

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

DOC: M1/04-0634

Date: September 29, 2004

Ref: Project 1703-D

Initial Contribution: Test Assertions for ANSI INCITS 358-2002 (BioAPI)

Editor's Contribution

Reply to: Mark Jerde
Phone: (301) 975-6622
Email: mjerde@nist.gov

BioAPI Conformance Assessment Features And Assertions

<i>Feature</i>				<i>Reference</i>
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
1 BioAPI Registry				2.2, 4.2, 4.2.3
1.1	1	To test entries to the module registry on BSP installation.	The BSP is not installed.	List of required registry entries: - Support for local operation, distributed operation, or both.
1.2	1	To test that the BSP posts to the module registry whether it supports Local operation, Distributed operation, or both.	TBD	TBD
2 BioSPI Module Load				3.3.1.1 (P. 97)
2.1	1	To test BioSPI_ModuleLoad with valid parameters.	Call BioSPI_ModuleLoad with valid parameters.	Return code BioAPI_OK.
2.2	1	To test BioSPI_ModuleLoad with invalid parameters.	Call BioSPI_ModuleLoad with one or more invalid parameters.	Return code other than BioAPI_OK.
2.3	1	To test BioSPI_ModuleLoad failure.	TBD	Return code BioAPI_H_FRAMEWORK_MODULE_LOAD_FAIL.
2.4	1	To test BioSPI_ModuleLoad with an invalid UUID.	TBD	Return code BioAPIERR_H_FRAMEWORK_INVALID_UUID.
3 BioSPI Module Unload				3.3.1.2
3.1	1	To test BioSPI_ModuleUnload with valid parameters.	BioSPI_ModuleLoad succeeded.	Return code BioAPI_OK.
3.2	1	To test BioSPI_ModuleUnload unmatched with BioSPI_ModuleLoad.	Initial state, or calling ModuleUnload twice.	Return code BioAPI_H_FRAMEWORK_MODULE_UNLOAD_FAILED.
3.3	1	To test BioSPI_ModuleUnload with an invalid UUID.	BioSPI_ModuleLoad succeeded.	Return code BioAPI_H_FRAMEWORK_INVALID_UUID.
4 BioSPI Module Attach				3.3.1.3
4.1	1	To test BioSPI_ModuleAttach with valid parameters.	TBD	Return code BioAPI_OK.
4.2	1	To test BioSPI_ModuleAttach with an invalid UUID.	TBD	Return code BioAPI_H_FRAMEWORK_INVALID_UUID.
4.3	1	To test BioSPI_ModuleAttach with an invalid BioAPI Version.	TBD	Return code other than BioAPI_OK.
5 BioSPI Detach				3.3.1.4
5.1	1	To test BioAPI_ModuleDetach with a valid handle.	TBD	Return code BioAPI_OK.
5.2	1	To test BioAPI_ModuleDetach with an invalid handle.	TBD	Return code other than BioAPI_OK.

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
6 BioSPI Free BIR Handle				3.3.2.1
6.1	1	To test BioSPI_FreeBIRHandle with a valid handle.	TBD	Return code BioAPI_OK.
6.2	1	To test BioSPI_FreeBIRHandle with an invalid handle.	TBD	Return code other than BioAPI_OK.
7 BioSPI Get BIR From Handle				3.3.2.2
7.1	1	To test BioSPI_GetBIRFromHandle with valid parameters.	TBD	1. Return code BioAPI_OK. 2. Valid BIR.
7.2	1	To test BioSPI_GetBIRFromHandle with an invalid ModuleHandle.	TBD	Return code other than BioAPI_OK.
7.3	1	To test BioSPI_GetBIRFromHandle with an invalid BIR Handle.	TBD.	Return code other than BioAPI_OK.
8 BioSPI Get Header from Handle				3.3.2.3
8.1	1	To test BioSPI_GetHeaderFromHandle with valid parameters.	TBD	1. Return code BioAPI_OK. 2. Valid BIR header.
8.2	1	To test BioSPI_GetHeaderFromHandle with an invalid ModuleHandle.	TBD	Return code other than BioAPI_OK.
8.3	1	To test BioSPI_GetHeaderFromHandle with an invalid BIR Handle.	TBD	Return code other than BioAPI_OK.
9 BioSPI Enable Events				3.3.3.1
9.1	1	To test BioSPI_EnableEvents with valid parameters.	TBD	Return code BioAPI_OK.
9.2	1	To test BioSPI_EnableEvents with an invalid ModuleHandle.	TBD	Return code other than BioAPI_OK.
9.3	1	To test BioSPI_EnableEvents with an invalid Events mask.	TBD	Return code other than BioAPI_OK.
10 BioSPI Set GUI Callbacks				3.3.3.2
NA				
11 BioSPI Set Stream Callbacks				3.3.3.3
NA				

