



# Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0

Committee Specification Draft ~~0304~~ /  
Public Review Draft ~~0203~~

~~23 February~~03 August 2011

## Specification URIs:

### This ~~version~~:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd03/biasprofile-v1.0-csprd03.doc>  
(Authoritative)  
<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd03/biasprofile-v1.0-csprd03.html>  
<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd03/biasprofile-v1.0-csprd03.pdf>~~Version~~

### Previous ~~version~~:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd02/biasprofile-v1.0-csprd02.doc>  
(Authoritative) ~~<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd02/biasprofile-v1.0-csprd02.pdf>~~  
<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd02/biasprofile-v1.0-csprd02.html>

~~<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd02/biasprofile-v1.0-csprd02.pdf>~~ ~~Previous~~  
~~Version:~~  
~~(Authoritative)~~

### Latest ~~V~~ersion:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0.doc> (Authoritative)  
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0.html>  
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0.pdf>

## Technical Committee:

OASIS Biometric Identity Assurance Services (BIAS) Integration TC

### Chair(s):

Cathy Tilton ([cathy.tilton@daon.com](mailto:cathy.tilton@daon.com), ~~Daon~~), Daon

### Editors:

**Kevin Mangold** ([kevin.mangold@nist.gov](mailto:kevin.mangold@nist.gov)) ~~Editor(s):~~

, NIST

Matthew Swayze ([matthew.swayze@daon.com](mailto:matthew.swayze@daon.com)), Daon

Cathy Tilton ([cathy.tilton@daon.com](mailto:cathy.tilton@daon.com), ~~Daon~~), Daon

### Additional Work Product artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schema: [soap-profile/v1.0/csprd03/cbeff.xsd](#)
- WSDL: [soap-profile/v1.0/csprd03/BIAS.wsdl](#)

#### Related ~~W~~ork:

This specification is related to:

- ANSI INCITS 442-2010~~8~~, Biometric Identity Assurance Services (BIAS)

#### Declared XML ~~Namespace(s):~~namespaces:

<http://docs.oasis-open.org/bias/ns/bias-1.0/>

<http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/>

#### Abstract:

This document specifies a SOAP profile that implements the BIAS abstract operations specified in INCITS 442 as SOAP messages.

#### Status:

This document was last revised or approved by ~~the~~ OASIS Biometric Identity Assurance Services (BIAS) Integration TC on the above date. The level of approval is also listed above. Check the “Latest ~~V~~ersion” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <http://www.oasis-open.org/committees/bias/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/bias/ipr.php>).

#### Citation ~~F~~ormat:

When referencing this specification the following citation format should be used:

~~[BIAS SOAP PROFILE]~~—~~BIASPROFILE~~

*Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0.*

~~23 February~~03 August 2011. OASIS Committee Specification ~~Draft 04~~ / Public Review Draft ~~03~~.  
<http://docs.oasis-open.org/bias/soap-profile/v1.0/csprd03/biasprofile-v1.0-csprd03.html>.

---

## Notices

Copyright © OASIS~~© 2009- Open~~ 2011. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full ~~Policy~~Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS" and "BIAS" are trademarks of ~~OASIS~~OASIS<sub>1</sub>, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

---

# Table of Contents

1	Introduction .....	8
1.1	Purpose/Scope .....	8
1.2	Overview .....	8
1.3	Background.....	8
1.4	Relationship to Other Standards .....	9
1.5	Terminology .....	9
1.6	References.....	10
1.6.1	Normative References.....	10
1.6.2	Non-Normative References.....	11
2	Design Concepts and Architecture (non-normative) .....	13
2.1	Philosophy .....	13
2.2	Context.....	13
2.3	Architecture.....	13
3	Data dictionary.....	16
3.1	Documentation Conventions.....	16
3.2	Common Elements .....	17
3.2.1	ApplicationIdentifier .....	17
3.2.2	ApplicationUserIdentifier .....	17
3.2.3	BaseBIRType .....	17
3.2.4	BIASBiometricDataType.....	17
3.2.5	BIASFaultCode.....	18
3.2.6	BIASFaultDetail .....	18
3.2.7	BIASIdentity.....	19
3.2.8	BIASIDType.....	19
3.2.9	BinaryBIR .....	19
3.2.10	BiographicDataItem Type.....	20
3.2.11	BiographicDataSetType .....	20
3.2.12	BiographicDataType .....	21
3.2.13	BiometricDataElementType.....	21
3.2.14	BiometricDataListType .....	22
3.2.15	CandidateListResultType .....	22
3.2.16	CandidateListType .....	22
3.2.17	CandidateType .....	23
3.2.18	CapabilityListType .....	23
3.2.19	CapabilityName .....	23
3.2.20	CapabilityType.....	24
3.2.21	CBEFF_BIR_ListType.....	24
3.2.22	CBEFF_BIR_Type.....	24
3.2.23	Classification .....	25
3.2.24	ClassificationAlgorithmType .....	25
3.2.25	ClassificationData.....	26
3.2.26	EncounterListType .....	26
3.2.27	FusionDecision.....	26

3.2.28	FusionInformationListType .....	26
3.2.29	FusionInformationType.....	26
3.2.30	FusionResult.....	27
3.2.31	FusionScore .....	27
3.2.32	GenericRequestParameters.....	27
3.2.33	IdentifySubjectResultType.....	27
3.2.34	InformationType .....	28
3.2.35	ListFilterType .....	28
3.2.36	MatchType.....	28
3.2.37	ProcessingOptionsType .....	28
3.2.38	ProductID.....	28
3.2.39	QualityData.....	29
3.2.40	ResponseStatus .....	29
3.2.41	ReturnCode .....	29
3.2.42	Score .....	29
3.2.43	TokenResultType .....	30
3.2.44	TokenType.....	30
3.2.45	URI_BIR .....	30
3.2.46	VendorIdentifier .....	30
3.2.47	Version .....	30
3.2.48	VersionType .....	31
3.2.49	XML_BIR .....	31
4	BIAS Messages .....	32
4.1	Primitive Operations .....	32
4.1.1	AddSubjectToGallery .....	32
4.1.2	CheckQuality .....	33
4.1.3	ClassifyBiometricData .....	35
4.1.4	CreateSubject.....	37
4.1.5	DeleteBiographicData .....	38
4.1.6	DeleteBiometricData .....	39
4.1.7	DeleteSubject .....	40
4.1.8	DeleteSubjectFromGallery .....	41
4.1.9	GetIdentifySubjectResults .....	43
4.1.10	IdentifySubject.....	45
4.1.11	ListBiographicData .....	47
4.1.12	ListBiometricData .....	50
4.1.13	PerformFusion.....	54
4.1.14	QueryCapabilities.....	56
4.1.15	RetrieveBiographicInformation.....	57
4.1.16	RetrieveBiometricInformation.....	59
4.1.17	SetBiographicData .....	61
4.1.18	SetBiometricData .....	63
4.1.19	TransformBiometricData .....	64
4.1.20	UpdateBiographicData .....	66
4.1.21	UpdateBiometricData .....	68

4.1.22	VerifySubject .....	69
4.2	Aggregate Operations .....	71
4.2.1	Enroll .....	71
4.2.2	GetEnrollResults .....	72
4.2.3	GetIdentifyResults .....	74
4.2.4	GetVerifyResults .....	75
4.2.5	Identify .....	77
4.2.6	RetrieveInformation .....	78
4.2.7	Verify .....	79
5	Message structure and rules .....	82
5.1	Purpose and constraints .....	82
5.2	Message requirements .....	83
5.3	Handling binary data .....	84
5.3.1	Base64 encoding .....	84
5.3.2	Use of XOP .....	84
5.4	Discovery .....	85
5.5	Identifying operations .....	85
5.5.1	Operation name element .....	85
5.5.2	WS-Addressing Action .....	86
5.6	Security .....	87
5.6.1	Use of SSL 3.0 or TLS 1.0 .....	87
5.6.2	Data Origin Authentication .....	87
5.6.3	Message Integrity .....	87
5.6.4	Message Confidentiality .....	87
5.6.5	CBEFF BIR security features .....	87
5.6.6	Security Considerations .....	88
5.6.7	Security of Stored Data .....	88
5.6.8	Key Management .....	88
5.7	Use with other WS* standards .....	88
5.8	Tailoring .....	88
6.	Error handling .....	90
6.1	BIAS operation return codes .....	90
6.2	SOAP fault codes .....	90
7.	Conformance .....	91
Annex A.	XML Schema .....	92
Annex B:	BIAS Patron format specification .....	174
B.1	Patron .....	174
B.2	Patron identifier .....	174
B.3	Patron format name .....	174
B.4	Patron format identifier .....	174
B.5	ASN.1 object identifier for this patron format .....	174
B.6	Domain of use .....	174
B.7	Version identifier .....	174
B.8	CBEFF version .....	174
B.9	General .....	175

B.10	Specification .....	175
B.11	Element <BIR> .....	176
B.11.1	Syntax .....	176
B.11.2	Semantics .....	176
B.12	Element <Version> .....	177
B.12.1	Syntax .....	177
B.12.2	Semantics .....	177
B.13	Element <CBEFFVersion> .....	178
B.13.1	Syntax .....	178
B.13.2	Semantics .....	178
B.14	Element <BIRInfo> .....	178
B.14.1	Syntax .....	178
B.14.2	Semantics .....	179
B.15	Element <BDBInfo> .....	180
B.15.1	Syntax .....	180
B.15.2	Semantics .....	182
B.16	Element <SBInfo> .....	186
B.16.1	Syntax .....	186
B.16.2	Semantics .....	187
B.17	Representation of Integers .....	187
B.18	Representation of Octet Strings .....	187
B.19	Representation of Date and Time of the Day .....	188
B.20	Representation of Universally Unique Identifiers .....	189
B.21	Patron format conformance statement .....	189
B.21.1	Identifying information .....	189
B.21.2	ISO/IEC 19785-1:2006/Amd 1:2010 to Patron Format Mapping .....	189
B.22	XML schema of the BIAS patron format .....	191
B.23	Sample BIR encoding .....	194
Annex C.	Use Cases (non-normative) .....	196
C.1	Verification Use Case .....	196
C.2	Asynchronous Verification Use Case .....	197
C.3	Primitive Verification Use Case .....	198
C.4	Identification Use Case .....	199
C.5	Biometric Enrollment Use Case .....	200
C.6	Primitive Enrollment Use Case .....	201
Annex D.	Samples (non-normative) .....	202
D.1	Create Subject Request/Response Example .....	202
D.2	Set Biographic Data Request/Response Example .....	204
D.3	Set Biometric Data Request/Response Example .....	205
Annex E.	Acknowledgements .....	208
Annex F.	Revision History .....	209

---

# 1 Introduction

## 1.1 Purpose/Scope

This Organization for the Advancement of Structured Information Standards (OASIS) Biometric Identity Assurance Services (BIAS) profile specifies how to use the eXtensible Markup Language (XML) [XML10] defined in ANSI INCITS 442-2010<sup>8</sup> – Biometric Identity Assurance Services [INCITS-BIAS] to invoke Simple Object Access Protocol (SOAP) -based services that implement BIAS operations. These SOAP-based services enable an application to invoke biometric identity assurance operations remotely in a Services Oriented Architecture (SOA) infrastructure.

Not included in the scope of BIAS is the incorporation of biometric authentication as an integral component of an authentication or security protocol. (However, BIAS services may be leveraged to implement biometric authentication in the future.)

## 1.2 Overview

In addition to this introduction, this standard includes the following:

- Clause 2 presents the design concepts and architecture for invoking SOAP-based services that implement BIAS operations.
- Clause 3 presents the namespaces necessary to implement this profile, INCITS BIAS data elements, and identifies relationships to external data definitions.
- Clause 4 specifies the content of the BIAS messages.
- Clause 5 presents the BIAS message structure, as well as rules and considerations for its application.
- Clause 6 presents information on error handling.
- Clause 7 specifies conformance requirements.
- Annexes include the OASIS BIAS XML schema/sample Web Service Definition Language (WSDL), [BIAS CBEFF Patron Format](#), use cases, sample code, acknowledgements, and the revision history of this profile.

## 1.3 Background

In late 2005/early 2006, a gap was identified in the existing biometric standards portfolio with respect to biometric services. The Biometric Identity Assurance Services standard proposal was for a collaborative effort between government and private industry to provide a services-based framework for delivering identity assurance capabilities, allowing for platform and application independence. This standard proposal required the attention of two major technical disciplines: biometrics and service architectures. The expertise of both disciplines was required to ensure the standard was technically sound, market relevant, and achieved widespread adoption. The International Committee for Information Technology Standards (INCITS) M1 provided the standards leadership relevant to biometrics, defining the “taxonomy” of biometric operations and data elements. OASIS provided the standards leadership relevant to service architectures with an initial focus on web services, defining the schema and SOAP messaging.

The driving requirements of the BIAS standard proposal were to provide the ability to remotely invoke biometric operations across an SOA infrastructure; to provide business level operations without constraining the application/business logic that implements those operations; to be as generic as possible – technology, framework, & application domain independent; and to provide basic capabilities that can be used to construct higher level, aggregate/composite operations.



## 1.4 Relationship to Other Standards

This OASIS BIAS profile comprises a companion standard to ANSI INCITS 442-2010<sup>8</sup> – Biometric Identity Assurance Services, which defines the BIAS requirements and taxonomy, specifying the identity assurance operations and the associated data elements. This OASIS BIAS profile specifies the design concepts and architecture, data model and data dictionary, message structure and rules, and error handling necessary to invoke SOAP-based services that implement BIAS operations.

Together, the BIAS standard and the BIAS profile provide an open framework for deploying and remotely invoking biometric-based identity assurance capabilities that can be readily accessed across an SOA infrastructure.

This relationship allows the leveraging of the biometrics and web services expertise of the two standards development organizations. Existing standards are available in both domains and many of these standards will provide the foundation and underlying capabilities upon which the biometric services depend.

## 1.5 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

The following additional terms and definitions are used:

Note: The terms and definitions specified in INCITS (InterNational Committee for Information Technology Standards) (Project 1823-D) also apply to this Standard.

BIAS operation and data element names are not defined here, but in their respective sections.

### **BIAS**

Biometric Identity Assurance Services

### **BIR**

Biometric Information Record

### **ESB**

Enterprise Service Bus

### **HTTP**

HyperText Transfer Protocol

### **HTTPS**

HyperText Transfer Protocol over SSL or HTTP Secure

### **IRI**

Internationalized Resource Identifier

### **SOA**

Service-Oriented Architecture

### **SOAP**

Simple Object Access Protocol

### **SSL**

Secure Sockets Layer

### **TLS**

Transport Layer Security

### **UDDI**

Universal Description, Discovery, and Integration

### **URI**

Uniform Resource Identifier

### **VPN**

Virtual Private Network

### **WSDL**

Web Services Description Language

### **WSS**

Web Services Security

## XML

eXtensible Markup Language

## CBEFF

Common Biometric Exchange Formats Framework - data elements and BIR formats specified in ISO/IEC 19785-1

## BIAS implementation

software entity that is capable of creating, processing, sending, and receiving BIAS messages

## BIAS endpoint

runtime entity, identified by an endpoint URI/IRI, capable of sending and receiving BIAS messages, and containing a running BIAS implementation

## BIAS message

message that can be sent from a BIAS endpoint to another BIAS endpoint through a BIAS link channel

## BIAS request message

BIAS message conveying a request for an action to be performed by the receiving BIAS endpoint

## BIAS response message

BIAS message conveying a response to a prior BIAS requestmessage

## 1.6 References

### 1.6.1 Normative References

[RFC2119] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997.

<http://www.ietf.org/rfc/rfc2119.txt>

[CBEFF] ISO/IEC19785-1:2006, Information technology – Common Biometric Exchange Formats Framework – Part 1: Data element specification - ~~Part 3: Patron format specifications~~, with Amendment 1:2010

<http://www.iso.org>

[DATE-TIME] ISO 8601:2004, Data elements and interchange formats — Information interchange — Representation of dates and times

<http://www.iso.org>

[INCITS-BIAS] ANSI INCITS 442-2010, Biometric Identity Assurance Services (BIAS), July 2010

<http://www.incits.org>

[IRI] M. Duerst, et al, Internationalized Resource Identifiers, W3C RFC3987, January 2005

<http://www.ietf.org/rfc/rfc3987.txt>

[SOAP11] Simple Object Access Protocol (SOAP) 1.1, 8 May 2000

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[URI] T. Berners-Lee, R. Fielding, L. Masinter, *Uniform Resource Identifiers (URI): Generic Syntax*, RFC 3986, MIT/LCS, U.C. Irvine, Xerox Corporation, January 2005.

<http://ietf.org/rfc/rfc3986>

140	<b>[UTF-8]</b>	ISO/IEC 10646:2003, Information technology — Universal Multiple-Octet Coded
141		Character Set (UCS)
142		<a href="http://www.iso.org">http://www.iso.org</a>
143	<b>[WS-Addr]</b>	W3C Recommendation, Web Services Addressing 1.0 - Core, and Web Services
144		Addressing 1.0 - SOAP Binding, 9 May 2006
145		<a href="http://www.w3.org/2002/ws/addr/">http://www.w3.org/2002/ws/addr/</a>
146	<b>[WS-I-Basic]</b>	Basic Profile Version 1.1, 10 April 2006
147		<a href="http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html">http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html</a>
148	<b>[WS-I-Bind]</b>	Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile
149		Version 1.0, 24 August 2004
150		<a href="http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html">http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html</a>
151	<b>[WSDL11]</b>	Web Services Description Language (WSDL) 1.1, 15 March 2001
152		<a href="http://www.w3.org/TR/2001/NOTE-wsdl-20010315">http://www.w3.org/TR/2001/NOTE-wsdl-20010315</a>
153	<b>[XML 10]</b>	Extensible Markup Language (XML) 1.0, 16 August 2006
154		<a href="http://www.w3.org/TR/2006/REC-xml-20060816/">http://www.w3.org/TR/2006/REC-xml-20060816/</a>
155	<b>[XOP]</b>	XML-binary Optimized Packaging, W3C Recommendation, 25 January 2005
156		<a href="http://www.w3.org/TR/2005/REC-xop10-20050125/">http://www.w3.org/TR/2005/REC-xop10-20050125/</a>

## 1.6.2 Non-Normative References

158	<b>[BioAPI]</b>	ISO/IEC 19784-1:2006, Information technology – Biometric Application
159		Programming Interface – Part 1: BioAPI Specification
160		<a href="http://www.iso.org">http://www.iso.org</a>
161	<b>[CBEFF-3]</b>	<u>ISO/IEC 19785-3:2007, Information technology – Common Biometric Exchange Formats</u>
162		<u>Framework – Part 3: Patron format specifications, with Amendment 1:2010</u>
163		<a href="http://www.iso.org">http://www.iso.org</a>
165	<b>[BIO SEC]</b>	ISO 19092 Financial services -- Biometrics -- Security framework
166		<a href="http://www.iso.org">http://www.iso.org</a>
167	<b>[EBTS-DOD]</b>	Department of Defense Electronic Biometric Transmission Specification, Version
168		2.0, 27 March 2009
169		<a href="http://www.biometrics.dod.mil/CurrentInitiatives/Standards/dodebts.aspx">http://www.biometrics.dod.mil/CurrentInitiatives/Standards/dodebts.aspx</a>
170	<b>[EBTS-FBI]</b>	IAFIS-DOC-01078-8.1, “Electronic Biometric Transmission Specification (EBTS)”,
171		Version 8.1, November 19, 2008, Federal Bureau of Investigation, Criminal Justice
172		Information Services Division
173		<a href="https://www.fbibiospecs.org">https://www.fbibiospecs.org</a>
174	<b>[EFTS]</b>	IAFIS-DOC-01078-7, “Electronic Fingerprint Transmission Specification (EFTS)”, Version
175		7.1, May 2, 2005, Federal Bureau of Investigation, Criminal Justice Information Services
176		Division
177		<a href="https://www.fbibiospecs.org">https://www.fbibiospecs.org</a>
178	<b>[HR-XML]</b>	HR-XML Consortium Library, 2007 April 15
179		<a href="http://www.hr-xml.org">http://www.hr-xml.org</a>
180	<b>[INT-I]</b>	Interpol Implementation of ANSI/NIST ITL1-2000, Ver 4.22b, October 28, 2005, The Interpol
181		AFIS Expert Group
182		<a href="http://www.interpol.int">http://www.interpol.int</a>
183	<b>[NIEM]</b>	National Information Exchange Model (NIEM), Ver 2.0, June 2007, US DOJ/DHS
184		<a href="http://www.niem.gov">http://www.niem.gov</a>

