
**Information technology — Conformance
testing for the biometric application
programming interface (BioAPI) —
Part 3:
Test assertions for BioAPI frameworks**

iTeh STANDARD PREVIEW
*Technologies de l'information — Essai de conformité pour l'interface de
programmation d'applications biométriques (BioAPI) —
Partie 3: Déclarations d'essai pour cadres BioAPI*
(standards.iteh.ai)

[ISO/IEC 24709-3:2011](https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011)

<https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011>

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 24709-3:2011](https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011)

<https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	v
Introduction.....	vi
1 Scope	1
2 Conformance	1
3 Normative references	1
4 Terms and definitions	1
5 Abbreviated terms	2
6 General principles	2
7 Testing the conformance of BioAPI frameworks	3
7.1 General	3
7.2 Configuration of test assertions	3
7.3 Test flow	6
7.4 Initialisation and termination	6
7.5 List of test assertions	7
7.6 BioAPI conformity statement	10
8 Test assertions	11
8.1 Descriptions of Test Tables	11
8.2 Descriptions of XML text	15
8.3 Common activities	16
8.4 Assertion 1.1 – BioAPI_Init	28
8.5 Assertion 1.2 – BioAPI_Terminate	30
8.6 Assertion 1.3 – BioAPI_GetFrameworkInfo	33
8.7 Assertion 1.4 – BioAPI_EnumBSPs	35
8.8 Assertion 1.5 – BioAPI_BSPLoad_And_BioSPI_BSPLoad	40
8.9 Assertion 1.6 – BioAPI_BSPUnload_And_BioSPI_BSPUnload	44
8.10 Assertion 1.7 – BioAPI_BSPAttach_And_BioSPI_BSPAttach	47
8.11 Assertion 1.8 – BioAPI_BSPDetach_And_BioSPI_BSPDetach	51
8.12 Assertion 1.9 – BioAPI_QueryUnits_And_BioSPI_QueryUnits	53
8.13 Assertion 1.10 – BioAPI_EnumBFPs	58
8.14 Assertion 1.11 – BioAPI_QueryBFPs_And_BioSPI_QueryBFPs	60
8.15 Assertion 1.12 – BioAPI_ControlUnit_And_BioSPI_ControlUnit	63
8.16 Assertion 2.1 – BioAPI_FreeBIRHandle_And_BioSPI_FreeBIRHandle	67
8.17 Assertion 2.2 – BioAPI_GetBIRFromHandle_And_BioSPI_GetBIRFromHandle	69
8.18 Assertion 2.3 – BioAPI_GetHeaderFromHandle_And_BioSPI_GetHeaderFromHandle	73
8.19 Assertion 3.1 – BioAPI_EnableEvents_And_BioSPI_EnableEvents	76
8.20 Assertion 3.2 – BioAPI_SetGUICallbacks_And_BioSPI_SetGUICallbacks	79
8.21 Assertion 4.1 – BioAPI_Capture_And_BioSPI_Capture	82
8.22 Assertion 4.2 – BioAPI_CreateTemplate_And_BioSPI_CreateTemplate	87
8.23 Assertion 4.3 – BioAPI_Process_And_BioSPI_Process	92
8.24 Assertion 4.4 – BioAPI_ProcessWithAuxBIR_And_BioSPI_ProcessWithAuxBIR	95
8.25 Assertion 4.5 – BioAPI_VerifyMatch_And_BioSPI_VerifyMatch	100
8.26 Assertion 4.6 – BioAPI_IdentifyMatch_And_BioSPI_IdentifyMatch	105
8.27 Assertion 4.7 – BioAPI_Enroll_And_BioSPI_Enroll	111
8.28 Assertion 4.8 – BioAPI_Verify_And_BioSPI_Verify	117
8.29 Assertion 4.9 – BioAPI_Identify_And_BioSPI_Identify	125
8.30 Assertion 4.10 – BioAPI_Import_And_BioSPI_Import	133
8.31 Assertion 4.11 – BioAPI_PresetIdentifyPopulation_And_BioSPI_PresetIdentifyPopulation	137
8.32 Assertion 5.1 – BioAPI_DbOpen_And_BioSPI_DbOpen	140

