

# DRAFT INTERNATIONAL STANDARD

## ISO/IEC DIS 30106-2

ISO/IEC JTC 1/SC 37

Secretariat: ANSI

Voting begins on:  
2019-09-23

Voting terminates on:  
2019-12-16

---

---

### Information technology — Object oriented BioAPI —

#### Part 2: Java implementation

*Technologies de l'information — Objet orienté BioAPI —*

*Partie 2: Mise en oeuvre Java*

ICS: 35.240.15

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/295e142a-3dd3-4420-9acd-a48af264e573/iso-iec-fdis-30106-2>

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.



Reference number  
ISO/IEC DIS 30106-2:2019(E)

© ISO/IEC 2019

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/295e142a-3dd3-4420-9acd-a48af264e573/iso-iec-fdis-30106-2>



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2019

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  
Fax: +41 22 749 09 47  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

Foreword .....	xii
Introduction.....	xiii
<b>1</b> <b>Scope</b> .....	<b>1</b>
<b>2</b> <b>Normative References</b> .....	<b>1</b>
<b>3</b> <b>BioAPI Java Package Structure</b> .....	<b>1</b>
<b>3.1</b> <b>Package org.bioapi</b> .....	<b>1</b>
<b>3.1.1</b> <b>Package description</b> .....	<b>1</b>
<b>3.1.2</b> <b>Structure</b> .....	<b>1</b>
<b>3.2</b> <b>Package org.bioapi.data</b> .....	<b>2</b>
<b>3.2.1</b> <b>Package description</b> .....	<b>2</b>
<b>3.2.2</b> <b>Structure</b> .....	<b>2</b>
<b>4</b> <b>Data types and constants</b> .....	<b>3</b>
<b>4.1</b> <b>Class ACBioParameters</b> .....	<b>3</b>
<b>4.1.1</b> <b>Description</b> .....	<b>3</b>
<b>4.1.2</b> <b>Method Summary</b> .....	<b>3</b>
<b>4.1.2.1</b> <b>int[] getChallenge()</b> .....	<b>3</b>
<b>4.1.2.2</b> <b>int[] getInitialBPUIOIndexOutput()</b> .....	<b>3</b>
<b>4.1.2.3</b> <b>int[] getSupremumBPUIOIndexOutput()</b> .....	<b>3</b>
<b>4.2</b> <b>Class BFPListElement</b> .....	<b>3</b>
<b>4.2.1</b> <b>Description</b> .....	<b>3</b>
<b>4.2.2</b> <b>Method Summary</b> .....	<b>3</b>
<b>4.2.2.1</b> <b>UUID getBFPID()</b> .....	<b>3</b>
<b>4.2.2.2</b> <b>UnitCategoryType getUnitCategory()</b> .....	<b>4</b>
<b>4.2.2.3</b> <b>void setBFPID(UUID bfpID)</b> .....	<b>4</b>
<b>4.2.2.4</b> <b>void setUnitCategory(UnitCategoryType unitCategory)</b> .....	<b>4</b>
<b>4.3</b> <b>Class BFPSchema</b> .....	<b>4</b>
<b>4.3.1</b> <b>Description</b> .....	<b>4</b>
<b>4.3.2</b> <b>Method Summary</b> .....	<b>4</b>
<b>4.3.2.1</b> <b>String getBFPDescription()</b> .....	<b>4</b>
<b>4.3.2.2</b> <b>Vector&lt;RegistryID&gt; getBFPSupportedFormats()</b> .....	<b>4</b>
<b>4.3.2.3</b> <b>UUID getBFPUUID()</b> .....	<b>4</b>
<b>4.3.2.4</b> <b>Vector&lt;BiometricType&gt; getFactorsMask()</b> .....	<b>4</b>
<b>4.3.2.5</b> <b>byte[] getFWProperty()</b> .....	<b>5</b>
<b>4.3.2.6</b> <b>UUID getFWPropertyID()</b> .....	<b>5</b>
<b>4.3.2.7</b> <b>String getPath()</b> .....	<b>5</b>
<b>4.3.2.8</b> <b>String getProductVersion()</b> .....	<b>5</b>
<b>4.3.2.9</b> <b>String getSpecVersion()</b> .....	<b>5</b>
<b>4.3.2.10</b> <b>UnitCategoryType getUnitCategory()</b> .....	<b>5</b>
<b>4.3.2.11</b> <b>String getVendor()</b> .....	<b>6</b>
<b>4.4</b> <b>Class BIR</b> .....	<b>6</b>
<b>4.4.1</b> <b>Description</b> .....	<b>6</b>
<b>4.4.2</b> <b>Method Summary</b> .....	<b>6</b>
<b>4.4.2.1</b> <b>void birFromArray(byte[] record)</b> .....	<b>6</b>
<b>4.4.2.2</b> <b>byte[] birToArray()</b> .....	<b>6</b>
<b>4.4.2.3</b> <b>void destroy()</b> .....	<b>6</b>
<b>4.4.2.4</b> <b>BiometricSubtype getBDBBiometricSubtype()</b> .....	<b>6</b>
<b>4.4.2.5</b> <b>BiometricType getBDBBiometricType()</b> .....	<b>7</b>
<b>4.4.2.6</b> <b>byte[] getBDBChallengeResponse()</b> .....	<b>7</b>
<b>4.4.2.7</b> <b>Date getBDBCreationDate()</b> .....	<b>7</b>

