INTERNATIONAL STANDARD

ISO/IEC 30106-1

First edition
2016-03-15

AMENDMENT 1
2019-05

Information technology — Object oriented BioAPI —

Part 1: Architecture

AMENDMENT 1: Additional specifications and conformance statements

Technologies de l'information — Objet orienté BioAPI —

Partie 1: Architecture

AMENDEMENT 1: Spécifications et déclarations de conformité complémentaires

Reference number
ISO/IEC 30106-1:2016/Amd.1:2019(E)

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

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

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

A list of all parts in the ISO/IEC 30106 series can be found on the ISO website.

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

Information technology — Object oriented BioAPI —

Part 1:
Architecture

AMENDMENT 1: Additional specifications and conformance statements

Page 1, Clause 2 Normative references

Update the listing for ISO/IEC 19785-3 to the following:

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

Page 2

Add new Clause 5 Conformance with the following text and renumber subsequent clauses.

> **Clause 5 Conformance**
>
> Those products that claim conformance with any of the parts in the ISO/IEC 30106 series, shall comply with the requirements stated in Annex A.

Page 8, Clause 6

Replace Clause 6 with the following:

Clause 6 Object Oriented BioAPI CBEFF Patron Formats

Object Oriented BioAPI is able to use biometric data coded as Self-Identifying BIRs, either Simple BIRs or Complex BIRs, following the structure and definition of CBEFF (i.e. ISO/IEC 19785). In particular, Object Oriented BioAPI shall use the following CBEFF Patron Formats:

— For Simple BIRs the patron format to be used is the one called "Self-identifying Tag-oriented Simple BIR", registered as Patron Format Owner 257 and Patron Format Type 12, and described in ISO/IEC 19785-3.

— For Complex BIRs the patron format to be used is the one called "Self-identifying Tag-oriented Complex BIR", registered as Patron Format Owner 257 and Patron Format Type 13, and described in ISO/IEC 19785-3.

The particular tagged format (e.g. TLV, XML or JSON) to be used will be determined by the application and/or platform used.

Page 29

Insert the following clause after 8.7, to become Clause 9.

Clause 9 Additional specifications

9.1 Standardized control codes for additional functionality

9.1.1 General

When a BSP or BFP requires the addition of further functionality to the one already defined in previous clauses, the method to be used is:

byte[] ControlUnit (int unitID, int controlCode, byte[] inputData)

The method can be used by different vendors, providing different functionalities. In order to minimize interoperability issues, the control codes shall be defined to allow both standardized behaviour, and proprietary functionality.

Therefore, control codes shall have the same format:

Code (in hex)	Description
00 01 XX XX	Standardized functionality for Archive units
00 02 XX XX	Standardized functionality for Processing units
00 03 XX XX	Standardized functionality for Comparison units
00 04 XX XX	Standardized functionality for Capture units
00 0X XX XX	RFU for SC37 standardized functionality
(01-FF) XX XX XX	Proprietary functionality (out of the scope of SC37 description)

9.1.2 Control codes for Archive Units

The following codes are standardized for Archive units:

Code	Input data	Output data	Description
00 01 00 01			

9.1.3 Control codes for Processing Units

The following codes are standardized for Archive units:

Code	Input data	Output data	Description
00 02 00 01	Byte threshold	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Change the quality threshold to accept a BIR as in input for Processing. The algorithm may have its own threshold and also rules to accept new thresholds (e.g. not accepting thresholds below a determined minimum value).

9.1.4 Control codes for Comparison Units

The following codes are standardized for Archive units:

Code	Input data	Output data	Description
00 03 00 01	int minimumFMR	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Change the comparison threshold to determine the minimum FMR to allow a match. The unit may have its own threshold and also rules to accept new thresholds (e.g. not accepting thresholds below a determined minimum value).

