DRAFT INTERNATIONAL STANDARD

ISO/IEC DIS 30106-3

ISO/IEC JTC **1**/SC **37**

Secretariat: **ANSI**

Voting begins on:
2014-12-04

Voting terminates on:
2015-03-04

Information Technology — Object oriented BioAPI —

Part 3:
C# implementation

Titre manque

ICS: 35.040

Reference number
ISO/IEC DIS 30106-3:2014(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member 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 shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 30106-3 was prepared by Technical Committee ISO/TC JTC1, *Information Technology*, Subcommittee SC 37, *Biometrics*.

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

ISO/IEC 30106 consists of the following parts, under the general title *Information Technology — Object Oriented BioAPI*:

— *Part 1: Architecture*

— *Part 2: Java Implementation*

— *Part 3: C# Implementation*

Introduction

In this part of the standard an application programming interface expressed in C# language is specified. C# is intended to be a simple, general-purpose, object oriented programming language that is aimed at enabling programmers to quickly build a wide range of applications for the Microsoft .NET platform.

One of the advantages of using C# is that, as it is designed for the CLI (Common Language Infrastructure), allows multiple high-level languages to be used on different computer platforms without being rewritten for specific architectures.

C# shares some features (overloading, some syntactic details, etc.) with C++ but includes new characteristics (reference and output parameters, enumerations, unified type system, etc). Besides, C# is very similar to Java (Interfaces, Exceptions, object-orientation, etc.), which implies that the structure of interfaces and namespaces ((which is the equivalent to packages in Java language)) is mostly the same as Java but, as expected, code implementation and compilation are different.

As Java implementation allows an easy use of Java BSPs, Java-based application servers or Java applets, C# is the best way to write windows desktop and web applications/services and provides an advanced and well designed remote framework.

Information Technology — Object Oriented BioAPI — Part 3: C# Implementation

1 Scope

The proposed standard will specify an interface of a BioAPI C# framework and BioAPI C# BSP which will mirror the corresponding components specified in ISO/IEC 30106-1.The semantic equivalence of this standard will be maintained with ISO/IEC 30106-2 (Java implementation). In spite of the differences in actual parameters passed between functions, the names and interface structure are the same.

2 BioAPI C# Namespace Structure

The BioAPI C# interface will be divided into several namespaces. The following is the namespace structure:

- Namespace BioAPI: Contains functions for the attach sessions and managing units.
- Namespace BioAPI.Data: Contains all the data structures.

2.1 Namespace BioAPI

2.1.1 Namespace description

This namespace contains all the components responsible for managing and executing the functionality of BioAPI, either in a Framework-free deployment or when an OO BioAPI Framework is used. Component Registry interface is also defined in this namespace.

2.1.2 Structure

The description of this namespace is given explaining a bottom-up structure. In clause 4, the interfaces needed to be implemented for each of the Unit types are explained. It is important to note that such interfaces do not refer to an implemented class by itself, as they accessible class will be either the Biometric Service Provider (BSP) or the Biometric Function Provider (BFP), but the specifications in such clause are common to the methods and properties to be added to the implemented BSP and/or BFP classes.

This will be followed by the specification of the implementation of the BFP (clause 5) and BSP (clause 6) interfaces. These two interfaces provide the lower layer interoperability level, equivalent to the SPI and BFPI interfaces in ISO/IEC 19784-1.

The higher layer of interoperability level is provided by the specification of the Framework (clause 7, with the Framework Interface and the Component Registry) and the Application interaction (clause 8, with the specification of the Exceptions and Callback functions). This provide the equivalence to the API interface in ISO/IEC 19784-1.

2.2 Namespace BioAPI.Data

2.2.1 Namespace description

This namespace contains all data structures needed for the implementation of a either a Framework-based version of OO BioAPI, or a Framework-free one.

2.2.2 Structure

Several data structures are provided to comply with the requirements of specifying this Intenational Standard. All the BioAPI.Data namespace is specified in clause 3, where all needed classes and enumerations are defined. This has to be complemented to the constants defined in Part 1 of this International Standard.

3 Data types and constants

3.1 Class ACBioParameters

3.1.1 Description

Structure that provides the information which is used to generate ACBio instances.

3.1.2 Properties Summary

— int[] Challenge {get;} – Challenge from the validator of a biometric verification when ACBio is used. This value shall be sent to the field controlValue of type ACBioContentInformation in ACBio instances.

— int[] InitialBPUIOIndexOutput {get;} – The initial value of BPU IO index which is to be assigned to the output from the BioAPI Unit , BFP, or BSP when the ACBio instances are generated. The range between InitialBPUIOIndexOutput and SupremumBPUIOIndexOutput shall be divided into the number of BSP Units and BFPs which are attached to the BSP, and assigned to the BSP Units and BSPs.

— int[] SupremumBPUIOIndexOutput {get;} – The supremum of BPU IO indexes which are to be assigned to the output from the BioAPI Unit , BFP, or BSP when the ACBio instances are generated.

3.2 Class BFPListElement

3.2.1 Description

Identifies a BFP by category and UUID. A list is returned by a BSP when queried for the installed BFPs that it supports.

3.2.2 Properties Summary

— UnitCategoryType UnitCategory { get; set; }: The category of the unit

— UUID BFPID { get; set; }: The UUID assigned to the BFP

3.3 Class BFPSchema [Serializable()]

3.3.1 Description

Represents the record in the component registry that defines the properties of the BFP installed in the system. Is a serializable class.

3.3.2 Properties Summary

— UUID BFPUUID {get;}: UUID of the BFP

— UnitCategoryType BFPCategory {get;}: Category of the BFP identified by the BFPUUID

— String BFPDescription {get;}: A NULL-terminated string containing a text description of the BFP