4.4.2.8	byte[] getBDBData() .....	7
4.4.2.9	RegistryID getBDBFormat().....	7
4.4.2.10	byte[] getBDBIndex() .....	7
4.4.2.11	ProcessedLevel getBDBProcessedLevel() .....	7
4.4.2.12	Purpose getBDBPurpose() .....	7
4.4.2.13	byte getBDBQuality() .....	8
4.4.2.14	Vector<Date> getBDBValidityPeriod().....	8
4.4.2.15	Date getBIRCreationDate() .....	8
4.4.2.16	byte[] getBIRCreator() .....	8
4.4.2.17	byte[] getBIRIndex() .....	8
4.4.2.18	byte[] getBIRAdditionalData() .....	8
4.4.2.19	Vector<Date> getBIRValidityPeriod() .....	8
4.4.2.20	byte getCBEFFVersion() .....	9
4.4.2.21	RegistryID getPatronFormat() .....	9
4.4.2.22	byte getPatronHeaderVersion().....	9
4.4.2.23	byte[] getSBData() .....	9
4.4.2.24	RegistryID getSBFormat().....	9
4.4.2.25	boolean hasBDBEncryption().....	9
4.4.2.26	boolean hasBDBIntegrity() .....	9
4.4.2.27	boolean isBIRSigned() .....	10
4.4.2.28	boolean isQualitySupported() .....	10
4.4.2.29	boolean isQualityKnown() .....	10
4.4.2.30	void setBDBBiometricSubtype(BiometricSubtype bdbBiometricSubtype) .....	10
4.4.2.31	void setBDBBiometricType(BiometricType bdbBiometricType).....	10
4.4.2.32	void setBDBChallengeResponse(byte bdbChallengeResponse).....	10
4.4.2.33	void setBDBCreationDate(Date bdbCreationDate) .....	10
4.4.2.34	void setBDBEncryption(boolean bdbEncryption).....	10
4.4.2.35	void setBDBFormat(RegistryID bdbFormat).....	10
4.4.2.36	void setBDBData(byte[] bdbData).....	11
4.4.2.37	void setBDBIndex(byte[] bdbIndex) .....	11
4.4.2.38	void setBDBIntegrity(boolean bdbIntegrity).....	11
4.4.2.39	void setBDBQuality(byte bdbQuality) .....	11
4.4.2.40	void setBDBProcessedLevel(ProcessedLevel bdbProcessedLevel) .....	11
4.4.2.41	void setBDBPurpose(Purpose bdbPurpose).....	11
4.4.2.42	void setBDBValidityPeriod(Vector<Date> bdbValidityPeriod).....	11
4.4.2.43	void setBIRCreationDate(Date birCreationDate).....	11
4.4.2.44	void setBIRCreator(byte[] birCreator) .....	12
4.4.2.45	void setBIRIndex(byte[] birIndex) .....	12
4.4.2.46	void setBIRAdditionalData(byte[] birAdditionalData).....	12
4.4.2.47	void setBIRValidityPeriod(Vector<Date> birValidityPeriod) .....	12
4.4.2.48	void setCBEFFVersion(byte cbeffVersion) .....	12
4.4.2.49	void setPatronFormat(RegistryID patronFormat) .....	12
4.4.2.50	void setPatronHeaderVersion(byte patronHeaderVersion) .....	12
4.4.2.51	void setSBData(byte[] sbData).....	13
4.4.2.52	void setSBFormat(RegistryID sbFormat).....	13
4.5	Class BSPSchema .....	13
4.5.1	Description .....	13
4.5.2	Method Summary .....	13
4.5.2.1	UUID getBSPAccessUUID() .....	13
4.5.2.2	String getBSPDescription() .....	13
4.5.2.3	Vector<RegistryID> getBSPSupportedAlgorithms().....	13
4.5.2.4	Vector<RegistryID> getBSPSupportedFormats().....	13
4.5.2.5	Vector<UUID> getBSPSupportedTransformOperation() .....	14
4.5.2.6	UUID getBSPUUID() .....	14
4.5.2.7	int getDefaultCalibrateTimeout().....	14
4.5.2.8	int getDefaultCaptureTimeout().....	14
4.5.2.9	int getDefaultEnrolTimeout() .....	14
4.5.2.10	int getDefaultIdentifyTimeout() .....	14
4.5.2.11	int getDefaultVerifyTimeout() .....	14
4.5.2.12	Vector<BiometricType> getFactorsMask() .....	15