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
12 BioSPI Stream Input Output				3.3.3.4
NA				
13 BioSPI Capture				3.3.4.1
13.1	2	To test that either an "intermediate" BIR or a "processed" BIR is returned for a specified process.	A valid biometric is presented to BioSPI_Capture.	Return code BioAPI_OK and a valid "intermediate" or "processed" BIR.
13.2	2	To test serialization of capture with timeout.	<ol style="list-style-type: none"> 1. Start test. 2. Prompt "Start 2nd BioAPI_Capture Application (Long Timeout)" and pause test. 3. Start 2nd application, observe GUI. 4. Continue test (Timeout shorter than 2nd application) 5. Wait for timeout. 	Return code BioAPIERR_BSP_TIMEOUT_EXPIRED.
13a Return of raw/audit data				3.3.4.1
13a.1	2	To test the return of "raw" data if the BSP supports return of raw/audit data.	A valid biometric is presented to BioSPI_Capture.	If the BSP supports AuditData, a handle to a "raw" BIR or a value of BioAPI_INVALID_BIR_HANDLE if no audit data is available. If AuditData is not supported, the BSP may return a handle value of BioAPI_UNSUPPORTED_BIR_HANDLE.
13b Return of quality in the captured BIR header				3.3.4.1, (2.1.46, 4.2.4.2)
13b.1	2	To test that Purpose is recorded in the header of the CapturedBIR.	A valid biometric is presented to BioSPI_Capture.	Return code BioAPI_OK and Purpose in the header of the CapturedBIR.
13c BIR signing (by BSP)				1.5, 2.1.7
13c.1	3	To test BIR signing if supported by the BSP.	If the BSP supports signing call BioSPI_Capture with valid parameters. Present a valid biometric.	Return code BioAPI_OK and a valid CapturedBIR that includes a signature.
13d BIR encryption (by BSP)				1.5, 2.1.7
13d.1	3	To test BIR encryption if supported by the BSP.	If the BSP supports BIR encryption call BioSPI_Enroll with valid parameters. Present a valid biometric.	Return code BioAPI_OK and a valid CapturedBIR.
13e Detection of Source Presence				4.2.4.2
NA				

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
13f Support of application control of the GUI				1.10, 4.2.4.2
		NA		
14 BioSPI Create Template				3.3.4.2
14.1	2	To test that a NewTemplate is created from a CapturedBIR.	BioSPI_Create_Template is passed a valid CapturedBIR.	Return code BioAPI_OK and a valid NewTemplate.
14a Accept input of stored template to return update/adapted template				3.3.4.2
14a.1	3	If the BSP supports template adaptation, test that the StoredTemplate remains unchanged.	BioSPI_Create_Template is passed a valid CapturedBIR and StoredTemplate.	Return code BioAPI_OK and an unchanged StoredTemplate.
14b Acceptance of payload for inclusion of enrollment BIR				3.3.4.2
14b.1	3	To test Payload if supported by the BSP.	If the BSP supports Payload, call BioSPI_CreateTemplate with valid parameters including Payload.	Return code BioAPI_OK and a valid NewTemplate that includes the Payload.
14b.2	3	If the BSP supports template adaptation and a BIR payload, test that the payload is not copied from StoredTemplate to NewTemplate.	BioSPI_Create_Template is passed a valid CapturedBIR and StoredTemplate. The StoredTemplate must contain a payload. The Payload parameter is NULL.	Return code BioAPI_OK and a valid NewTemplate that does not contain a payload.
14c Return of quality in the processed BIR header				3.3.4.2, (2.1.46, 4.2.4.2)
14c.1	3	To test the return of quality in the enrollment BIR header if supported by the BSP.	If the BSP supports return of quality, call BioSPI_CreateTemplate with valid parameters.	Return code BioAPI_OK and a valid NewTemplate that includes the quality.
14d BIR signing (by BSP)				1.5, 2.1.7
14d.1	3	To test BIR signing if supported by the BSP.	If the BSP supports signing call BioSPI_CreateTemplate with valid parameters.	Return code BioAPI_OK and a valid NewTemplate that includes a signature.
14e BIR encryption (by BSP)				1.5, 2.1.7
14e.1	3	To test BIR encryption if supported by the BSP.	If the BSP supports BIR encryption call BioSPI_CreateTemplatel with valid parameters.	Return code BioAPI_OK and a valid NewTemplate.

