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

Document: M1/07-0207
Date: March 05, 2007
Reply to: Shimon Modi
Phone: 765-409-6634
Email: shimon@purdue.edu

BioAPI Java, Revision 0

INCITS Project 1829-D

InterNational Committee for Information Technology Standards (INCITS)
INCITS Secretariat, Information Technology Industry Council (ITI)
1250 Eye St. NW, Suite 200, Washington, DC 20005
Telephone 202-737-8888; Fax 202-638-4922
email: incits@itic.org

Title: Biometric application programming interface – Java (BioAPI Java)

Source: INCITS M1

Date: March 05, 2007

Revision:

Revision	Date	M1 Document #	Comments
0	March 5, 2007	M1/07-0207	

Project Editor:
Shimon Modi
Purdue University
shimon@purdue.edu
765-409-6634

Contents

Foreword

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

In the field of Biometrics, INCITS has established the Technical Committee M1. Standards developed by this Technical Committee have reached consensus throughout the development process and have been thoroughly reviewed through several Public Review processes. In addition, the INCITS Executive Board and the ANSI Board of Standards Review have approved this American National Standard for Publication as an INCITS Standard.

Introduction

The existing versions of BioAPI, the ANSI version-INCITS 358 and ISO/IEC 19784-1, specify an application programming interface expressed in the C language. The use of this language ensures the wide applicability and implementability of BioAPI across multiple computing platforms and application domains. This API is an appropriate fit for applications written in the same language, and is adequate for applications written in C++.

Lack of portability of applications written in C is an issue that can be answered by a standard Java version of BioAPI. A C API does not work well with applications written in Java, C#, and other programming languages. In particular, the use of a C API from within a Java application is very unnatural, and requires certain programming artifices that introduce complications in the application, thus increasing the cost of application development and maintenance. If a standard Java version of BioAPI were available, the development of Java applications that use a standard biometric API would be easier and cheaper than it is today.

A standard Java version of BioAPI would also allow the development of Java BSPs that are intended for loading into a Java-based application server to perform verification and/or identification operations. In those application servers, a Java BioAPI framework and Java BSPs would work better than their C counterparts.

Another area in which a standard Java version of BioAPI would be useful is that of small computing devices based on Java, where (as on the large application servers mentioned above) a Java BioAPI framework and Java BSPs would fit better than their C counterparts.

This standard is expected to have the following impact:

- Enable creation of many new BioAPI applications by developers more comfortable with Java
- Create a market of Java components that are standard Java BioAPI BSPs, and that can be used (along with a Java BioAPI framework) in Java environments such as Java-based application servers, Java applets, or small Java-based devices; and
- Increase the level of adoption of BioAPI by enabling Java application developers to access C BSPs (as if they were Java BSPs) through special versions of the BioAPI framework that will bridge a standard Java BioAPI framework to a standard C BioAPI framework.

1 Scope

The proposed standard will specify an interface of a Java BioAPI framework and Java BioAPI BSP which will mirror the corresponding components specified in ISO/IEC 19784-1. Therefore, the position occupied by the proposed standard within the general picture of biometrics standards will be the same position that ISO/IEC 19784-1 occupies, the only difference being the programming language of the interfaces. The concepts such as BioAPI unit, component registry, etc. are present in this standard and will have the same meaning as in ISO/IEC 19784-1. The semantic equivalence of this standard will be maintained with ISO/IEC 19784-1, but there are differences in actual parameters passed between functions and the sequence of function calls. These differences exist to take advantage of the object oriented benefits of Java.

2 Conformance

Annex A specifies the conformance requirements for systems/components claiming conformance to this standard.

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.