4.5.2.13	byte[] getHostingEndpointIRI() .....	15
4.5.2.14	int getMaxBSPDbSize() .....	15
4.5.2.15	int getMaxIdentify() .....	15
4.5.2.16	int getMaxNumEnrollInstances() .....	15
4.5.2.17	int getMaxAdditionalDataSize() .....	15
4.5.2.18	Vector<BSPSchemaOperations> getOperations() .....	16
4.5.2.19	Vector<BSPSchemaOptions> getOptions() .....	16
4.5.2.20	String getPath() .....	16
4.5.2.21	int getAdditionalDataPolicy() .....	16
4.5.2.22	String getProductVersion() .....	16
4.5.2.23	String getSpecVersion() .....	16
4.5.2.24	String getVendor() .....	16
4.6	Class Candidate .....	17
4.6.1	Description .....	17
4.6.2	Method Summary .....	17
4.6.2.1	int getFMRAchieved() .....	17
4.6.2.2	UUID getKey() .....	17
4.6.2.3	void setFMRAchieved(int fmrAchieved) .....	17
4.6.2.4	void setKey(UUID key) .....	17
4.7	Class DataTypes .....	17
4.7.1	Description .....	17
4.7.2	Enumerations .....	17
4.7.2.1	BiometricSubtype .....	18
4.7.2.2	BiometricType .....	18
4.7.2.3	BIRDatabaseAccess .....	18
4.7.2.4	BSPSchemaOperations .....	19
4.7.2.5	BSPSchemaOptions .....	19
4.7.2.6	EventKind .....	20
4.7.2.7	Facility .....	20
4.7.2.8	GUIEnrolType .....	20
4.7.2.9	GUIMoment .....	20
4.7.2.10	GUIOperation .....	20
4.7.2.11	GUIResponse .....	20
4.7.2.12	GUISuboperation .....	21
4.7.2.13	ProcessedLevel .....	21
4.7.2.14	Purpose .....	21
4.7.2.15	ResultOptions .....	21
4.7.2.16	SecurityOptionsType .....	22
4.7.2.17	UnitCategoryType .....	22
4.7.2.18	UnitIndicatorStatus .....	22
4.7.2.19	UnitPowerMode .....	22
4.8	Class Date .....	22
4.8.1	Description .....	22
4.8.2	Method Summary .....	22
4.8.2.1	int getDayOfMonth() .....	22
4.8.2.2	int getHour() .....	23
4.8.2.3	int getMinute() .....	23
4.8.2.4	int getMonth() .....	23
4.8.2.5	int getSecond() .....	23
4.8.2.6	int getYear() .....	23
4.8.2.7	boolean isLowerOrEqual (int day, int month, int year) .....	23
4.8.2.8	boolean isLowerOrEqual (int day, int month, int year, int hour, int minute, int second) .....	23
4.8.2.9	boolean isLowerOrEqual (Date date) .....	23
4.8.2.10	boolean isHigherOrEqual (int day, int month, int year) .....	23
4.8.2.11	boolean isHigherOrEqual (int day, int month, int year, int hour, int minute, int second) .....	23
4.8.2.12	boolean isHigherOrEqual (Date date) .....	23
4.8.2.13	void setDayOfMonth(int dayOfMonth) .....	24
4.8.2.14	void setHour(int hour) .....	24