8.33	Assertion 5.2 – BioAPI_DbClose_And_BioSPI_DbClose	143
8.34	Assertion 5.3 – BioAPI_DbCreate_And_BioSPI_DbCreate	146
8.35	Assertion 5.4 – BioAPI_DbDelete_And_BioSPI_DbDelete	150
8.36	Assertion 5.5 – BioAPI_DbSetMarker_And_BioSPI_DbSetMarker	153
8.37	Assertion 5.6 – BioAPI_DbFreeMarker_And_BioSPI_DbFreeMarker	156
8.38	Assertion 5.7 – BioAPI_DbStoreBIR_And_BioSPI_DbStoreBIR	159
8.39	Assertion 5.8 – BioAPI_DbGetBIR_And_BioSPI_DbGetBIR	162
8.40	Assertion 5.9 – BioAPI_DbGetNextBIR_And_BioSPI_DbGetNextBIR	166
8.41	Assertion 5.10 – BioAPI_DbDeleteBIR_And_BioSPI_DbDeleteBIR	169
8.42	Assertion 6.1 – BioAPI_SetPowerMode_And_BioSPI_SetPowerMode	172
8.43	Assertion 6.2 – BioAPI_SetIndicatorStatus_And_BioSPI_SetIndicatorStatus	175
8.44	Assertion 6.3 – BioAPI_GetIndicatorStatus_And_BioSPI_GetIndicatorStatus	178
8.45	Assertion 6.4 – BioAPI_CalibrateSensor_And_BioSPI_CalibrateSensor	181
8.46	Assertion 7.1 – BioAPI_Cancel_And_BioSPI_Cancel	183
8.47	Assertion 7.2 – BioAPI_Free_And_BioSPI_Free	186
8.48	Assertion 8.1 – BioAPI_Util_InstallBSP	188
8.49	Assertion 8.2 – BioAPI_Util_InstallBFP	192
Annex A (informative) Test procedures for the multiple-component support		196
A.1	General	196
A.1.1	Principles	196
A.2	Test procedures	197
A.2.1	Role of API/SPI routing for the applications and the BSPs	197
Bibliography		200

iTeh STANDARD PREVIEW
(standards.iteh.ai)

<https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011>

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.

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

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

ISO/IEC 24709-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

ISO/IEC 24709 consists of the following parts, under the general title *Information technology — Conformance testing for the biometric application programming interface (BioAPI)*:

- *Part 1: Methods and procedures* [ISO/IEC 24709-3:2011](https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011)
- *Part 2: Test assertions for biometric service providers*
- *Part 3: Test assertions for BioAPI frameworks*

The following part is under preparation:

- *Part 4: Test assertions for BioAPI applications*

Introduction

The test assertions specified in this part of ISO/IEC 24709 enable a user of this part of ISO/IEC 24709 (such as a testing laboratory) to test the conformance to ISO/IEC 19784-1 (BioAPI2.0) of any BioAPI framework that claims to be a conforming implementation of that International Standard.

The organization of the test assertions in this part of ISO/IEC 24709 reflects the structure of Annex A of ISO/IEC 19784-1:2006, which specifies conformance to BioAPI for various types of implementations (BSPs, frameworks, and applications) and for BSPs belonging to several conformance subclasses.

This part of ISO/IEC 24709 contains test assertions for testing conformance of BioAPI frameworks to claim compliance to the BioAPI specification defined by ISO/IEC 19784-1:2006. The assertions are further organized according to conformance subclass (if any) and claimed support of optional features.

Each test assertion exercises one or more (possibly elementary) features of an implementation under test. Assertions are placed into packages (one or more assertions per package) as required by the assertion language.

Clause 6 specifies general principles.

Clause 7 specifies the principles and the testing mechanism for the conformance testing for BioAPI frameworks in addition to listing up the test assertions to be used in this conformance testing model.

Clause 8 specifies the assertions to be used in the conformance testing model for BioAPI frameworks.

[ISO/IEC 24709-3:2011](https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011)

<https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011>

Information technology — Conformance testing for the biometric application programming interface (BioAPI) —

Part 3: Test assertions for BioAPI frameworks

1 Scope

This part of ISO/IEC 24709 defines a number of test assertions written in the assertion language specified in ISO/IEC 24709-1:2007.

This part of ISO/IEC 24709 specifies all the test assertions that are to be executed for conformance testing of BioAPI frameworks claiming conformance to ISO/IEC 19784-1 (BioAPI 2.0).