9.1.5 Control codes for Capture Units

The following codes are standardized for Archive units:

Code	Input data	Output data	Description
00 04 00 01	Message (string)	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Message to be shown by the acquisition unit to request a trait to be presented
00 04 00 02	Message (string)	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Message to be shown by the acquisition unit to ask the user to wait while presenting the trait
00 04 00 03	Message (string)	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Message to be shown by the acquisition unit to ask the user to remove the trait
00 04 00 04	Message (string)	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Message to be shown by the acquisition unit to ask the user to retry the presentation of the trait
00 04 01 (01 – 04)	Byte string with the acoustic information of the message	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Same as codes 00 04 00 (01 – 04) but instead of a text message, an acoustic signal.

Code	Input data	Output data	Description
00 04 00 03	Message (string)	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Message to be shown by the acquisition unit to ask the user to remove the trait
00 04 00 04	Message (string)	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Message to be shown by the acquisition unit to ask the user to retry the presentation of the trait
00 04 01 (01 – 04)	Byte string with the acoustic information of the message	00 – OK 01 – Not accepted by unit 02 – Not available in unit 03 – Other error	Same as codes 00 04 00 (01 – 04) but instead of a text message, an acoustic signal.

Add new Annex A.

Annex A
(normative)

Conformance statements

A.1 General

Conformance to this document falls into the following three classes:

- OO BioAPI conformant biometric application
- OO BioAPI conformant BioAPI Framework
- OO BioAPI conformant BSP, comprising one of the following sub-classes:
  - OO BioAPI conformant Verification BSP
  - OO BioAPI conformant Identification BSP
  - OO BioAPI conformant Capture BSP
  - OO BioAPI conformant Verification Engine
  - OO BioAPI conformant Identification Engine

Conformance requirements for biometric applications, OO BioAPI Frameworks, and for BSPs are defined in A.2, A.3, and A.4, respectively.

NOTE Conformance of BFPs is not addressed in this document.

A.2 OO BioAPI Conformant Biometric Application

To claim compliance to the OO BioAPI specification, a biometric application shall, for each OO BioAPI function call utilized, invoke that operation consistently with this document. That is, all input parameters shall be present and valid. The application shall accept all valid output parameters and return values.

The biometric application shall conform to the call dependencies identified for the functions.

A.3 OO BioAPI Conformant Framework

The OO BioAPI Framework component serves the following general purposes:

a) BSP loading.

b) BSP and BFP management.

c) Component registry maintenance and management.

d) Handling of event notifications from BSPs, and sending those event notifications to (possibly multiple) event handlers in applications that have loaded that BSP.

e) Supporting API calls related to the installation or de-installation of OO BioAPI components, with appropriate update of the component registry.

f) Supporting queries from a BSP about installed BFPs.

To claim conformance to the OO BioAPI specification, an OO BioAPI Framework shall:

a) Provide component management functions as specified in **bioapi package** definition of following parts.

b) Provide component registry services in accordance with **ComponentRegistry interface definition**.

c) Conform to the data structures as defined in **data package defined in followings parts** and the error codes as defined in **BioAPIException class** when implementing a) through c), above.

d) Handle event notifications as defined in **EventHandler (bioapi package) and Event (data package)** and interfaces as defined in **GUI interface**.

A conformant OO BioAPI Framework is required to support ALL options identified in this document, since it will provide services to applications and BSPs that may implement any of those options.

A.4 OO BioAPI Conformant BSPs

A.4.1 General

To claim conformance to the OO BioAPI specification, BSPs shall implement mandatory functions for their conformance sub-class, as defined below. BSPs claim conformance to one of the conformance sub-classes specified in A.1.

BSPs shall accept all valid input parameters and return valid outputs. Optional capabilities and returns are not required to claim conformance; but any optional functions or parameters that are implemented shall be implemented in accordance with the specification requirements. Additional parameters shall not be required.

The BSP installation process shall perform the population of all required component registry entries.