4.8.2.15	void setMinute(int minute).....	24
4.8.2.16	void setMonth(int month) .....	24
4.8.2.17	void setSecond(int second).....	24
4.8.2.18	void setYear(int year) .....	25
4.9	Class FrameworkSchema .....	25
4.9.1	Description .....	25
4.9.2	Method Summary.....	25
4.9.2.1	UUID getFrameworkUUID() .....	25
4.9.2.2	String getFWDescription() .....	25
4.9.2.3	byte[] getFWProperty().....	25
4.9.2.4	UUID getFWPropertyID() .....	25
4.9.2.5	byte[] getPath() .....	25
4.9.2.6	String getProductVersion().....	26
4.9.2.7	String getSpecVersion().....	26
4.9.2.8	String getVendor() .....	26
4.10	Class GUIBitmap.....	26
4.10.1	Description .....	26
4.10.2	Method Summary.....	26
4.10.2.1	BIRSubtype getBIRSubtype() .....	26
4.10.2.2	int getHeight() .....	26
4.10.2.3	byte[][] getPixel() .....	26
4.10.2.4	int getWidth().....	27
4.11	Class IdentifyPopulation .....	27
4.11.1	Description .....	27
4.11.2	Method Summary.....	27
4.11.2.1	void addMember(PopulationMember member) .....	27
4.11.2.2	void destroy() .....	27
4.11.2.3	Vector<PopulationMember> getIdentifyPopulation() .....	27
4.11.2.4	bool isBound() .....	27
4.11.2.5	void unbind() .....	28
4.12	Class PopulationMember .....	28
4.12.1	Description .....	28
4.12.2	Method Summary.....	28
4.12.2.1	UUID getKey().....	28
4.12.2.2	BIR getTemplate().....	28
4.12.2.3	void setKey(UUID key) .....	28
4.12.2.4	void setTemplate(BIR template).....	28
4.13	Class RegistryID .....	28
4.13.1	Description .....	28
4.13.2	Method Summary.....	29
4.13.2.1	short getOwner().....	29
4.13.2.2	short getType() .....	29
4.13.2.3	void setOwner(short owner).....	29
4.13.2.4	void setType(short type).....	29
4.14	Class SecurityProfileType .....	29
4.14.1	Description .....	29
4.14.2	Method Summary.....	29
4.14.2.1	Vector<BSPSchemaOptions> acBioOption() .....	29
4.14.2.2	byte[] getENCInfo() .....	30
4.14.2.3	byte[] getHASHAlgForACBio() .....	30
4.14.2.4	byte[] getMACAlgForACBio() .....	30
4.14.2.5	byte[] getMACInfo() .....	30
4.14.2.6	byte[] getSIGNAlg() .....	30
4.14.2.7	byte[] getSIGNAlgForACBio().....	30
4.14.2.8	Vector<SecurityOptionsType> getSupportedSecurityOptions() .....	30
4.15	Class UnitList.....	31
4.15.1	Description .....	31
4.15.2	Method Summary.....	31
4.15.2.1	void add(UnitListElement unitListElement).....	31
4.15.2.2	int getUnitID (UnitCategoryType unitCategoryType) .....	31