Feature				Reference
Num	Priority	Purpose	Scenario	Expected Results
15 BioSPI Process				3.3.4.3
15.1	2	If the attached BSP has processing capability, test that the BSP builds a ProcessedBIR from an "intermediate" CapturedBIR.	BioSPI_Process is passed a valid "intermediate" CapturedBIR.	Return code BioAPI_OK and a valid ProcessedBIR.
15.2	2	If the attached BSP does not have processing capability, test that the BSP returns a NULL ProcessedBIR from an "intermediate" CapturedBIR.	BioSPI_Process is passed a valid "intermediate" CapturedBIR.	Return code BioAPI_OK and ProcessedBIR is NULL.
15a Return of quality in the processed BIR header				3.3.4.3, (2.1.46, 4.2.4.2)
15a.1	3	To test the return of quality in the ProcessedBIR header if supported by the BSP.	If the BSP supports return of quality, call BioSPI_Process with valid parameters.	Return code BioAPI_OK and a valid ProcessedBIR that includes the quality in its header.
15b BIR signing (by BSP)				1.5, 2.1.7
15b.1	3	To test BIR signing if supported by the BSP.	If the BSP supports signing call BioSPI_Process with valid parameters.	Return code BioAPI_OK and a valid ProcessedBIR that includes a signature.
15c BIR encryption (by BSP)				1.5, 2.1.7
15c.1	3	To test BIR encryption if supported by the BSP.	If the BSP supports BIR encryption call BioSPI_Process with valid parameters.	Return code BioAPI_OK and a valid ProcessedBIR.
16 BioSPI Verify Match				3.3.4.4
16.1	2	To test BioSPI_VerifyMatch with valid parameters.	Valid parameters are passed to BioSPI_VerifyMatch.	Return code BioAPI_OK. Result can be either BioAPI_TRUE or BioAPI_FALSE. FARAchieved must be valid.
16.2	2	To test failure on unspecified FAR.	BioSPI_VerifyMatch is passed valid parameters except for MaxFARRequested.	Return code other than BioAPI_OK.
16.3	2	To test failure when both FAR and FRR are specified but FARPrecedence is not specified.	BioSPI_VerifyMatch is passed valid parameters (including both FAR and FRR) except for FARPrecedence..	Return code other than BioAPI_OK.
16a Set threshold using FRR criteria				3.3.4.4
NA				
16b Model/template adaptation				3.3.4.4
16b.1	3	To test Model/template adaptation if supported by the BSP.	If the BSP supports Model/template adaptation, call BioSPI_VerifyMatch with valid parameters.	Return code BioAPI_OK and a valid AdaptedBIR.

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
16c Return of continuous scores				1.7
NT				
16d Return of achieved FRR score				3.3.4.4
16d.1	3	To test return of achieved FRR score if supported by the BSP.	If the BSP supports the return of achieved FRR score, call BioSPI_VerifyMatch with valid parameters. ProcessedBIR and Stored template should be a "match."	Return code BioAPI_OK and a valid FRRAchieved.
16d.2	3	To test that FRRAchieved returns -2 if the BSP does not support the return of achieved FRR score.	If the BSP does not support the return of achieved FRR score, call BioSPI_VerifyMatch with valid parameters.	Return code BioAPI_OK and a FRRAchieved value of -2.
16e Return of payload				3.3.4.4
16e.1	3	To test Return of payload if supported by the BSP and verification succeeds.	If the BSP supports the return of payload, call BioSPI_VerifyMatch with valid parameters. ProcessedBIR and Stored template should be a "match."	Return code BioAPI_OK, Result of BioAPI_TRUE and a valid Payload.
16e.2	3	To test the non Return of payload if verification fails.	Call BioSPI_VerifyMatch with valid parameters. ProcessedBIR and Stored template should be a "non-match."	Return code BioAPI_OK, Result of BioAPI_FALSE and an invalid Payload.
16e.3	3	To test that Payload returns -1 if the BSP does not support payload carry.	If the BSP does not support payload: 1. Call BioSPI_VerifyMatch with valid parameters. ProcessedBIR and Stored template should be a "non-match." 2. Call BioSPI_VerifyMatch with valid parameters. ProcessedBIR and Stored template should be a "match."	1. Return code BioAPI_OK, Result of BioAPI_FALSE and a Payload of -1. 2. Return code BioAPI_OK, Result of BioAPI_TRUE and a Payload of -1.
17 BioSPI Identify Match				3.3.4.5
NA				
17a Set threshold using FRR criteria				3.3.4.5
NA				