Test assertions specified in this part of ISO/IEC 24709 are not claimed to be exhaustive (see also ISO/IEC 24709-1:2007, Clause 6). Implementations of BioAPI 2.0 that are tested according to the methodology specified in ISO/IEC 24709-1:2007 and with test assertions specified in this part of ISO/IEC 24709 can (only) claim conformance to those aspects of ISO/IEC 19784-1 that are covered by these test assertions.

[ISO/IEC 24709-3:2011](https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011)

<https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a6003f/iso-iec-24709-3-2011>

2 Conformance

Implementations (BioAPI conformance test suites) claiming conformance to this part of ISO/IEC 24709 shall be able to process all the test assertions specified in Clause 8 according to the methodology specified in ISO/IEC 24709-1:2007 and the general principles and provisions specified in Clauses 6 and 7.

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*

ISO/IEC 24709-1:2007, *Information technology — Conformance testing for the biometric application programming interface (BioAPI) — Part 1: Methods and procedures*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19784-1:2006, ISO/IEC 24709-1:2007 and the following apply.

4.1 API/SPI routing
feature provided by the BioAPI Framework to handle multiple applications and/or multiple biometric service providers (BSPs), with which a BioAPI call from an application is correctly given to the BSP specified by the application, and with which a BioSPI return from a BSP is correctly given to the application that specified the BSP

5 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

API	application programming interface
BIR	biometric information record
BSP	biometric service provider
CBEFF	common biometric exchange format framework
FMR	false match rate
FPI	function provider interface
GUI	graphic user interface
ID	identity/identification/identifier
SPI	service provider interface
UUID	universally unique identifier
BCS	BioAPI conformity statement
CTS	BioAPI conformance test suite
IUT	implementation under test

6 General principles

6.1 The test assertions listed in Clause 7 and specified in Clause 8 are based on the conformance testing methodology specified in ISO/IEC 24709-1:2007, and can only be used in the context of that methodology. The assertions are written in the assertion language specified in ISO/IEC 24709-1:2007, which is part of that methodology.

6.2 An important concept of the conformance testing methodology specified in ISO/IEC 24709-1:2007 is the existence of three conformance testing models (see ISO/IEC 24709-1:2007, Clause 6).

- a) the conformance testing model for BioAPI applications
- b) the conformance testing model for BioAPI frameworks
- c) the conformance testing model for BioAPI BSPs

6.3 Each testing model is concerned with testing one of the three standard components of the BioAPI architecture (see ISO/IEC 24709-1:2007, Clause 6). Clause 8 of this part of ISO/IEC 24709 specifies a number of test assertions for the conformance testing model for BioAPI frameworks. This part of ISO/IEC 24709 is not concerned with the conformance testing models for BioAPI applications, BSPs and corresponding test assertions.

6.4 In the conformance testing model for BioAPI frameworks, a special testing component (called the "framework-testing application") shall replace the normal application, and another special testing component (called the "framework-testing BSP") shall replace the normal BSP. (See 6.2.5.2 of ISO/IEC 24709-1:2007).

7 Testing the conformance of BioAPI frameworks

7.1 General

7.1.1 This subclause describes the principles of the test conditions of the conformance test for BioAPI frameworks and the principles to create pass/fail results.

7.1.2 The principles of the test conditions of the BioAPI frameworks under test are as follows:

- All the BioAPI functions and all the related BioSPI functions shall be tested.
- All the parameters defined in each function shall be tested.
- All the values that are able to be specified in each of the parameters shall be tested.
- All the capabilities in the BSP schema of the testing BSP that are related to the test case shall be tested.

7.1.3 The principles to create pass/fails results of the test cases of the BioAPI frameworks are as follows:

- The return value shall be checked. If ISO/IEC 19784-1:2006 does not specify an error value for a specific BioAPI function, the test shall pass if one of the possible error values has been returned.
- As for the output parameters, they shall be checked if ISO/IEC 19784-1:2006 declares that the BioAPI framework sets the values to the output parameters.
- As for the parameters that are transferred from the BioAPI function to the BioSPI function by the BioAPI framework, the test assertions shall check if all the parameters are correctly transferred.

NOTE 1 In this edition of ISO/IEC 24709-3, the test assertions do not contain (i) the test cases with combinations of various parameters, (ii) the test cases with various sequences of BioAPI functions, (iii) the test cases with the callback functions, (iv) the test cases related to asynchronous behaviors, (v) error cases derived from the incorrect implementation of the testing BSP and (vi) the test cases that are only relevant to optional features of the testing BSP. The excluded test cases will be considered in the next edition of this part of ISO/IEC 24709. The excluded test cases will be considered in the next edition of this part of ISO/IEC 24709.