185 **[RFC2246]** T. Dierks & C. Allen, *The TLS Protocol, Version 1.0*, January 1999  
 186 <http://www.ietf.org/rfc/rfc2246.txt>  
 187 **[RFC2617]** J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*,  
 188 June 1999  
 189 <http://www.ietf.org/rfc/rfc2617.txt>

190 **[RFC3280]** R. Housley, et al, Internet X.509 Public Key Infrastructure Certificate and  
 191 Certificate Revocation List (CRL) Profile, April 2002  
 192 <http://www.ietf.org/rfc/rfc3280.txt>

193 **[SAML]** Security Assertion Markup Language (SAML), Oasis Standard, March 2005  
 194 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

195 **[SAML SEC]** Security and Privacy Considerations for the OASIS Security Assertion Markup  
 196 Language (SAML) V2.0, Oasis Standard, 15 March 2005  
 197 <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>

198 **[SSL3]** SSL 3.0 Specification  
 199 <http://www.freessoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM>

200 **[WSS]** Web Services Security: SOAP Message Security 1.1, (WS-Security 2004),  
 201 OASIS Standard Specification, 1 February 2006  
 202 [http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)  
 203 [SOAPMessageSecurity.pdf](http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)

204 **[X509]** X.509: Information technology - Open Systems Interconnection - The Directory:  
 205 Public-key and attribute certificate frameworks, ITU-T, August 2005  
 206 <http://www.itu.int/rec/T-REC-X.509-200508-I>

207 **[xNAL]** Customer Information Quality Specifications Version 3.0: Name (xNL), Address (xAL), Name  
 208 and Address (xNAL) and Party (xPIL), Committee Specification 02, 20 September 2008  
 209 [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ciq](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq)

## 2 Design Concepts and Architecture (non-normative)

### 2.1 Philosophy

Rather than define a totally new and unique messaging protocol for biometric services, this specification instead defines a method for using existing biometric and Web services standards to exchange biometric data and perform biometric operations.

### 2.2 Context

Today, biometric systems are being developed which collect, process, store and match biometric data for a variety of purposes. In many cases, data and/or capabilities need to be shared between systems or systems serve a number of different client stakeholders. As architectures move towards services-based frameworks, access to these biometric databases and services is via a Web services front-end. However, lack of standardization in this area has led implementers to develop customized services for each system/application.

BIAS is intended to provide a common, yet flexible, Web services interface that can be used within both closed and open SOA systems. Figure 1, below, depicts the context in which the BIAS messages will be implemented.

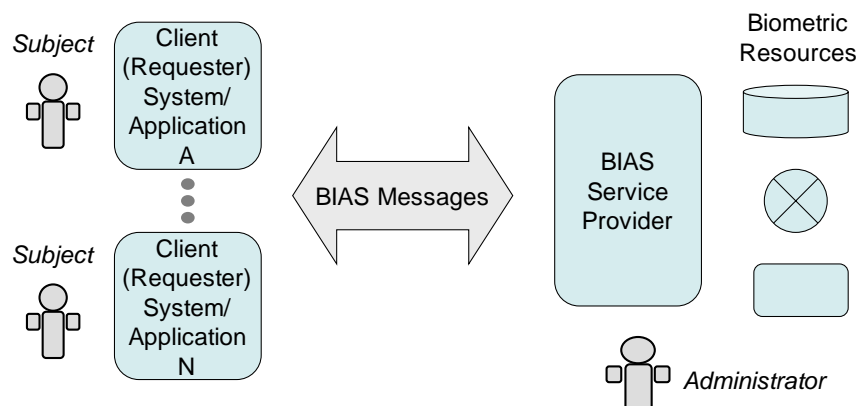


Figure 1. BIAS Context

The clients (requesters) may use standard discovery mechanisms (i.e., UDDI directories) to discover the BIAS service provider (implementation) or, particularly in closed systems, the URI/IRI and WSDL for the service provider may be known *a priori* by the client BIAS application developer.

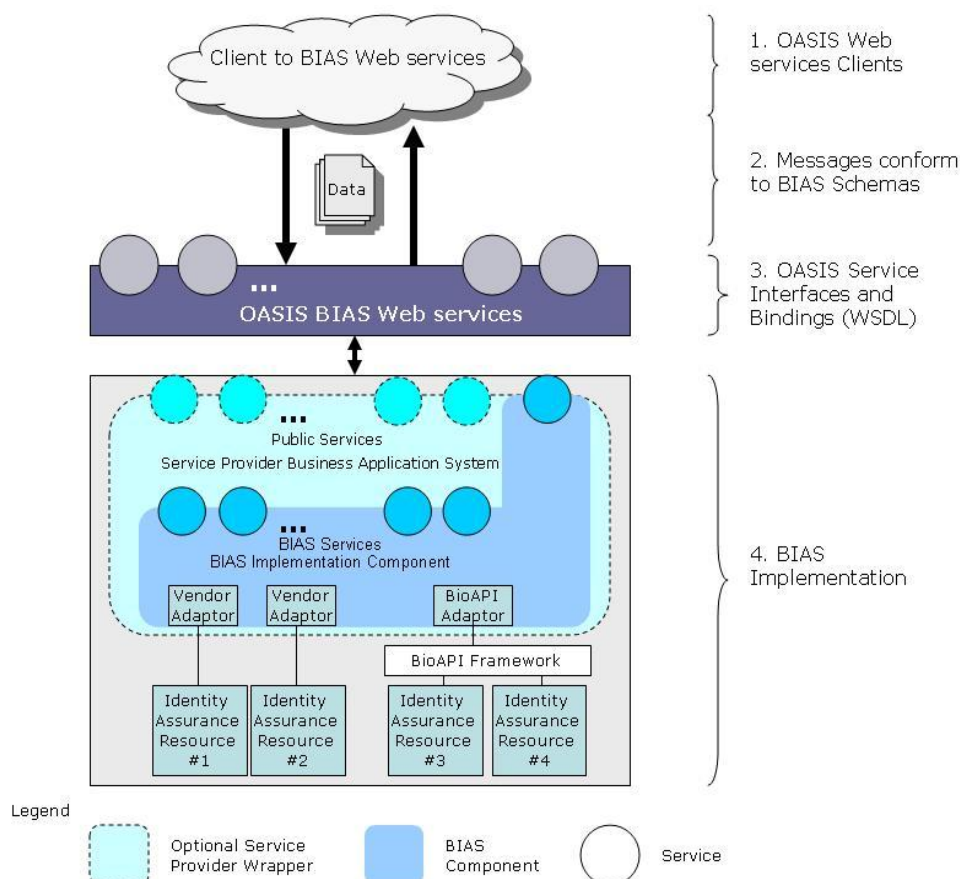
### 2.3 Architecture

BIAS Web services are intended to be used within systems employing a services framework, such as a services-oriented architecture (SOA) (although implementations are not limited to this environment). As such, it is recognized that the clients may interact directly with the BIAS service provider or layers may exist between the client and the service provider, for example as an ESB or other application layer.

The BIAS Architecture as shown in Figure 2, in which:

- A Client request to the BIAS Web services may be triggered by a human interaction OR any proxy system such as an ESB.
- Client sends and receives SOAP messages that conform to the BIAS schemas
- Calls to the BIAS Implementation use OASIS Service Interfaces and Bindings (via WSDL)

- The BIAS implementation maps the service call to the appropriate internal API or set of APIs and returns data according to the service interface.
- Note that services are represented as circles.



**Figure 2. Representative BIAS Architecture**

NOTE: It is possible that BIAS may also be used between the service provider and the managed resource (e.g., a biometric matcher).

At the heart of the BIAS ~~messaging protocol~~SOAP Profile are the concepts of BIAS messages and endpoints.

### **BIAS implementation**

A BIAS implementation is a software entity that is capable of creating, processing, sending, and receiving BIAS messages. This standard does not define requirements for the BIAS implementation other than defining the messages and protocols used by the endpoints.

### **BIAS messages**

A BIAS message is a one that can be sent from a BIAS endpoint to another BIAS endpoint over a TCP/IP link.

### **BIAS endpoints**

266 | A BIAS endpoint is a runtime entity, uniquely identified and accessed by an endpoint URI/IRI [URI] [IRI]<sub>5.1</sub>.  
267 | capable of sending and receiving BIAS messages.  
268 | ~~Note that when~~NOTE: When not publicly and directly exposed, the endpoints for purposes of this  
269 | specification are the BIAS service provider exposing BIAS services and the component that directly  
270 | interacts with that service provider, e.g., the business application or ESB, rather than the ultimate  
271 | end client requester.

## 3 Data dictionary

This section describes the BIAS data elements used within BIAS messages (as defined in Clause 4). Common data elements are defined for use in one or more operations. These include common data types or return codes. BIAS data elements are defined in ANSI INCITS 442-2010. The elements, complex types and simple types described for the BIAS messages belong to the following namespace: <http://docs.oasis-open.org/bias/ns/bias-1.0/>. See ~~Annex A~~AnnexA for the XML schema.

NOTE: Biographic and biometric data included in a native XML format MAY contain elements referencing external namespaces (e.g., ansi-nist).

### 3.1 Documentation Conventions

Each common element has a section describing its content. Likewise, each operation has a section describing the request and response messages and the associated input and output parameters. The input and output of each message and the comment elements are detailed in a table as described in the figure below. Each field that forms part of the message request/response is detailed in the table.

Header Name	Description	Values	Value Meaning
Field	The name of the field.		
Type	The XML schema type of the field.		
#	The cardinality of the field	1	One occurrence
		0..1	Zero or one occurrence
		0..*	Zero or more occurrences
		1..*	One or more occurrences
?	Defines if the field must be present.	Y	Yes – is always required
		N	No – is not always required, an optional field.
		C	Conditional – requirement is dependent on system or message conditions.
Meaning	Gives a short description of the field's use		

Figure 3. BIAS Message Input/Output Dictionary Table Headings

#### **Fields Hierarchy Explained:**

To denote the field hierarchy the symbol  $\uparrow$  is used to denote the child-of relationship.

All string types/elements MUST consist of ISO/IEC 10646 (Unicode) characters encoded in UTF-8 [UTF-8] (see ISO/IEC 10646:2003, Annex D).



## 3.2 Common Elements

### 3.2.1 ApplicationIdentifier

Type:	string
Description:	Identifies an application.
Min Length:	1
Max Length:	255

### 3.2.2 ApplicationUserIdentifier

Type:	string
Description:	Identifies an application user or instance.
Min Length:	1
Max Length:	255

### 3.2.3 BaseBIRType

Type:	Schema complexType
Description:	Base type for all BIR subtypes; see <a href="#">BinaryBIR</a> , <a href="#">URI_BIR</a> , and <a href="#">XML_BIR</a> for currently available types.

### 3.2.4 BIASBiometricDataType

Field	Type	#	?	Meaning
BIASBiometricDataType			Y	Wraps the various BIAS biometric types. The operations that use this type specify which elements are required.
↑ BIRList	<a href="#">CBEFF_BIR_ListType</a>	0..1	N	A list of CBEFF-BIR elements.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	0..1	N	Contains biometric information in either a non-XML or an XML representation.
↑ InputBIR	<a href="#">CBEFF_BIR_Type</a>	0..1	N	Maps to specific INCITS BIAS elements as required by that specification.
↑ ReferenceBIR	<a href="#">CBEFF_BIR_Type</a>	0..1	N	Maps to specific INCITS BIAS elements as required by that specification.
↑ BiometricDataList	<a href="#">BiometricDataListType</a>	0..1	N	A list of biometric data elements.

### 297 **3.2.5 BIASFaultCode**

<b>Type:</b>	String
<b>Description:</b>	Error code referenced in a SOAP fault.

### 298 **BIASFaultCode Enumeration Values**

Value	Description
UNKNOWN_ERROR	The service failed for an unknown reason.
UNSUPPORTED_CAPABILITY	A requested capability is not supported by the service implementation.
INVALID_INPUT	The data in a service input parameter is invalid.
BIR_QUALITY_ERROR	Biometric sample quality is too poor for the service to succeed.
INVALID_BIR	The input BIR is empty or in an invalid or unrecognized format.
BIR_SIGNATURE_FAILURE	The service could not validate the signature, if used, on the input BIR.
BIR_DECRYPTION_FAILURE	The service could not decrypt an encrypted input BIR.
INVALID_ENCOUNTER_ID	The input encounter ID is empty or in an invalid format.
INVALID_SUBJECT_ID	The input subject ID is empty or in an invalid format.
UNKNOWN_SUBJECT	The subject referenced by the input subject ID does not exist.
UNKNOWN_GALLERY	The gallery referenced by the input gallery ID does not exist.
UNKNOWN_ENCOUNTER	The encounter referenced by the input encounter ID does not exist.
UNKNOWN_BIOGRAPHIC_FORMAT	The biographic data format is not known or not supported.
UNKNOWN_IDENTITY_CLAIM	The identity referenced by the input identity claim does not exist.
INVALID_IDENTITY_CLAIM	The identity claim requested is already in use.
NONEXISTANT_DATA	The data requested for deletion does not exist.

### 299 **NOTES:**

- 300 (1) See Clause 6 (Error handling) for an explanation of BIAS faults and return codes.
- 301 (2) Service provider MAY define additional values specific to their service implementation.
- 302 (3) See section 5.5 for additional information on BIAS security.

### 303 **3.2.6 BIASFaultDetail**

Field	Type	#	?	Meaning
BIASFaultDetail			Y	Defines the error information associated with a SOAP fault.
↗ BIASFaultType	<a href="#">BIASFaultCode</a>	1	Y	References an error code.
↗ BIASFaultMessage	string	1	Y	Provides a brief explanation of the fault.

Field	Type	#	?	Meaning
BIASFaultDescription	string	0..1	N	Provides detailed information about a BIAS fault, such as trace details.

### 3.2.7 BIASIdentity

Field	Type	#	?	Meaning
BIASIdentity			Y	Defines a single element for encapsulating the data associated with an Identity. Includes the Identity's reference identifiers, biographic data, and biometric data.  The operations that use this type specify which elements are required.
SubjectID	<a href="#">BIASIDType</a>	0..1	C	A system unique identifier for a subject. Required as input to many operations.
IdentityClaim	<a href="#">BIASIDType</a>	0..1	N	An identifier by which a subject is known to a particular gallery or population group.
EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
EncounterList	<a href="#">EncounterListType</a>	0..1	N	A list of encounters associated with a subject.
BiographicData	<a href="#">BiographicDataType</a>	0..1	N	An Identity's biographic data.
BiographicDataElements	<a href="#">BiographicDataType</a>	0..1	N	An Identity's biographic data elements that are stored in the implementing system.
BiometricData	<a href="#">BIASBiometricDataType</a>	0..1	N	An Identity's biometric data.

### 3.2.8 BIASIDType

Type:	string
Description:	A BIAS Identifier.

### 3.2.9 BinaryBIR

Field	Type	#	?	Meaning
BinaryBIR	BaseBIRType		Y	Defines a BIR type of Binary

Field	Type	#	?	Meaning
↑ Binary	base64Binary	1	Y	BIR information in base64 binary format

### 3.2.10 BiographicDataItem

Field	Type	#	?	Meaning
BiographicDataItem			Y	Defines a single biographic data element.
↑ Name	string	1	Y	The name of the biographic data item.
↑ Type	string	1	Y	The data type for the biographic data item.
↑ Value	string	0..1	N	The value assigned to the biographic data item.

NOTE: This element can be used to transmit scanned identity documents or document information (e.g., passports, driver's license, birth certificates, utility bills, etc. required to establish an identity).

### 3.2.11 BiographicDataSet

Field	Type	#	?	Meaning
BiographicDataSet			Y	Defines a set of biographic data that is formatted according to the specified format.
↑ name	string	1	Y	The name of the biographic data format. Use these names for common formats: FBI-EFTS [EFTS], FBI-EBTS [EBTS-FBI], DOD-EBTS [EBTS-DOD], INT-I [INT-I], NIEM [NIEM], xNAL [xNAL], HR-XML [HR-XML].
↑ version	string	0..1	N	The version of the biographic data format (e.g., "7.1" for FBI-EFTS or "2.0" for NIEM).
↑ source	string	1	Y	Reference to a URI/IRI describing the biographic data format. For example: (FBI-EFTS and FBI-EBTS) www.fbibiospecs.org, (DOD-EBTS) www.biometrics.dod.mil, (INT-I) www.interpol.int, (NIEM) www.niem.gov, (xNAL) www.oasis-open.org, (HR-XML) www.hr-xml.org.
↑ type	string	1	Y	The biographic data format type. Use these types for common formats: ASCII (e.g., for non-XML versions of FBI-EFTS, FBI-EBTS, DOD-EBTS, or INT-I), XML (e.g., for NIEM, xNAL, and HR-XML or future versions of FBI-EBTS).
↑ unspecified	any	0..*	N	Biographic data formatted according to a specific format.

NOTE: Biographic data formats are not limited to those listed. The string value is not enumerated. If one of the common types are used, it ~~MUST be~~ **MUST be** indicated by the specified name values; however, the service provider MAY offer other formats. See INCITS 442 for further information.

### 3.2.12 BiographicDataType

Field	Type	#	?	Meaning
BiographicDataType			Y	Defines a set of biographic data elements, utilizing either the <a href="#">BiographicDataItem</a> to represent a list of elements or the <a href="#">BiographicDataSetType</a> to represent a complete, formatted set of biographic information. One of the following elements must be present.
↑ LastName	string	0..1	N	The last name of a subject.
↑ FirstName	string	0..1	N	The first name of a subject.
↑ BiographicDataItems	<a href="#">BiographicDataItem</a>	0..1	N	A list of biographic data elements.
↑ BiographicDataItem	<a href="#">BiographicDataItem</a>	1..*	N	A single biographic data element.
↑ BiographicDataSet	<a href="#">BiographicDataSetType</a>	0..1	N	A set of biographic data information.

NOTE: The implementer is given three choices for encoding biographic data:

- Encode only first and last name using the defined fields within BiographicDataType
- Define a list of biographic data elements using the BiographicDataItem
- Use a pre-defined set of biographic data (e.g., as specified in another standard) using the BiographicDataSetType.

See also INCITS 442, section 8.1 for further information.

### 3.2.13 BiometricDataElementType

Field	Type	#	?	Meaning
BiometricDataElement			Y	Provides descriptive information about biometric data, such as the biometric type, subtype, and format, contained in the BDB of the CBEFF-BIR.
↑ BiometricType	<del>iso-iec19785-3-</del> <del>7oasis_cbeff:</del> MultipleTypesType	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↑ BiometricTypeCount	positiveInteger	0..1	N	The number of biometric records having the biometric type recorded in the biometric type field.

