

1 InterNational Committee for Information Technology Standards
2 INCITS Secretariat, Information Technology Industry Council (ITI)
3 1250 Eye St. NW, Room 200, Washington, DC 20005
4 Telephone 202-737-8888; Fax 202-638-4922
5 email: incits@itic.org
6
7
8

9 **DOC: M1/06-0893**

10 Date: October 25, 2006
11 Source: Ad Hoc Group for Use of BioAPI to Support Tenprint Capture
12 Reply to: David Benini
13 Phone: (781) 687-0306
14 Email: dbenini@aware.com
15
16
17

18 **Working draft for "Tenprint Capture Using BioAPI" standard (Project 1832-D)**
19

1 **INCITS Project 1832-D**

2

3 InterNational Committee for Information Technology Standards

4 INCITS Secretariat, Information Technology Industry Council (ITI)

5 1250 Eye St. NW, Suite 200, Washington, DC 20005

6 Telephone 202-737-8888; Fax 202-638-4922

7 email: incits@itic.org

8

9

10 **Title:** Tenprint Capture Using BioAPI

11

12 **Source:** Project Editor

13

14 **Date:** October 25, 2006

15

16 **Revision:** 3

17

18

Revision	Date	M1 Doc. #	Comments
1	April 5, 2006	376	Baseline text
2	Sept 26, 2006	738	Working draft
3	October 25, 2006	893	Working draft incorporating M060892 comments

19

20 **Project editor:**

21 David Benini

22 Aware, Inc.

23 dbenini@aware.com

24 (781) 687-0306

1 **Foreword**

2 INCITS (The International Committee for Information Technology Standards) is the ANSI
3 recognized Standards Development Organization for information technology within the United
4 States of America. Members of INCITS are drawn from Government, Corporations, Academia
5 and other organizations with a material interest in the work of INCITS and its Technical
6 Committees. INCITS does not restrict membership and attracts participants in its technical work
7 from 13 different countries, and operates under the rules of the American National Standards
8 Institute.

9 In the field of Biometrics, INCITS has established the Technical Committee M1. Standards
10 developed by this Technical Committee have reached consensus throughout the development
11 process and have been thoroughly reviewed through several Public Review processes.

12 This document is submitted to INCITS M1-Biometrics by the M1.2 Ad Hoc Group on Tenprint
13 Capture Using BioAPI (AHGTCUB).

1	Table of Contents	
2	0 INTRODUCTION	1
3	1 SCOPE	1
4	2 CONFORMANCE	1
5	3 NORMATIVE REFERENCES	1
6	4 TERMS AND DEFINITIONS	2
7	4.1 CAPTURE CONTROL BLOCK (CCB)	2
8	4.2 CAPTURE OUTPUT BLOCK (COB)	2
9	4.3 ROLLED PRINTS	2
10	4.4 SLAP PRINTS.....	2
11	4.5 TENPRINT CAPTURE	2
12	5 REQUIREMENTS	2
13	6 BIOAPI FUNCTION CALLS	3
14	6.1 BIOAPI_CONTROLUNIT()	3
15	6.1.1 Usage	3
16	6.1.2 Prototype	3
17	6.1.3 Parameters.....	3
18	6.1.4 Additional Error Return Codes.....	4
19	6.2 BIOAPI_CAPTURE()	4
20	6.2.1 Usage	4
21	6.2.2 Prototype	4
22	6.2.3 Parameters.....	4
23	6.2.4 Additional Error Return Codes.....	5
24	6.3 BIOAPI_QUERYUNITS()	5
25	6.3.1 Usage	5
26	6.3.2 Prototype	5
27	6.3.3 Parameters.....	5
28	6.3.4 BioAPI_UNIT_SCHEMA.....	6
29	7 CAPTURE CONTROL BLOCKS (CCB)	7
30	8 RECORD TYPES AND FIELDS	8
31	8.1 CONTROL AND REQUIRED FIELDS	8
32	8.2 IMAGE HEIGHT AND WIDTH	9
33	8.3 IMAGE DATA.....	9
34	8.4 AMPUTATED OR BANDAGED FINGERS	9
35	8.5 CAPTURE RESOLUTION	10
36	8.6 IMAGE COMPRESSION	10
37	8.7 SELECTING IMAGES TO BE CAPTURED.....	10
38	8.8 CALCULATION OF FINGER SEGMENT POSITION.....	10
39	8.9 IMAGE QUALITY.....	11
40	8.10 BITS PER PIXEL.....	11
41	9 RETURNED BIR	11

