select whether finger segment position calculation should be done on the image. When this field is sent empty, the segment position calculation shall be performed and the results stored in this field.

As currently designed, this API does not enable selection of the segmentation algorithm by the application. It does, however, enable selection of the segmentation quality algorithm (generating the "estimated correctness score" as described in ANSI/NIST ITL-1 2007) as described in the following section.

9.9 Segmentation quality

Segmentation quality calculation is possible through use of Type-14 records for fingerprint images. When the Type-14 SQM field is passed in empty, the segmentation quality calculation should be done for the fingers in the image. The Type-14 SQM field may be used to select which segmentation quality algorithm is to be used. When used for this purpose, the first information item for each finger shall be filled in with the finger position. The second information item, which is the "estimated correctness score", shall be empty. The third and fourth items identifying the segmentation quality algorithm shall be present.

9.10 Image quality

9.10.1 General

Fingerprint quality calculation is possible through use of Type-14 records for fingerprint images. The Type-14 NQM or FQM field may be used to select which quality algorithm is to be used. If the NQM field is used then an NFIQ quality score shall be set. If the FQM field is used in the CCB, the second item, which is the "quality score", shall be empty. The third and fourth items identifying the algorithm shall be present. It is put upon the application to determine how best to interpret and act upon the quality score information provided by the BSP.

9.10.2 Setting quality thresholds

Individual finger capture quality thresholds are set as follows:

NFIQ: In the NQM field (14.022), the finger number(s) and corresponding minimum quality threshold value(s) are set. (For example, if a value of NFIQ 3 or better/lower is required, the value for that finger would be set to 3 meaning <=3.) Note that NFIQ uses a scale where "1" represents the best quality and "5" represents the worst quality.

Other: If an alternate quality metric has been selected by the application, then the FQM (14.024) field is used and a threshold value between 0-100 set. (For example, a quality setting of 75 would mean values >=75 are acceptable.)

9.11 Bits per pixel

For Type-14 records the BPX field is mandatory for CCBs. It shall always contain the value “8”. COBs shall always have this field set to “8” in Type-14 records.

9.12 Date printed

The ANSI/NIST ITL 1-2007 standard allows user defined fields in the Type-2 record. The Type-2 record field 2.038 shall be used to indicate the date that the subject was fingerprinted. The date shall appear as 8-digits in the format CCYYMMDD. The CCYY characters shall represent the year of the transaction; the MM characters shall be the tens and units values of the month and the DD characters shall be the day of the month. For example "20000103" represents January 3, 2000.

9.13 Image capture equipment

The ANSI/NIST ITL 1-2007 standard allows user defined fields in the Type-2 record. The Type-2 record field 2.067 shall be used to log the make, model, and serial number of the equipment used to acquire the images. The three items are separated by the separator character described in the ANSI/NIST ITL-1 2007 standard.

10 Returned BIR

The captured data shall be returned as *CapturedBIR as a result of the ***BioAPI_Capture*** call. The data shall be formatted as a CBEFF-compliant BioAPI BIR structure, consisting of a BIR header, a BDB, and optionally, a security block.

10.1 BIR header

The BIR header shall consist of a set of metadata describing the content of the “opaque” data block or BDB. Values to be used for mandatory elements are as specified below:

Table 4 - BIR header elements with fixed values

BIR Header Element	Value
BioAPI_BIR_DATA_TYPE	0x01 (Raw)
BioAPI_BIR_BIOMETRIC_DATA_FORMAT	Format Owner: 0x001B (INCITS Technical Committee M1) Format Type: 0x8013 (ANSI/NIST ITL-1 2007 record collection for Tenprint Capture)
BioAPI_BIR_BIOMETRIC_TYPE	0x00000008 (Fingerprint)
BioAPI_BIR_SUBTYPE	0x00 (No subtype available)

All other header values are set at the discretion of the BSP in accordance with the base standard.