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
17b Return of continuous scores				1.7
		NT		
17c Return of achieved FRR score				3.3.4.5
		NA		
17d Support of binning				3.3.4.5
		NA		
18 BioSPI Enroll				3.3.4.6
18.1	1	To test BioSPI_Enroll with valid parameters.	BioSPI_Enroll is called with valid parameters and a valid biometric is presented.	Return code BioAPI_OK and NewTemplate is valid..
18.2	1	To test BioSPI_Enroll with an invalid Purpose.	BioSPI_Enroll is called with valid parameters, except Purpose, which is invalid.. A valid biometric is presented.	Return code BioAPIERR_BSP_PURPOSE_NOT_SUPPORTED.
18.3	1	To test BioSPI_Enroll timeout feature.	BioSPI_Enroll is called with valid parameters but is not presented with a biometric.	Return code BioAPIERR_BSP_TIMEOUT_EXPIRED.
18.4	1	To test failure on negative Timeout other than the value -1.	BioSPI_Enroll is passed valid parameters except Timeout, with is a negative number other than -1.	Return code other than BioAPI_OK.
18a Template update				3.3.4.6
18a.1	3	Test template adaptation if supported by the BSP.	BioSPI_Enroll is passed valid parameters, including StoredTemplate.. A valid biometric is presented.	Return code BioAPI_OK and a valid NewTemplate.
18b Acceptance of payload for inclusion of enrollment BIR				3.3.4.6
18b.0	3 (?)	To test failure on Payload too large if Payload supported by the BSP.	BioSPI_Enroll is passed valid parameters except Payload, whose size exceeds the maximum payload size specified in the module registry.	Return code other than BioAPI_OK.
18b.1	3 (?)	To test Payload.	BioSPI_Enroll is passed valid parameters, including Payload.. A valid biometric is presented.	Return code BioAPI_OK and a valid NewTemplate that contains the Payload.

Feature			Scenario	Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>		<i>Expected Results</i>
18c Return of raw/audit data				3.3.4.6
18c.1	2	To test AuditData.	BioAPI_Enroll is passed valid parameters. A valid biometric is presented.	One of the following: 1. Return code BioAPI_OK and valid AuditData. 2. BioAPI_UNSUPPORTED_BIR_HANDLE if the BSP does not support audit data. 3. BioAPI_INVALID_BIR_HANDLE if the BSP supports audit data but no audit data is available.
18d Return of quality in the enrollment BIR header				3.3.4.6, (2.1.46, 4.2.4.2)
18d.1	2	To test the return of quality in the enrollment BIR header if supported by the BSP.	If the BSP supports return of quality, call BioSPI_Enroll with valid parameters. Present a valid biometric.	Return code BioAPI_OK and a valid NewTemplate that includes the quality in its header.
18e Support of application control of the GUI				1.10, 4.2.4.2
NA				
18f BIR signing (by BSP)				1.5, 2.1.7, 4.2.4.2
18f.1	3	To test BIR signing if supported by the BSP.	If the BSP supports signing call BioSPI_Enroll with valid parameters. Present a valid biometric.	Return code BioAPI_OK and a valid NewTemplate that includes a signature.
18g BIR encryption (by BSP)				1.5, 2.1.7
18g.1	3	To test BIR encryption if supported by the BSP.	If the BSP supports BIR encryption call BioSPI_Enroll with valid parameters. Present a valid biometric.	Return code BioAPI_OK and a valid NewTemplate.
18h BSP client/server comms				1.6, 4.2.4.2
NA				
19 BioSPI Verify				3.3.4.7
19.1	1	To test BioSPI_Verify with valid parameters.	Valid parameters are passed to BioSPI_Verify.	Return code BioAPI_OK. Result can be either BioAPI_TRUE or BioAPI_FALSE. FARAchieved must be valid.
19.2	1	To test failure on unspecified FAR.	BioSPI_Verify is passed valid parameters except for MaxFARRequested.	Return code other than BioAPI_OK.