1	9.1	BIR HEADER.....	11
2	9.2	BDB FORMAT	11
3	ANNEX A: EXAMPLE CAPTURE CONTROL BLOCKS (CCB) FOR ACQUISITION OF		
4	IMAGES (INFORMATIVE)..... 13		
5	A.1	THREE SLAPS (FANC TOT)	13
6	A.1.1	Type-1 Record.....	13
7	A.1.2	Type-2 Record (IDC 0).....	13
8	A.1.3	Type-14 Record (IDC 1).....	13
9	A.1.4	Type-14 Record (IDC 2).....	13
10	A.1.5	Type-14 Record (IDC 3).....	14
11	A.2	TEN PLAIN IMPRESSIONS (NO TOT)	14
12	A.2.1	Type-1 Record.....	14
13	A.2.2	Type-4 Record (IDC 0).....	15
14	A.2.3	Type-4 Record (IDC 1).....	15
15	A.2.4	Type-4 Record (IDC 2).....	15
16	A.2.5	Type-4 Record (IDC 3).....	15
17	A.2.6	Type-4 Record (IDC 4).....	16
18	A.2.7	Type-4 Record (IDC 5).....	16
19	A.2.8	Type-4 Record (IDC 6).....	16
20	A.2.9	Type-4 Record (IDC 7).....	17
21	A.2.10	Type-4 Record (IDC 8).....	17
22	A.2.11	Type-4 Record (IDC 9).....	17
23	ANNEX B: BIOAPI CALLING SEQUENCE EXAMPLE (INFORMATIVE)..... 19		
24			

1 0 Introduction

2 Currently the implementation of tenprint capture (a local/client-side operation) is done in a
3 proprietary fashion, without interchangeability, interoperability, or process standardization in
4 mind. With an increase in the need for and deployment of large-scale identity management and
5 credentialing systems (which generally require a tenprint capture as part of the identity vetting
6 and background checking process), a method of performing this operation based on existing
7 technical interface standards is required. The BioAPI standard is the standard most suited for
8 this purpose; however, there are a number of areas where options exist or that implementation
9 details are left to the vendor or application. This standard is needed to further constrain the use
10 of the BioAPI standard in the tenprint capture context and to provide additional guidance that
11 will improve both usability and interoperability.

12 1 Scope

13 This standard specifies requirements for the use of ISO/IEC 19784-1, BioAPI Specification (also
14 known as BioAPI 2.0), a software interface standard, for the purpose of performing a tenprint
15 capture operation.

16 This may include one or more of the following:

- 17 1. Identification of BioAPI functions to be utilized and the order (if any) in which they are to
18 be called
- 19 2. Specification of values for function parameters
- 20 3. Definition of GUI (graphical user interface) events (for use with an application controlled
21 GUI)
- 22 4. User interface specifications for use with a BSP (biometric service provider) controlled
23 GUI
- 24 5. Sample calling sequences and example inputs/outputs

25 2 Conformance

26 Biometric Service Providers (BSPs) that claim conformance to this standard shall satisfy the
27 normative requirements as described in Clauses 5 through 9.

28 3 Normative References

29 The following normative references are critical to the proper understanding and application of
30 this standard. For dated references, only the edition cited applies. For undated references, the
31 latest edition of the referenced document (including any amendments) applies.

32

- 33 1. ANSI/NIST-ITL 1-2006 (draft) "Data Format for the Interchange of Fingerprint, Facial, &
34 Other Biometric Information", October 19, 2006
- 35 2. "Wavelet Scalar Quantization (WSQ) Grayscale Fingerprint Image Compression
36 Specification", December 19, 1997 (IAFIS-IC-0010(V3)), Federal Bureau of Investigation
- 37 3. ISO/IEC 19784-1 Information Technology – Biometric Application Programming Interface
38 – Part 1: BioAPI Specification
 - 39 a. Amd 1, BioGUI Specification (PDAM)

1 4. ISO/IEC 19785-1, Common Biometric Exchange Formats Framework – Part 1: Data
2 Element Specification

3 5. ISO/IEC 19785-2, Common Biometric Exchange Formats Framework – Part2:
4 Procedures for the Operation of the Biometric Registration Authority

5 **4 Terms and Definitions**

6 **4.1 Capture control block (CCB)**

7 A formatted data structure containing placeholders for data objects desired for return by BSP.

8 **4.2 Capture output block (COB)**

9 A formatted data structure containing data objects returned by the BSP and also placeholders for those data objects
10 that could not be returned by the BSP.