- ISO/IEC 19784-1:2006, Information Technology – Biometric Application Programming Interface – Part 1: BioAPI Specification
- ANSI/INCITS 398-2005, American National Standard – Information technology – Common Biometric Exchange Formats Framework (CBEFF)
- ISO/IEC 19785-1:2006, Information Technology – Common Biometric Exchange Formats Framework – Part 1: Data Element Specification
- ISO/IEC 19785-2:2006, Information Technology – Common Biometric Exchange Formats Framework – Part 2: Procedures for the Operation of the Biometric Registration Authority

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. These terms are defined in ISO/IEC 19784-1.

4.1
adaptation
template adaptation

use of a BIR produced from a newly captured and verified biometric sample to automatically update or refresh a reference template

NOTE: This procedure is used to minimize the effects of template aging.

4.2
attach session

temporary association between an application, a single BSP, and a set of BioAPI Units that are managed either directly or indirectly by that BSP

4.3
BioAPI component

component of the BioAPI architecture with a defined interface that can be supplied by a separate vendor and which is subject to conformance testing

NOTE: BioAPI components include BioAPI applications, the BioAPI Framework, BSPs, and BFPs.

4.4
BioAPI Function Provider - BFP
component that manages one or more BioAPI Units of a specific category

NOTE 1: Interfaces to BioAPI Function Providers are standardized in subsequent parts of ISO/IEC 19784.

NOTE 2: BFPs are categorized according to the categories of BioAPI Units that they manage (see clause 6.2.4).

4.5
BioAPI Unit

abstraction of a hardware or software resource that is directly managed by a BSP or BFP

NOTE: BioAPI Units are categorized (see clause 6.2.2) and include sensor units, archive units, matching-algorithm units and processing-algorithm units.

4.6
biometric (used only as an adjective)

pertaining to the field of biometrics

4.7
Biometric Data Block - BDB

block of data with a defined format that contains one or more biometric samples or biometric templates

NOTE 1: ISO/IEC 19784 does not support BDB formats that are not an integral multiple of eight bits.

NOTE 2: There is no requirement that a BDB format be self-delimiting.

NOTE 3: Each part of ISO/IEC 19794 standardises one or more BDB formats. Vendor specific formats can also be specified and identified.

NOTE 4: Within BioAPI, the BDB is “opaque” to the application and is therefore sometimes referred to as an opaque biometric data block.

4.8
Biometric Information Record - BIR

data structure containing one or more BDBs, together with information identifying the BDB formats, and possibly further information such as whether a BDB is signed or encrypted (see ISO/IEC 19785-1)

NOTE: This part of ISO/IEC 19784 defines a BIR format (see clause 7.4) that supports only a single BDB. ISO/IEC 19785-1 defines a more general BIR format that supports multiple BDBs within the BIR, and the above definition is used in common by the two International Standards. When the term BIR is used in this part of ISO/IEC 19784, it normally refers to the specific BIR format defined by BioAPI (see Annex B), not to an arbitrary BIR. The term BioAPI BIR is used where clarity is needed.

4.8.1
reference BIR

BIR whose BDB(s) contain one or more biometric templates

4.8.2
sample BIR

BIR whose BDB(s) contain only biometric samples that are not templates

4.9
biometric sample

information obtained from a biometric sensor, either directly or after further processing

NOTE: See also raw biometric sample, intermediate biometric sample, and processed biometric sample.

4.9.1
biometric template

biometric sample or combination of biometric samples that is suitable for storage as a reference for future comparison

4.9.2
intermediate biometric sample

biometric sample obtained by processing a raw biometric sample, intended for further processing

4.9.3
processed biometric sample

biometric sample suitable for comparison

4.9.4
raw biometric sample

biometric sample obtained directly from a biometric sensor

NOTE: The formats for raw biometric samples are not currently standardised, and depend on the nature of the biometric device and the vendor of that device. They may in the future be standardised as part of the standardisation of specific biometric devices.

4.9.5
reference template

biometric template that has been stored

4.10
biometric sensor

biometric hardware used to capture raw biometric samples from a subject

NOTE: The term ‘biometric device’ is used interchangeably with this term.