Feature			Reference	
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
19.3	1	To test failure when both FAR and FRR are specified but FARPrecedence is not specified.	BioSPI_VerifyMatch is passed valid parameters (including both FAR and FRR) except for FARPrecedence.	Return code other than BioAPI_OK.
19a Set threshold using FRR criteria				3.3.4.7
NA				
19b Model/template adaptation				3.3.4.7
19b.1	3	To test Model/template adaptation if supported by the BSP.	If the BSP supports Model/template adaptation, call BioSPI_Verify with valid parameters.	Return code BioAPI_OK and a valid AdaptedBIR.
19c Return of continuous scores				1.7
NT				
19d Return of achieved FRR score				3.3.4.7
NA				
19e Return of payload				3.3.4.7
19e.1	3	To test Return of payload if supported by the BSP and verification succeeds.	If the BSP supports the return of payload, call BioSPI_Verify with valid parameters. ProcessedBIR and Stored template should be a "match."	Return code BioAPI_OK, Result of BioAPI_TRUE and a valid Payload.
19e.2	3	To test the non Return of payload if verification fails.	Call BioSPI_Verify with valid parameters. ProcessedBIR and Stored template should be a "non-match."	Return code BioAPI_OK, Result of BioAPI_FALSE and an invalid Payload.
19e.3	3	To test that Payload returns -1 if the BSP does not support payload carry.	If the BSP does not support payload: 1. Call BioSPI_Verify with valid parameters. ProcessedBIR and Stored template should be a "non-match." 2. Call BioSPI_Verify with valid parameters. ProcessedBIR and Stored template should be a "match."	1. Return code BioAPI_OK, Result of BioAPI_FALSE and a Payload of -1. 2. Return code BioAPI_OK, Result of BioAPI_TRUE and a Payload of -1.
19f Return of raw/audit data				3.3.4.7
19f.1	2	To test the return of "raw" data if the BSP supports return of raw/audit data.	If the BSP supports the return of raw/audit data, call BioSPI_Verify with valid parameters, including AuditData.	Return code BioAPI_OK and a valid AuditData.

Feature		Scenario	Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Expected Results</i>
19g Support of application control of the GUI			1.10, 4.2.4.2
		NA	
19h BIR signing (by BSP)			1.5, 2.1.7
19h.1	3	To test BIR signing if both BIR signing and Model/template adaptation are supported by the BSP.	If the BSP supports both BIR signing and Model/template adaptation, call BioSPI_Verify with valid parameters, including AdaptedBIR. The presented biometric should "match" the StoredTemplate.
			Return code BioAPI_OK, Result of BioAPI_TRUE and a valid, signed AdaptedBIR.
19i BIR encryption (by BSP)			1.5, 2.1.7
19i.1	3	To test BIR encryption if both BIR encryption and Model/template adaptation are supported by the BSP.	If the BSP supports both BIR encryption and Model/template adaptation, call BioSPI_Verify with valid parameters, including AdaptedBIR. The presented biometric should "match" the StoredTemplate.
			Return code BioAPI_OK, Result of BioAPI_TRUE and a valid AdaptedBIR.
19j BSP client/server comms			1.6, 4.2.4.2
		NA	
20 BioSPI Identify			3.3.4.8
		NA	
20a Set threshold using FRR criteria			3.3.4.8
		NA	
20b Return of continuous scores			1.7
		NA	
20c Return of achieved FRR score			3.3.4.8
		NA	