BSPs shall possess a valid and unique UUID that is associated with a specific BSP product and version.

The UUID may be self-generated (see ISO/IEC 9834-8) and should (but need not) be the same on multiple systems where the same BSP product/version is installed.

BIRs generated by the BSP shall conform to the data structures **BIR interface of data package** (they shall be BioAPI BIRs). BSPs shall only return BIR object containing a registered FormatOwner with an associated valid FormatType (see **relevant interfaces in following parts**).

BSPs shall perform error handling as defined in **BioAPIException class.**

All BSPs shall support basic Component Management **bioapi package**), Utility (**BSPSchema and FramewoirkSchema interfaces**) and Event (**EventHandler interface**) operations. Callback (**GUI interfaces**), BioAPI Unit (**Unit Interface**) and Database (**BIRDatabase interface**) operations are optional.

The following table is a summary of BSP conformance requirements by subclass of BSP. Details are provided in the following sub-clauses. A.4.6 addresses conformance with respect to optional capabilities.

Table A.1 — BSP/BFP conformance sub-classes

Function	Verification BSP/BFP	Identification BSP/BFP	Capture BSP/BFP	Verification engine	Identification engine	Framework
Component Management Functions						
org.bioapi.Framework.loadBSP	X	X	X	X	X	
org.bioapi.ComponentRegistry.refresh	X	X	X	X	X	
org.bioapi.ComponentRegistry.install	X	X	X	X	X	

Table A.1 *(continued)*

Function	Verification BSP/BFP	Identification BSP/BFP	Capture BSP/BFP	Verification engine	Identification engine	Framework
org.bioapi.ComponentRegistry.uninstall	X	X	X	X	X	
org.bioapi.BSP.getUnits(Query<UnitSchema> query)	X	X	X			
org.bioapi.BSP.getBFPs(Query<BFPSchema> query)						
Callback and Event Functions						
org.bioapi.AttackSession.enableEvents	X	X	X	X	X	
org.bioapi.AttackSession.setGuiObservers						
Biometric Functions						
org.bioapi.Sensor.capture			X			
org.bioapi.Processing.createTemplate				X	X	
org.bioapi.Processing.process(BIR capturedBIR, BIR.Format ouputFormat)				X	X	
org.bioapi.Processing.process(BIR captureBIR, BIR auxiluaryBIR, BIR.Format outputFormat)						
org.bioapi.Matching.verify				X	X	
org.bioapi.Matching.identify					X	
org.bioapi.AttachSession.enroll	X	X				
org.bioapi.AttachSession.verify	X	X				
org.bioapi.AttachSession.identify		X				
Org.bioapi.AttachSession.importBIR						
Org.bioapi.Matching.presetIdentifyPopulation						
Database Functions						
org.bioapi.Archive.openDatabase						
org.bioapi.BIRDatabase.close				X	X	
org.bioapi.Archive.createDatabase				X	X	
org.bioapi.Archive.deleteDatabase						
org.bioapi.BIRDatabase.Market.terminate				X	X	
org.bioapi.BIRDatabase.storeBIR						
org.bioapi.BIRDatabase.getSingleBIR						
Org.bioapi.BIRDatabase.getBIRs						
Org.bioapi.BIRDatabase.deleteBIR						
BioAPI Unit Functions						
org.bioapi.Unit.getIndicatorStatus						
org.bioapi.Unit.setIndicatorStatus						
org.bioapi.Unit.setPowerMode						
org.bioapi.Sensor.calibrate						
Utility Functions						
org.bioapi.AttachSession.cancel	X	X	X	X	X	
org.bioapi.AttachSession.terminate, org.bioapi.BSP.terminate, org.bioapi.Framework.terminate	X	X	X	X	X	

A.4.2 OO BioAPI Conformant Verification BSPs

A.4.2.1 General

Verification BSPs are those which are capable of performing 1:1 matching (or authentication), but not 1:N identification matching.