
**Information technology — Object
oriented BioAPI —**

Part 3:
C# implementation

*Technologies de l'information — Objet orienté BioAPI —
Partie 3: Mise en oeuvre de C#*

This document is a preview generated by EMS



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Contents

	Page
Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 BioAPI C# namespace structure	1
3.1 Overall structure.....	1
3.2 Namespace BioAPI.....	1
3.2.1 Namespace description.....	1
3.2.2 Structure.....	1
3.3 Namespace BioAPI.Data.....	2
3.3.1 Namespace description.....	2
3.3.2 Structure.....	2
4 Data types and constants	2
4.1 Class ACBioParameters.....	2
4.1.1 Description.....	2
4.1.2 Properties summary.....	2
4.2 Class BFPListElement.....	2
4.2.1 Description.....	2
4.2.2 Properties summary.....	2
4.3 Class BFPSchema [Serializable()].....	3
4.3.1 Description.....	3
4.3.2 Properties summary.....	3
4.3.3 Method summary.....	3
4.4 Class BIR.....	3
4.4.1 Description.....	3
4.4.2 Properties summary.....	4
4.4.3 Method summary.....	5
4.5 Class BSPSchema [Serializable()].....	5
4.5.1 Description.....	5
4.5.2 Properties summary.....	6
4.5.3 Method summary.....	7
4.6 Class Candidate.....	7
4.6.1 Description.....	7
4.6.2 Properties summary.....	7
4.7 Class DataTypes.....	7
4.7.1 Description.....	7
4.7.2 Enumerations.....	8
4.8 Class date.....	14
4.8.1 Description.....	14
4.8.2 Properties summary.....	15
4.8.3 Methods summary.....	15
4.9 Class FrameworkSchema.....	15
4.9.1 Description.....	15
4.9.2 Properties summary.....	15
4.9.3 Method summary.....	16
4.10 Class GUIBitmap.....	16
4.10.1 Description.....	16
4.10.2 Properties.....	16
4.10.3 Method summary.....	16
4.11 Class Identifypopulation.....	17
4.11.1 Description.....	17
4.11.2 Properties summary.....	17
4.11.3 Method summary.....	17

4.12	Class PopulationMember	17
	4.12.1 Description	17
	4.12.2 Properties summary	17
4.13	Class RegistryID	18
	4.13.1 Description	18
	4.13.2 Properties summary	18
4.14	Class SecurityProfileType	18
	4.14.1 Description	18
	4.14.2 Properties summary	18
	4.14.3 Method summary	18
4.15	Class UnitList	19
	4.15.1 Description	19
	4.15.2 Properties summary	19
	4.15.3 Methods summary	19
4.16	Class UnitListElement	19
	4.16.1 Description	19
	4.16.2 Properties summary	19
4.17	Class UnitSchema	19
	4.17.1 Description	19
	4.17.2 Properties summary	20
	4.17.3 Method summary	20
4.18	Class UUID [Serializable()]	20
	4.18.1 Description	20
	4.18.2 Properties	21
5	Object-oriented interfaces for supporting BioAPI_Units	21
5.1	General	21
5.2	Interface IArchive	21
	5.2.1 Description	21
	5.2.2 Method summary	22
5.3	Interface IComparison	24
	5.3.1 Description	24
	5.3.2 Method summary	25
5.4	Interface IProcessing	27
	5.4.1 Description	27
	5.4.2 Method summary	28
5.5	Interface ISensor	29
	5.5.1 Description	29
	5.5.2 Method summary	29
6	BFP level	30
6.1	Interface IBFP	30
	6.1.1 Description	30
	6.1.2 Imported interfaces	30
	6.1.3 Properties summary	30
	6.1.4 Events summary	30
	6.1.5 Method summary	31
7	BSP level	33
7.1	Interface IBSP	33
	7.1.1 Description	33
	7.1.2 Imported interfaces	33
	7.1.3 Properties summary	33
	7.1.4 Events summary	33
	7.1.5 Method summary	33
8	Framework level	40
8.1	Interface IComponentRegistry	40
	8.1.1 Description	40
	8.1.2 Method summary	41

8.2	Interface IFramework.....	42
8.2.1	Description.....	42
8.2.2	Inherited interfaces.....	42
8.2.3	Properties summary.....	42
8.2.4	Method summary.....	43
9	Application interaction.....	47
9.1	Class BioAPIException: Exception.....	47
9.1.1	Description.....	47
9.1.2	Constructor summary.....	48
9.1.3	Properties summary.....	48
9.1.4	Method summary.....	49
9.2	Callback functions.....	49
9.2.1	Description.....	49
9.2.2	Callback functions specification.....	50
Annex A (informative) Calling sequence examples and sample code.....		55
Bibliography.....		56

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.

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).

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword - Supplementary information](#)

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 37, *Biometrics*.

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 ISO/IEC 30106, 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 Common Language Infrastructure (CLI), it 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

This part of ISO/IEC 30106 specifies 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 part of ISO/IEC 30106 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 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 30106-1, *Information technology — BioAPI for object oriented programming languages — Part 1: Architecture*

3 BioAPI C# namespace structure

3.1 Overall structure

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

- namespace BioAPI: contains functionality to manage units, BSPs, BFPs, the framework and applications;
- namespace BioAPI.Data: contains all the data structures.

3.2 Namespace BioAPI

3.2.1 Namespace description

This namespace contains all the components responsible for managing and executing the functionality of BioAPI. Component registry interface is also defined in this namespace.

3.2.2 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 the 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.