Feature			Scenario	Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>		<i>Expected Results</i>
20d Support of binning				3.3.4.8
	NA			
20e Return of raw/audit data				3.3.4.8
	NA			
20f Support of application control of the GUI				1.10, 4.2.4.2
	NA			
20g BIR signing (by BSP)				1.5, 2.1.7
	NA			
20h BIR encryption (by BSP)				1.5, 2.1.7
	NA			
20i BSP client/server comms				1.6, 4.2.4.2
	NA			
21 BioSPI Import				3.3.4.9
21.1	3	To test BioSPI_Import with valid parameters.	Call BioSPI_Import with valid parameters.	Return code of BioAPI_OK and a valid ConstructedBIR.
21.2	3	To test BioSPI_Import with invalid parameters.	Call BioSPI_Import with one or more invalid parameters.	Return code other than BioAPI_OK and an invalid ConstructedBIR.
22 BioSPI Set Power Mode				3.3.4.10
	NA			
23 BioSPI db Open				3.3.5.1
23.1	3	To test BioSPI_DbOpen with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbOpen with valid parameters.	Return code BioAPI_OK, a valid DbHandle and a valid Cursor.

Feature

Num Priority Purpose

Scenario

Reference

Expected Results

23.2	3	To test BioSPI_DbOpen with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbOpen with one or more invalid parameters.	Return code other than BioAPI_OK, an invalid DbHandle and an invalid Cursor.
------	---	--	--	--

24 BioSPI db Close

3.3.5.2

24.1	3	To test BioSPI_DbClose with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbClose with valid parameters.	Return code BioAPI_OK.
24.2	3	To test BioSPI_DbClose with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbClose with one or more invalid parameters.	Return code other than BioAPI_OK.

25 BioSPI db Create

3.3.5.3

25.0	3	To test BioSPI_DbCreate existing database protection if supported by the BSP.	If supported by the BSP, call BioSPI_DbCreate with valid parameters, with DbName the name of an existing database.	Return code of BioAPIERR_BSP_DATABASE_ALREADY_EXISTS and an invalid DbHandle.
25.1	3	To test BioSPI_DbCreate with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbCreate with valid parameters.	Return code BioAPI_OK and a valid DbHandle.
25.2	3	To test BioSPI_DbCreate with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbCreate with one or more invalid parameters.	Return code other than BioAPI_OK and a DbHandle of BioAPI_DB_INVALID_HANDLE.

26 BioSPI db Delete

3.3.5.4

26.1	3	To test BioSPI_DbDelete with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbDelete with valid parameters on an existing, closed database.	Return code of BioAPI_OK and the database deleted.
26.2	3	To test that BioSPI_DbDelete does not delete an open database if supported by the BSP.	If supported by the BSP, open a database, then call BioSPI_DbDelete with a valid ModuleHandle and DbName the name of the open database.	Return code BioAPIERR_BSP_DATABASE_IS_OPEN and the database not deleted.
26.3	3	To test BioSPI_DbDelete with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbDelete with one or more invalid parameters.	Return code other than BioAPI_OK and the database not deleted if a valid database name was passed into BioSPI_DbDelete.

27 BioSPI db Set Cursor

3.3.5.5

27.1	3	To test BioSPI_DbSetCursor with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbSetCursor with valid parameters.	Return code of BioAPI_OK and a valid Cursor.
27.2	3	To test BioSPI_DbSetCursor with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbSetCursor with one or more invalid parameters.	Return code other than BioAPI_OK and an invalid Cursor.
27.3	3	To test BioSPI_DbSetCursor "record not found" if supported by the BSP.	If supported by the BSP, call BioSPI_DbSetCursor with valid parameters but KeyValue not in the database.	Return code of BioAPIERR_BSP_RECORD_NOT_FOUND.