NOTE 2 A test case for the multiple component support of the BioAPI Framework is described in Annex A as a recommendation for the Conformance Test Suite to implement the test for the API/SPI routing feature of the BioAPI Framework.

NOTE 3 As for the error handling of invalid parameters, ISO/IEC 19784-1:2006 makes allowance for the freedom of implementation for BioAPI frameworks and BSPs. This text shall take into account the following cases to create the test assertions conformant to ISO/IEC 19784-1:2006:

- (a) In the specification of ISO/IEC 19784-1:2006, it does not necessarily specify one error value to be returned from the BioAPI function if there is an incorrect value specified in one of the parameters in the BioAPI function. In such cases, the test assertions only check if one of the possible error values is returned and do not define in the test assertion that a specific error value shall be returned.
- (b) There is no specific description in ISO/IEC 19784-1:2006 whether the BioAPI framework shall do the parameter check or the testing BSP shall do it. Therefore in this text, the test assertions do not care about which component (BioAPI framework or BSP) has found the errors when it checked the parameters. Therefore, the test assertions do not care about the error codes which includes the highest 8 bits that is allowed to be set by the BioAPI framework or by the testing BSP. It also do not care about whether the BioAPI framework calls the corresponding BioSPI function even after there is an error in the parameters in the BioAPI functions. The followings are the examples of the error cases that the BioAPI framework can positively handle the error without calling the corresponding BioSPI function.
 1. The value in the parameter is irrelevant; the value not relevant to the enrollment purpose such as BioAPI_PURPOSE_AUDIT is set to the Purpose parameter in the BioAPI_Enroll function.
 2. The value is not supported by the BSP; even though the BSP does not support the feature to specify Subtype, the value for the Subtype parameter such as BioAPI_BIR_RIGHT is given.

7.2 Configuration of test assertions

7.2.1 A test assertion consists of the three tables that exist per BioAPI function under test, and the XML text that exists per BioAPI function or BioSPI function.

7.2.2 The three tables that comprise a part of the test assertions are (i) Default Input Table, which assembles the default values for all the input parameters of the BioAPI function under test, (ii) Test Condition Table, which assembles all the conditions given during the BioAPI framework is tested, and (iii) Expected Result Table, which is used to create pass/fail results by comparing the expected test results in the table with the values given from the BioAPI framework. In Test Condition Table, each row shows a test case and each column shows (i) the values given to the parameters of the BioAPI function, (ii) the BSP Schema information related to the test case and (iii) the return value from the BSP via the BioSPI function. In Expected Result Table, each row shows a test case same as Test Condition Table and each column shows the information that is used to make a pass/fail decision by referring to it. To make a decision, the return value and the output parameters related to the test case are used.

NOTE In the test assertions in each subclause in Chapter 8, it simply uses the words Test Condition Table and Expected Result Table without putting the suffixes such as the BioAPI and/or BioSPI function names to avoid redundancy.

7.2.3 Before calling a BioAPI function to be tested, the testing application shall read the values described in the Default Input Table first then read one of test conditions described in the Test Condition Table by picking up one of the rows in the table, which means that all the input parameters are set by the testing application by referring to the Default Input Table but one of the parameters is overwritten by the value described in the Test Condition Table. The testing application shall repeat reading the two tables every time it executes a test case.

7.2.4 In Default Input Table, the input parameter names and the input parameter values for the BioAPI function under test are described. The sequence of the parameters are the same as the one described in ISO/IEC 24709-1.

7.2.5 In Test Condition Table, the following information is described.

- (a) Input parameter name and input parameter value: Describes the parameters given to the BioAPI function to be tested.
- (b) Supported options in BSP Schema: Chooses the options in BioAPI_OPERATIONS_MASK and BioAPI_OPTIONS_MASK in the BSP Schema that are related to the test case to show these options are supported or not in the test case. The detail of Test Condition Table is described in 7.3 and Clause 8.
- (c) Return value (from BioSPI): Shows a value to be returned from the testing BSP after the BioSPI function is invoked by the BioAPI framework. One return value that is thought to be adequate is selected from the possible return values and described in the table.

7.2.6 In Expected Result Table, each row includes the return value and one of the names of the parameters and its value to make a decision of the pass/fail results. The detail of Expected Result Table is shown in 7.3 and Clause 8.

