

---

---

**Information technology — Object  
oriented BioAPI —**

**Part 3:  
C# implementation**

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





**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, 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  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier; Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

	Page
<b>Foreword</b> .....	<b>vi</b>
<b>Introduction</b> .....	<b>vii</b>
<b>1 Scope</b> .....	<b>1</b>
<b>2 Normative references</b> .....	<b>1</b>
<b>3 Terms and definitions</b> .....	<b>1</b>
<b>4 BioAPI C# Namespace Structure</b> .....	<b>1</b>
4.1 Overall structure.....	1
4.2 Namespace BioAPI.....	1
4.2.1 Namespace description.....	1
4.2.2 Structure.....	2
4.3 Namespace BioAPI.Data.....	2
4.3.1 Namespace description.....	2
4.3.2 Structure.....	2
<b>5 Data types and constants</b> .....	<b>2</b>
5.1 Class ACBioParameters.....	2
5.1.1 Description.....	2
5.1.2 Properties summary.....	2
5.2 Class BFPListElement.....	3
5.2.1 Description.....	3
5.2.2 Properties summary.....	3
5.3 Class BFPSchema [Serializable()].....	3
5.3.1 Description.....	3
5.3.2 Properties summary.....	3
5.3.3 Method summary.....	4
5.4 Class BIR.....	4
5.4.1 Description.....	4
5.4.2 Properties summary.....	4
5.4.3 Method summary.....	5
5.5 Class BSPSchema [Serializable()].....	6
5.5.1 Description.....	6
5.5.2 Properties Summary.....	6
5.5.3 Method summary.....	7
5.6 Class Candidate.....	7
5.6.1 Description.....	7
5.6.2 Properties summary.....	8
5.7 Class DataTypes.....	8
5.7.1 Description.....	8
5.7.2 Enumerations.....	8
5.8 Class Date.....	15
5.8.1 Description.....	15
5.8.2 Properties summary.....	15
5.8.3 Methods summary.....	16
5.9 Class FrameworkSchema.....	16
5.9.1 Description.....	16
5.9.2 Properties summary.....	17
5.9.3 Method summary.....	17
5.10 Class GUIBitmap.....	17
5.10.1 Description.....	17
5.10.2 Properties.....	17
5.10.3 Method summary.....	17
5.11 Class Identifypopulation.....	18
5.11.1 Description.....	18

5.11.2	Properties summary	18
5.11.3	Method summary	18
5.12	Class PopulationMember	18
5.12.1	Description	18
5.12.2	Properties summary	18
5.13	Class RegistryID	19
5.13.1	Description	19
5.13.2	Properties summary	19
5.14	Class SecurityProfileType	19
5.14.1	Description	19
5.14.2	Properties summary	19
5.14.3	Method summary	19
5.15	Class UnitList	20
5.15.1	Description	20
5.15.2	Properties summary	20
5.15.3	Methods summary	20
5.16	Class UnitListElement	20
5.16.1	Description	20
5.16.2	Properties summary	20
5.17	Class UnitSchema	21
5.17.1	Description	21
5.17.2	Properties summary	21
5.17.3	Method summary	21
5.18	Class UUID [Serializable()]	22
5.18.1	Description	22
5.18.2	Properties	22
<b>6</b>	<b>Object oriented interfaces for supporting BioAPI_Units</b>	<b>22</b>
6.1	General	22
6.2	Interface IArchive	22
6.2.1	Description	22
6.2.2	Method summary	22
6.3	Interface IComparison	26
6.3.1	Description	26
6.3.2	Method summary	27
6.4	Interface IProcessing	29
6.4.1	Description	29
6.4.2	Method summary	30
6.5	Interface ISensor	31
6.5.1	Description	31
6.5.2	Method summary	32
<b>7</b>	<b>BFP level</b>	<b>33</b>
7.1	Interface IBFP	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
<b>8</b>	<b>BSP level</b>	<b>36</b>
8.1	Interface IBSP	36
8.1.1	Description	36
8.1.2	Imported interfaces	36
8.1.3	Properties summary	36
8.1.4	Events summary	36
8.1.5	Method summary	36
<b>9</b>	<b>Framework level</b>	<b>43</b>
9.1	Interface IComponentRegistry	43

9.1.1	Description.....	43
9.1.2	Method summary .....	43
9.2	Interface IFramework.....	44
9.2.1	Description.....	44
9.2.2	Inherited interfaces.....	45
9.2.3	Properties summary.....	45
9.2.4	Method summary .....	45
<b>10</b>	<b>Application interaction.....</b>	<b>49</b>
10.1	class BioAPIException : Exception.....	49
10.1.1	Description.....	49
10.1.2	Constructor summary .....	49
10.1.3	Properties summary.....	50
10.1.4	Method summary .....	50
10.2	Callback functions.....	51
10.2.1	Description.....	51
10.2.2	Callback functions specification.....	51
<b>Annex A (informative) Calling sequence examples and sample code .....</b>		<b>57</b>
<b>Bibliography.....</b>		<b>58</b>

## 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](http://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](http://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](http://www.iso.org/iso/foreword.html).

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

This second edition cancels and replaces the first edition (ISO/IEC 30106-3:2016), which has been technically revised.

The main changes compared to the previous edition are as follows:

- correction of typing errors;
- addition of AnalyseQuality method.

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](http://www.iso.org/members.html).

## Introduction

This document specifies an application programming interface expressed in C# language. 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), 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) with C++ but also includes new characteristics (reference and output parameters, enumerations, unified type system). Furthermore, C# is very similar to Java (interfaces, exceptions, object-orientation), 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 document specifies an interface of a BioAPI C# framework and BioAPI C# BSP which mirror the corresponding components specified in ISO/IEC 30106-1. The semantic equivalence of this document 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 are referred to in the text in such a way that some or all of their content constitutes requirements 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 10646:2017, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 30106-1, *Information technology — Object oriented BioAPI — Part 1: Architecture*

### 3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

### 4 BioAPI C# Namespace Structure

#### 4.1 Overall structure

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

- Namespace BioAPI: Contains functionality to manage units, BSPs, BFPs, the Framework and Applications.
- Namespace BioAPI.Data: Contains all the data structures.

#### 4.2 Namespace BioAPI

##### 4.2.1 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.