4.11
Biometric Service Provider
BSP

component that provides biometric services to an application through a defined interface by managing one or more BioAPI Units directly, or through interfaces to BioAPI Function Providers

4.12
biometrics (noun)

automated recognition of individuals based on their behavioural and biological characteristics

4.13
callback

mechanism by which a component that exposes an API invokes a function within a component that uses that API, where the address of that function has been previously passed as an input parameter of an API function call

NOTE: This mechanism enables a BioAPI component to communicate with another BioAPI component other than by invoking an API function, usually in response to an event or interrupt.

4.14
component registry

information maintained by the BioAPI Framework concerning the BioAPI components that are available on a biometric system

4.15
encrypt
encryption

(reversible) transformation of data by a cryptographic algorithm to produce ciphertext; that is, to hide the information content (protect the confidentiality) of the data

NOTE 1: Encryption algorithms consist of two processes: encryption (or encipherment) which transforms plaintext into ciphertext, and decryption (or decipherment) which transforms ciphertext to plaintext.

NOTE 2: Encryption may be used for either security or privacy reasons.

4.16
enrollment

process of collecting one or more biometric samples from an individual, and the subsequent construction of a biometric reference template which can then be used to verify or determine the individual's identity

NOTE: The reference template would normally be stored by a biometric application, a BSP supporting an archive BioAPI Unit, or both.

4.17
False Match Rate
FMR

measure of the probability that a biometric matching process will incorrectly identify an individual or will fail to reject an impostor

NOTE 1: Within BioAPI, FMR is used as a means of specifying scores and thresholds (see clause C.4).

NOTE 2: Historically, False Acceptance Rate (FAR) has also been used with a similar definition, but FMR is the preferred term in International Standards. Similarly for False Rejection Rate (FRR), as opposed to the preferred False Non-Match Rate (FNMR).

4.18
handle

parameter returned by a BioAPI function (A say) that can be used by the BioAPI application in a subsequent function call to identify a BioAPI component or data element within the component A

NOTE: Types of handles include:

BIR_Handle, generated by a BSP to select or access a BIR within that BSP.

BSP Attach Session Handle, for an attach session.

DB_Handle, generated by a BSP to select or access a BIR database controlled by that BSP.

4.19
identify
identification

one-to-many process of comparing a submitted biometric sample against a reference population to determine whether the submitted biometric sample matches any of the reference templates in that reference population in order to determine the identity of the enrollee whose template was matched

NOTE: This is often called an "identification match" or "identifymatch".

4.20
match
matching

one-to-one process of comparing a submitted biometric sample against a single biometric reference template and scoring the level of similarity.

NOTE 1: An accept or reject decision would then normally be based upon whether this score exceeds a given threshold.

NOTE 2: Matching algorithms and their effect on False Match Rate and False Non-Match Rate scores are currently not standardised.

NOTE 3: See also identify (4.21) and verify (4.29).

4.21
payload

data, provided at the time of enrolment and associated with a reference template, which can be released upon a successful biometric verification.

NOTE: Examples of payloads include user names, accounts, passwords, cryptographic keys, or digital certificates (see clause C.5).

4.22
score
scoring

value indicating the degree of similarity or correlation between a biometric sample and a biometric reference template

4.23
security block

block of data with a defined format that contains security information (for example, related to encryption or integrity) related to a BIR (see ISO/IEC 19785-1)

4.24
self-contained device

combination device which includes a biometric sensor and all or part of the BSP functionality

NOTE: A self-contained device may include the ability to not only capture a biometric, but also to process, match, and/or store it. This functionality is typically implemented in hardware/firmware.

4.25
signature
digital signature

data appended to, or a cryptographic transformation of, a data unit that allows the recipient of the data unit to prove the origin and integrity of the data unit and protect against forgery, e.g. by the recipient

NOTE: Digital signatures may be used for purposes of authentication, data integrity, and non-repudiation

4.26
threshold

predefined value which establishes the degree of similarity or correlation (that is, a score) necessary for a biometric sample to be deemed a match with a biometric reference template

4.27
universally unique identifier
UUID
A 128-bit value generated in accordance with ISO/IEC 9834-8 and providing unique values between systems and over time

4.28
verify
verification

one-to-one process of comparing a single submitted biometric sample against a biometric reference template to determine whether the submitted biometric sample matches the reference template

NOTE: This is often called a "verification match" or "verifymatch".