7.2.7 In the XML texts, it sets up the parameters from Default Input Table described in 7.2.4 and Test Condition Table described in 7.2.5 using the <input> element before calling the BioAPI function to be tested, then makes a pass/fail decision by obtaining the expected results from Expected Result Table described in 7.2.6 using the <input> element after the BioAPI function returns to the testing application. Each test case has the above mentioned logic in common for one BioAPI function or one BioSPI function, so in principle, there is only one XML text for one BioAPI function or one BioSPI function without having any exception that is to be applied to a particular test case.

7.2.8 The structure of CTS for BioAPI frameworks is depicted in Figure 1.

7.2.8.1 CTS consists of the testing application, the testing BSP, XML text, the two test tables and the BioAPI framework under test. Both the testing application and the testing BSP read the XML text and the test tables and run the script after translating it to the binaries executable on a computer. In addition, the testing application and the testing BSP check the behavior of the BioAPI framework through the information given by the BioAPI framework and make a pass/fail decision.

NOTE Setting the values given in the Test Condition Table to the variables defined in the XML text is responsible for CTS and it is not in the scope of this part of ISO/IEC 24709.

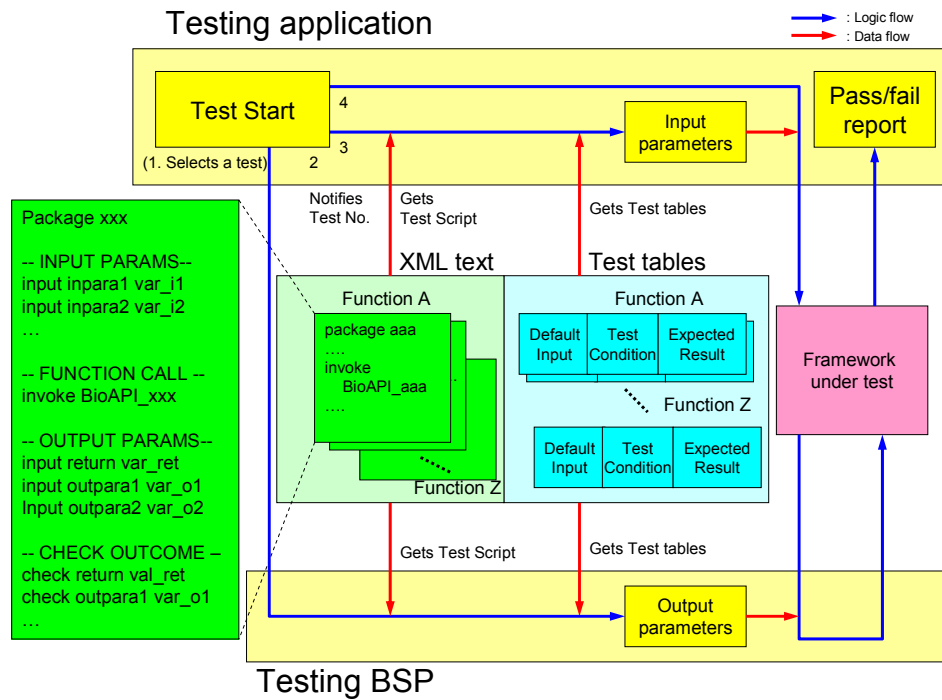


Figure 1 — Structure of CTS for BioAPI framework

7.2.8.2 In accordance with the descriptions in Test Condition Table, the testing BSP shall have a capability to change the members of BioAPI_OPERATIONS_MASK and BioAPI_OPTIONS_MASK, which are part of the BSP Schema. The members of the BSP Schema related to the conformance test for BioAPI frameworks are shown in Table 1.

ISO/IEC 24709-3:2011
<https://standards.iteh.ai/catalog/standards/sist/acd5f699-553d-49fb-a2ea-4dce25a00031/iso-iec-24709-3-2011>
 Table 1 — Members of BSP Schema related to 24709-3