4.16	Class UnitListElement.....	31
4.16.1	Description .....	31
4.16.2	Method Summary .....	31
4.16.2.1	UnitCategoryType getUnitCategory() .....	31
4.16.2.2	int getUnitID().....	31
4.16.2.3	void setUnitCategory(UnitCategoryType unitCategory) .....	31
4.16.2.4	void setUnitID(int unitID) .....	32
4.17	Class UnitSchema .....	32
4.17.1	Description .....	32
4.17.2	Method Summary .....	32
4.17.2.1	UUID getBSPUUID().....	32
4.17.2.2	String getFirmwareVersion().....	32
4.17.2.3	String getHardwareSerialNumber() .....	32
4.17.2.4	String getHardwareVersion() .....	32
4.17.2.5	int getMaxBSPDbSize().....	32
4.17.2.6	int getMaxIdentify() .....	33
4.17.2.7	Vector<SecurityProfileType> getSecurityProfile().....	33
4.17.2.8	String getSoftwareVersion().....	33
4.17.2.9	Vector<EventKind> getSupportedEvents() .....	33
4.17.2.10	UnitCategoryType getUnitCategory().....	33
4.17.2.11	int getUnitID().....	33
4.17.2.12	UUID getUnitManagerUUID() .....	33
4.17.2.13	UUID getUnitProperties() .....	34
4.17.2.14	byte[] getUnitProperty() .....	34
4.17.2.15	UUID getUnitPropertyID() .....	34
4.17.2.16	String getVendorInformation() .....	34
4.17.2.17	boolean isAuthenticatedHardware() .....	34
4.17.2.18	void setBSPUUID(UUID bspUUID) .....	34
4.17.2.19	void setUnitID(int unitID) .....	34
4.18	Class UUID .....	35
4.18.1	Description .....	35
5	Object oriented interfaces for supporting BioAPI_Units .....	36
5.1	Introduction.....	36
5.2	Interface Archive .....	36
5.2.1	Description .....	36
5.2.2	Method Summary .....	36
5.2.2.1	void closeDatabase (int unitID).....	36
5.2.2.2	void deleteBIR (int unitID, UUID key) .....	36
5.2.2.3	BIR getSingleBIR (int unitID, UUID key, Vector<ResultOptions> resultOptions).....	36
5.2.2.4	Vector<UUID> listUUIDs (int unitID) .....	37
5.2.2.5	void openDatabase (int unitID, byte[] databaseID, BIRDatabaseAccess access) .....	37
5.2.2.6	UUID storeBIR (int unitID, BIR biometricReference) .....	37
5.2.2.7	UUID storeBIR (int unitID, BIR biometricReference, byte[] auxiliaryData) .....	38
5.2.2.8	void storeBIR (int unitID, BIR biometricReference, UUID key) .....	38
5.2.2.9	void storeBIR (int unitID, BIR biometricReference, byte[] auxiliaryData, UUID key) .....	38
5.2.2.10	IdentifyPopulation newIdentifyPopulation (int unitID) .....	39
5.2.2.11	IdentifyPopulation newIdentifyPopulation (int unitID, byte[] query) .....	39
5.2.2.12	IdentifyPopulation newIdentifyPopulation (int unitID, Vector<UUID> uuidList) .....	39
5.3	Interface Comparison.....	39
5.3.1	Description .....	39
5.3.2	Method Summary .....	40
5.3.2.1	Vector<Candidate> identify (int unitID, int maxFMRrequested, BIR processedBIR, boolean binning, int maxResults, int timeout).....	40