Field	Type	#	?	Meaning
↑ BiometricSubType	<del>iso-iec19785-3-</del> <del>7oasis_cbeff:</del> SubtypeType	0..1	N	More specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF.
↑ BDBFormatOwner	positiveInteger	1	Y	Identifies the standards body, working group, industry consortium, or other CBEFF biometric organization that has defined the format for the biometric data.
↑ BDBFormatType	positiveInteger	1	Y	Identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field.

~~NOTE: XML biometric metadata (BIR header elements) is aligned with ISO/IEC 19785-3, XML Patron Format.~~

### **3.2.14 BiometricDataListType**

Field	Type	#	?	Meaning
BiometricDataListType			Y	A list of biometric data elements.
↑ BiometricDataElement	<a href="#">3.2.13</a> BiometricDataElementType	0..*	N	Data structure containing information about a biometric record.

### **3.2.15 CandidateListResultType**

Field	Type	#	?	Meaning
CandidateListResultType			Y	Defines a set of candidates, utilizing the <a href="#">CandidateType</a> to represent each element in the set.
↑ CandidateList	<a href="#">3.2.16</a> CandidateListType	1	Y	The candidate list.

### **3.2.16 CandidateListType**

Field	Type	#	?	Meaning
CandidateListType			Y	Defines a set of candidates, utilizing the <a href="#">CandidateType</a> to represent each element in the set.
↑ Candidate	<a href="#">CandidateType</a>	0..*	N	A single candidate.

### 329 **3.2.17 CandidateType**

Field	Type	#	?	Meaning
CandidateType			Y	Defines a single candidate as a possible match in response to a biometric identification request.
↑ Score	<a href="#">Score</a>	0..1	N	The match score.
↑ Rank	integer	1	Y	The rank of the candidate in relation to other candidates for the same biometric identification operation.
↑ BiographicData	<a href="#">BiographicDataType</a>	0..1	N	Biographic data associated with the candidate match.
↑ BIRList	<a href="#">CBEFF_BIR_ListType</a>	1	Y	Biometric data associated with the candidate match.

### 330 **3.2.18 CapabilityListType**

Field	Type	#	?	Meaning
CapabilityListType			Y	Defines a set of capabilities.
↑ Capability	<a href="#">CapabilityType</a>	0..*	N	A single capability.

### 331 **3.2.19 CapabilityName**

<b>Type:</b>	string
<b>Description:</b>	A list of capability items.

### 332 **CapabilityName Enumeration Values**

Value	Description
AggregateInputDataOptional	A data element accepted as optional input by the implementing system for the aggregate services.
AggregateInputDataRequired	A data element required as input by the implementing system for the aggregate services.
AggregateProcessingOption	A processing option supported by the implementing system for the aggregate services.
AggregateReturnData	A data element returned by the implementing system for the aggregate services.
AggregateServiceDescription	Describes the processing logic of an aggregate service supported by the implementing system.
BiographicDataSet	Identifies a biographic data set supported by the implementing system.
CBEFFPatronFormat	A patron format supported by the implementing system.
ClassificationAlgorithmType	A classification algorithm type supported by the implementing system.
ConformanceClass	Identifies the conformance class of the BIAS implementation.

Value	Description
Gallery	A gallery or population group supported by the implementing system.
IdentityModel	Identifies whether the implementing system is person-centric or encounter-centric based.
MatchScore	Identifies the use of match scores returned by the implementing system.
QualityAlgorithm	A quality algorithm vendor and algorithm vendor product ID supported by the implementing system.
SupportedBiometric	A biometric type supported by the implementing system.
TransformOperation	A transform operation type supported by the implementing system.

### 333 **3.2.20 CapabilityType**

Field	Type	#	?	Meaning
CapabilityType			Y	Defines a single capability supported by an implementing system.
↑ CapabilityName	<a href="#">CapabilityName</a>	1	Y	The name of the capability.
↑ CapabilityID	string	0..1	N	An identifier assigned to the capability by the implementing system.
↑ CapabilityDescription	string	0..1	N	A description of the capability.
↑ CapabilityValue	string	0..1	N	A value assigned to the capability.
↑ CapabilitySupportingValue	string	0..1	N	A secondary value supporting the capability.
↑ CapabilityAdditionalInfo	string	0..1	N	Contains additional information for the supported capability.

### 334 **3.2.21 CBEFF\_BIR\_ListType**

Field	Type	#	?	Meaning
CBEFF_BIR_ListType			Y	A list of CBEFF-BIR elements.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	0..*	N	CBEFF structure containing information about a biometric sample.

### 335 **3.2.22 CBEFF\_BIR\_Type**

Field	Type	#	?	Meaning
CBEFF_BIR_Type			Y	Represents biometric information, with either a non-XML or XML representation.
↑ FormatOwner	positiveInteger	1	Y	Identifies the Patron format owner.
↑ FormatType	positiveInteger	1	Y	Identifies the Patron format type.



Field	Type	#	?	Meaning
↑ BIR_Information		0..1	N	Describes what is contained in a BIR.
↑ BIR_Info	<del>iso-iec19785-3-7oasis_cbeff:</del> BIRInfoType	0..1	N	Contains information about the CBEFF-BIR.
↑ BDB_Info	<del>iso-iec19785-3-7oasis_cbeff:</del> BDBInfoType	0..1	N	Contains information about the BDB in a simple CBEFF-BIR.
↑ SB_Info	<del>iso-iec19785-3-7oasis_cbeff:</del> SBInfoType	0..1	N	Contains information about the security block, if used, in a simple CBEFF-BIR.
↑ BIR	BaseBIRType	1	Y	One of the following sub-elements must be present: <a href="#">BinaryBIR</a> , <a href="#">URI_BIR</a> , or <a href="#">XML_BIR</a> .

NOTE: The implementer is given three choices for encoding a BIR:

- As an XML BIR (following the XML Patron format as specified in ~~Part 3, Clause 13~~[Annex B](#))
- As a reference to a URI (from which the receiver would retrieve the actual BIR)
- As a complete Base64 encoded binary (non-XML) BIR.

The latter two alternatives can use any CBEFF Patron Format. The optional BIR\_Information provides a mechanism for exposing metadata associated with a BIR format that is not easily decoded (i.e., a non-XML BIR). See section 5.3 for more information on handling of binary data within BIAS and INCITS 442, Clause 8.2, for more information on representing biometric data.

NOTE:

- (1) XML BIRs MUST conform to ~~ISO/IEC 19785-3 (clause 13, the XML Patron Format); patron format in Annex B~~; however, non-XML (binary) and URI BIRs MAY implement any CBEFF patron format.
- (2) It is RECOMMENDED that only registered CBEFF patron formats be used; however, in closed systems, this may not be required.

### **3.2.23 Classification**

<b>Type:</b>	string
<b>Description:</b>	The result of a classification.

### **3.2.24 ClassificationAlgorithmType**

<b>Type:</b>	string
<b>Description:</b>	Type of classification algorithm that was used to perform the classification.

### 3.2.25 ClassificationData

Field	Type	#	?	Meaning
ClassificationData			Y	Contains information on classification results and the algorithm used to determine the classification.
↑ Classification	<a href="#">Classification</a>	1	Y	The result of the classification.
e ↑ ClassificationAlgorithmType	<a href="#">ClassificationAlgorithmType</a>	1	Y	Identifies the type of classification algorithm that was used to perform the classification.

### 3.2.26 EncounterListType

Field	Type	#	?	Meaning
EncounterListType			Y	Defines a set of encounters.
↑ EncounterID	<a href="#">BIASIDType</a>	0..*	N	The identifier of an encounter.

### 3.2.27 FusionDecision

Type:	string
Description:	The match decision assigned by the matching algorithm

### 3.2.28 FusionInformationListType

Field	Type	#	?	Meaning
FusionInformationListType			Y	Contains at a minimum two sets of fusion input elements, as input to the <a href="#">PerformFusion</a> operation.
↑ FusionElement	<a href="#">FusionInformationType</a>	2..*	Y	A set of fusion information.

### 3.2.29 FusionInformationType

Field	Type	#	?	Meaning
FusionInformationType			Y	Represents the information necessary to perform a fusion operation.
↑ BiometricType	<del>iso-iec19785-3-</del> <a href="#">7oasis_cbeff</a> :MultipleTypesType	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
e ↑ BiometricSubType	<del>iso-iec19785-3-</del> <a href="#">7oasis_cbeff</a> :SubtypeType	0..1	N	More specifically defines the type of biometric data stored in the biometric record.

Field	Type	#	?	Meaning
AlgorithmOwner	string	1	Y	The owner or vendor of the algorithm used to determine the score or decision.
AlgorithmType	string	1	Y	The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision.
FusionResult	<a href="#">FusionResult</a>	0..1	C	Either <a href="#">FusionScore</a> or a <a href="#">FusionDecision</a> element MUST be used.

### 3.2.30 FusionResult

<b>Type:</b>	complexType
<b>Description:</b>	The base type for any resulting types which indicate the status of a Fusion operation

### 3.2.31 FusionScore

<b>Type:</b>	<a href="#">Score</a>
<b>Description:</b>	The similarity score assigned by the matching algorithm.

### 3.2.32 GenericRequestParameters

Field	Type	#	?	Meaning
GenericRequestParameters			Y	Common request parameters that can be used to identify the requester.
Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested.

NOTE: See section 5.4 for alternatives for identifying the requested BIAS operation in a BIAS SOAP message.

### 3.2.33 IdentifySubjectResultType

<b>Description:</b>	A base type for all types that could be returned from the IdentifySubject operation
---------------------	---

366 **3.2.34 InformationType**

Field	Type	#	?	Meaning
InformationType			Y	Allows for an unlimited number of data element types, and it does not specify nor require any particular data element.
↑ unspecified	any	0..*	N	

367 **3.2.35 ListFilterType**

Field	Type	#	?	Meaning
ListFilterType			Y	Provides a method to filter the amount of information returned in a search of biometric data.
↑ BiometricTypeFilters		1	Y	
↑ BiometricTypeFilter	<del>iso-iec19785-3-7oasis_cbeff</del> :MultipleTypesType	1..*	Y	Limits the returned information to a specific type of biometric, as defined by CBEFF.
e ↑ IncludeBiometricSubType	boolean	1	Y	A Boolean flag indicating if biometric subtype information should be returned.

368 **3.2.36 MatchType**

<b>Type:</b>	boolean
<b>Description:</b>	The result of a fusion method.

369 **3.2.37 ProcessingOptionsType**

Field	Type	#	?	Meaning
ProcessingOptionsType			Y	BIAS aggregate operations support the ability to include various processing options which direct and possibly control the business logic for that operation. The ProcessingOptionsType provides a method to represent those options. Processing options SHOULD be defined by the implementing system.
↑ Option	string	0..*	N	An option supported by the implementing system.

370 **3.2.38 ProductID**

<b>Type:</b>	string
<b>Description:</b>	The vendor's ID for a particular product.

371 **3.2.39 QualityData**

Field	Type	#	?	Meaning
QualityData			Y	Contains information about a biometric sample's quality and the algorithm used to compute the quality.
↑ QualityScore	<del>iso-iec19785-3-7oasis_cbeff</del> :QualityType	0..1	N	The quality of a biometric sample.
↑ AlgorithmVendor	<a href="#">VendorIdentifier</a>	1	Y	The vendor of the quality algorithm used to determine the quality score.
↑ AlgorithmVendorProductID	<a href="#">ProductID</a>	1	Y	The vendor's ID for the algorithm used to determine the quality.
↑ AlgorithmVersion	<a href="#">VersionType</a>	0..1	N	The version of the algorithm used to determine the quality.

372 **3.2.40 ResponseStatus**

Field	Type	#	?	Meaning
ResponseStatus			Y	
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

373 **3.2.41 ReturnCode**

<b>Type:</b>	unsignedLong
<b>Description:</b>	Return value specifying success or other condition.

374  
375 **ReturnCode Enumeration Values**

Value	Description
0	Success

376 **3.2.42 Score**

<b>Type:</b>	float
<b>Description:</b>	Match result or quality score.

377 NOTE: Matching scores MAY be in a standardized or proprietary form in terms of value range and  
378 interpretation. Quality scores, however, follow the definition found in ~~ISO/IEC 19785-3, Clause~~  
379 ~~43Annex B~~.

### 3.2.43 TokenResultType

Field	Type	#	?	Meaning
TokenResultType			Y	Defines a token that is returned for asynchronous processing.
↑ TokenType	<a href="#">TokenType</a>	1	Y	Defines a token that is returned for asynchronous processing.

### 3.2.44 TokenType

Field	Type	#	?	Meaning
TokenType			Y	Defines a token that is returned for asynchronous processing.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

NOTE: Date/time format is defined in INCITS 442 and is consistent with the date format specified in [ISO/IEC 19785-3 Annex B](#) and ISO 8601 [DATE-TIME]. See also Annex A for schema definition.

### 3.2.45 URI\_BIR

Field	Type	#	?	Meaning
URI_BIR	BaseBIRType		Y	Defines a BIR type of Binary
↑ URI	anyURI	1	Y	The URI of the BIR

### 3.2.46 VendorIdentifier

Type:	string
Description:	Identifies a vendor.

NOTE: Vendor identifiers are registered with IBIA as the CBEFF registration authority (see ISO/IEC 19785-2). Registered biometric organizations are listed at:  
[http://www.ibia.org/cbeff/\\_biometric\\_org.php](http://www.ibia.org/cbeff/_biometric_org.php).

### 3.2.47 Version

Field	Type	#	?	Meaning
Version			Y	For a description or definition of each data element, see the referenced CBEFF standards in the <a href="#">3.2.22</a> CBEFF_BIR_Typeschema.
↑ major	nonNegativeInteger	1	Y	
↑ minor	nonNegativeInteger	1	Y	

392 **3.2.48 VersionType**

<b>Type:</b>	string
<b>Description:</b>	The version of a component.

393 **3.2.49 XML\_BIR**

Field	Type	#	?	Meaning
XML_BIR	BaseBIRType		Y	Defines a BIR type of Binary
↑ XML	<del>iso-iec19785-3-</del> Oasis_cbeff:BIRType	1	Y	BIR information in XML format

394

## 4 BIAS Messages

This section describes the BIAS messages implementing BIAS operations as defined in ANSI INCITS 442-2010. The operations are listed alphabetically, with each operation containing a request and a response message. The tables follow the conventions described in section 3.1.

### Primitive Operations

#### 4.1.1 AddSubjectToGallery

[AddSubjectToGalleryRequest](#)

[AddSubjectToGalleryResponse](#)

The AddSubjectToGallery operation registers a subject to a given gallery or population group. As an OPTIONAL parameter, the value of the claim to identity by which the subject is known to the gallery MAY be specified. This claim to identity MUST be unique across the gallery. If no claim to identity is specified, the subject ID (assigned with the [CreateSubject](#) operation) will be used as the claim to identity. Additionally, in the encounter-centric model, the encounter ID associated with the subject's biometrics that will be added to the gallery MUST be specified.

#### Request Message

Field	Type	#	?	Meaning
AddSubjectToGallery			Y	Register a subject to a given gallery or population group.
↑ AddSubjectToGalleryRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "AddSubjectToGallery".
↑ GalleryID	<a href="#">BIASIDType</a>	1	Y	The identifier of the gallery or population group to which the subject will be added.
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	The identity to add to the gallery.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.



Field	Type	#	?	Meaning
↑ IdentityClaim	<a href="#">BIASIDType</a>	0..1	N	An identifier by which a subject is known to a particular gallery or population group. (This could be a username or account number, for example.)
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

## 410 Response Message

Field	Type	#	?	Meaning
AddSubjectToGalleryResponse			Y	The response to an AddSubjectToGallery operation.
↑ AddSubjectToGalleryResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

## 411 4.1.2 CheckQuality

### 412 [CheckQualityRequest](#)

### 413 [CheckQualityResponse](#)

414 The CheckQuality operation returns a quality score for a given biometric. The biometric input is provided  
415 in a CBEFF basic structure or CBEFF record, which in this specification is called a CBEFF-BIR. The  
416 algorithm vendor and algorithm vendor product ID MAY be optionally provided in order to request a  
417 particular algorithm's use in calculating the biometric quality. If an algorithm vendor is provided then the  
418 algorithm vendor product ID is REQUIRED. If no algorithm vendor is provided, the implementing system  
419 will provide the algorithm vendor and algorithm vendor product ID that were used to calculate the  
420 biometric quality as output parameters.

## 421 Request Message

Field	Type	#	?	Meaning
CheckQuality			Y	Calculate a quality score for a given biometric.

Field	Type	#	?	Meaning
↑ CheckQualityRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "CheckQuality".
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	Data structure containing a single biometric sample for which a quality score is to be determined.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	1	Y	The biometric sample.
↑ Quality	<a href="#">QualityData</a>	0..1	N	Specifies a particular algorithm vendor and vendor product ID.
↑ AlgorithmVendor	<a href="#">VendorIdentifier</a>	1	Y	The vendor of the quality algorithm used to determine the quality score.
↑ AlgorithmVendorProductID	<a href="#">ProductID</a>	1	Y	The vendor's ID for the algorithm used to determine the quality.

422 **Response Message**

Field	Type	#	?	Meaning
CheckQualityResponse			Y	The response to a CheckQuality operation.
↑ CheckQualityResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ QualityInfo	<a href="#">QualityData</a>	1	Y	Contains the quality information for the submitted biometric sample.
↑ QualityScore	<del>iso-iec19785-3-7</del> <a href="#">oasis_cbeff:QualityType</a>	0..1	N	The quality of a biometric sample.
↑ AlgorithmVendor	<a href="#">VendorIdentifier</a>	1	Y	The vendor of the quality algorithm used to determine the quality score.
↑ AlgorithmVendorProductID	<a href="#">ProductID</a>	1	Y	The vendor's ID for the algorithm used to determine the quality.
↑ AlgorithmVersion	<a href="#">VersionType</a>	1	Y	The version of the algorithm used to determine the quality.

423 **4.1.3 ClassifyBiometricData**424 [ClassifyBiometricDataRequest](#)425 [ClassifyBiometricDataResponse](#)

426 The ClassifyBiometricData operation attempts to classify a biometric sample. For example, a fingerprint  
 427 biometric sample may be classified as a whorl, loop, or arch (or other classification classes and sub-  
 428 classes).

429 To obtain the types of classification algorithms and classes, see the [QueryCapabilities](#) operation.

430 **Request Message**

Field	Type	#	?	Meaning
ClassifyBiometricData			Y	Classifies a biometric sample.
↑ ClassifyBiometricDataRequest		1	Y	

Field	Type	#	?	Meaning
↑ GenericRequestParameter s	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "ClassifyBiometricData".
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	Data structure containing a single biometric sample for which the classification is to be determined.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	1	Y	The biometric sample.

## 431 Response Message

Field	Type	#	?	Meaning
ClassifyBiometricDataResponse			Y	The response to a ClassifyBiometricData operation, containing the classification of a biometric sample.
↑ ClassifyBiometricDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ClassificationData	<a href="#">ClassificationData</a>	1	Y	Information on the results and type of classification performed.
↑ Classification	<a href="#">Classification</a>	1	Y	The result of the classification.

Field	Type	#	?	Meaning
↑ ClassificationAlgorithmType	<a href="#">ClassificationAlgorithmType</a>	1	Y	Identifies the type of classification algorithm that was used to perform the classification.