11 **4.3 Rolled prints**

12 An image of a single finger, or set of single fingers, obtained by rolling the finger from the
13 extreme left side to the extreme right side (or extreme right side to the extreme left side) in order
14 to obtain as much information from the fingertip as possible (i.e., “nail-to-nail”).

15 **4.4 Slap prints**

16 An image of one or all three “plain impressions” obtained by laying the 4 left fingers, 4 right
17 fingers, or two thumbs flat on a capture surface (e.g., platen of a fingerprint scanner).

18 **4.5 Tenprint capture**

19 The electronic capture of ten human fingerprints, normally for the purposes of law enforcement
20 or background checking.

21 **5 Requirements**

22 This standard describes requirements for a BioAPI 2.0 compliant BSP to support tenprint
23 capture using livescan devices.

24 Selection of the finger images captured and the processing done on the images shall be done
25 through use of a capture control block (CCB) based upon the ANSI/NIST ITL 1-2006 standard.
26 This CCB contains incomplete data fields for each desired image. Each CCB shall have empty,
27 invalid fields for information that the application requires the BSP to generate and output as a
28 capture output block (COB). This COB shall be based upon the ANSI/NIST ITL 1-2006
29 standard and may contain multiple Type-4 or Type-14 fingerprint image records.

30 Functionality associated with amputated or bandaged fingers is not directly supported by the
31 ANSI/NIST ITL 1-2006 standard. See Section 8.4, Amputated or Bandaged Fingers, which
32 defines how this standard addresses the need for this functionality.

33 Two BioAPI interface functions shall be used to perform the tenprint data acquisition:

- 34 1. BioAPI_ControlUnit()
- 35 2. BioAPI_Capture()

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

1 See Section 6: BioAPI Function Calls.

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

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

7 **6 BioAPI Function Calls**

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

10 **6.1 BioAPI_ControlUnit()**

11 **6.1.1 Usage**

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

16 **6.1.2 Prototype**

```
17 BioAPI_RETURN BioAPI BioAPI_ControlUnit  
18     (BioAPI_HANDLE BSPHandle,  
19     BioAPI_UNIT_ID UnitID,  
20     uint32_t ControlCode  
21     const BioAPI_DATA *InputData,  
22     BioAPI_Data *OutputData);
```

23 **6.1.3 Parameters**

24 **6.1.3.1 BSPHandle (input)**

25 The handle of the attached BSP.

26 **6.1.3.2 UnitID (input)**

27 ID of the BioAPI Unit.

28 **6.1.3.3 ControlCode (input)**

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

31 **6.1.3.4 InputData (input)**

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

1 **6.1.3.5 OutputData (output)**

2 No output is needed. The buffer address will be NULL and length set to 0 within the
3 BioAPI_DATA structure. Usage of the OutputData parameter to return diagnostic data about
4 the CCB should be considered.

5 **6.1.4 Additional Error Return Codes**

6 Error codes should be assigned to indicate errors detected in the CCB passed in the InputData
7 parameter.

8 **6.2 BioAPI_Capture()**

9 **6.2.1 Usage**

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

14 **6.2.2 Prototype**

```
15 BioAPI_RETURN BioAPI BioAPI_Capture  
16     (BioAPI_HANDLE BSPHandle,  
17      BioAPI_BIR_PURPOSE Purpose,  
18      BioAPI_BIR_SUBTYPE Subtype,  
19      BioAPI_BIR_BIOMETRIC_DATA_FORMAT *OutputFormat,  
20      BioAPI_BIR_HANDLE *CapturedBIR,  
21      Int32_t Timeout,  
22      BioAPI_BIR_HANDLE *AuditData);
```

23 **6.2.3 Parameters**

24 **6.2.3.1 BSPHandle (input)**

25 The handle of the attached BSP.

26 **6.2.3.2 Purpose (input)**

27 A value indicating the purpose of the biometric data capture. For the purposes of tenprint
28 capture, the value BioAPI_PURPOSE_ENROLL shall be used.