Number	Members	Value
1	BSPUuid	Depends on CTS
2	Description	Depends on CTS
3	SpecVersion	0x20 (Version 2.0)
4	ProductVersion	Depends on CTS
5	Vendor	Depends on CTS
6	FactorsMask (Biometric Type)	0x00000001 (BioAPI_TYPE_MULTIPLE)
7	Operations	Depends on Test Condition Table
8	Options	Depends on Test Condition Table
9	PayloadPolicy	Depends on CTS
10	MaxPayloadSize	1024 (bytes)
11	DefaultVerifyTimeout	10000 (milliseconds)
12	DefaultIdentifyTimeout	10000 (milliseconds)
13	DefaultCaptureTimeout	10000 (milliseconds)
14	DefaultEnrollTimeout	10000 (milliseconds)
15	DefaultCalibrateTimeout	10000 (milliseconds)
16	MaxBSPDbSize	10240 (bytes)
17	MaxIdentify	0xFFFFFFFF (Unlimited)

7.3 Test flow

7.3.1 Prior to executing a test, one test case is selected (see the number 1 in Figure –1).

7.3.2 The testing application notifies the testing BSP the information which indicates a unique test case is going to be executed (see the number 2 in Figure 1). The implementation of the notification is not in the scope of this part of ISO/IEC 24709, so it depends on each CTS. (e.g. The testing BSP supports a function specific to the purpose of the notification that is only known by the testing application, then it calls the function prior to the invocation of the BioAPI function to be tested so the testing BSP can find which test case has been selected.) With such a feature for notification, the testing BSP can prepare for the test case by reading the corresponding XML text and the test tables in advance and create the appropriate BSP Schema and the return value for the BioSPI function in case the BioAPI framework invokes the BioSPI function during the test.

7.3.3 The testing application reads the corresponding XML text and the test tables same as the testing BSP does, and prepares the parameters of the BioAPI function to be tested (see the number 3 in Figure 1).

NOTE 7.3.2 and 7.3.3 can be in no particular order.

7.3.4 The testing application calls the BioAPI function after setting the parameters corresponding to the selected test case (see the number 4 in Figure -1). It depends on the implementation of the BioAPI framework whether to return an error or not when it finds a contradiction between one of the specified parameters and one of the capabilities in BioAPI_OPERATIONS_MASK or BioAPI_OPTIONS_MASK in the BSP Schema. In the former case, the BioAPI framework returns an error value to the testing application without calling the testing BSP. A pass/fail report will be created by the testing application (see 7.3.7).

7.3.5 In 7.3.4, if the BioAPI framework calls the BioSPI function after it is invoked from the testing application via the BioAPI call, the testing BSP checks the validity of the behavior of the BioAPI framework by checking the parameters with the expected results described in Expected Result Table. (In many cases, it checks whether the parameters in the BioSPI function are the same as the parameters in the BioAPI function or not.) If it finds out that the parameters in the BioSPI function are different from the description in Expected Result Table, it creates a fail report and terminates the test case.

7.3.6 If the test assertion finds out that the parameters of the BioSPI functions are correct, the testing BSP sets a return value by referring to the corresponding area in Test Condition Table and gives the control to the BioAPI framework, then it returns to the testing application.

7.3.7 The testing application checks whether the information returned from the BioAPI framework is consistent with the descriptions in Expected Result Table. If all the members of the information are correct, it creates a pass report. If one or more members is not correct, it creates a fail report.

7.4 Initialisation and termination

7.4.1 All the test cases shall include the initialisation of the BioAPI framework and the testing BSP. By initialising these components, each test is independent from the other test cases by avoiding any influence from the tests done prior to the test.

7.4.2 The initialisation includes the BioAPI_Init and the BioAPI_Util_InstallBSP so that, in addition to the initialisation of the BioAPI framework, the capabilities of the testing BSP corresponding to the test case will be created in the BSP Schema in the component registry every time the test starts. (The testing BSP knows which capabilities it shall have in the BSP Schema by the notification of the test number from the testing application.) The termination includes the BioAPI_Util_InstallBSP and the BioAPI_Terminate, so the component registry will be deleted and the BioAPI framework will be terminated every time the test finishes.

7.5 List of test assertions

7.5.1 The following tables list the test assertions for the BioAPI and BioSPI functions in accordance with the categories of the BioAPI functions described in ISO/IEC19784-1:2006.

- a) Component Management Functions
- b) Data Handle Operations
- c) Callback and Event Handling Operations
- d) Biometric Operations
- e) Database Operations
- f) BioAPI Unit Operations
- g) Utility Functions
- h) Component Registry Functions

NOTE Successful processing of all applicable test assertions is prima facie evidence that the implementation satisfies the applicable requirements of ISO/IEC19784-1, but does not establish this, as the assertions are not (and cannot be) an exhaustive test of conformance (see also ISO/IEC 29709-1:2007, Clause 6).