Feature		Scenario	Reference
Num	Priority Purpose		Expected Results
28 BioSPI db Free Cursor			3.3.5.6
28.1	3 To test BioSPI_DbFreeCursor with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbFreeCursor with valid parameters.	Return code BioAPI_OK.
28.2	3 To test BioSPI_DbFreeCursor with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbFreeCursor with one or more invalid parameters.	Return code other than BioAPI_OK.
28.3	3 To test BioSPI_DbFreeCursor "invalid cursor" if supported by the BSP.	If supported by the BSP, call BioSPI_DbFreeCursor with a valid ModuleHandle and an invalid Cursor.	Return code BioAPIERR_BSP_CURSOR_IS_INVALID.
29 BioSPI db Store BIR			3.3.5.7
29.1	3 To test BioSPI_DbStoreBIR with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbStoreBIR with valid parameters.	Return code BioAPI_OK.
29.2	3 To test BioSPI_DbStoreBIR with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbStoreBIR with one or more invalid parameters.	Return code other than BioAPI_OK.
30 BioSPI db Get BIR			3.3.5.8
30.1	3 To test BioSPI_DbGetBIR with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbGetBIR with valid parameters.	Return code BioAPI_OK.
30.2	3 To test BioSPI_DbGetBIR with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbGetBIR with one or more invalid parameters.	Return code other than BioAPI_OK.
30.3	3 To test BioSPI_DbGetBIR "record not found" if supported by the BSP.	If supported by the BSP, call BioSPI_DbGetBIR with valid parameters and KeyValue not in the DbHandle database.	Return code BioAPIERR_BSP_RECORD_NOT_FOUND.
31 BioSPI db Get Next BIR			3.3.5.9
31.1	3 To test BioSPI_DbGetNextBIR with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbGetNextBIR with valid parameters.	Return code BioAPI_OK.
31.2	3 To test BioSPI_DbGetNextBIR with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbGetNextBIR with one or more invalid parameters.	Return code other than BioAPI_OK.

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
32 BioSPI db Query BIR				3.3.5.10
32.1	3	To test BioSPI_DbQueryBIR with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbQueryBIR with valid parameters.	Return code BioAPI_OK.
32.2	3	To test BioSPI_DbQueryBIR with invalid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbQueryBIR with one or more invalid parameters.	Return code other than BioAPI_OK.
33 BioSPI db Delete BIR				3.3.5.11
33.1	3	To test BioSPI_DbDeleteBIR with valid parameters if supported by the BSP.	If supported by the BSP, call BioSPI_DbDeleteBIR with valid parameters.	Return code BioAPI_OK.
33.2	3	If supported by the BSP, call BioSPI_DbQueryBIR with invalid parameters.	supported by the BSP, call BioSPI_DbDeleteBIR with one or more invalid parameters.	Return code other than BioAPI_OK.
101 BSP must implement all mandatory functions IAW SPI				A.2
101.1	1	To test that the BSP implements all mandatory functions: BioSPI_ModuleLoad, BioSPI_ModuleUnload, BioSPI_ModuleAttach, BioSPI_ModuleDetach, BioSPI_FreeBIRHandle, BioSPI_GetBIRFromHandle, BioSPI_GetHeaderFromHandle, BioSPI_Enroll and BioSPI_Verify. Identification BSPs must also implement BioSPI_Identify. If the BSP is a Client/Server BSP, these functions can be executed Locally or Remotely: BioSPI_Enroll, BioSPI_Verify, and (if an Identification BSP) BioSPI_Identify.	TBD	TBD
102 BSP accepts all valid input param and return valid outputs				A.2
102.1	1	To test that the BSP accepts all valid inputs to all mandatory functions and implemented optional functions, and returns valid output from all mandatory functions and implemented optional functions.	TBD	TBD
103 Options implemented must be in accordance to spec (as shown in column 2 & 3 of Table)				A.2
103.1	2 & 3	To test that all options implemented are correct.	Call all options (CA Test Cases) in turn, tracking the results.	Each implemented option should succeed according to its Test Cases.
104 BSP provide all registry entries				A.2
104.1	1	To test that the BSP provides all registry entries.	TBD	TBD

Feature				Reference
<i>Num</i>	<i>Priority</i>	<i>Purpose</i>	<i>Scenario</i>	<i>Expected Results</i>
105 BSP has UUID according to data definition				A.2
105.1	1	To test the BSP's UUID	TBD	The UUID conforms to the specification.
106 Conformant data structures -- (Biometric data according to 2.1 & 3.2 including the BIR)				A.2
106.1	1	To test conformance of data structures.	TBD	TBD
107 Registered valid Format Owner and Format Type				A.2
107.1	1	To test that Format Owner and Format Type are registered and valid.	TBD	Format Owner and Format Type are registered and valid.
108 Error handling according to 2.3				A.2
108.1	1	To test that the BSP implements error handling in accordance with 2.3.	TBD	TBD
109 If GUI BSP must provide it				A.2
109.1	1	To test that the BSP provides a GUI for the capture portion of the Verify operation.	TBD	TBD