29 **6.2.3.3 Subtype (input/optional)**

30 Specifies which subtype to capture. This parameter does not have the flexibility to describe the
31 range of possible finger image sets that might be captured. A value of
32 BioAPI_NO_SUBTYPE_AVAILABLE (0x00) shall be used to indicate that the BSP is to select
33 the subtypes using the current CCB.

34 **6.2.3.4 OutputFormat (input/optional)**

35 Specifies which BDB (biometric data block) format to use for the returned CapturedBIR. Format
36 Owner and Format Type shall be set to 0x001 and B8013 respectively to designate that the BIR

1 (biometric information record) shall consist of a CCB. Alternatively a NULL pointer may be used
2 to indicate that output format selection will be handled by the BSP.

3 **6.2.3.5 CapturedBIR (output)**

4 A handle to a BIR containing captured data, the CCB. See Section 9 for a specification of this
5 BIR.

6 **6.2.3.6 Timeout (input)**

7 Specifies the timeout value (in milliseconds) for the capture operation. If this timeout is reached,
8 the function returns an error, and no results. This value can be any positive number. A -1 value
9 means the BSP's default timeout value will be used.

10 **6.2.3.7 AuditData (output/optional)**

11 There may be no need for this parameter. A handle value of
12 `BioAPI_UNSUPPORTED_BIR_HANDLE` shall be used to indicate that AuditData is not
13 available.

14 **6.2.4 Additional Error Return Codes**

15 It may not be necessary to add additional error codes. BioAPI already includes codes such as
16 `BioAPIERR_UNABLE_TO_CAPTURE` to indicate that the BSP could not collect the data that it
17 is meant to.

18 **6.3 BioAPI_QueryUnits()**

19 **6.3.1 Usage**

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

23 **6.3.2 Prototype**

```
24 BioAPI_RETURN BioAPI BioAPI_QueryUnits  
25     (const BioAPI_UUID *BSPUuid,  
26      BioAPI_UNIT_SCHEMA **UnitSchemaArray,  
27      uint32_t *NumberOfElements);
```

28 **6.3.3 Parameters**

29 **6.3.3.1 BSPUuid (input)**

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

32 **6.3.3.2 UnitSchemaArray (output)**

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

1 **6.3.3.3 NumberOfElements (output)**

2 A pointer to the number of elements in the array.

3 **6.3.4 BioAPI_UNIT_SCHEMA**

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

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

10 **6.3.4.1 UnitProperties**

11 UUID indicating a set of properties of the BioAPI Unit. The indicated set can either be specified
12 by each vendor or follow a related standard. The UUID “ba38567f-66be-468c-9ada-
13 0b1eeb0cc7fb” shall indicate that the unit supports the profile/standard for “Tenprint Capture
14 Using BioAPI”.

15 **6.3.4.2 UnitPropertyID**

16 UUID of the format of the following Unit property structure. The UUID “10568e83-c7f8-418d-
17 9913-4e80f01aecfb” shall represent the format ID for Tenprint Scanner Properties.

18 **6.3.4.3 UnitProperty**

19 Address and length of a memory buffer containing the Unit property describing the BioAPI Unit.
20 The format and content of the Unit property can either be specified by the vendor or be specified
21 in a related standard. For tenprint scanners conforming to this standard, the Unit property
22 structure is defined as follows.

23 Sequential, bit-oriented fields, all of which must be present. Big-endian (network) byte ordering
24 shall be used.

25 *NOTE: UnitProperty is of type BioAPI_DATA which includes a length value, so it does not need to be restated here.*

26 **Table 1. UnitProperty Structure**

Field Name	Type	Size	Valid 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
Segmentation	Boolean	1B	0x00, 0x01	'01' indicates device can calculate bounding boxes
Finger extraction	Boolean	1B	0x00, 0x01	'01' indicates device can extract individual fingerprints from a slap image (return segmented flats)
Number of quality algorithms supported	Integer	1B	0-255	Number of quality algorithm identifiers contained in the next field

Field Name	Type	Size	Valid Values	Notes
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 4-bytes (see Note 1)	Array of unique identifiers for each compression algorithm supported by the device
Number of segmentation algorithms supported	Integer	1B	0-255	Number of segmentation algorithm identifiers contained in the next field
Segmentation algorithms supported	Integer (array)	Var.	Each algorithm is described by 4-bytes (see Note 1)	Array of unique identifiers for each segmentation algorithm supported by the device
Number of image resolutions supported	Integer	1B	0-255	Number of resolution values contained in the next field
Image resolutions supported	Integer	2B	0-64535	Value of each resolutions (in ppi) supported by the device (e.g., 500)
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