#### 4.1.4 CreateSubject

[CreateSubjectRequest](#)

[CreateSubjectResponse](#)

The CreateSubject operation creates a new subject record and associates a subject ID to that record. As an optional parameter, the subject ID MAY be specified by the caller. If no subject ID is specified, the CreateSubject operation will generate one.

#### Request Message

Field	Type	#	?	Meaning
CreateSubject			Y	
↑ CreateSubjectRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "CreateSubject".

#### Response Message

Field	Type	#	?	Meaning
CreateSubjectResponse			Y	The response to a CreateSubject operation, containing the subject ID of the new subject record.
↑ CreateSubjectResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.

#### 4.1.5 DeleteBiographicData

[DeleteBiographicDataRequest](#)

[DeleteBiographicDataResponse](#)

The DeleteBiographicData operation erases all of the biographic data associated with a given subject record. In the encounter-centric model the operation erases all of the biographic data associated with a given encounter, and therefore the encounter ID MUST be specified.

When deleting data, BIAS implementations MAY completely erase the information in order to prevent the ability to reconstruct a record in whole or in part, or they MAY track and record the deleted information for auditing and/or quality control purposes.

#### Request Message

Field	Type	#	?	Meaning
DeleteBiographicData			Y	Erase all of the biographic data associated with a given subject record or, in the encounter-centric model, with a given encounter.
↑ DeleteBiographicDataRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteBiographicData"
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.

Field	Type	#	?	Meaning
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

## 450 Response Message

Field	Type	#	?	Meaning
DeleteBiographicDataResponse			Y	The response to a DeleteBiographicData operation.
↑ DeleteBiographicDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

## 451 4.1.6 DeleteBiometricData

### 452 [DeleteBiometricDataRequest](#)

### 453 [DeleteBiometricDataResponse](#)

454 The DeleteBiometricData operation erases all of the biometric data associated with a given subject  
 455 record. In the encounter-centric model the operation erases all of the biometric data associated with a  
 456 given encounter, and therefore the encounter ID MUST be specified.

457 When deleting data, BIAS implementations MAY completely erase the information in order to prevent the  
 458 ability to reconstruct a record in whole or in part, or they MAY track and record the deleted information for  
 459 auditing and/or quality control purposes.

## 460 Request Message

Field	Type	#	?	Meaning
DeleteBiometricData			Y	Erase all of the biometric data associated with a given subject record or, in the encounter-centric model, with a given encounter.
↑ DeleteBiometricDataRequest		1	Y	

Field	Type	#	?	Meaning
↑ GenericRequestParameter s	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteBiometricData".
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject.  Required for encounter-centric models.

## 461 Response Message

Field	Type	#	?	Meaning
DeleteBiometricDataResponse			Y	The response to a DeleteBiometricData operation.
↑ DeleteBiometricDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

## 462 4.1.7 DeleteSubject

### 463 [DeleteSubjectRequest](#)

### 464 [DeleteSubjectResponse](#)

465 The DeleteSubject operation deletes an existing subject record and, in an encounter-centric model, any  
466 associated encounter information from the system. This operation also removes the subject from any  
467 registered galleries.



468 When deleting a subject, BIAS implementations MAY completely erase the subject information in order to  
 469 prevent the ability to reconstruct a record or records in whole or in part, or they MAY track and record the  
 470 deleted information for auditing and/or quality control purposes.

## 471 Request Message

Field	Type	#	?	Meaning
DeleteSubject			Y	Delete an existing subject record and, in an encounter-centric model, any associated encounter information.
⌞ DeleteSubjectRequest		1	Y	
⌞ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
⌞ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
⌞ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
⌞ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteSubject".
⌞ Identity	<a href="#">BIASIdentity</a>	1	Y	The identity of the subject to delete.
⌞ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.

## 472 Response Message

Field	Type	#	?	Meaning
DeleteSubjectResponse			Y	The response to a DeleteSubject operation.
⌞ DeleteSubjectResponsePackage		1	Y	
⌞ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
⌞ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
⌞ Message	string	0..1	N	A short message corresponding to the return code.

## 473 4.1.8 DeleteSubjectFromGallery

### 474 [DeleteSubjectFromGalleryRequest](#)

475 [DeleteSubjectFromGalleryResponse](#)

476 The DeleteSubjectFromGallery operation removes the registration of a subject from a gallery or  
 477 population group. The subject is identified by either the subject ID or the claim to identity that was  
 478 specified in the [AddSubjectToGallery](#) operation.

479 **Request Message**

Field	Type	#	?	Meaning
DeleteSubjectFromGallery			Y	Remove the registration of a subject from a gallery or population group.
↳ DeleteSubjectFromGalleryRequest		1	Y	
↳ GenericRequestParameter s	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↳ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteSubjectFromGallery".
↳ GalleryID	<a href="#">BIASIDType</a>	1	Y	The identifier of the gallery or population group from which the subject will be deleted.
↳ Identity	<a href="#">BIASIdentity</a>	1	Y	The identity to remove from the gallery.
↳ SubjectID	<a href="#">BIASIDType</a>	0..1	C	A system unique identifier for a subject. Required if an Identity Claim is not provided.
↳ IdentityClaim	<a href="#">BIASIDType</a>	0..1	C	An identifier by which a subject is known to a particular gallery or population group. Required if a Subject ID is not provided.

## 480 Response Message

Field	Type	#	?	Meaning
DeleteSubjectFromGalleryResponse			Y	The response to a DeleteSubjectFromGallery operation.
↳ DeleteSubjectFromGalleryResponsePackage				
↳ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↳ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.

## 481 4.1.9 GetIdentifySubjectResults

482 [GetIdentifyResultsRequest](#)

483 [GetIdentifySubjectResultsResponse](#)

484 The GetIdentifySubjectResults operation retrieves the identification results for the specified token. This  
 485 operation is used in conjunction with the [IdentifySubject](#) operation. If the IdentifySubject operation is  
 486 implemented as an asynchronous service, the implementing system returns a token and the  
 487 GetIdentifySubjectResults operation is used to poll for the results of the original IdentifySubject request.

## 488 Request Message

Field	Type	#	?	Meaning
GetIdentifySubjectResults			Y	Retrieve the identification results for a specified token, which was returned by the <a href="#">IdentifySubject</a> operation.
↳ GetIdentifySubjectResultsRequest		1	Y	
↳ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↳ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.

Field	Type	#	?	Meaning
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetIdentifySubjectResults".
↑ Token	<a href="#">TokenType</a>	1	Y	A value used to retrieve the results of an IdentifySubject request.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

## 489 Response Message

Field	Type	#	?	Meaning
GetIdentifySubjectResultsResponse			Y	The response to a GetIdentifySubjectResults operation, which includes a candidate list.
↑ GetIdentifySubjectResultsResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ CandidateList	<a href="#">CandidateListType</a>	1	Y	A rank-ordered list of candidates that have a likelihood of matching the input biometric sample.
↑ Candidate	<a href="#">CandidateType</a>	0..*	N	A single candidate.
↑ Score	<a href="#">Score</a>	0..1	N	The match score.
↑ BiographicData	<a href="#">BiographicDataType</a>	0..1	N	Biographic data associated with the candidate match.

Field	Type	#	?	Meaning
↑ BIRList	<a href="#">CBEFF_BIR_ListType</a>	1	Y	Biometric data associated with the candidate match.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	0..*	N	CBEFF structure containing information about a biometric sample.

#### 4.1.10 IdentifySubject

[IdentifySubjectRequest](#)

[IdentifySubjectResponse](#)

The IdentifySubject operation performs an identification search against a given gallery for a given biometric, returning a rank-ordered candidate list of a given maximum size.

If the IdentifySubject operation is implemented as a synchronous service, the implementing system immediately processes the request and returns the results in the candidate list. If the IdentifySubject operation is implemented as an asynchronous service, the implementing system returns a token, which is an indication that the request is being handled asynchronously. In this case, the [GetIdentifySubjectResults](#) operation is used to poll for the results of the IdentifySubject request.

#### Request Message

Field	Type	#	?	Meaning
IdentifySubject			Y	Perform an identification search against a given gallery for a given biometric.
↑ IdentifySubjectRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "IdentifySubject".
↑ GalleryID	<a href="#">BIASIDType</a>	1	Y	The identifier of the gallery or population group which will be searched.

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Contains the BIR, a data structure containing the biometric sample for the search.
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	An Identity's biometric data.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	1	Y	Contains biometric information in either a non-XML or an XML representation.
↑ MaxListSize	positiveInteger	1	Y	The maximum size of the candidate list that should be returned.

## 501 Response Message

Field	Type	#	?	Meaning
IdentifySubjectResponse			Y	The response to an IdentifySubject operation, returning a rank-ordered candidate list.
↑ IdentifySubjectResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ CandidateList	<a href="#">CandidateListResultType</a> (see <a href="#">IdentifySubjectResultType</a> )	0..1	C	A rank-ordered list of candidates that have a likelihood of matching the input biometric sample (i.e., exceed the system threshold).  Rank ordering is from highest to lowest match score.  Returned with successful synchronous request processing.
↑ Candidate	<a href="#">CandidateType</a>	0..*	N	A single candidate.
↑ Score	string	0..1	N	The match score.

Field	Type	#	?	Meaning
↑ BiographicData	<a href="#">BiographicDataType</a>	0..1	N	Biographic data associated with the candidate match.
↑ BIRList	<a href="#">CBEFF_BIR_ListType</a>	1	Y	Biometric data associated with the candidate match.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	0..*	N	CBEFF structure containing information about a biometric sample.
↑ Token	<a href="#">TokenResultType</a> (see <a href="#">IdentifySubjectResultType</a> )	0..1	C	A token used to retrieve the results of the IdentifySubject operation. Returned with asynchronous request processing.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

NOTES:

- (1) In the event that the number of candidates exceeding the threshold exceeds the MaxListSize, the system will determine which candidate is included in the last position of the rank ordered candidate list (i.e., in the event of a tie).
- (2) Requesters MAY NOT change the system thresholds.

## 4.1.11 ListBiographicData

[ListBiographicDataRequest](#)

[ListBiographicDataResponse](#)

The ListBiographicData operation lists the biographic data elements stored for a subject using the Biographic Data Elements output parameter. Note that no actual biographic data is returned by this operation (see the [RetrieveBiographicInformation](#) operation to obtain the biographic data). In the encounter-centric model, an encounter ID MAY be specified to indicate that only the biographic data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the operation returns the list of encounter IDs which contain biographic data using the Encounter List output parameter, and the Biographic Data Elements output parameter is empty.

## Request Message

Field	Type	#	?	Meaning
ListBiographicData			Y	Lists the biographic data elements stored for a subject.

Field	Type	#	?	Meaning
↑ ListBiographicDataRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "ListBiographicData".
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	N	The identifier of an encounter associated with the subject.

## 518 Response Message

Field	Type	#	?	Meaning
ListBiographicDataResponse			Y	The response to a ListBiographicData request, containing a list of biographic data elements stored for a subject. In the encounter-centric model, the biographic data elements for a specific encounter are returned. If an encounter ID is not specified and encounter data exists for the subject, the list of encounter IDs which contain biographic data is returned.
↑ ListBiographicDataResponsePackage		1	Y	



Field	Type	#	?	Meaning
↳ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↳ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ Identity	<a href="#">BIASIdentity</a>	1	Y	Contains a list of biographic data elements associated with a subject or encounter; non-empty if the service was successful, biographic data exists, and either (a) the person-centric model is being used or (b) the encounter-centric model is being used and an encounter identifier was specified.
↳ BiographicDataElements	<a href="#">BiographicDataType</a>	0..1	C	An Identity's biographic data elements that are stored in the implementing system.
↳ BiographicDataItem	<a href="#">BiographicDataItem</a>	0..*	N	A single biographic data element.
↳ Name	string	1	Y	The name of the biographic data item.
↳ Type	string	1	Y	The data type for the biographic data item.
↳ EncounterList	<a href="#">EncounterListType</a>	0..1	C	A list of encounter ID's associated with a subject and which contain biographic data; non-empty if the service was successful, biographic data exists, the encounter-centric model is being used, and an encounter identifier was not specified.

Field	Type	#	?	Meaning
↑ EncounterID	<a href="#">BIASIDType</a>	0..*	N	The identifier of an encounter.

#### 4.1.12 ListBiometricData

[ListBiometricDataRequest](#)

[ListBiometricDataResponse](#)

The ListBiometricData operation lists the biometric data elements stored for a subject using the Biometric Data List output parameter. Note that no actual biometric data is returned by this operation (see the [RetrieveBiometricInformation](#) operation to obtain the biometric data). In the encounter-centric model, an encounter ID MAY be specified to indicate that only the biometric data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the operation returns the list of encounter IDs which contain biometric data using the Encounter List output parameter, and the Biometric Data List output parameter is empty.

An optional parameter MAY be used to indicate a filter on the list of returned data. Such a filter may indicate that only biometric types should be listed (e.g., face, finger, iris, etc.) or that only biometric subtypes for a particular biometric type should be listed (e.g., all fingerprints: left slap, right index, etc.). If a filter is not specified, all biometric type and biometric subtype information are listed (e.g., left index finger, right iris, face frontal, etc.).

#### Request Message

Field	Type	#	?	Meaning
ListBiometricData			Y	Lists the biometric data elements stored for a subject.
↑ ListBiometricDataRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "ListBiometricData".

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	N	The identifier of an encounter associated with the subject.
↑ ListFilterType	<a href="#">ListFilterType</a>	0..1	N	Indicates what biometric information should be returned.
↑ BiometricTypeFilter	<del>iso-iec19785-3-7:Multiple-types</del> oasis_cbeff:MultipleTypesType	1..*	Y	Limits the returned information to a specific type of biometric, as defined by CBEFF.
↑ IncludeBiometricSubType	boolean	1	Y	A Boolean flag indicating if biometric subtype information should be returned.

**Response Message**

Field	Type	#	?	Meaning
ListBiometricDataResponse			Y	The response to a ListBiometricData operation, containing a list of biometric data elements stored for a subject. In the encounter-centric model, the biometric data elements for a specific encounter are returned. If an encounter ID is not specified and encounter data exists for the subject, the list of encounter IDs which contain biometric data is returned.
↳ ListBiometricDataResponsePackage		1	Y	
↳ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↳ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ Identity	<a href="#">BIASIdentity</a>	0..1	N	Includes a list of biometric data elements associated with a subject or encounter or a list of encounter ID's associated with a subject and which contain biometric data.
↳ BiometricData	<a href="#">BIASBiometricDataType</a>	0..1	C	An Identity's biometric data.

Field	Type	#	?	Meaning
↑ BiometricDataList	<a href="#">BiometricDataListType</a>	0..1	N	A list of biometric data elements.
↑ BiometricDataElement	<a href="#">BiometricDataElementType</a>	1..*	Y	Data structure containing information about a biometric record.
↑ BiometricType	<del>iso-iec19785-3-</del> <del>7oasis_cbeff:</del> MultipleTypesType	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↑ BiometricTypeCount	positiveInteger	0..1	N	The number of biometric records having the biometric type recorded in the biometric type field.
↑ BiometricSubType	<del>iso-iec19785-3-</del> <del>7oasis_cbeff:</del> SubtypeType	0..1	N	More specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF.
↑ BDBFormatOwner	positiveInteger	1	Y	Identifies the standards body, working group, industry consortium, or other CBEFF biometric organization that has defined the format for the biometric data.
↑ BDBFormatType	positiveInteger	1	Y	Identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field.

Field	Type	#	?	Meaning
↑ EncounterList	<a href="#">EncounterListType</a>	0..1	C	A list of encounter ID's associated with a subject and which contain biometric data; non-empty if the service was successful, biometric data exists, the encounter-centric model is being used, and an encounter identifier was not specified.
↑ EncounterID	<a href="#">BIASIDType</a>	1..*	Y	The identifier of an encounter.

### 4.1.13 PerformFusion

[PerformFusionRequest](#)

[PerformFusionResponse](#)

The PerformFusion operation accepts either match score or match decision information and creates a fused match result. The [FusionInformationListType](#), through the [FusionInformationType](#), provides specific elements for match score input and match decision input. The fusion method and processes are left to the implementing system.

### Request Message

Field	Type	#	?	Meaning
PerformFusion			Y	Accepts either match score or match decision information and creates a fused match result.
↑ PerformFusionRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.

Field	Type	#	?	Meaning
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "PerformFusion".
↑ FusionInput	<a href="#">FusionInformationListType</a>	1	Y	Score or decision input information to the fusion method.
↑ FusionElement	<a href="#">FusionInformationType</a>	2..*	Y	A set of fusion information.
↑ BiometricType	<del>iso-iec19785-3-</del> <del>7oasis_cbeff:</del> MultipleTypes Type	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↑ BiometricSubType	<del>iso-iec19785-3-</del> <del>7oasis_cbeff:</del> SubtypeType	0..1	N	More specifically defines the type of biometric data stored in the biometric record.
↑ AlgorithmOwner	string	1	Y	The owner or vendor of the algorithm used to determine the score or decision.
↑ AlgorithmType	string	1	Y	The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision.
↑ FusionResult	<a href="#">FusionResult</a>	0..1	C	Either <a href="#">FusionScore</a> or a <a href="#">FusionDecision</a> element MUST be used.

## 544 Response Message

Field	Type	#	?	Meaning
PerformFusionResponse			Y	The response to the PerformFusion operation.
↑ PerformFusionResponsePackage		1	Y	

Field	Type	#	?	Meaning
↳ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↳ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ Match	<a href="#">MatchType</a>	1	1	Indicates the result of the fusion method.

#### 4.1.14 QueryCapabilities

[QueryCapabilitiesRequest](#)

[QueryCapabilitiesResponse](#)

The QueryCapabilities operation returns a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation. Refer to Annex A in the INCITS BIAS standard [[INCITS-BIAS](#)] for conformance requirements regarding which capability names an implementation must use in the QueryCapabilities operation.

#### Request Message

Field	Type	#	?	Meaning
QueryCapabilities			Y	Returns a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation.
↳ QueryCapabilitiesRequest		1	Y	
↳ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↳ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "QueryCapabilities".



553 **Response Message**

Field	Type	#	?	Meaning
QueryCapabilitiesResponse			Y	The response to a QueryCapabilities operation.
↑ QueryCapabilitiesResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ CapabilityList	<a href="#">CapabilityListType</a>	1	Y	A list of capabilities supported by the BIAS implementation.
↑ Capability	<a href="#">CapabilityType</a>	0..*	N	A single capability.
↑ CapabilityName	<a href="#">CapabilityName</a>	1	Y	The name of the capability.
↑ CapabilityID	string	0..1	N	An identifier assigned to the capability by the implementing system.
↑ CapabilityDescription	string	0..1	N	A description of the capability.
↑ CapabilityValue	string	0..1	N	A value assigned to the capability.
↑ CapabilitySupportingValue	string	0..1	N	A secondary value supporting the capability.
↑ CapabilityAdditionalInfo	string	0..1	N	Contains additional information for the supported capability.

554 **4.1.15 RetrieveBiographicInformation**555 [RetrieveBiographicInformationRequest](#)556 [RetrieveBiographicInformationResponse](#)

557 The RetrieveBiographicInformation operation retrieves the biographic data associated with a subject ID.

558 In the encounter-centric model, the encounter ID MAY be specified and the operation will return the

559 biographic data associated with that encounter. If the encounter ID is not specified in the encounter-

560 centric model, the operation returns the biographic information associated with the most recent  
 561 encounter.