10.2 BDB format

The Biometric Data Block (body of the BIR) shall contain the COB as defined in clause 8. The BDB shall therefore consist of a sequence of logical record types whose syntax is defined by ANSI/NIST ITL-1 2007. Additionally, the following rules shall apply:

- A single Type-1 record is mandatory, shall be the first record in the sequence, and shall indicate what records follow.
- A single Type-2 record is optional. In the case of use of the Type-4 record, the Type-2 record shall be used to indicate missing fingers.
- One or more Type-4 or Type-14 records shall be present, in the quantity and order as specified in the Type-1 record.

See Annex A for example record sets (BDB content) for the return of three slap images and for ten individual plain impressions.

11 Error codes

The following subclauses define additional error codes.

11.1 Unable to parse

BioAPI_TPERR_UNABLE_TO_PARSE (0x020000) – The reference template could not be parsed in a ControlUnit() call. This would be due to a syntax error in the reference template. It can also be used when a more specific error is not available.

11.2 Roll capture not supported

BioAPI_TPERR_ROLLS_UNSUPPORTED (0x020001) – The BSP does not support roll capture and a roll capture request was made in a ControlUnit() call.

11.3 Slap capture not supported

BioAPI_TPERR_SLAPS_UNSUPPORTED (0x020002) – The BSP does not support slap capture and a slap capture request was made in a ControlUnit() call.

11.4 Segment position calculation not supported

BioAPI_TPERR_SEGMENT_UNSUPPORTED (0x020003) – The BSP does not support the calculation of finger segment positions and a request was made in a ControlUnit() call.

11.5 Quality algorithm not supported

BioAPI_TPERR_QUALITY_ALG_UNSUPPORTED (0x020005) – The BSP does not support the quality algorithm specified and a request was made in a ControlUnit() call.

11.6 Compression algorithm not supported

BioAPI_TPERR_COMPRESS_ALG_UNSUPPORTED (0x020006) – The BSP does not support the compression algorithm specified and a request was made in a ControlUnit() call.

11.7 Segment position calculation algorithm not supported

BioAPI_TPERR_SEG_ALG_UNSUPPORTED (0x020007) – The BSP does not support segment position calculation algorithm requested and a request was made in a ControlUnit() call.

11.8 Image resolution not supported

BioAPI_TPERR_RESOLUTION_UNSUPPORTED (0x020008) – The BSP does not support the image resolution requested and a request was made in a ControlUnit() call.

11.9 Type-4 records not supported

BioAPI_TPERR_TYPE4_UNSUPPORTED (0x020009) – The BSP does not support Type 4 records and a request was made in a ControlUnit() call.

11.10 Type-14 records not supported

BioAPI_TPERR_TYPE14_UNSUPPORTED (0x020008) – The BSP does not support Type 14 records and a request was made in a ControlUnit() call.

11.11 Finger position not supported

BioAPI_TPERR_FINGER_POS_UNSUPPORTED (0x020009) – The BSP does not support the finger position and a request was made in a ControlUnit() call.

12 Example capture orders

The following steps describe an example of capture order.

- Right slaps
- Left slaps
- Right flat thumb
- Left flat thumb
- Right rolled thumb
- Right rolled index finger
- Right rolled middle finger
- Right rolled ring finger
- Right rolled little finger
- Left rolled thumb
- Left rolled index finger
- Left rolled middle finger
- Left rolled ring finger
- Left rolled little finger

The following processing steps describe an example for the capture processing of each image.

- Live image view
- Auto capture and auto contrast assessment
- Image capture
- Image segmentation
- Image quality check(s)
- Image sequence check(s)
- Operator review

Annex A
(informative)

Example capture control blocks (CCB) for acquisition of images

A.1 Three slaps (FANC TOT)

A.1.1 Type-1 record

Field	Contents	Comments
LEN	1.001:60GS	Mandatory length field
VER	1.002:0400GS	Indicates using 2007 version of standard
CNT	1.003:1US4RS2US0RS 14US1RS14US2RS14US3GS	File content indicating one Type-2 Record and three Type-14 records
TOT	1.004:FANCFS	Indicates type of transaction

A.1.2 Type-2 record (IDC 0)

Field	Contents	Comments
LEN	2.001:18GS	Mandatory length field
IDC	2.002:00FS	Mandatory IDC field

A.1.3 Type-14 record (IDC 1)