1
 2 *Note 1 – Quality algorithm identifiers are structured as follows (analogous to the representation of data format*
 3 *identifiers):*
 4 *<algorithm owner> - 2-byte integer (hex) value representing the vendor/developer of the algorithm (e.g.,*
 5 *NIST, Vendor A) as registered with the IBIA as a product owner.*
 6 *<algorithm type> - 2-byte integer (hex) value representing the specific algorithm (i.e., product code) as*
 7 *assigned by the <algorithm owner>.*

8 **7 Capture Control Blocks (CCB)**

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

13 A BSP shall never remove valid data contained in a CCB, but rather only correct missing or
 14 invalid data. This allows CCBs to contain some valid data that is not related to the capture
 15 process. For example the CCB could include a Type-2 record that already had all the

1 demographic data filled in. The BSP must then include this filled-in Type-2 record in its output
2 BDB.

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

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

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

14 **8 Record Types and Fields**

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

18 **8.1 Control and Required Fields**

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

23 **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)
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 EFTS 7.1 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)
2.084	AMP	O	Indicates fingers which cannot be printed, see section 8.4
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 section 8.7
4.004	FGP	M	Finger/image to capture, see section 8.7
4.005	ISR	M	Image scanning resolution, see section 8.5
4.006	HLL	M	Image width, see section 8.2
4.007	VLL	M	Image height, see section 8.2

Record/Field	Ident	Cond Code	Purpose
4.008	GCA	M	Compression type (none/WSQ), see section 8.4
4.009	DAT	M	Image data, see section 8.3
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 section 8.7
14.006	HLL	M	Image width, see section 8.2
14.007	VLL	M	Image height, see section 8.2
14:008	SLC	M	Sampling units for capture resolution, see section 8.5
14:009	HPS	M	Horizontal capture resolution, see section 8.5
14:010	VPS	M	Vertical capture resolution, see section 8.5
14:011	CGA	M	Compression type, see section 8.6
14:012	BPX	M	Bits per pixel for capture, see section 8.10
14:013	FGP	M	Finger/image to capture, see section 8.7
14:021	SEG	O	Finger segmentation positions, see section 8.8
14.022	IQM	O	NFIQ finger image quality, see section 8.9
14.023	ASM	O	Finger segmentation positions (algorithm indicated), see section 8.8
14.024	AFM	O	Finger image quality (algorithm indicated), see section 8.9
14.999	DAT	M	Image data, see section 8.3

1

2 **8.2 Image Height and Width**

3 For the Type-4 and the Type-14 records, the HLL and VLL fields indicate the dimensions of the
 4 image. For Type-4 or Type-14 records occurring in the CCB these fields are mandatory. Zero
 5 values in these fields indicate that the BSP should select the size of the captured image. Non-
 6 zero values can be used with BSPs that are capable of sizing images, to specify the desired
 7 size of the captured image. When non-zero values are used in the CCB, BSP’s should return
 8 an error if they are unable to satisfy the specified image size.

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

11 **8.3 Image Data**

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

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

18 **8.4 Amputated or Bandaged Fingers**

19 The ANSI/NIST ITL 1-2006 standard allows user defined fields in the Type 2 record. The Type
 20 2 record field 2.084 shall be used to indicate fingers not available for image capture, e.g. due to
 21 amputation or bandaging.

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

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

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

18 **8.5 Capture Resolution**

19 The Type-14 SLC, HPS, VPS fields may be used to indicate the capture resolution. These
20 fields can be used to select either 500 or 1000 ppi image capture. If the BSP is not capable of
21 capturing at the specified resolution it shall return an error.

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

25 **8.6 Image Compression**

26 The Type-14 CGA or the Type-4 GCA field shall be used to select the type of compression to
27 perform on the images. Values for compression method should be taken from Table 1 of the
28 ANSI/NIST-ITL 1-2006 standard. For example, using a value of "NONE" in the Type-14 CGA
29 field would indicate that the image data should be stored in raw format.