## 562 Request Message

Field	Type	#	?	Meaning
RetrieveBiographicInformation			Y	Retrieves the biographic data associated with a subject ID.
⌞ RetrieveBiographicInformationRequest		1	Y	
⌞ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
⌞ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
⌞ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
⌞ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "RetrieveBiographicInformation".
⌞ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
⌞ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
⌞ EncounterID	<a href="#">BIASIDType</a>	0..1	N	The identifier of an encounter associated with the subject.

## 563 Response Message

Field	Type	#	?	Meaning
RetrieveBiographicInformationResponse			Y	The response to a RetrieveBiographicInformation operation.
⌞ RetrieveBiographicInformationResponsePackage		1	Y	

Field	Type	#	?	Meaning
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Includes the set of biographic data associated with a subject.
↑ BiographicData	<a href="#">BiographicDataType</a>	1	Y	An Identity's biographic data. One of the following elements MUST be present.
↑ LastName	string	0..1	C	The last name of a subject.
↑ FirstName	string	0..1	C	The first name of a subject.
↑ BiographicDataItem	<a href="#">BiographicDataItemType</a>	0..*	C	A single biographic data element.
↑ BiographicDataSet	<a href="#">BiographicDataItemType</a>	0..1	C	A set of biographic data information.

#### 4.1.16 RetrieveBiometricInformation

[RetrieveBiometricInformationRequest](#)

[RetrieveBiometricInformationResponse](#)

The RetrieveBiometricInformation operation retrieves the biometric data associated with a subject ID. In the encounter-centric model, the encounter ID MAY be specified and the operation will return the biometric data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the operation returns the biometric information associated with the most recent encounter. The operation provides an OPTIONAL input parameter to specify that only biometric data of a certain type should be retrieved.

#### Request Message

Field	Type	#	?	Meaning
RetrieveBiometricInformation			Y	Retrieves the biometric data associated with a subject ID.

Field	Type	#	?	Meaning
↑ RetrieveBiometricInformationRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "RetrieveBiometricInformation".
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	N	The identifier of an encounter associated with the subject.
↑ BiometricType	<del>iso-iec19785-3-7:Multiple-types</del> <del>oasis_cbeff:MultipleTypes</del> <a href="#">pesType</a>	0..1	N	The type of biological or behavioral data to retrieve.

## 574 Response Message

Field	Type	#	?	Meaning
RetrieveBiometricInformationResponse			Y	The response to a RetrieveBiometricInformation operation.
↑ RetrieveBiometricInformationResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Includes the biometric data associated with a subject.
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	An Identity's biometric data.
↑ BIRList	<a href="#">CBEFF_BIR_ListType</a>	1	Y	A list of CBEFF-BIR elements.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	0..*	N	CBEFF structure containing information about a biometric sample.

#### 4.1.17 SetBiographicData

[SetBiographicDataRequest](#)

[SetBiometricDataResponse](#)

The SetBiographicData operation associates biographic data to a given subject record. The identity model of the system determines whether the biographic information should replace any existing biographic information (person-centric model) or if a new encounter should be created and associated with the subject (encounter-centric model). For encounter-centric models, the encounter ID MAY be specified by the caller in order to link biographic and biometric information (assuming biometric information was previously associated using the [SetBiometricData](#) operation). If the encounter ID is omitted for the encounter-centric model, the operation returns a system-assigned encounter ID.

#### Request Message

Field	Type	#	?	Meaning
SetBiographicData			Y	Associates biographic data to a given subject record.
↑ SetBiographicDataRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "SetBiographicData".

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to store.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	N	The identifier of an encounter associated with the subject.
↑ BiographicData	<a href="#">BiographicDataType</a>	1	Y	An Identity's biographic data. One of the following elements MUST be present.
↑ LastName	string	0..1	C	The last name of a subject.
↑ FirstName	string	0..1	C	The first name of a subject.
↑ BiographicDataItem	<a href="#">BiographicDataItem</a>	0..*	C	A single biographic data element.
↑ BiographicDataSet	<a href="#">BiographicDataSetType</a>	0..1	C	A set of biographic data information.

## 586 Response Message

Field	Type	#	?	Meaning
SetBiographicDataResponse			Y	The response to a SetBiographicData operation.
↑ SetBiographicDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	0..1	C	In an encounter-centric model, identifies the encounter ID assigned to a new encounter.
↑ EncounterID	<a href="#">BIASIDType</a>	1	Y	The identifier of an encounter associated with the subject.

#### 4.1.18 SetBiometricData

[SetBiometricDataRequest](#)

[SetBiometricDataResponse](#)

The SetBiometricData operation associates biometric data to a given subject record. The identity model of the system determines whether the biometric information should replace any existing biometric information (person-centric model) or if a new encounter should be created and associated with the subject (encounter-centric model). For encounter-centric models, the encounter ID MAY be specified by the caller in order to link biographic and biometric information (assuming biographic information was previously associated using the [SetBiographicData](#) operation). If the encounter ID is omitted for the encounter-centric model, the operation returns a system-assigned encounter ID.

#### Request Message

Field	Type	#	?	Meaning
SetBiometricData			Y	Associates biometric data to a given subject record.
↑ SetBiometricDataRequest		1	Y	
↑ GenericRequestParameter s	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "SetBiometricData".
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biometric data to store.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.

Field	Type	#	?	Meaning
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	N	The identifier of an encounter associated with the subject.
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	An Identity's biometric data.
↑ BIRList	<a href="#">CBEFF_BIR_ListType</a>	1	Y	A list of CBEFF-BIR elements.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	1..*	Y	CBEFF structure containing information about a biometric sample.

## 598 Response Message

Field	Type	#	?	Meaning
SetBiometricDataResponse			Y	The response to a SetBiometricData operation.
↑ SetBiometricDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	<a href="#">BIASIdentity</a>	0..1	C	In an encounter-centric model, identifies the encounter ID assigned to a new encounter.
↑ EncounterID	<a href="#">BIASIDType</a>	1	Y	The identifier of an encounter associated with the subject.

## 599 4.1.19 TransformBiometricData

600 [TransformBiometricDataRequest](#)

601 [TransformBiometricDataResponse](#)

602 The TransformBiometricData operation transforms or processes a given biometric in one format into a  
603 new target format.



## 604 Request Message

Field	Type	#	?	Meaning
TransformBiometricData			Y	Transforms or processes a given biometric in one format into a new target format.
↑ TransformBiometricDataRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "TransformBiometricData".
↑ InputBIR	<a href="#">CBEFF_BIR_Type</a>	1	Y	Data structure containing the biometric information to be transformed.
↑ TransformOperation	unsignedLong	1	Y	Value indicating the type of transformation to perform.
↑ TransformControl	string	0..1	N	Specifies controls for the requested transform operation.  Note: This could be a compression ratio, target data format, etc.

605 NOTE: The values for TransformOperation and TransformControl are implementation specific.

## 606 Response Message

Field	Type	#	?	Meaning
TransformBiometricDataResponse			Y	The response to a TransformBiometricData operation.
↑ TransformBiometricDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
↰ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↰ Message	string	0..1	N	A short message corresponding to the return code.
↰ OutputBIR	<a href="#">CBEFF_BIR_Type</a>	0..1	N	Data structure containing the new, transformed biometric information.

## 4.1.20 UpdateBiographicData

[UpdateBiographicDataRequest](#)

[UpdateBiographicDataResponse](#)

The UpdateBiographicData operation updates the biographic data for an existing subject record. The operation replaces any existing biographic data with the new biographic data. In the encounter-centric model, the encounter ID MUST be specified.

### Request Message

Field	Type	#	?	Meaning
UpdateBiographicData			Y	Updates the biographic data for a given subject record.
↰ UpdateBiographicDataRequest		1	Y	
↰ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↰ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↰ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↰ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "UpdateBiographic Data".

Field	Type	#	?	Meaning
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to update.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
↑ BiographicData	<a href="#">BiographicDataType</a>	1	Y	An Identity's biographic data. One of the following elements MUST be present.
↑ LastName	string	0..1	C	The last name of a subject.
↑ FirstName	string	0..1	C	The first name of a subject.
↑ BiographicDataItem	<a href="#">BiographicDataItem</a>	0..*	C	A single biographic data element.
↑ BiographicDataSet	<a href="#">BiographicDataSetType</a>	0..1	C	A set of biographic data information.

## 614 Response Message

Field	Type	#	?	Meaning
UpdateBiographicDataResponse			Y	The response to an UpdateBiographicData operation.
↑ UpdateBiographicDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.

Field	Type	#	?	Meaning
↑ Message	string	0..1	N	A short message corresponding to the return code.

## 4.1.21 UpdateBiometricData

[UpdateBiometricDataRequest](#)

[UpdateBiometricDataResponse](#)

The UpdateBiometricData operation updates the biometric data for an existing subject record. The operation includes an OPTIONAL parameter indicating if the new biometric sample should be merged with the existing biometric sample. If this parameter is set to “False” or is not used in the request, the operation replaces the existing biometric sample with the new biometric sample. In the encounter-centric model, the encounter ID MUST be specified.

### Request Message

Field	Type	#	?	Meaning
UpdateBiometricData			Y	Updates a single biometric sample for a given subject record.
↑ UpdateBiometricDataRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: “UpdateBiometricData”.
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biometric data to update.
↑ SubjectID	<a href="#">BIASIDType</a>	1	Y	A system unique identifier for a subject.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

Field	Type	#	?	Meaning
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	An Identity's biometric data.
↑ BIR	<a href="#">CBEFF_BIR_Type</a>	1	Y	Contains biometric information in either a non-XML or an XML representation.
↑ Merge	boolean	0..1	N	Value indicating if the input biometric sample should be merged with any existing biometric information.

## 624 Response Message

Field	Type	#	?	Meaning
UpdateBiometricDataResponse			Y	The response to an UpdateBiometricData operation.
↑ UpdateBiometricDataResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

## 625 4.1.22 VerifySubject

626 [VerifySubjectRequest](#)

627 [VerifySubjectResponse](#)

628 The VerifySubject operation performs a 1:1 verification match between a given biometric and either a  
629 claim to identity in a given gallery or another given biometric. As such either the Identity Claim or  
630 Reference BIR input parameters are REQUIRED.

## 631 Request Message

Field	Type	#	?	Meaning
VerifySubject			Y	Performs a 1:1 verification match between a given biometric and either a claim to identity in a given gallery or another given biometric.
↑ VerifySubjectRequest		1	Y	

Field	Type	#	?	Meaning
s ↗ GenericRequestParameter	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↗ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↗ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↗ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "VerifySubject".
↗ GalleryID	<a href="#">BIASIDType</a>	0..1	C	The identifier of the gallery or population group of which the subject must be a member. Required if an Identity Claim is provided.
↗ Identity	<a href="#">BIASIdentity</a>	1	Y	Includes the identifying information and/or input and reference biometric samples.
↗ IdentityClaim	<a href="#">BIASIDType</a>	0..1	C	An identifier by which a subject is known to a particular gallery or population group. Required if a Reference BIR is not provided.
↗ BiometricData	<a href="#">BIASBiometricDataType</a>	1	Y	An Identity's biometric data.
↗ InputBIR	<a href="#">CBEFF_BIR_Type</a>	1	Y	Maps to specific INCITS BIAS elements as required by that specification.
↗ ReferenceBIR	<a href="#">CBEFF_BIR_Type</a>	0..1	C	Maps to specific INCITS BIAS elements as required by that specification. Required if an Identity Claim is not provided.

632 **Response Message**

Field	Type	#	?	Meaning
VerifySubjectResponse			Y	The response to a VerifySubject operation.
└ VerifySubjectResponsePackage		1	Y	
└┐ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
└┐ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
└┐ Message	string	0..1	N	A short message corresponding to the return code.
└┐ Match	boolean	0..1	N	Indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR.
└┐ Score	<a href="#">Score</a>	0..1	N	The score if the biometric information matched.

633 **Aggregate Operations**634 **4.2.1 Enroll**635 [EnrollRequest](#)636 [EnrollResponse](#)

637 The Enroll operation adds a new subject or, in an encounter-centric model, a new encounter to the  
638 system. This may be accomplished in a number of different ways according to system requirements  
639 and/or resources. If the Enroll operation is implemented as a synchronous service, the implementing  
640 system immediately processes the request and returns the results in the Return Data parameter. If the  
641 Enroll operation is implemented as an asynchronous service, the implementing system returns a token in  
642 the Return Data parameter, which is an indication that the request is being handled asynchronously. In  
643 this case, the [GetEnrollResults](#) operation is used to poll for the results of the Enroll request.

644 **Request Message**

Field	Type	#	?	Meaning
Enroll			Y	Adds a new subject or, in an encounter-centric model, a new encounter to the system.
└ EnrollRequest		1	Y	

Field	Type	#	?	Meaning
↑ GenericRequestParameter s	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "Enroll".
↑ ProcessingOptions	<a href="#">ProcessingOptionsType</a>	1	Y	Options that guide how the aggregate service request is processed.
↑ Option	string	0..*	N	An option supported by the implementing system.
↑ InputData	<a href="#">InformationType</a>	1	Y	Contains the input data for the operation, as required by the implementing system.

## 645 Response Message

Field	Type	#	?	Meaning
EnrollResponse			Y	The response to an Enroll operation.
↑ EnrollResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.

## 646 4.2.2 GetEnrollResults

### 647 [GetEnrollResultsRequest](#)

### 648 [GetEnrollResultsResponse](#)

649 The GetEnrollResults operation retrieves the enrollment results for the specified token. This operation is  
650 used in conjunction with the [Enroll](#) operation. If the Enroll operation is implemented as an asynchronous  
651 service, the implementing system returns a token and the GetEnrollResults operation is used to poll for  
652 the results of the original Enroll request.



653 If the service provider implements an asynchronous Enroll operation, then it MUST also implement the  
 654 GetEnrollResults operation.

## 655 Request Message

Field	Type	#	?	Meaning
GetEnrollResults			Y	Retrieves the enrollment results for the specified token.
↑ GetEnrollResultsRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetEnrollResults".
↑ Token	<a href="#">TokenType</a>	1	Y	A value used to retrieve the results of the Enroll request.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

## 656 Response Message

Field	Type	#	?	Meaning
GetEnrollResultsResponse			Y	The response to a GetEnrollResults operation.
↑ GetEnrollResultsResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.

### 4.2.3 GetIdentifyResults

[GetIdentifyResultsRequest](#)

[GetIdentifyResultsResponse](#)

The GetIdentifyResults operation retrieves the identification results for the specified token. This operation is used in conjunction with the [Identify](#) operation. If the Identify operation is implemented as an asynchronous service, the implementing system returns a token and the GetIdentifyResults operation is used to poll for the results of the original Identify request.

If the service provider implements an asynchronous Identify operation, then it MUST also implement the GetIdentifyResults operation.

### Request Message

Field	Type	#	?	Meaning
GetIdentifyResults			Y	Retrieves the identification results for the specified token
↑ GetIdentifyResultsRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetIdentifyResults".
↑ Token	<a href="#">TokenType</a>	1	Y	A value used to retrieve the results of the Identify request.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to

Field	Type	#	?	Meaning
				retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

## 668 Response Message

Field	Type	#	?	Meaning
GetIdentifyResultsResponse			Y	The response to a GetIdentifyResults operation.
↑ GetIdentifyResultsResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.

## 669 4.2.4 GetVerifyResults

### 670 [GetVerifyResultsRequest](#)

### 671 [GetVerifyResultsResponse](#)

672 The GetVerifyResults operation retrieves the verification results for the specified token. This operation is  
673 used in conjunction with the [Verify](#) operation. If the Verify operation is implemented as an asynchronous  
674 service, the implementing system returns a token and the GetVerifyResults operation is used to poll for  
675 the results of the original Verify request.

676 If the service provider implements an asynchronous Verifyoperation, then it MUST also implement the  
677 GetVerifyResults operation.

## 678 Request Message

Field	Type	#	?	Meaning
GetVerifyResults			Y	Retrieves the verification results for the specified token
↑ GetVerifyResultsRequest		1	Y	

Field	Type	#	?	Meaning
s ↗ GenericRequestParameter	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↗ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↗ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↗ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetVerifyResults".
↗ Token	<a href="#">TokenType</a>	1	Y	A value used to retrieve the results of the Verify request.
↗ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↗ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

## 679 Response Message

Field	Type	#	?	Meaning
GetVerifyResultsResponse			Y	The response to a GetVerifyResults operation.
↗ GetVerifyResultsResponsePackage		1	Y	
↗ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↗ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↗ Message	string	0..1	N	A short message corresponding to the return code.
↗ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.

Field	Type	#	?	Meaning
↑ Match	boolean	0..1	N	Indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR.
↑ Score	<a href="#">Score</a>	0..1	N	The score if the biometric information matched.

## 4.2.5 Identify

[IdentifyRequest](#)

[IdentifyResponse](#)

The Identify operation performs an identification function according to system requirements and/or resources. If the Identify operation is implemented as a synchronous service, the implementing system immediately processes the request and returns the results in the Return Data parameter. If the Identify operation is implemented as an asynchronous service, the implementing system returns a token in the Return Data parameter, which is an indication that the request is being handled asynchronously. In this case, the [GetIdentifyResults](#) operation is used to poll for the results of the Identify request.

## Request Message

Field	Type	#	?	Meaning
Identify			Y	Performs an identification function.
IdentifyRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "Identify".
↑ ProcessingOptions	<a href="#">ProcessingOptionsType</a>	1	Y	Options that guide how the aggregate service request is processed.
↑ Option	string	0..*	N	An option supported by the implementing system.
↑ InputData	<a href="#">InformationType</a>	1	Y	Contains the input data for the aggregate services.

## 690 Response Message

Field	Type	#	?	Meaning
IdentifyResponse			Y	The response to an Identify operation.
↑ IdentifyResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.

## 691 4.2.6 RetrieveInformation

### 692 [RetrieveInformationRequest](#)

### 693 [RetrieveInformationResponse](#)

694 The RetrieveInformation operation retrieves requested information about a subject, or in an encounter-  
695 centric model about an encounter. In a person-centric model, this operation can be used to retrieve both  
696 biographic and biometric information for a subject record. In an encounter-centric model, this operation  
697 can be used to retrieve biographic and/or biometric information for either a single encounter or all  
698 encounters. Either a subject ID or encounter ID MUST be specified.

## 699 Request Message

Field	Type	#	?	Meaning
RetrieveInformation			Y	Retrieves requested information about a subject or encounter.
↑ RetrieveInformationRequest		1	Y	
↑ GenericRequestParameters	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "RetrieveInformation".

Field	Type	#	?	Meaning
↑ ProcessingOptions	<a href="#">ProcessingOptionsType</a>	1	Y	Options that guide how the aggregate service request is processed, and MAY identify what type(s) of information should be returned.
↑ Option	string	0..*	N	An option supported by the implementing system.
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Includes the identifier of the subject or encounter.
↑ SubjectID	<a href="#">BIASIDType</a>	0..1	C	A system unique identifier for a subject. Required if an Encounter ID is not provided.
↑ EncounterID	<a href="#">BIASIDType</a>	0..1	C	The identifier of an encounter associated with the subject. Required if a Subject ID is not provided.

## 700 Response Message

Field	Type	#	?	Meaning
RetrieveInformationResponse			Y	Response to a RetrieveInformation operation.
↑ RetrieveInformationResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.

## 701 4.2.7 Verify

### 702 [VerifyRequest](#)

## [VerifyResponse](#)

The Verify operation performs a 1:1 verification function according to system requirements and/or resources. Either the Identity Claim or Reference BIR input parameters are REQUIRED. If the Verify operation is implemented as a synchronous service, the implementing system immediately processes the request and returns the results in the Return Data parameter. If the Verify operation is implemented as an asynchronous service, the implementing system returns a token in the Return Data parameter, which is an indication that the request is being handled asynchronously. In this case, the [GetVerifyResults](#) operation is used to poll for the results of the Verify request.