7.5.1.1 For the Component Management Functions, the implementation shall be tested by executing all of the following assertions (in order):

Table 2 — Test assertions for component management functions

Number	Assertion Name	References in 19784-1:2006	Package
1.1	BioAPI_Init	8.1.1, 11.2.3	4839c860-7929-11de-8a39-0800200c9a66
1.2	BioAPI_Terminate	8.1.2	8782cd50-7929-11de-8a39-0800200c9a66
1.3	BioAPI_GetFrameworkInfo	8.1.3	b3a468d0-7929-11de-8a39-0800200c9a66
1.4	BioAPI_EnumBSPs	8.1.4	ce45e240-7929-11de-8a39-0800200c9a66
1.5	BioAPI_BSPLoad_And_BioSPI_BSPLoad	8.1.5, 9.3.1	f481f070-7929-11de-8a39-0800200c9a66
1.6	BioAPI_BSPUnload_And_BioSPI_BSPUnload	8.1.6, 9.3.1.2	1067a9b0-792a-11de-8a39-0800200c9a66
1.7	BioAPI_BSPAttach_And_BioSPI_BSPAttach	8.1.7, 9.3.1.3	2ae45d10-792a-11de-8a39-0800200c9a66
1.8	BioAPI_BSPDetach_And_BioSPI_BSPDetach	8.1.8, 9.3.1.4	4149b370-792a-11de-8a39-0800200c9a66
1.9	BioAPI_QueryUnits_And_BioSPI_QueryUnits	8.1.9, 9.3.1.5	507a4030-792a-11de-8a39-0800200c9a66
1.10	BioAPI_EnumBFPs	8.1.10	62eb03d0-792a-11de-8a39-0800200c9a66
1.11	BioAPI_QueryBFPs_And_BioSPI_QueryBFPs	8.1.11, 9.3.1.6	70d92580-792a-11de-8a39-0800200c9a66
1.12	BioAPI_ControlUnit_And_BioSPI_ControlUnit	8.1.12, 9.3.1.7	819d98b0-792a-11de-8a39-0800200c9a66

7.5.1.2 For the Data Handle Operation Functions, the implementation shall be tested by executing all of the following assertions (in order):

Table 3 — Test assertions for data handle operation functions

Number	Assertion Name	References in 19784-1:2006	Package
2.1	BioAPI_FreeBIRHandle_And_BioSPI_FreeBIRHandle	8.2.1, 9.3.2.1	94a32240-792a-11de-8a39-0800200c9a66
2.2	BioAPI_GetBIRFromHandle_And_BioSPI_GetBIRFromHandle	8.2.2, 9.3.2.2	ca10cea0-792a-11de-8a39-0800200c9a66
2.3	BioAPI_GetHeaderFromHandle_And_BioSPI_GetHeaderFromHandle	8.2.3, 9.3.2.3	d9332a90-792a-11de-8a39-0800200c9a66

7.5.1.3 For the Callback and Event Operation Functions, the implementation shall be tested by executing all of the following assertions (in order):

Table 4 — Test assertions for callback and event operation functions

Number	Assertion Name	References in 19784-1:2006	Package
3.1	BioAPI_EnableEvents_And_BioSPI_EnableEvents	8.3.1, 9.3.3.1	c0c4abd0-792c-11de-8a39-0800200c9a66
3.2	BioAPI_SetGUICallbacks_And_BioSPI_SetGUICallbacks	8.3.2, 9.3.3.2	f0a2b310-792c-11de-8a39-0800200c9a66

ISO/IEC 24709-3:2011

7.5.1.4 For the Biometric Operation Functions, the implementation shall be tested by executing all of the following assertions (in order):