30 **8.7 Selecting Images to be Captured**

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

35 **8.8 Calculation of Finger Segment Position**

36 Fingerprint segmentation is possible through use of Type-14 records for fingerprint images. The
37 Type-14 SEG or ASM field may be used to select whether finger segmentation should be done
38 on the image. When either of these fields are sent empty, the segmentation would be
39 performed and the results stored in this field. Using the ASM field allows the type of
40 segmentation algorithm used to be selected. If the ASM field is used in the CCB, the second
41 segmentation quality score item should be empty, but the algorithm item should be filled in so
42 that the algorithm type will get selected.

8.9 Image Quality

Fingerprint quality calculation is possible through use of Type-14 records for fingerprint images. The Type-14 IQM or AFM field may be used to select which quality algorithm is to be used. If the IQM field is used then NFIQ quality will always be done. If the AFM field is used in the CCB, the second quality score item should be left empty, but the algorithm item should be filled in so that the algorithm type will get selected.

8.10 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 Returned BIR

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

9.1 BIR Header

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

Table 3. BIR Header Elements

BIR Header Element	Value
BioAPI_BIR_DATA_TYPE	0x01 (Raw)
BioAPI_BIR_BIOMETRIC_DATA_FORMAT	Format Owner: 0x001B Format Type: 0x8013
BioAPI_BIR_PURPOSE	3 (Enroll)
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.

9.2 BDB Format

The Biometric Data Block (body of the BIR) shall contain the COB in accordance with ANSI/NIST–ITL 1-2006 syntax. 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, but can be used to indicate missing fingers, if the BSP supports this capability.
- One or more Type-4 or Type-14 records shall be present, in the quantity and order as specified in the Type-1 record.

- 1 *NOTE: This sequence of ANSI/NIST-ITL 1-2006 logical record types comprises the BDB identified by the*
- 2 *Format ID 0x001B8013.*
- 3 See Annex A for example record sets (BDB content) for the return of three slap images and for
- 4 ten individual flat images.

ANNEX A: EXAMPLE CAPTURE CONTROL BLOCKS (CCB) FOR ACQUISITION OF IMAGES (INFORMATIVE)

A.1 Three Slaps (FANC TOT)

A.1.1 Type-1 Record

Field	Contents	Comments
LEN	1.001:60Gs	Mandatory length field
VER	1.002:0300Gs	Indicates using 2000 version of standard
CNT	1.003:1us4rs2us0rs 14us1rs14us2rs14us3gs	File content indicating one Type-2 Record and three Type-14 records
TOT	1.004:FANCgs	Indicates type of transaction

A.1.2 Type-2 Record (IDC 0)

Field	Contents	Comments
LEN	2.001:30Gs	Mandatory length field
IDC	2.002:00Gs	Mandatory IDC field
AMP	2.084:01usUPGs	Indicates right thumb cannot be printed

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

Field	Contents	Comments
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

1

2 **A.1.5 Type-14 Record (IDC 3)**

3

Field	Contents	Comments
LEN	14.001:125Gs	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
SEG	14:021:Gs	Empty indicating finger segment positions should be calculated
DAT	14.999:Fs	Empty indicating image capture required

4

5 **A.2 Ten Plain Impressions (no TOT)**6 **A.2.1 Type-1 Record**

7

Field	Contents	Comments
LEN	1.001:71Gs	Mandatory length field
VER	1.002:0300Gs	Indicates using 2000 version of standard
CNT	1.003:1Us10Rs4Us0Rs 4Us1Rs4Us2Rs4Us3Rs 4Us4Rs4Us5Rs4Us6Rs 4Us7Rs4Us8Rs4Us9Fs	File content indicating ten Type-4 records

8

1 **A.2.2 Type-4 Record (IDC 0)**

2

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x00	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x01FFFFFFFF	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

3

4 **A.2.3 Type-4 Record (IDC 1)**

5

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x01	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x02FFFFFFFF	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

6

7 **A.2.4 Type-4 Record (IDC 2)**

8

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x02	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x03FFFFFFFF	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

9

10 **A.2.5 Type-4 Record (IDC 3)**

11

Field	Contents	Comments
LEN	0x00000012	Mandatory length field

Field	Contents	Comments
IDC	0x03	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x04FFFFFFFF	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

1

2 **A.2.6 Type-4 Record (IDC 4)**

3

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x04	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x05FFFFFFFF	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

4

5 **A.2.7 Type-4 Record (IDC 5)**

6

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x05	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x06FFFFFFFF	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