## Request Message

Field	Type	#	?	Meaning
Verify			Y	Performs a 1:1 verification function.
↑ VerifyRequest		1	Y	
↑ GenericRequestParameter s	<a href="#">GenericRequestParameters</a>	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	<a href="#">ApplicationIdentifier</a>	0..1	N	Identifies the requesting application.
↑ ApplicationUser	<a href="#">ApplicationUserIdentifier</a>	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "Verify".
↑ ProcessingOptions	<a href="#">ProcessingOptionsType</a>	1	Y	Options that guide how the aggregate service request is processed.
↑ Option	string	0..*	N	An option supported by the implementing system.
↑ InputData	<a href="#">InformationType</a>	1	Y	Contains the input data for the aggregate services.
↑ Identity	<a href="#">BIASIdentity</a>	1	Y	Includes either the Identity Claim or Reference BIR.
↑ IdentityClaim	<a href="#">BIASIDType</a>	0..1	C	An identifier by which a subject is known to a particular gallery or population group. Required if a Reference BIR is not provided.
↑ BiometricData	<a href="#">BIASBiometricDataType</a>	0..1	N	An Identity's biometric data.



Field	Type	#	?	Meaning
↑ ReferenceBIR	<a href="#">CBEFF_BIR_Type</a>	0..1	C	Maps to specific INCITS BIAS elements as required by that specification. Required if an Identity Claim is not provided.
↑ GalleryID	<a href="#">BIASIDType</a>	0..1	C	The identifier of the gallery or population group of which the subject must be a member. Required if an Identity Claim is provided.

## 712 Response Message

Field	Type	#	?	Meaning
VerifyResponse			Y	The response to a Verify operation.
↑ VerifyResponsePackage		1	Y	
↑ ResponseStatus	<a href="#">ResponseStatus</a>	1	Y	Returned status for the operation.
↑ Return	<a href="#">ReturnCode</a>	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	<a href="#">InformationType</a>	0..1	N	Contains the output data for the response.
↑ Match	boolean	0..1	N	Indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR.
↑ Score	<a href="#">Score</a>	0..1	N	The score if the biometric information matched.

713

## 5 Message structure and rules

BIAS operations and data elements are defined in XML in the INCITS 422 BIAS standard. This OASIS standard further specifies the full XML schema (see AnnexA) and specifies how this XML is packaged and exchanged as SOAP messages.

Annex A provides a WSDL of operations and structures aggregated from all the conformance classes, both synchronous and asynchronous. A specific implementation's WSDL must only expose its respective operations and structures. For example, for a Class 5-only conformant implementation, all of the primitive operations must not be exposed as operations (with the exception of QueryCapabilities) unless that functionality is supported. Additionally, the WSDL exposed by an implementation shall not contain instances of `xsd:any`, `xsd:anyType`, or `xsd:anyAttribute`; these instances must be replaced with explicit schema contents. An example is the XML complex type, `InformationType`, which has `xsd:any` as its only child. This type is used to represent implementation-specific input data and return data. The children of `InformationType` must be replaced with explicit content. Doing so removes the ability to transmit unexpected or arbitrary data. Also, it provides a clear definition of information that a client needs to provide to the server, or expect to receive, to optimally perform an operation.

SOAP 1.1 messages consist of three elements: an envelope, header data, and a message body. BIAS request-response elements MUST be enclosed within the SOAP message body. The general structure of the BIAS SOAP message is shown in Figure 4, below. The data model for BIAS is addressed in Section 3 and BIAS messages in Section 4.

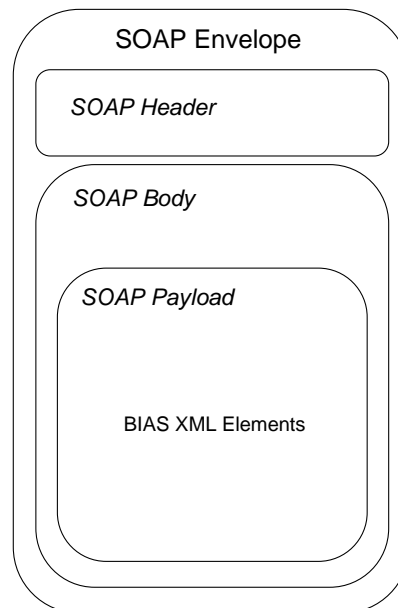


Figure 4. BIAS SOAP Structure

Biometric data, regardless of native format, is carried as a binary structure. As such, options exist on how this data is carried within the SOAP structure. It can be carried as embedded Base-64 objects or [XOP] can be used – this standard allows for either method (See section 5.3).

### 5.1 Purpose and constraints

This document defines a SOAP profile describing how the XML elements defined in INCITS 442 are to be used as the payload of a SOAP message and the rules for structuring and exchanging such messages. Philosophical tenets include:

- SOAP messages will carry BIAS XML [XML 10] payloads.
- SOAP messages will follow WS-I and will deviate only when absolutely necessary.
- Message structures and interchanges will be kept as simple as possible – “nice to have” features will be addressed in future revisions.
- XML schemas will be produced based on INCITS 442.
- BIAS will support a broad range of application domains.
- BIAS will allow for a variety of biometric and biographic data formats to be used
- Only the SOAP messaging will be defined – no message protocols or client/server agents will be defined.
- Basic usage/formatting rules (beyond WS-I) will be defined.
- Existing biometric and Web services standards will be leveraged wherever possible.
- Sample WSDL and use cases will be provided as an aid in implementation.
- Use of basic SOAP will allow all other compatible WS\* standards (and discovery mechanisms) to be used in conjunction with BIAS messaging.
- BIAS will support both secure (i.e., using existing security mechanisms such as WS-Security, SAML, etc.) and non-secure implementations.
- Generic biometric operations will be defined – use of biometrics within a Web services authentication protocol is not addressed.
- OASIS namespace rules will be followed, though some external schemas MAY also be referenced.

## 5.2 Message requirements

BIAS SOAP messages MUST conform to [WS-I-Basic] and [WS-I-Bind]. A single BIAS SOAP message MUST contain only one BIAS service request (or single BIAS service response). Binary components of BIAS messages are already Base-64 encoded and therefore do not need to be conveyed as SOAP attachments (though XOP MAY be utilized).

The system model used for BIAS conversations over SOAP is a simple request-response model. BIAS comprises both synchronous and asynchronous operations, with the majority being of the former type. Asynchronous operations are implemented through message pairs. That is, there are separate messages to request the operation and to request the results of the operation. These have been defined for those operations that are likely to take significant time to complete. For example, an identify operation can be implemented as either a synchronous or asynchronous service as follows:

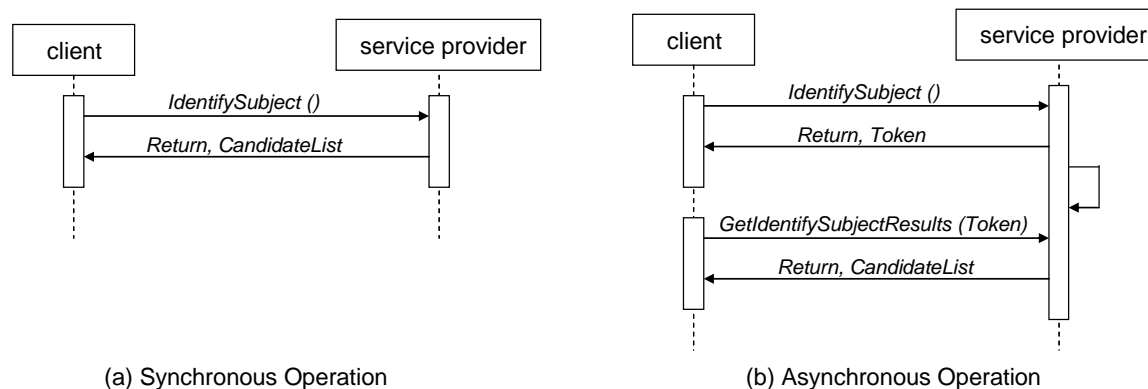


Figure 5. Example of Synchronous and Asynchronous BIAS Operations

The basic process for using SOAP for BIAS operations is:

1. A system entity acting as a BIAS requester transmits a BIAS request element within the body of a SOAP message to a system entity acting as a BIAS responder. The BIAS requester MUST NOT include more than one BIAS request per SOAP message or include any additional XML elements in the SOAP body.
2. The BIAS responder MUST return either a BIAS response element within the body of another SOAP message or generate a SOAP fault. The BIAS responder MUST NOT include more than one BIAS response per SOAP message or include any additional XML elements in the SOAP body. If a BIAS responder cannot, for some reason, process a BIAS request, it MUST generate a SOAP fault. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)
3. On receiving a BIAS response in a SOAP message, the BIAS requester MUST NOT send a fault code or other error messages to the BIAS responder. Since the format for the message interchange is a simple request-response pattern, adding additional items such as error conditions would needlessly complicate the protocol.

SOAP 1.1 also defines an optional data encoding system. This system is not used within the BIAS SOAP binding. This means that BIAS messages can be transported using SOAP without re-encoding from the “standard” BIAS schema to one based on the SOAP encoding.

NOTE: [SOAP11] references an early draft of the XML Schema specification including an obsolete namespace. BIAS requesters SHOULD generate SOAP documents referencing only the final XML schema namespace. BIAS responders MUST be able to process both the XML schema namespace used in [SOAP11] as well as the final XML schema namespace.

## 5.3 Handling binary data

BIAS messages frequently contain binary data (e.g., biometric data, scanned identity documents, etc.). Two methods are provided for dealing with this:

- Embedded Base64 encoding
- XOP [XOP]

Use of SOAP with Attachments (SWA) is deprecated.

### 5.3.1 Base64 encoding

This method is the default method for including binary data. Binary data is Base64 encoded and included between the tags in the XML SOAP body for the appropriate data elements. Data elements using this method are indicated as such in the schema.

As an example, the CBEFF\_BIR\_Type includes, as one of the BIR types, BinaryBIR of type base64binary.

```
<xsd:element name="BinaryBIR" type="xs:base64Binary" />
```

However, even an XML\_BIR as defined within [CBEFF3-Part 3.1] contains a biometric data block (BDB) which may be entirely binary (most common),

```
<xs:element name="bdb" type="xs:base64Binary" minOccurs="0"/>
```

or contain an element which is binary (e.g., an image within an XML BDB).

### 5.3.2 Use of XOP

When XOP is used, the binary content is replaced with a reference (URI) to an attachment (i.e., MIME) which contains that “stripped” content via an xop:include. The advantage of this method is overall

825 message size during transmission since the overhead of the embedded Base64 is not present (since the  
826 MIME attachment contains the native binary format).  
827 Use of XOP is generally transparent to the developer, other than in how they configure their toolset. Most  
828 frameworks support this; however, there is a possibility of mismatch if the transmitter supports and uses  
829 XOP but the receiver does not.

## 830 **5.4 Discovery**

831 BIAS implementers (service providers) MUST provide WSDL [[WSDL11](#)] to describe their  
832 implementations. This WSDL MAY or may not be made public via a standard discovery mechanism  
833 (such as UDDI) or other method.

834 In addition, it is REQUIRED that the BIAS implementation include the [QueryCapabilities](#) operation to  
835 provide dynamic information regarding BIAS capabilities, options, galleries, etc. that are supported.

## 836 **5.5 Identifying operations**

837 Receivers of BIAS SOAP messages require a method of easily identifying the operation being requested  
838 (or response being provided). This SHOULD be possible without the receiver needing to infer it from the  
839 sum of the elements provided within the body of the SOAP message. The BIAS SOAP profile allows for  
840 two methods of identifying BIAS operations:

- 841 • Explicit named element in body of the SOAP message
- 842 • Use of WS-Addressing Action element

### 843 **5.5.1 Operation name element**

844 The BIAS message sender (requester) will include within the body of the BIAS SOAP message an XML  
845 element <BIASOperationName>. The receiver (service provider) can search for this tag within a received  
846 BIAS SOAP message to determine what operation is being requested. There is no requirement related to  
847 the ordering of this element within the message, though it is RECOMMENDED that it be included early in  
848 the message to aid in human readability.

849 An example of this method for the CreateSubject operation is shown below:

```
850
851 POST /bias HTTP/1.1
852 Host: www.acme.com
853 Content-Type: application/soap+xml; charset="utf-8"
854 Content-Length: nnnn
855 SOAPAction: ""
856 <?xml version="1.0"?>
857 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
858     <soap:Body>
859         <tns:CreateSubjectRequest                                xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
860             <tns:GenericRequestParameters>
861                 <tns:Application>BIAS Application</tns:Application>
862                 <tns:ApplicationUser>BIAS User</tns:ApplicationUser>
863                 <tns:BIASOperationName>CreateSubject</tns:BIASOperationName>
864             </tns:GenericRequestParameters>
865             <tns:Identity>
```

```

868         <tns:SubjectID>123456789</tns:SubjectID>
869     </tns:Identity>
870 </tns:CreateSubjectRequest>
871 </soap:Body>
872 </soap:Envelope>

```

## 873 **5.5.2 WS-Addressing Action**

874 WS-Addressing [WS-Addr] provides a mechanism for including action information inside any SOAP  
875 message. The information is in the SOAP Header. The WS-Addressing 'Action' element is used to  
876 indicate the intent of the message. The value is a URI/IRI identifying that intent; however, there are no  
877 restrictions on the format or specificity of the URI/IRI nor a requirement that it can be resolved. Adoption  
878 of this option also requires that the WS-Addressing 'To', 'ReplyTo', and 'MessageID' elements are  
879 supplied, as they are mandatory elements in a request-reply message pattern as used within BIAS.  
880 Response messages would also need to use WS-Addressing, requiring the 'To' (matching the 'ReplyTo'  
881 element in the request), 'RelatesTo' (matching the 'MessageID' element in the request), and  
882 'RelationshipType' (default value to "wsa:Reply") elements.

883 Use of WS-Addressing is OPTIONAL in this profile as is this method of using the 'Action' field for this  
884 purpose. However, when BIAS is used within an environment using WS-Addressing, it is  
885 RECOMMENDED that this approach for use of the 'Action' field to carry the BIAS operation name is  
886 employed, either alone or in combination with the BIASOperationName approach described in section  
887 5.4.5.1.

888 An example for a message request for the CreateSubject operation would look like the following:

```

889
890     POST /bias HTTP/1.1
891     Host: www.acme.com
892     Content-Type: application/soap+xml; charset="utf-8"
893     Content-Length: nnnn
894     SOAPAction: ""
895
896     <?xml version="1.0"?>
897     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
898         xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
899         <soap:Header>
900             <wsa:MessageID>some-ID</wsa:MessageID>
901             <wsa:ReplyTo>
902                 <wsa:Address>response-URI</wsa:Address>
903             </wsa:ReplyTo>
904             <wsa:To>destination-URI</wsa:To>
905             <wsa:Action>CreateSubject</wsa:Action>
906         </soap:Header>
907         <soap:Body>
908             <tns:CreateSubjectRequest
909                 xmlns:tns="http://docs.oasis-
910 open.org/bias/bias-1.0/" />
911         </soap:Body>
912     </soap:Envelope>

```

913

## 914 **5.6 Security**

915 The end-points that exchange SOAP messages (or handle the contents of the BIAS operations) are  
916 expected to be protected and trusted such that message-level security mechanisms may not be required.  
917 The use of SSL (HTTPS) or VPN technology that provides end-point to end-point security is  
918 RECOMMENDED and MAY be sufficient in some cases. Other mechanisms such as Signed XML or  
919 WSS [WSS] could also be implemented.

920 Unless stated otherwise, the following security statements apply to all BIAS bindings.

### 921 **5.6.1 Use of SSL 3.0 or TLS 1.0**

922 Unless otherwise specified, in any BIAS binding's use of SSL 3.0 [SSL3] or TLS1.0 [RFC2246], servers  
923 MUST authenticate clients using a X.509 v3 certificate [X509]. The client MUST establish server identity  
924 based on contents of the certificate (typically through examination of the certificate's subject DN field,  
925 subjectAltName attribute, etc.).

926 Use of transport level security in the form of SSL or TLS is OPTIONAL but highly RECOMMENDED. Use  
927 of these mechanisms alone may not be sufficient for end-to-end integrity and confidentiality, however  
928 (see 5.56.3 and 5.56.4 below).

### 929 **5.6.2 Data Origin Authentication**

930 Authentication of both the BIAS requester and the BIAS responder associated with a message is  
931 OPTIONAL and depends on the environment of use: Authentication mechanisms available at the SOAP  
932 message exchange layer or from the underlying substrate protocol (for example, in many bindings the  
933 SSL/TLS or HTTP protocol) MAY be utilized to provide data origin authentication.

934 Transport authentication will not meet end-to-end origin authentication requirements in bindings where  
935 the BIAS SOAP message passes through an intermediary – in this case, message authentication is  
936 RECOMMENDED.

937 Note that SAML [SAML] MAY be used as the mechanism for parties to authenticate to one another.

### 938 **5.6.3 Message Integrity**

939 Message integrity of both BIAS requests and BIAS responses is OPTIONAL and depends on the  
940 environment of use. The security layer in the underlying substrate protocol or a mechanism at the SOAP  
941 message exchange layer MAY be used to ensure message integrity.

942 Transport integrity will not meet end-to-end integrity requirements in bindings where the BIAS SOAP  
943 message passes through an intermediary – in this case, message integrity is RECOMMENDED.

### 944 **5.6.4 Message Confidentiality**

945 Message confidentiality of both BIAS requests and BIAS responses is OPTIONAL and depends on the  
946 environment of use. The security layer in the underlying substrate protocol or a mechanism at the SOAP  
947 message exchange layer MAY be used to ensure message confidentiality.

948 Transport confidentiality will not meet end-to-end confidentiality requirements in bindings where the BIAS  
949 SOAP message passes through an intermediary.

950 NOTE: Biometric and biographic data is likely to contain personal information the confidentiality of  
951 which SHOULD be protected accordingly. See INCITS 442, section 6.3 for further discussion.

### 952 **5.6.5 CBEFF BIR security features**

953 Within BIAS, biometric data is transferred within a CBEFF BIR (as defined in ISO/IEC 19785-1). CBEFF  
954 provides for the optional encryption of the Biometric Data Block (BDB) of the BIR and for the integrity of  
955 the entire BIR. If implemented, this is indicated in the BIR header. The BIR structure defines an optional



Security Block which MAY contain a digital signature (or message authentication code), encryption parameters (e.g., key name, algorithm, etc.), and/or other security related data. Such protections are associated with an individual BIR and are separate from any other protections provided at the message level.

### **5.6.6 Security Considerations**

Before deployment, each combination of authentication, message integrity, and confidentiality mechanisms SHOULD be analyzed for vulnerability in the context of the specific protocol exchange and the deployment environment.

Special care should be given to the impact of possible caching on security.

IETF RFC 2617 [RFC2617] describes possible attacks in the HTTP environment when basic or message digest authentication schemes are used.

Many of the security considerations identified in [SAML SEC] MAY also apply.

ISO/IEC 19092 [BIO SEC] describes a security framework for biometric systems including a minimum set of security requirements addressing integrity, authenticity, and confidentiality of biometric information during transmission and storage. These SHOULD be considered as part of an overall risk management approach.

NOTE: The requirements of ISO/IEC 19092, though useful across many application domains, are required for most biometric system implementations in the financial services environment. Application of this standard would make the requirements of sections 5.5.3 through 5.5.5 mandatory rather than optional. This is highly RECOMMENDED for any high security environment or where privacy concerns exist.