Field	Contents	Comments
LEN	14.001:125GS	Mandatory length field
IDC	14.002:01GS	Mandatory IDC field
IMP	14.003:0GS	Request for livescan plain impression type
HLL	14.006:0GS	Zero indicating image width should be filled in
VLL	14.007:0GS	Zero indicating image height should be filled in
SLC	14:008:1GS	Indicates sampling units are pixels per inch
HPS	14:009:500GS	Indicates capture at 500ppi horizontally
VPS	14:010:500GS	Indicates capture at 500ppi vertically
CGA	14:011:WSQGS	Indicates store using WSQ compression
BPX	14:012:8GS	Indicates 8-bits per pixel for capture
FGP	14:013:13GS	Indicates capture plain right four fingers
SEG	14:021:GS	Empty indicating finger segment positions should be calculated
DAT	14.999:FS	Empty indicating image capture required

A.1.4 Type-14 record (IDC 2)

Field	Contents	Comments
LEN	14.001:125GS	Mandatory length field
IDC	14.002:02GS	Mandatory IDC field
IMP	14.003:0GS	Request for livescan plain impression type
HLL	14.006:0GS	Zero indicating image width should be filled in
VLL	14.007:0GS	Zero indicating image height should be filled in
SLC	14:008:1GS	Indicates sampling units are pixels per inch
HPS	14:009:500GS	Indicates capture at 500ppi horizontally
VPS	14:010:500GS	Indicates capture at 500ppi vertically
CGA	14:011:WSQGS	Indicates store using WSQ compression
BPX	14:012:8GS	Indicates 8-bits per pixel for capture
FGP	14:013:14GS	Indicates capture plain left four fingers
SEG	14:021:GS	Empty indicating finger segment positions should be calculated
DAT	14.999:FS	Empty indicating image capture required

A.1.5 Type-14 record (IDC 3)

Field	Contents	Comments
LEN	14.001:138GS	Mandatory length field
IDC	14.002:03GS	Mandatory IDC field
IMP	14.003:0GS	Request for livescan plain impression type
HLL	14.006:0GS	Empty indicating image width should be filled in
VLL	14.007:0GS	Empty indicating image height should be filled in
SLC	14:008:1GS	Indicates sampling units are pixels per inch
HPS	14:009:500GS	Indicates capture at 500ppi horizontally
VPS	14:010:500GS	Indicates capture at 500ppi vertically
CGA	14:011:WSQGS	Indicates store using WSQ compression
BPX	14:012:8GS	Indicates 8-bits per pixel for capture
FGP	14:013:15GS	Indicates capture plain double thumb image
AMP	14:018:01USUPGS	Indicates right thumb cannot be printed
SEG	14:021:GS	Empty indicating finger segment positions should be calculated
DAT	14.999:FS	Empty indicating image capture required

A.2 Ten plain impressions (no TOT)

A.2.1 Type-1 record

Field	Contents	Comments
LEN	1.001:76GS	Mandatory length field
VER	1.002:0400GS	Indicates using 2007 version of standard
CNT	1.003:1US11RS2US0RS4US 1RS 4US2RS4US3RS4US4RS 4US5RS4US6RS4US7RS 4US8RS4US9RS4US10FS	File content indicating one Type-2 record and ten Type 4 records

A.2.2 Type-2 record (IDC 0)

Field	Contents	Comments
LEN	2.001:18GS	Mandatory length field
IDC	2.002:00FS	Mandatory IDC field

A.2.3 Type-4 record (IDC 1)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x00	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x01FFFFFFFFFF	Request for right thumb
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.4 Type-4 record (IDC 2)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x01	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x02FFFFFFFFFF	Request for right index finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.5 Type-4 record (IDC 3)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x02	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x03FFFFFFFFFF	Request for right middle finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.6 Type-4 record (IDC 4)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x03	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x04FFFFFFFFFF	Request for right ring finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.7 Type-4 record (IDC 5)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x04	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x05FFFFFFFFFF	Request for right little finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.8 Type-4 record (IDC 6)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x05	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x06FFFFFFFFFF	Request for left thumb
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.9 Type-4 record (IDC 7)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x06	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x07FFFFFFFFFF	Request for left index finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.10 Type-4 record (IDC 8)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x07	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x08FFFFFFFFFF	Request for left middle finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.11 Type-4 record (IDC 9)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x08	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x09FFFFFFFFFF	Request for left ring finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