Table 5 — Test assertions for biometric operation functions

Number	Assertion Name	References in 19784-1:2006	Package
4.1	BioAPI_Capture_And_BioSPI_Capture	8.4.1, 9.3.4.1	aaec0fa0-792d-11de-8a39-0800200c9a66
4.2	BioAPI_CreateTemplate_And_BioSPI_CreateTemplate	8.4.2, 9.3.4.2	b851a060-792d-11de-8a39-0800200c9a66
4.3	BioAPI_Process_And_BioSPI_Process	8.4.3, 9.3.4.3	c9a0a050-792d-11de-8a39-0800200c9a66
4.4	BioAPI_ProcessWithAuxBIR_And_BioSPI_ProcessWithBIR	8.4.4, 9.3.4.4	eab95fc0-792d-11de-8a39-0800200c9a66
4.5	BioAPI_VerifyMatch_And_BioSPI_VerifyMatch	8.4.5, 9.3.4.5	f66559a0-792d-11de-8a39-0800200c9a66
4.6	BioAPI_IdentifyMatch_And_BioSPI_IdentifyMatch	8.4.6, 9.3.4.6	028794f0-792e-11de-8a39-0800200c9a66
4.7	BioAPI_Enroll_and_BioSPI_Enroll	8.4.7, 9.3.4.7	12892d50-792e-11de-8a39-0800200c9a66
4.8	BioAPI_Verify_And_BioSPI_Verify	8.4.8, 9.3.4.8	233d9af0-792e-11de-8a39-0800200c9a66
4.9	BioAPI_Identify_And_BioSPI_Identify	8.4.9, 9.3.4.9	30b63e80-792e-11de-8a39-0800200c9a66

4.10	BioAPI_Import_And_BioSPI_Import	8.4.10, 9.3.4.10	4019c220-792e-11de-8a39-0800200c9a66
4.11	BioAPI_PresetIdentifyPopulation_And_BioSPI_PresetIdentifyPopulation	8.4.11, 9.3.4.11	55440070-792e-11de-8a39-0800200c9a66

7.5.1.5 For the Database Operation Functions, the implementation shall be tested by executing all of the following assertions (in order):

Table 6 — Test assertions for database operation functions

Number	Assertion Name	References in 19784-1:2006	Package
5.1	BioAPI_DbOpen_And_BioSPI_DbOpen	8.5.1, 9.3.5.1	9f31c870-792e-11de-8a39-0800200c9a66
5.2	BioAPI_DbClose_And_BioSPI_DbClose	8.5.2, 9.3.5.2	aa8f2d20-792e-11de-8a39-0800200c9a66
5.3	BioAPI_DbCreate_And_BioSPI_DbCreate	8.5.3, 9.3.5.3	b5f8ede0-792e-11de-8a39-0800200c9a66
5.4	BioAPI_DbDelete_And_BioSPI_DbDelete	8.5.4, 9.3.5.4	c60b0100-792e-11de-8a39-0800200c9a66
5.5	BioAPI_DbSetMarker_And_BioSPI_DbSetMarker	8.5.5, 9.3.5.5	d06aa4c0-792e-11de-8a39-0800200c9a66
5.6	BioAPI_DbFreeMarker_And_BioSPI_DbFreeMarker	8.5.6, 9.3.5.6	e4647960-792e-11de-8a39-0800200c9a66
5.7	BioAPI_DbStoreBIR_And_BioSPI_DbStoreBIR	8.5.7, 9.3.5.7	f2614110-792e-11de-8a39-0800200c9a66
5.8	BioAPI_DbGetBIR_And_BioSPI_DbGetBIR	8.5.8, 9.3.5.8	020724e0-792f-11de-8a39-0800200c9a66
5.9	BioAPI_DbGetNextBIR_And_BioSPI_DbGetNextBIR	8.5.9, 9.3.5.9	13003ca0-792f-11de-8a39-0800200c9a66
5.10	BioAPI_DbDeleteBIR_And_BioSPI_DbDeleteBIR	8.5.10, 9.3.5.10	2068daa0-792f-11de-8a39-0800200c9a66

7.5.1.6 For the BioAPI Unit Operation Functions, the implementation shall be tested by executing all of the following assertions (in order):

Table 7 — Test assertions for BioAPI unit operation functions

Number	Assertion Name	References in 19784-1:2006	Package
6.1	BioAPI_SetPowerMode_And_BioSPI_SetPowerMode	8.6.1, 9.3.6.1	5cda0770-792f-11de-8a39-0800200c9a66
6.2	BioAPI_SetIndicatorStatus_And_BioSPI_SetIndicatorStatus	8.6.2, 9.3.6.2	678e0cc0-792f-11de-8a39-0800200c9a66
6.3	BioAPI_GetIndicatorStatus_And_BioSPI_GetIndicatorStatus	8.6.3, 9.3.6.3	72b67ec0-792f-11de-8a39-0800200c9a66
6.4	BioAPI_CalibrateSensor_And_BioSPI_CalibrateSensor	8.6.4, 9.3.6.4	833adbb0-792f-11de-8a39-0800200c9a66