### **5.6.7 Security of Stored Data**

This specification does not address security considerations for stored data. It is the purview of the BIAS service provider to implement security mechanisms and protect data at rest as per their own security policies.

### **5.6.8 Key Management**

This specification does not address key management considerations with respect to implementation of cryptographic security mechanisms (e.g., for authenticity, integrity, or confidentiality).

## **5.7 Use with other WS\* standards**

The intent of specifying SOAP bindings for BIAS messages is to enable the full range of existing Web services standards to be able to be applied. Some may be normative while others can be optionally applied (i.e., WS-Security, WS-Addressing). Still others may require additional profiling to be used in an interoperable manner (e.g., WS-Notification); this is left to a future revision. However, the intent is to avoid specifying anything in the first, base version that would preclude the use of such standards in the future.

## **5.8 Tailoring**

This standard provides for a common method of implementing biometric Web services; however, it does not guarantee interoperability in a specific application. In some cases further tailoring or profiling of this standard may be required in order to further constrain the implementation options available.

NOTE: As an example, BIAS allows for a number of different biographic and biometric data formats to be used, whereas a given application/domain MAY wish to limit this to a small set or just one of each type. Other examples (not comprehensive) include:

- Identification of a subset of BIAS operations to be used
- Specification of security features to be implemented (e.g., SSL, CBEFF BIR encryption, etc.)
- Choice of operation name identification method



- 1000
  - Choice of BIR type to be used (XML, non-XML, or URI)
- 1001
  - Further definition of aggregate operations
- 1002
  - Use (or not) of the encounter model
- 1003
  - Use (or not) of asynchronous operations
- 1004
  - Process sequences
- 1005
  - Implementation specific values (e.g., Transform operations/controls)

---

## 6. Error handling

There are two levels of errors that can be returned in an error response: system and service errors.

- System-level errors occur when the implementing system cannot service a request. They could result due to an internal logic error or because the implementing system does not support a particular request.
- Service-level errors occur when there is a problem transmitting or representing the service request. They could result due to an invalid service request or because of a communications error.

The INCITS BIAS standard defines the error condition codes for system-level errors.

- If successful, a response message (containing a return code) will be generated.
- If unsuccessful, a SOAP fault message (containing a fault code) will be generated.

### 6.1 BIAS operation return codes

If a BIAS operation is successful, a response (service output) will be sent to the requester by the service provider. Each response message contains a response status (see section 3.2.37) and return code (see section 3.2.38) along with any response data as defined for that operation, if any. A response code of '0' indicates success.

### 6.2 SOAP fault codes

If a BIAS operation is unsuccessful, no BIAS response message is sent. Instead a SOAP fault message is returned.

Every Web service (operation) described in the BIAS WSDL may result in a fault message that will be returned in the response by the service provider in the event of an error. The fault message contains a FaultCode element as defined by the SOAP 1.1 specification (see section 3.2.5). The fault message MUST contain a Detail element in a common format, as described by the BIASFault element (see section 3.2.6).

The schema provided in Annex A defines "BIASFaultCode" and "BIASFaultDetail" types as well as "BIASFault", "BIASFaultType", "BIASFaultMessage" and "BIASFaultDescription" elements.

The list of defined BIAS fault codes is provided in section 3.2.5. Note that BIAS service providers MAY define additional fault codes unique to their service.

NOTE: See also section 5.2 for additional information on message returns and faults.

---

## 7. Conformance

Implementations claiming conformance to this standard, MUST implement, at a minimum, all mandatory requirements and provisions set forth in Clauses 3, 4, 5 and 6. If such implementations claim conformance to any OPTIONAL requirements and provisions stated in Clauses 3, 4, 5 and 6, these requirements and provisions MUST be implemented as set forth in these Clauses.

INCITS 442 [\[INCITS-BIAS\]](#) (Annex A) specifies five BIAS conformance classes. For each class, a set of mandatory BIAS operations is identified in order for implementations (BIAS service providers) to claim conformance. These categories are:

- Class 1: Full Primitive Services Implementation
- Class 2: Full Aggregate Services Implementation
- Class 3: Limited Primitive Services Implementation
- Class 4: Minimum Primitive Services Implementation
- Class 5: Minimum Aggregate Services Implementation

In addition, the minimum capability information to be returned in response to a Query Capabilities request (the only mandatory BIAS operation across all 5 classes) is specified for each class.

These conformance classes and their associated requirements apply to this BIAS SOAP Profile.

There are no minimum set of operations required to be implemented by BIAS requesters; however, any operations implemented must conform to the requirements of Clauses 3 and 4 and those requirements within Clause 5 that are mandatory and are not specific to BIAS responders.

---

## Annex A. XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Document : BIAS.xml
Created on : December 17, 2009, 9:31 AM
Last Edit : December 6, 2010 by Kevin Mangold, National Institute of Standards and Technology
Description:
This document represents the WSDL definitions of BIAS Profile.
-->
<wsdl:definitions
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:iso-iec19785-3-7="urn:oid:1.1.19785.0.257.1.7.0"
  xmlns:oasis_cbeff="http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/"
  xmlns:tns="http://docs.oasis-open.org/bias/ns/bias-1.0/"
  targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">
  <wsdl:types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">
      <xsd:import namespace="urn:oid:1.1.19785.0.257.1.7.0" http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/" schemaLocation="cbeff.xsd" />
      <xsd:complexType name="AggregateRequestTemplate">
        <xsd:annotation>
          <xsd:documentation>Base template for BIAS aggregate service requests.</xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
          <xsd:extension base="tns:RequestTemplate">
            <xsd:sequence>
              <xsd:element name="ProcessingOptions" type="tns:ProcessingOptionsType" minOccurs="1" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Options that guide how the aggregate service request is processed.</xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="InputData" type="tns:InformationType" minOccurs="1" maxOccurs="1">
                <xsd:annotation>
```

```

1093 <xsd:documentation>Contains the input data for the aggregate service request.</xsd:documentation>
1094 </xsd:annotation>
1095 </xsd:element>
1096 </xsd:sequence>
1097 </xsd:extension>
1098 </xsd:complexContent>
1099 </xsd:complexType>
1100 <xsd:complexType name="AggregateResponseTemplate">
1101 <xsd:annotation>
1102 <xsd:documentation>Base template for BIAS aggregate service responses.</xsd:documentation>
1103 </xsd:annotation>
1104 <xsd:complexContent>
1105 <xsd:extension base="tns:ResponseTemplate">
1106 <xsd:sequence>
1107 <xsd:element name="ReturnData" type="tns:InformationType" minOccurs="0" maxOccurs="1">
1108 <xsd:annotation>
1109 <xsd:documentation>Contains the output data for the aggregate service response.</xsd:documentation>
1110 </xsd:annotation>
1111 </xsd:element>
1112 </xsd:sequence>
1113 </xsd:extension>
1114 </xsd:complexContent>
1115 </xsd:complexType>
1116 <xsd:simpleType name="ApplicationIdentifier">
1117 <xsd:annotation>
1118 <xsd:documentation>Identifies an application.</xsd:documentation>
1119 </xsd:annotation>
1120 <xsd:restriction base="xsd:string">
1121 <xsd:minLength value="1"/>
1122 <xsd:maxLength value="255"/>
1123 </xsd:restriction>
1124 </xsd:simpleType>
1125 <xsd:simpleType name="ApplicationUserIdentifier">
1126 <xsd:annotation>
1127 <xsd:documentation>Identifies an application user or instance.</xsd:documentation>
1128 </xsd:annotation>
1129 <xsd:restriction base="xsd:string">
1130 <xsd:minLength value="1"/>
1131 <xsd:maxLength value="255"/>
1132 </xsd:restriction>
1133 </xsd:simpleType>

```

```

1134 <xsd:complexType name="BIASBiometricDataType">
1135 <xsd:annotation>
1136 <xsd:documentation>Wraps the various BIAS biometric types.</xsd:documentation>
1137 </xsd:annotation>
1138 <xsd:sequence>
1139 <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="0" maxOccurs="1">
1140 <xsd:annotation>
1141 <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1142 </xsd:annotation>
1143 </xsd:element>
1144 <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1145 <xsd:annotation>
1146 <xsd:documentation>Contains biometric information in either a non-XML and an XML
1147 representation.</xsd:documentation>
1148 </xsd:annotation>
1149 </xsd:element>
1150 <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1151 <xsd:annotation>
1152 <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
1153 specification.</xsd:documentation>
1154 </xsd:annotation>
1155 </xsd:element>
1156 <xsd:element name="ReferenceBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1157 <xsd:annotation>
1158 <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
1159 specification.</xsd:documentation>
1160 </xsd:annotation>
1161 </xsd:element>
1162 <xsd:element name="BiometricDataList" type="tns:BiometricDataListType" minOccurs="0" maxOccurs="1">
1163 <xsd:annotation>
1164 <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1165 </xsd:annotation>
1166 </xsd:element>
1167 </xsd:sequence>
1168 </xsd:complexType>
1169 <xsd:simpleType name="BIASFaultCode">
1170 <xsd:annotation>
1171 <xsd:documentation></xsd:documentation>
1172 </xsd:annotation>
1173 <xsd:restriction base="xsd:string">
1174 <xsd:enumeration value="UNKNOWN_ERROR">

```

1175 <xsd:annotation>  
 1176 <xsd:documentation>The service failed for an unknown reason.</xsd:documentation>  
 1177 </xsd:annotation>  
 1178 </xsd:enumeration>  
 1179 <xsd:enumeration value="UNSUPPORTED\_CAPABILITY">  
 1180 <xsd:annotation>  
 1181 <xsd:documentation>A requested capability is not supported by the service  
 1182 implementation.</xsd:documentation>  
 1183 </xsd:annotation>  
 1184 </xsd:enumeration>  
 1185 <xsd:enumeration value="INVALID\_INPUT">  
 1186 <xsd:annotation>  
 1187 <xsd:documentation>The data in a service input parameter is invalid.</xsd:documentation>  
 1188 </xsd:annotation>  
 1189 </xsd:enumeration>  
 1190 <xsd:enumeration value="BIR\_QUALITY\_ERROR">  
 1191 <xsd:annotation>  
 1192 <xsd:documentation>Biometric sample quality is too poor for the service to succeed.</xsd:documentation>  
 1193 </xsd:annotation>  
 1194 </xsd:enumeration>  
 1195 <xsd:enumeration value="INVALID\_BIR">  
 1196 <xsd:annotation>  
 1197 <xsd:documentation>The input BIR is empty or in an invalid or unrecognized format.</xsd:documentation>  
 1198 </xsd:annotation>  
 1199 </xsd:enumeration>  
 1200 <xsd:enumeration value="BIR\_SIGNATURE\_FAILURE">  
 1201 <xsd:annotation>  
 1202 <xsd:documentation>The service could not validate the signature, if used, on the input BIR.</xsd:documentation>  
 1203 </xsd:annotation>  
 1204 </xsd:enumeration>  
 1205 <xsd:enumeration value="BIR\_DECRYPTION\_FAILURE">  
 1206 <xsd:annotation>  
 1207 <xsd:documentation>The service could not decrypt an encrypted input BIR.</xsd:documentation>  
 1208 </xsd:annotation>  
 1209 </xsd:enumeration>  
 1210 <xsd:enumeration value="INVALID\_ENCOUNTER\_ID">  
 1211 <xsd:annotation>  
 1212 <xsd:documentation>The input encounter ID is empty or in an invalid format.</xsd:documentation>  
 1213 </xsd:annotation>  
 1214 </xsd:enumeration>  
 1215 <xsd:enumeration value="INVALID\_SUBJECT\_ID">

1216 <xsd:annotation>  
1217 <xsd:documentation>The input subject ID is empty or in an invalid format.</xsd:documentation>  
1218 </xsd:annotation>  
1219 </xsd:enumeration>  
1220 <xsd:enumeration value="UNKNOWN\_SUBJECT">  
1221 <xsd:annotation>  
1222 <xsd:documentation>The subject referenced by the input subject ID does not exist.</xsd:documentation>  
1223 </xsd:annotation>  
1224 </xsd:enumeration>  
1225 <xsd:enumeration value="UNKNOWN\_GALLERY">  
1226 <xsd:annotation>  
1227 <xsd:documentation>The gallery referenced by the input gallery ID does not exist.</xsd:documentation>  
1228 </xsd:annotation>  
1229 </xsd:enumeration>  
1230 <xsd:enumeration value="UNKNOWN\_ENCOUNTER">  
1231 <xsd:annotation>  
1232 <xsd:documentation>The encounter referenced by the input encounter ID does not exist.</xsd:documentation>  
1233 </xsd:annotation>  
1234 </xsd:enumeration>  
1235 <xsd:enumeration value="UNKNOWN\_BIOGRAPHIC\_FORMAT">  
1236 <xsd:annotation>  
1237 <xsd:documentation>The biographic data format is not known or not supported.</xsd:documentation>  
1238 </xsd:annotation>  
1239 </xsd:enumeration>  
1240 <xsd:enumeration value="UNKNOWN\_IDENTITY\_CLAIM">  
1241 <xsd:annotation>  
1242 <xsd:documentation>The identity referenced by the input identity claim does not exist.</xsd:documentation>  
1243 </xsd:annotation>  
1244 </xsd:enumeration>  
1245 <xsd:enumeration value="INVALID\_IDENTITY\_CLAIM">  
1246 <xsd:annotation>  
1247 <xsd:documentation>The identity claim requested is already in use.</xsd:documentation>  
1248 </xsd:annotation>  
1249 </xsd:enumeration>  
1250 <xsd:enumeration value="NONEXISTANT\_DATA">  
1251 <xsd:annotation>  
1252 <xsd:documentation>The data requested for deletion does not exist.</xsd:documentation>  
1253 </xsd:annotation>  
1254 </xsd:enumeration>  
1255 </xsd:restriction>  
1256 </xsd:simpleType>



```

1257 <xsd:complexType name="BIASFaultDetail">
1258 <xsd:annotation>
1259 <xsd:documentation>Defines the error information associated with a SOAP fault.</xsd:documentation>
1260 </xsd:annotation>
1261 <xsd:sequence>
1262 <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultType" type="tns:BIASFaultCode">
1263 <xsd:annotation>
1264 <xsd:documentation>References an error code.</xsd:documentation>
1265 </xsd:annotation>
1266 </xsd:element>
1267 <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultMessage" type="xsd:string">
1268 <xsd:annotation>
1269 <xsd:documentation>Provides an explanation of the fault.</xsd:documentation>
1270 </xsd:annotation>
1271 </xsd:element>
1272 <xsd:element minOccurs="0" maxOccurs="1" name="BIASFaultDescription" type="xsd:string">
1273 <xsd:annotation>
1274 <xsd:documentation>Provides detailed information about a BIAS fault, such as trace details.</xsd:documentation>
1275 </xsd:annotation>
1276 </xsd:element>
1277 </xsd:sequence>
1278 </xsd:complexType>
1279 <xsd:complexType name="BIASIdentity">
1280 <xsd:annotation>
1281 <xsd:documentation>
1282 Defines a single element for encapsulating the data associated
1283 with an Identity. Includes the Identity's reference identifiers,
1284 biographic data, and biometric data.
1285 </xsd:documentation>
1286 </xsd:annotation>
1287 <xsd:sequence>
1288 <xsd:element name="SubjectID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1289 <xsd:annotation>
1290 <xsd:documentation>A system unique identifier for a subject.</xsd:documentation>
1291 </xsd:annotation>
1292 </xsd:element>
1293 <xsd:element name="IdentityClaim" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1294 <xsd:annotation>
1295 <xsd:documentation>An identifier by which a subject is known to a particular gallery or population
1296 group.</xsd:documentation>
1297 </xsd:annotation>

```

```

1298 </xsd:element>
1299 <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1300 <xsd:annotation>
1301 <xsd:documentation>The identifier of an encounter associated with the subject, required for encounter-centric
1302 models.</xsd:documentation>
1303 </xsd:annotation>
1304 </xsd:element>
1305 <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
1306 <xsd:annotation>
1307 <xsd:documentation>A list of encounters associated with a subject.</xsd:documentation>
1308 </xsd:annotation>
1309 </xsd:element>
1310 <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1311 <xsd:annotation>
1312 <xsd:documentation>An Identity's biographic data.</xsd:documentation>
1313 </xsd:annotation>
1314 </xsd:element>
1315 <xsd:element name="BiographicDataElements" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1316 <xsd:annotation>
1317 <xsd:documentation>An Identity's biographic data elements that are stored in the implementing
1318 system.</xsd:documentation>
1319 </xsd:annotation>
1320 </xsd:element>
1321 <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="0" maxOccurs="1">
1322 <xsd:annotation>
1323 <xsd:documentation>An Identity's biometric data.</xsd:documentation>
1324 </xsd:annotation>
1325 </xsd:element>
1326 </xsd:sequence>
1327 </xsd:complexType>
1328 <xsd:simpleType name="BIASIDType">
1329 <xsd:annotation>
1330 <xsd:documentation>A BIAS identifier</xsd:documentation>
1331 </xsd:annotation>
1332 <xsd:restriction base="xsd:string"/>
1333 </xsd:simpleType>
1334 <xsd:complexType name="BiographicDataItem">
1335 <xsd:annotation>
1336 <xsd:documentation>Defines a single biographic data element.</xsd:documentation>
1337 </xsd:annotation>
1338 <xsd:sequence>

```

```

1339 <xsd:element name="Name" type="xsd:string" minOccurs="1" maxOccurs="1">
1340 <xsd:annotation>
1341 <xsd:documentation>The name of the biographic data item.</xsd:documentation>
1342 </xsd:annotation>
1343 </xsd:element>
1344 <xsd:element name="Type" type="xsd:string" minOccurs="1" maxOccurs="1">
1345 <xsd:annotation>
1346 <xsd:documentation>The data type for the biographic data item.</xsd:documentation>
1347 </xsd:annotation>
1348 </xsd:element>
1349 <xsd:element name="Value" type="xsd:string" minOccurs="0" maxOccurs="1">
1350 <xsd:annotation>
1351 <xsd:documentation>The value assigned to the biographic data item.</xsd:documentation>
1352 </xsd:annotation>
1353 </xsd:element>
1354 </xsd:sequence>
1355 </xsd:complexType>
1356 <xsd:complexType name="BiographicDataSetType">
1357 <xsd:annotation>
1358 <xsd:documentation>Defines a set of biographic data that is formatted according to the specified
1359 format.</xsd:documentation>
1360 </xsd:annotation>
1361 <xsd:sequence>
1362 <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">
1363 <xsd:annotation>
1364 <xsd:documentation>The name of the biographic data format. Use these names for common formats: FBI-EFTS,
1365 FBI-EBTS, DOD-EBTS, INT-I, NIEM, xNAL, HR-XML.</xsd:documentation>
1366 </xsd:annotation>
1367 </xsd:element>
1368 <xsd:element name="version" type="xsd:string" minOccurs="0" maxOccurs="1">
1369 <xsd:annotation>
1370 <xsd:documentation>The version of the biographic data format (e.g., "7.1" for FBI-EFTS or "2.0" for
1371 NIEM).</xsd:documentation>
1372 </xsd:annotation>
1373 </xsd:element>
1374 <xsd:element name="source" type="xsd:string" minOccurs="1" maxOccurs="1">
1375 <xsd:annotation>
1376 <xsd:documentation>Reference to a URI/IRI describing the biographic data format. For example: (FBI-EFTS)
1377 www.fbibiospecs.org, (DOD-EBTS) www.biometrics.dod.mil, (INT-I) www.interpol.int, (NIEM) www.niem.gov,
1378 (xNAL) www.oasis-open.org, (HR-XML) www.hr-xml.org.</xsd:documentation>
1379 </xsd:annotation>
1380 </xsd:element>

```