A.2.12 Type-4 record (IDC 10)

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x09	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x0AFFFFFFFFFF	Request for left little finger
ISR	0x01	Request for native scanning resolution to be used
HLL	0x0000	0 indicating image width should be filled in
VLL	0x0000	0 indicating image height should be filled in
GCA	0x01	Request for storage using WSQ compression
DAT		Empty indicating image capture required

Annex B
(informative)

BioAPI calling sequence example

```
void Example() {

    static BioAPI_VERSION BioAPIVersion = 0x21;
    static BioAPI_UUID bspUuid = ...;
        /* some UUID identifying a particular BSP */
    static BioAPI_UUID controlCodeConfigureSensorUnitFromANSINISTTemplate = ...;
        /* some UUID specified in the Standard */
    BioAPI_UNIT_LIST_ELEMENT unitList[1];
    BioAPI_RETURN returnValue;
    BioAPI_HANDLE bspHandle;
    BioAPI_UNIT_ID sensorUnitID;
    BioAPI_DATA inputData, outputData;
    BioAPI_BIR_BIOMETRIC_DATA_FORMAT bdbFormat;
    BioAPI_BIR_HANDLE capturedBIRLeftSlap;
    BioAPI_BIR_SUBTYPE subtype;
    BioAPI_BIR bir;

    returnValue = BioAPI_Init(BioAPIVersion);

    returnValue = BioAPI_BSPLoad(&bspUuid, EventHandler, NULL);

    sensorUnitID = ...;
        /* some integer identifying a sensor unit capable of tenprint capture */

    unitList[0].UnitCategory = BioAPI_CATEGORY_SENSOR;
    unitList[0].UnitId = sensorUnitID;

    returnValue = BioAPI_BSPAttach(&bspUuid, BioAPIVersion, &unitList, 1, &bspHandle);

    inputData.Data = ...;      /* address of a buffer containing the input ANSI/NIST file */
    inputData.Length = ...;   /* length of the data in the buffer */

    returnValue = BioAPI_ControlUnit(bspHandle,
        sensorUnitID,
        &controlCodeConfigureSensorUnitFromANSINISTTemplate,
        &inputData,
        &outputData);

    if(outputData.Data != NULL) {
        ...;   /* do something with the data returned by ControlUnit */
        BioAPI_Free(outputData.Data);
        outputData.Data = NULL;
        outputData.Length = 0;
    }

    bdbFormat.FormatOwner = ...;   /* BDB format owner of the ANSI/NIST file */
    bdbFormat.FormatType = ...;    /* BDB format type of the ANSI/NIST file */

    subtype = BioAPI_BIR_SUBTYPE_LEFT_INDEXFINGER_BIT |
        BioAPI_BIR_SUBTYPE_LEFT_MIDDLEFINGER_BIT |
        BioAPI_BIR_SUBTYPE_LEFT_RINGFINGER_BIT |
        BioAPI_BIR_SUBTYPE_LEFT_LITTLEFINGER_BIT;
    returnValue = BioAPI_Capture(bspHandle,
        BioAPI_PURPOSE_ENROLL,
        subtype,
        &bdbFormat,
        &capturedBIRLeftSlap,
        -1, NULL);

    returnValue = BioAPI_GetBIRFromHandle(bspHandle,
        capturedBIRLeftSlap,
        &bir);

    ... = bir.BiometricData.Data;   /* address of a buffer containing the output COB file */
    ... = bir.BiometricData.Length; /* length of the data in the buffer */

    ...;   /* use the data in the buffer */

    BioAPI_Free(bir.BiometricData.Data);
    BioAPI_BSPDetach(&bspUuid);
    BioAPI_BSPUnload(&bspUuid, EventHandler, NULL);
    BioAPI_Terminate();
}
```

ICS 35.040

Price based on 25 pages