5.3.2.2	Vector<Candidate> identify (int unitID, int maxFMRrequested, BIR processedBIR, Vector<BIR> auxiliaryBIRs, boolean binning, int maxResults, int timeout).....	40
5.3.2.3	void presetIdentifyPopulation (int unitID, IdentifyPopulation population).....	40
5.3.2.4	boolean verify (int unitID, int maxFMRrequested, BIR processedBIR, BIR referenceTemplate, Vector<ResultOptions> options) .....	41
5.3.2.5	boolean verify (int unitID, int maxFMRrequested, BIR processedBIR, BIR referenceTemplate, Vector<BIR> auxiliaryBIRs, Vector<ResultOptions> options).....	41
5.3.2.6	BIR getAdaptedBIR(int unitID) .....	41
5.3.2.7	int getFMRAchieved(int unitID).....	42
5.4	Interface Processing .....	42
5.4.1	Description .....	42
5.4.2	Method Summary.....	42
5.4.2.1	BIR createTemplate ((int unitID, BIR capturedBIR, BIR referenceTemplate, RegistryID outputFormat, byte[] additionalData) .....	42
5.4.2.2	BIR createTemplate (int unitID, Vector<BIR> capturedBIRs, BIR referenceTemplate, RegistryID outputFormat, byte[] additionalData) .....	42
5.4.2.3	BIR createTemplate ((int unitID, BIR capturedBIR, BIR referenceTemplate, Vector<BIR> auxBIRs, RegistryID outputFormat, byte[] additionalData) .....	42
5.4.2.4	BIR createTemplate (int unitID, Vector<BIR> capturedBIRs, BIR referenceTemplate, Vector<BIR> auxBIRs, RegistryID outputFormat, byte[] additionalData).....	42
5.4.2.5	BIR process (int unitID, BIR capturedBIR, RegistryID outputFormat) .....	43
5.4.2.6	BIR process (int unitID, BIR captureBIR, Vector<BIR> auxiliaryBIRs, RegistryID outputFormat).....	43
5.4.2.7	byte AnalyseQuality (int unitID, BIR capturedBIR) .....	43
5.5	Interface Sensor.....	43
5.5.1	Description .....	43
5.5.2	Method Summary.....	44
5.5.2.1	void calibrate (int unitID, int timeout).....	44
5.5.2.2	BIR capture (int unitID, Vector<Purpose> purpose, BiometricSubtype biometricSubtype, RegistryID outputFormat, int timeout, Vector<ResultOptions> resultOptions).....	44
5.5.2.3	UnitIndicatorStatus getIndicatorStatus(int unitID).....	44
5.5.2.4	void setIndicatorStatus (int unitID, UnitIndicatorStatus indicatorStatus).....	44
6	BFP level.....	46
6.1	Interface BFP.....	46
6.1.1	Description .....	46
6.1.2	Imported Interfaces.....	46
6.1.3	Method Summary.....	46
6.1.3.1	void bfpLoad (BFPEventListener bfpEventListener, BFPGUIProgressListener bfpGUIProgressListener) .....	46
6.1.3.2	byte[] controlUnit (int unitID, int controlCode, byte[] inputData) .....	47
6.1.3.3	byte[] getACBioInstance(int unitID).....	47
6.1.3.4	byte[] getAuxiliaryData(int unitID) .....	47
6.1.3.5	BFPSchema getBFPSchema().....	48
6.1.3.6	Vector<UnitSchema> queryUnits () .....	48
6.1.3.7	void setPowerMode (int unitID, UnitPowerMode powerMode) .....	48
7	BSP level.....	49
7.1	Interface BSP.....	49
7.1.1	Description .....	49
7.1.2	Imported Interfaces.....	49
7.1.3	Method Summary.....	49
7.1.3.1	void bspLoad (BSPEventListener bspEventListener, BFPEventListener bfpEventListener, BFPGUIProgressListener bfpGUIProgressListener, BFPEnumerationListener bfpEnumerationListener) .....	49
7.1.3.2	void bspUnload () .....	50
7.1.3.3	byte checkQuality (BIR inputBIR, RegistryID qualityAlgorithmID).....	50