5 Symbols and Abbreviated Terms

API – Application Programming Interface
BDB – Biometric Data Block
BFP – BioAPI Function Provider
BIR – Biometric Information Record
BSP – Biometric Service Provider
CBEFF – Common Biometric Exchange Formats Framework
FMR – False Match Rate
FPI – Function Provider Interface
GUI – Graphical User Interface
ID – Identity/Identification/Identifier
MOC – Match on Card
PID – Product ID
SB - Security Block
NOTE: This term and abbreviation is imported from ISO/IEC 19785-1.
SBH – Standard Biometric Header
NOTE: This term and abbreviation is imported from ISO/IEC 19785-1.
SPI – Service Provider Interface
UUID – Universally Unique Identifier

6 BioAPI Java Package Structure

The BioAPI Java interface will be divided into several packages. The following is the package structure:

- package org.bioapi
- package org.bioapi.data

Annex D contains a table of current implementation status for each interface and class.

6.1 package org.bioapi

This package has the following interfaces and classes:

6.1.1 Interface org.bioapi.Archive

6.1.2 Enum org.bioapi.AttachSession.EnrollResult

6.1.3 Interface org.bioapi.AttachSession.EnrollResult.Options

6.1.4 Interface org.bioapi.AttachSession.VerifyResult

6.1.5 Enum org.bioapi.AttachSession.VerifyResult.Options

6.1.6 Interface org.bioapi.AttachSession.IdentifyResult

6.1.7 Enum org.bioapi.AttachSession.IdentifyResult.Options

6.1.8 Class org.bioapi.BioAPIException

6.1.9 Enum org.bioapi.BioAPIException.Facility

6.1.10 Interface org.bioapi.BIRDatabase

6.1.11 Enum org.bioapi.BIRDatabase.Access

6.1.12 Interface org.bioapi.BIRDatabase.Record

6.1.13 Interface org.bioapi.BIRDatabase.Marker

6.1.14 Interface org.bioapi.BSP

6.1.15 Interface org.bioapi.BSP.UnitSet

6.1.16 Interface org.bioapi.ComponentRegistry

6.1.17 Enum org.bioapi.ComponentRegistry.Action

6.1.18 Interface org.bioapi.Unit

6.1.19 Enum org.bioapi.Unit.Category

6.1.20 Enum org.bioapi.Unit.IndicatorStatus

6.1.21 Enum org.bioapi.Unit.PowerMode

6.1.22 Interface org.bioapi.EventHandler

6.1.23 Interface org.bioapi.Framework

6.1.24 Class org.bioapi.FrameworkFactory

6.1.25 Interface org.bioapi.GUIImageObserver

6.1.26 Interface org.bioapi.GUIImageObserver.Bitmap

6.1.27 Interface org.bioapi.GUIStateObserver

6.1.28 Enum org.bioapi.GUIStateObserver.Response

6.1.29 Interface org.bioapi.GUIStateObserver.State

6.1.30 Interface org.bioapi.Matching

6.1.31 Interface org.bioapi.Matching.IdentifyResult

6.1.32 Interface org.bioapi.Matching.VerifyResult

6.1.33 Enum org.bioapi.Matching.VerifyResult.Options

6.1.34 Interface org.bioapi.Processing

6.1.35 Interface org.bioapi.Processing.ProcessResult

6.1.36 Interface org.bioapi.Processing.CreateTemplateResult

6.1.37 Interface org.bioapi.Query

6.1.38 Interface org.bioapi.Sensor

6.1.39 Interface org.bioapi.Sensor.CaptureResult

6.1.40 Enum org.bioapi.Sensor.CaptureResult.Options

6.2 Package org.bioapi.data

This package has the following interfaces and classes:

6.2.1 Interface org.bioapi.data.BFPSchema

6.2.2 Interface org.bioapi.data.BIR

6.2.3 Enum org.bioapi.data.BIR.Purpose

6.2.4 Interface org.bioapi.data.BIR.Subtype

6.2.5 Enum org.bioapi.data.BIR.Subtype.Instance

6.2.6 Enum org.bioapi.data.BIR.ProcessedLevel

6.2.7 Interface org.bioapi.data.BIR.BiometricType

6.2.8 Enum org.bioapi.data.BIR.BiometricType.Type

6.2.9 Interface org.bioapi.data.BIR.OwnerTypePair

6.2.10 Interface org.bioapi.data.BIR.Format

6.2.11 Interface org.bioapi.data.BIR.SecurityBlockFormat

6.2.12 Interface org.bioapi.data.BIR.ProductID

6.2.13 Interface org.bioapi.data.BIR.Quality

6.2.14 Interface org.bioapi.data.BIR.DTG

6.2.15 Interface org.bioapi.data.BSPSchema

6.2.16 Interface org.bioapi.data.BSPSchema.Operations

6.2.17 Enum org.bioapi.data.BSPSchema.Operations.Operation

6.2.18 Interface org.bioapi.data.BSPSchema.Options

6.2.19 Enum org.bioapi.data.BSPSchema.Options.Option

6.2.20 Interface org.bioapi.data.Candidates

6.2.21 Interface org.bioapi.data.Candidates.Candidate

6.2.22 Interface org.bioapi.data.DataFactory

6.2.23 Interface org.bioapi.data.Date

6.2.24 Interface org.bioapi.data.Event

6.2.25 Enum org.bioapi.data.Event.Kind

6.2.26 Interface org.bioapi.data.FMR

6.2.27 Interface org.bioapi.data.FrameworkSchema

6.2.28 Interface org.bioapi.data.IdentifyPopulation

6.2.29 Interface org.bioapi.data.Payload

6.2.30 Interface org.bioapi.data.Time

6.2.31 Interface org.bioapi.data.UnitSchema

6.3 Package org.bioapi.net

This package has the following interface:

6.3.1 Class org.bioapi.net.IRI

Annex A
(normative)

Conformance

Annex B
(normative)

CBEFF Patron Format Specification: BioAPI Patron Format

Annex C
(informative)

Calling Sequence Examples and Sample Code

Annex D
(informative)

Current Code Status

This is an informative annex with current implementation status for each interface and class in the API. This annex will be removed upon completion of this project.

Interface/Class	Implementation Status
Interface org.bioapi.Archive	Implemented but untested.
Enum org.bioapi.AttachSession.EnrollResult	Implemented but untested.
nterface org.bioapi.AttachSession.EnrollResult.Options	Implemented but untested.
Interface org.bioapi.AttachSession.VerifyResult	Implemented but untested.
Enum org.bioapi.AttachSession.VerifyResult.Options	Implemented but untested.
Interface org.bioapi.AttachSession.IdentifyResult	Implemented but untested.
Enum org.bioapi.AttachSession.IdentifyResult.Options	Implemented but untested.
Class org.bioapi.BioAPIException	Implemented but untested.
Enum org.bioapi.BioAPIException.Facility	Implemented but untested.
Interface org.bioapi.BIRDatabase	Implemented but untested.
Enum org.bioapi.BIRDatabase.Access	Implemented but untested.
Interface org.bioapi.BIRDatabase.Record	Implemented but untested.
Interface org.bioapi.BIRDatabase.Marker	Implemented but untested.
Interface org.bioapi.BSP	Implemented but untested.
Interface org.bioapi.BSP.UnitSet	Implemented but untested.
Interface org.bioapi.ComponentRegistry	Implemented but untested.