7

8 **A.2.8 Type-4 Record (IDC 6)**

9

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x06	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x07FFFFFFFF	Request for left index finger

Field	Contents	Comments
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

1

2 **A.2.9 Type-4 Record (IDC 7)**

3

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x07	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x08FFFFFFFF	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

4

5 **A.2.10 Type-4 Record (IDC 8)**

6

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x08	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x09FFFFFFFF	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

7

8 **A.2.11 Type-4 Record (IDC 9)**

9

Field	Contents	Comments
LEN	0x00000012	Mandatory length field
IDC	0x09	Mandatory IDC field
IMP	0x00	Request for livescan plain impression type
FGP	0x0AFFFFFFFF	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

Field	Contents	Comments
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: BIOAPI CALLING SEQUENCE EXAMPLE (INFORMATIVE)

```
1
2
3 void Example() {
4
5     static BioAPI_VERSION BioAPIVersion = 0x21;
6     static BioAPI_UUID bspUuid = ...;
7         /* some UUID identifying a particular BSP */
8     static BioAPI_UUID controlCodeConfigureSensorUnitFromEFTSTemplate = ...;
9         /* some UUID specified in the Standard */
10    BioAPI_UNIT_LIST_ELEMENT unitList[1];
11    BioAPI_RETURN returnValue;
12    BioAPI_HANDLE bspHandle;
13    BioAPI_UNIT_ID sensorUnitID;
14    BioAPI_DATA inputData, outputData;
15    BioAPI_BIR_BIOMETRIC_DATA_FORMAT bdbFormat;
16    BioAPI_BIR_HANDLE capturedBIRLeftSlap;
17    BioAPI_BIR_SUBTYPE subtype;
18    BioAPI_BIR bir;
19
20    returnValue = BioAPI_Init(BioAPIVersion);
21
22    returnValue = BioAPI_BSPLoad(&bspUuid, EventHandler, NULL);
23
24    sensorUnitID = ...;
25        /* some integer identifying a sensor unit capable of tenprint capture */
26
27    unitList[0].UnitCategory = BioAPI_CATEGORY_SENSOR;
28    unitList[0].UnitId = sensorUnitID;
29
30    returnValue = BioAPI_BSPAttach(&bspUuid, BioAPIVersion, &unitList, 1, &bspHandle);
31
32    inputData.Data = ...; /* address of a buffer containing the input EFTS file */
33    inputData.Length = ...; /* length of the data in the buffer */
34
35    returnValue = BioAPI_ControlUnit(bspHandle,
36        sensorUnitID,
37        &controlCodeConfigureSensorUnitFromEFTSTemplate,
38        &inputData,
39        &outputData);
40
41    if(outputData.Data != NULL) {
42        ...; /* do something with the data returned by ControlUnit */
43        BioAPI_Free(outputData.Data);
44        outputData.Data = NULL;
45        outputData.Length = 0;
46    }
47
48    bdbFormat.FormatOwner = ...; /* BDB format owner of the ANSI/NIST/EFTS file */
49    bdbFormat.FormatType = ...; /* BDB format type of the ANSI/NIST/EFTS file */
50
51    subtype = BioAPI_BIR_SUBTYPE_LEFT_INDEXFINGER_BIT |
52        BioAPI_BIR_SUBTYPE_LEFT_MIDDLEFINGER_BIT |
53        BioAPI_BIR_SUBTYPE_LEFT_RINGFINGER_BIT |
```

```
1         BioAPI_BIR_SUBTYPE_LEFT_LITTLEFINGER_BIT;
2
3     returnValue = BioAPI_Capture(bspHandle,
4         BioAPI_PURPOSE_ENROLL,
5         subtype,
6         &bdbFormat,
7         &capturedBIRLeftSlap,
8         -1, NULL);
9
10    returnValue = BioAPI_GetBIRFromHandle(bspHandle,
11        capturedBIRLeftSlap,
12        &bir);
13
14    ... = bir.BiometricData.Data; /* address of a buffer containing the output EFTS file */
15    ... = bir.BiometricData.Length; /* length of the data in the buffer */
16
17    ...; /* use the data in the buffer */
18
19    BioAPI_Free(bir.BiometricData.Data);
20    BioAPI_BSPDetach(&bspUuid);
21    BioAPI_BSPUnload(&bspUuid, EventHandler, NULL);
22    BioAPI_Terminate();
23 }
24
```