7.1.3.4	byte[] controlUnit (int unitID, int controlCode, byte[] inputData).....	50
7.1.3.5	UUID enrol (UnitList unitList, int numberOfPresentations, int numberOfAttempts, int numberOfTransactions, int qualityThreshold, int maxFmrRequested, int timeout, Vector<Purpose> purposes, BiometricSubtype biometricSubtype, RegistryID outputFormat, byte[] additionalData) .....	51
7.1.3.6	UUID enrol (UnitList unitList, int numberOfPresentations, int numberOfAttempts, int numberOfTransactions, int qualityThreshold, int maxFmrRequested, int maxFmrRequestedForUpdate, UUID referenceId, int timeout, Vector<Purpose> purposes, BiometricSubtype biometricSubtype, RegistryID outputFormat, byte[] additionalData) .....	51
7.1.3.7	UUID enrol (UnitList unitList, Vector<BIR> capturedBirs, int maxFmrRequested, int timeout, Vector<Purpose> purposes, BiometricSubtype biometricSubtype, RegistryID outputFormat, byte[] additionalData) .....	51
7.1.3.8	UUID enrol (UnitList unitList, Vector<BIR> capturedBirs, int maxFmrRequested, int maxFmrRequestedForUpdate, UUID referenceId, int timeout, Vector<Purpose> purposes, BiometricSubtype biometricSubtype, RegistryID outputFormat, byte[] additionalData) .....	51
7.1.3.9	byte[] getACBiInstance(int unitId).....	52
7.1.3.10	byte[] getAdditionalData(int unitID).....	52
7.1.3.11	BSPSchema getBSPSchema().....	52
7.1.3.12	Vector<Candidate> identifyAggregated (UnitList unitList, int maxFMRrequested, BiometricSubtype biometricSubtype, boolean binning, int maxResults, int timeout, Vector<ResultOptions> options).....	53
7.1.3.13	Vector<Candidate> identifyAggregated (UnitList unitList, BIR inputBIR, int maxFMRrequested, BiometricSubtype biometricSubtype, boolean binning, int maxResults, int timeout, Vector<ResultOptions> .....	53
7.1.3.14	Vector<Candidate> identifyAggregated (UnitList unitList, Vector<BIR> inputBIRs, int maxFMRrequested, BiometricSubtype biometricSubtype, boolean binning, int maxResults, int timeout, Vector<ResultOptions> options) .....	53
7.1.3.15	Vector<BFPListElement> queryBFPs ().....	54
7.1.3.16	Vector<BFPListElement> queryBFPs (Vector<UnitCategoryType> unitCategories).....	54
7.1.3.17	Vector<UnitSchema> queryUnits ().....	54
7.1.3.18	Vector<UnitSchema> queryUnits (Vector<UnitCategoryType> unitCategories).....	54
7.1.3.19	void setPowerMode (int unitID, UnitPowerMode powerMode).....	54
7.1.3.20	void subscribeToGUIEvents (GUISelectEventListener guiSelectEventListener, GUIStateEventListener guiStateEventListener, GUIProgressEventListener guiProgressEventListener) .....	54
7.1.3.21	void unsubscribeFromGUIEvents ().....	55
7.1.3.22	boolean verifyAggregated (UnitList unitList, int maxFMRrequested, BIR referenceTemplate, BiometricSubtype subtype, int timeout, Vector<ResultOptions> options) .....	55
7.1.3.23	boolean verifyAggregated (UnitList unitList, int maxFMRrequested, UUID referenceKey, BiometricSubtype subtype, int timeout, Vector<ResultOptions> options) .....	55
7.1.3.24	boolean verifyAggregated (UnitList unitList, int maxFMRrequested, BIR inputBIR, BIR referenceTemplate, BiometricSubtype subtype, int Timeout, Vector<ResultOptions> options).....	55
7.1.3.25	boolean verifyAggregated (UnitList unitList, int maxFMRrequested, BIR inputBIR, UUID referenceKey, BiometricSubtype subtype, int timeout, Vector<ResultOptions> options) .....	55
8	Framework Level .....	57
8.1	Interface ComponentRegistry .....	57
8.1.1	Description .....	57
8.1.2	Method Summary .....	57