Interface org.bioapi.Unit	Implemented but untested.
Enum org.bioapi.Unit.Category	Implemented but untested.
Enum org.bioapi.Unit.IndicatorStatus	Implemented but untested.
Enum org.bioapi.Unit.PowerMode	Implemented but untested.
Interface org.bioapi.EventHandler	Unimplemented.
Interface org.bioapi.Framework	Unimplemented.
Class org.bioapi.FrameworkFactory	Unimplemented.
Class org.bioapi.FrameworkFactory	Unimplemented.
Interface org.bioapi.GUIImageObserver	Unimplemented.
Interface org.bioapi.GUIStateObserver	Unimplemented.
Enum org.bioapi.GUIStateObserver.Response	Unimplemented.
Interface org.bioapi.GUIStateObserver.State	Unimplemented.
Interface org.bioapi.Matching	Unimplemented.
Interface org.bioapi.Matching.IdentifyResult	Unimplemented.
Interface org.bioapi.Matching.VerifyResult	Unimplemented.
Enum org.bioapi.Matching.VerifyResult.Options	Unimplemented.
Interface org.bioapi.Processing	Unimplemented.
Interface org.bioapi.Processing.ProcessResult	Unimplemented.
Interface org.bioapi.Processing.CreateTemplateResult	Unimplemented.
Interface org.bioapi.Query	Unimplemented.
Interface org.bioapi.Sensor	Unimplemented.
Interface org.bioapi.Sensor.CaptureResult	Unimplemented.
Enum org.bioapi.Sensor.CaptureResult.Options	Unimplemented.

Interface org.bioapi.data.BFPSchema	Unimplemented.
Interface org.bioapi.data.BIR	Unimplemented.
Enum org.bioapi.data.BIR.Purpose	Unimplemented.
Interface org.bioapi.data.BIR.Subtype	Unimplemented.
Enum org.bioapi.data.BIR.Subtype.Instance	Unimplemented.
Enum org.bioapi.data.BIR.ProcessedLevel	Unimplemented.
Interface org.bioapi.data.BIR.BiometricType	Unimplemented.
Enum org.bioapi.data.BIR.BiometricType.Type	Unimplemented.
Interface org.bioapi.data.BIR.OwnerTypePair	Unimplemented.
Interface org.bioapi.data.BIR.Format	Unimplemented.
Interface org.bioapi.data.BIR.SecurityBlockFormat	Unimplemented.
Interface org.bioapi.data.BIR.ProductID	Unimplemented.
Interface org.bioapi.data.BIR.Quality	Unimplemented.
Interface org.bioapi.data.BIR.DTG	Unimplemented.
Interface org.bioapi.data.BSPSchema	Unimplemented.
Interface org.bioapi.data.BSPSchema.Operations	Unimplemented.
Enum org.bioapi.data.BSPSchema.Operations.Operation	Unimplemented.
Interface org.bioapi.data.BSPSchema.Options	Unimplemented.
Enum org.bioapi.data.BSPSchema.Options.Option	Unimplemented.
Interface org.bioapi.data.Candidates	Unimplemented.
Interface org.bioapi.data.Candidates.Candidate	Unimplemented.
Interface org.bioapi.data.DataFactory	Unimplemented.
Interface org.bioapi.data.Date	Unimplemented.

Interface org.bioapi.data.Event	Unimplemented.
Interface org.bioapi.data.Date	Unimplemented.
Enum org.bioapi.data.Event.Kind	Unimplemented.
Interface org.bioapi.data.FMR	Unimplemented.
Interface org.bioapi.data.FrameworkSchema	Unimplemented.
Interface org.bioapi.data.IdentifyPopulation	Unimplemented.
Interface org.bioapi.data.Payload	Unimplemented.
Interface org.bioapi.data.Time	Unimplemented.
Interface org.bioapi.data.UnitSchema	Unimplemented.
Class org.bioapi.net.IRI	Unimplemented.