8.1.2.1	void installBFP (BFPSchema bfpSchema, boolean update).....	57
8.1.2.2	void installBSP (BSPSchema bspSchema, boolean update).....	57
8.1.2.3	void uninstallBFP (UUID bfpUUID) .....	57
8.1.2.4	void uninstallBSP (UUID bspUUID) .....	58
8.2	Interface Framework.....	58
8.2.1	Description .....	58
8.2.2	Inherited interfaces.....	58
8.2.3	Method Summary.....	58
8.2.3.1	void bspLoad (UUID bspID, BFPEventListener bfpEventListener, BFPGUIProgressListener bfpGUIProgressListener, BFPEnumerationListener bfpEnumerationListener, String context) .....	58
8.2.3.2	void bspUnload (UUID bspID, String context) .....	59
8.2.3.3	void enableEventNotifications (UUID bspID, Vector<EventKind> events) .....	60
8.2.3.4	Vector<BFPSchema> enumBFPs () .....	60
8.2.3.5	Vector<BSPSchema> enumBSPs () .....	61
8.2.3.6	FrameworkSchema getFrameworkSchema().....	61
8.2.3.7	void init (String version) .....	61
8.2.3.8	Vector<BFPListElement> queryBFPs (UUID bspUUID) .....	61
8.2.3.9	Vector<UnitSchema> queryUnits (UUID bspUUID).....	62
8.2.3.10	void terminate ().....	62
9	Application interaction .....	63
9.1	class BioAPIException extends Exception .....	63
9.1.1	Description .....	63
9.1.2	Constructor Summary .....	63
9.1.2.1	public BioAPIException (Facility source, int code) .....	63
9.1.2.2	public BioAPIException (Facility source, int code, String message) .....	63
9.1.2.3	public BioAPIException (Facility source, int code, String message, Exception cause) .....	63
9.1.2.4	public BioAPIException (Facility source, int code, Exception cause).....	63
9.1.3	Method Summary.....	64
9.1.3.1	public int getErrorCode () .....	64
9.1.3.2	public Facility getSource ().....	64
9.2	GUI callback functions .....	64
9.2.1	Description .....	64
9.2.2	Callback interface specification .....	64
9.2.2.1	Interface GUIProgressEventListener .....	64
9.2.2.2	Interface GUISelectEventListener .....	65
9.2.2.3	Interface GUIStateEventListener.....	66
10	BSP Interaction .....	68
10.1	Interface BSPEventListener.....	68
10.1.1	Method Summary.....	68
10.1.1.1	boolean bspEventCallback (UUID bfpUUID, int unitID, UnitSchema unitSchema, EventKind event).....	68
11	BFP Interaction .....	69
11.1	Interface BFPEnumerationListener.....	69
11.1.1	Method Summary.....	69
11.1.1.1	Vector<BFPSchema> bfpEnumerationCallback() .....	69
11.2	Interface BFPEventListener .....	69
11.2.1	Method Summary.....	69
11.2.1.1	boolean bfpEventCallback (UUID bfpUUID, int unitID, UnitSchema unitSchema, EventKind event).....	69
11.3	Interface BFPGUIProgressEventListener .....	69
11.3.1	Method Summary.....	69
11.3.1.1	void bfpGUIProgressEventCallback (int unitID, String context, Vector<GUIBitmap> bitmaps, byte response) .....	69
Annex A (informative)	Java Requirements .....	71
Annex B (informative)	Calling Sequence Examples and Sample Code.....	72

B.1	Reference Implementation.....	72
B.2	API Architecture .....	72

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/295e142a-3dd3-4420-9acd-a48af264e573/iso-iec-fdis-30106-2>

## 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 30106-2 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 Java language is specified. Java 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 multiple platforms.

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

**NOTE** Although the best practices of Java programming states that variables should be written in smallcase letters, in the case of symbols, such as BSP or BFPs, it has been kept as uppercase letters.

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/295e142a-3dd3-4420-9acd-a48af264e573/iso-iec-fdis-30106-2>