```

1381 <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1">
1382 <xsd:annotation>
1383 <xsd:documentation>The biographic data format type. Use these types for common formats: ASCII (e.g., for non-
1384 XML versions of FBI-EFTS, FBI-EBTS, DOD-EFTS, or INT-I), XML (e.g., for NIEM, xNAL, and HR-XML or future version
1385 of FBI-EBTS).</xsd:documentation>
1386
1387 </xsd:annotation>
1388 </xsd:element>
1389 <xsd:any namespace="##any">
1390 <xsd:annotation>
1391 <xsd:documentation>Biographic data formatted according to a specific format.</xsd:documentation>
1392 </xsd:annotation>
1393 </xsd:any>
1394 </xsd:sequence>
1395 </xsd:complexType>
1396 <xsd:complexType name="BiographicDataType">
1397 <xsd:annotation>
1398 <xsd:documentation>
1399 Defines a set of biographic data elements, utilizing either the
1400 BiographicDataItem type to represent a list of elements or the
1401 BiographicDataSet type to represent a complete, formatted set of
1402 biographic information.
1403 </xsd:documentation>
1404 </xsd:annotation>
1405 <xsd:sequence>
1406 <xsd:element name="LastName" type="xsd:string" minOccurs="0" maxOccurs="1">
1407 <xsd:annotation>
1408 <xsd:documentation>The last name of a subject.</xsd:documentation>
1409 </xsd:annotation>
1410 </xsd:element>
1411 <xsd:element name="FirstName" type="xsd:string" minOccurs="0" maxOccurs="1">
1412 <xsd:annotation>
1413 <xsd:documentation>The first name of a subject.</xsd:documentation>
1414 </xsd:annotation>
1415 </xsd:element>
1416 <xsd:element name="BiographicDataItems" minOccurs="0">
1417 <xsd:complexType>
1418 <xsd:sequence>
1419 <xsd:element name="BiographicDataItem" type="tns:BiographicDataItem" maxOccurs="unbounded">
1420 <xsd:annotation>
1421 <xsd:documentation>A single biographic data element.</xsd:documentation>

```

```

1422 </xsd:annotation>
1423 </xsd:element>
1424 </xsd:sequence>
1425 </xsd:complexType>
1426 </xsd:element>
1427 <xsd:element name="BiographicDataSet" type="tns:BiographicDataSetType" minOccurs="0" maxOccurs="1">
1428 <xsd:annotation>
1429 <xsd:documentation>A set of biographic data information.</xsd:documentation>
1430 </xsd:annotation>
1431 </xsd:element>
1432 </xsd:sequence>
1433 </xsd:complexType>
1434 <xsd:complexType name="BiometricDataElementType">
1435 <xsd:annotation>
1436 <xsd:documentation>
1437 Provides descriptive information about biometric data, such as
1438 the biometric type, subtype, and format, contained in the BDB of
1439 the CBEFF-BIR.
1440 </xsd:documentation>
1441 </xsd:annotation>
1442 <xsd:sequence>
1443 | <xsd:element name="BiometricType" type="iso-iec19785-3-7oasis\_cbeff:MultipleTypesType" minOccurs="1"
1444 maxOccurs="1">
1445 <xsd:annotation>
1446 <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1447 CBEFF.</xsd:documentation>
1448 </xsd:annotation>
1449 </xsd:element>
1450 <xsd:element name="BiometricTypeCount" type="xsd:positiveInteger" minOccurs="0" maxOccurs="1">
1451 <xsd:annotation>
1452 <xsd:documentation>The number of biometric records having the biometric type recorded in the biometric type
1453 field.</xsd:documentation>
1454 </xsd:annotation>
1455 </xsd:element>
1456 | <xsd:element name="BiometricSubType" type="iso-iec19785-3-7oasis\_cbeff:SubtypeType" minOccurs="0"
1457 maxOccurs="1">
1458 <xsd:annotation>
1459 <xsd:documentation>More specifically defines the type of biometric data stored in the biometric record, as
1460 defined by CBEFF.</xsd:documentation>
1461 </xsd:annotation>
1462 </xsd:element>
1463 <xsd:element name="BDBFormatOwner" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">

```

```

1464 <xsd:annotation>
1465 <xsd:documentation>Identifies the standards body, working group, industry consortium, or other CBEFF biometric
1466 organization that has defined the format for the biometric data.</xsd:documentation>
1467 </xsd:annotation>
1468 </xsd:element>
1469 <xsd:element name="BDBFormatType" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">
1470 <xsd:annotation>
1471 <xsd:documentation>Identifies the specific biometric data format specified by the CBEFF biometric organization
1472 recorded in the BDB Format Owner field.</xsd:documentation>
1473 </xsd:annotation>
1474 </xsd:element>
1475 </xsd:sequence>
1476 </xsd:complexType>
1477 <xsd:complexType name="BiometricDataListType">
1478 <xsd:annotation>
1479 <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1480 </xsd:annotation>
1481 <xsd:sequence>
1482 <xsd:element name="BiometricDataElement" type="tns:BiometricDataElementType" minOccurs="0"
1483 maxOccurs="unbounded">
1484 <xsd:annotation>
1485 <xsd:documentation>Data structure containing information about a biometric record.</xsd:documentation>
1486 </xsd:annotation>
1487 </xsd:element>
1488 </xsd:sequence>
1489 </xsd:complexType>
1490 <xsd:complexType name="CandidateListType">
1491 <xsd:annotation>
1492 <xsd:documentation>
1493 Defines a set of candidates, utilizing the Candidate Type to
1494 represent each element in the set.
1495 </xsd:documentation>
1496 </xsd:annotation>
1497 <xsd:sequence>
1498 <xsd:element name="Candidate" type="tns:CandidateType" minOccurs="0" maxOccurs="unbounded">
1499 <xsd:annotation>
1500 <xsd:documentation>A single candidate.</xsd:documentation>
1501 </xsd:annotation>
1502 </xsd:element>
1503 </xsd:sequence>
1504 </xsd:complexType>

```

```

1505 <xsd:complexType name="CandidateType">
1506 <xsd:annotation>
1507 <xsd:documentation>
1508 Defines a single candidate as a possible match in response to a
1509 biometric identification request.
1510 </xsd:documentation>
1511 </xsd:annotation>
1512 <xsd:sequence>
1513 <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
1514 <xsd:annotation>
1515 <xsd:documentation>The match score.</xsd:documentation>
1516 </xsd:annotation>
1517 </xsd:element>
1518 <xsd:element name="Rank" type="xsd:integer" minOccurs="1" maxOccurs="1">
1519 <xsd:annotation>
1520 <xsd:documentation>The rank of the candidate in relation to other candidates for the same biometric
1521 identification operation.</xsd:documentation>
1522 </xsd:annotation>
1523 </xsd:element>
1524 <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1525 <xsd:annotation>
1526 <xsd:documentation>Biographic data associated with the candidate match.</xsd:documentation>
1527 </xsd:annotation>
1528 </xsd:element>
1529 <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="1" maxOccurs="1">
1530 <xsd:annotation>
1531 <xsd:documentation>Biometric data associated with the candidate match.</xsd:documentation>
1532 </xsd:annotation>
1533 </xsd:element>
1534 </xsd:sequence>
1535 </xsd:complexType>
1536 <xsd:complexType name="CapabilityListType">
1537 <xsd:annotation>
1538 <xsd:documentation>Defines a set of capabilities.</xsd:documentation>
1539 </xsd:annotation>
1540 <xsd:sequence>
1541 <xsd:element name="Capability" type="tns:CapabilityType" minOccurs="0" maxOccurs="unbounded">
1542 <xsd:annotation>
1543 <xsd:documentation>A single capability.</xsd:documentation>
1544 </xsd:annotation>
1545 </xsd:element>

```

```

1546 </xsd:sequence>
1547 </xsd:complexType>
1548 <xsd:simpleType name="CapabilityName">
1549 <xsd:annotation>
1550 <xsd:documentation>A list of capability items.</xsd:documentation>
1551 </xsd:annotation>
1552 <xsd:restriction base="xsd:string">
1553 <xsd:enumeration value="AggregateInputDataOptional">
1554 <xsd:annotation>
1555 <xsd:documentation>A data element accepted as optional input by the implementing system for the aggregate
1556 services.</xsd:documentation>
1557 </xsd:annotation>
1558 </xsd:enumeration>
1559 <xsd:enumeration value="AggregateInputDataRequired">
1560 <xsd:annotation>
1561 <xsd:documentation>A data element required as input by the implementing system for the aggregate
1562 services.</xsd:documentation>
1563 </xsd:annotation>
1564 </xsd:enumeration>
1565 <xsd:enumeration value="AggregateProcessingOption">
1566 <xsd:annotation>
1567 <xsd:documentation>A processing option supported by the implementing system for the aggregate
1568 services.</xsd:documentation>
1569 </xsd:annotation>
1570 </xsd:enumeration>
1571 <xsd:enumeration value="AggregateReturnData">
1572 <xsd:annotation>
1573 <xsd:documentation>A data element returned by the implementing system for the aggregate
1574 services.</xsd:documentation>
1575 </xsd:annotation>
1576 </xsd:enumeration>
1577 <xsd:enumeration value="AggregateServiceDescription">
1578 <xsd:annotation>
1579 <xsd:documentation>Describes the processing logic of an aggregate service supported by the implementing
1580 system.</xsd:documentation>
1581 </xsd:annotation>
1582 </xsd:enumeration>
1583 <xsd:enumeration value="BiographicDataSet">
1584 <xsd:annotation>
1585 <xsd:documentation>Identifies a biographic data set supported by the implementing
1586 system.</xsd:documentation>
1587 </xsd:annotation>

```



```

1588 </xsd:enumeration>
1589 <xsd:enumeration value="CBEFFPatronFormat">
1590 <xsd:annotation>
1591 <xsd:documentation>A patron format supported by the implementing system.</xsd:documentation>
1592 </xsd:annotation>
1593 </xsd:enumeration>
1594 <xsd:enumeration value="ClassificationAlgorithmType">
1595 <xsd:annotation>
1596 <xsd:documentation>A classification algorithm type supported by the implementing system.</xsd:documentation>
1597 </xsd:annotation>
1598 </xsd:enumeration>
1599 <xsd:enumeration value="ConformanceClass">
1600 <xsd:annotation>
1601 <xsd:documentation>Identifies the conformance class of the BIAS implementation.</xsd:documentation>
1602 </xsd:annotation>
1603 </xsd:enumeration>
1604 <xsd:enumeration value="Gallery">
1605 <xsd:annotation>
1606 <xsd:documentation>A gallery or population group supported by the implementing system.</xsd:documentation>
1607 </xsd:annotation>
1608 </xsd:enumeration>
1609 <xsd:enumeration value="IdentityModel">
1610 <xsd:annotation>
1611 <xsd:documentation>Identifies whether the implementing system is person-centric or encounter-centric
1612 based.</xsd:documentation>
1613 </xsd:annotation>
1614 </xsd:enumeration>
1615 <xsd:enumeration value="MatchScore">
1616 <xsd:annotation>
1617 <xsd:documentation> Identifies the use of match scores returned by the implementing
1618 system.</xsd:documentation>
1619 </xsd:annotation>
1620 </xsd:enumeration>
1621 <xsd:enumeration value="QualityAlgorithm">
1622 <xsd:annotation>
1623 <xsd:documentation>A quality algorithm vendor and algorithm vendor product ID supported by the implementing
1624 system.</xsd:documentation>
1625 </xsd:annotation>
1626 </xsd:enumeration>
1627 <xsd:enumeration value="SupportedBiometric">
1628 <xsd:annotation>

```

1629 <xsd:documentation>A biometric type supported by the implementing system.</xsd:documentation>  
 1630 </xsd:annotation>  
 1631 </xsd:enumeration>  
 1632 <xsd:enumeration value="TransformOperation">  
 1633 <xsd:annotation>  
 1634 <xsd:documentation>A transform operation type supported by the implementing system.</xsd:documentation>  
 1635 </xsd:annotation>  
 1636 </xsd:enumeration>  
 1637 </xsd:restriction>  
 1638 </xsd:simpleType>  
 1639 <xsd:complexType name="CapabilityType">  
 1640 <xsd:annotation>  
 1641 <xsd:documentation>Defines a single capability supported by an implementing system.</xsd:documentation>  
 1642 </xsd:annotation>  
 1643 <xsd:sequence>  
 1644 <xsd:element name="CapabilityName" type="tns:CapabilityName" minOccurs="1" maxOccurs="1">  
 1645 <xsd:annotation>  
 1646 <xsd:documentation>The name of the capability.</xsd:documentation>  
 1647 </xsd:annotation>  
 1648 </xsd:element>  
 1649 <xsd:element name="CapabilityID" type="xsd:string" minOccurs="0" maxOccurs="1">  
 1650 <xsd:annotation>  
 1651 <xsd:documentation>An identifier assigned to the capability by the implementing system.</xsd:documentation>  
 1652 </xsd:annotation>  
 1653 </xsd:element>  
 1654 <xsd:element name="CapabilityDescription" type="xsd:string" minOccurs="0" maxOccurs="1">  
 1655 <xsd:annotation>  
 1656 <xsd:documentation>A description of the capability.</xsd:documentation>  
 1657 </xsd:annotation>  
 1658 </xsd:element>  
 1659 <xsd:element name="CapabilityValue" type="xsd:string" minOccurs="0" maxOccurs="1">  
 1660 <xsd:annotation>  
 1661 <xsd:documentation>A value assigned to the capability.</xsd:documentation>  
 1662 </xsd:annotation>  
 1663 </xsd:element>  
 1664 <xsd:element name="CapabilitySupportingValue" type="xsd:string" minOccurs="0" maxOccurs="1">  
 1665 <xsd:annotation>  
 1666 <xsd:documentation>A secondary value supporting the capability.</xsd:documentation>  
 1667 </xsd:annotation>  
 1668 </xsd:element>  
 1669 <xsd:element name="CapabilityAdditionalInfo" type="xsd:string" minOccurs="0" maxOccurs="1">

```

1670 <xsd:annotation>
1671 <xsd:documentation> Contains additional information for the supported capability.</xsd:documentation>
1672 </xsd:annotation>
1673 </xsd:element>
1674 </xsd:sequence>
1675 </xsd:complexType>
1676 <xsd:complexType name="CBEFF_BIR_ListType">
1677 <xsd:annotation>
1678 <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1679 </xsd:annotation>
1680 <xsd:sequence>
1681 <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="unbounded">
1682 <xsd:annotation>
1683 <xsd:documentation>CBEFF structure containing information about a biometric sample.</xsd:documentation>
1684 </xsd:annotation>
1685 </xsd:element>
1686 </xsd:sequence>
1687 </xsd:complexType>
1688 <xsd:complexType name="CBEFF_BIR_Type">
1689 <xsd:annotation>
1690 <xsd:documentation>Represents biometric information, with either a non-XML or XML
1691 representation.</xsd:documentation>
1692 </xsd:annotation>
1693 <xsd:sequence>
1694 <xsd:element name="FormatOwner" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
1695 <xsd:element name="FormatType" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
1696 <xsd:element name="BIR_Information" minOccurs="0" maxOccurs="1">
1697 <xsd:complexType>
1698 <xsd:sequence>
1699 <xsd:element name="BIR_Info" type="iso-iec19785-3-7oasis_cbeff:BIRInfoType" minOccurs="0" maxOccurs="1"/>
1700 <xsd:element name="BDB_Info" type="iso-iec19785-3-7oasis_cbeff:BDBInfoType" minOccurs="0"
1701 maxOccurs="1"/>
1702 <xsd:element name="SB_Info" type="iso-iec19785-3-7oasis_cbeff:SBInfoType" minOccurs="0" maxOccurs="1"/>
1703 </xsd:sequence>
1704 </xsd:complexType>
1705 </xsd:element>
1706 <xsd:element name="BIR" type="tns:BaseBIRType" />
1707 </xsd:sequence>
1708 </xsd:complexType>
1709 <xsd:complexType name="BaseBIRType">
1710 </xsd:complexType>

```

```

1711 <xsd:complexType name="BinaryBIR">
1712 <xsd:complexContent>
1713 <xsd:extension base="tns:BaseBIRType">
1714 <xsd:sequence>
1715 <xsd:element name="Binary" type="xsd:base64Binary" />
1716 </xsd:sequence>
1717 </xsd:extension>
1718 </xsd:complexContent>
1719 </xsd:complexType>
1720 <xsd:complexType name="URI_BIR">
1721 <xsd:complexContent>
1722 <xsd:extension base="tns:BaseBIRType">
1723 <xsd:sequence>
1724 <xsd:element name="URI" type="xsd:anyURI" />
1725 </xsd:sequence>
1726 </xsd:extension>
1727 </xsd:complexContent>
1728 </xsd:complexType>
1729 <xsd:complexType name="XML_BIR">
1730 <xsd:complexContent>
1731 <xsd:extension base="tns:BaseBIRType">
1732 <xsd:sequence>
1733 <xsd:element name="XML" type="iso-iec19785-3-7oasis_cbeff:BIRType" />
1734 </xsd:sequence>
1735 </xsd:extension>
1736 </xsd:complexContent>
1737 </xsd:complexType>
1738 <xsd:simpleType name="Classification">
1739 <xsd:annotation>
1740 <xsd:documentation>The result of a classification.</xsd:documentation>
1741 </xsd:annotation>
1742 <xsd:restriction base="xsd:string"/>
1743 </xsd:simpleType>
1744 <xsd:simpleType name="ClassificationAlgorithmType">
1745 <xsd:annotation>
1746 <xsd:documentation>Type of classification algorithm that was used to perform the
1747 classification.</xsd:documentation>
1748 </xsd:annotation>
1749 <xsd:restriction base="xsd:string"/>
1750 </xsd:simpleType>
1751 <xsd:complexType name="ClassificationData">

```

```

1752 <xsd:annotation>
1753 <xsd:documentation>Contains information on classification results and the algorithm used to determine the
1754 classification.</xsd:documentation>
1755 </xsd:annotation>
1756 <xsd:sequence>
1757 <xsd:element name="Classification" type="tns:Classification" minOccurs="1" maxOccurs="1">
1758 <xsd:annotation>
1759 <xsd:documentation>The result of the classification.</xsd:documentation>
1760 </xsd:annotation>
1761 </xsd:element>
1762 <xsd:element name="ClassificationAlgorithmType" type="tns:ClassificationAlgorithmType" minOccurs="1"
1763 maxOccurs="1">
1764 <xsd:annotation>
1765 <xsd:documentation>Identifies the type of classification algorithm that was used to perform the
1766 classification.</xsd:documentation>
1767 </xsd:annotation>
1768 </xsd:element>
1769 </xsd:sequence>
1770 </xsd:complexType>
1771 <xsd:complexType name="EncounterListType">
1772 <xsd:annotation>
1773 <xsd:documentation>Defines a set of encounters.</xsd:documentation>
1774 </xsd:annotation>
1775 <xsd:sequence>
1776 <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="unbounded">
1777 <xsd:annotation>
1778 <xsd:documentation>The identifier of an encounter.</xsd:documentation>
1779 </xsd:annotation>
1780 </xsd:element>
1781 </xsd:sequence>
1782 </xsd:complexType>
1783 <xsd:complexType name="FusionInformationListType">
1784 <xsd:annotation>
1785 <xsd:documentation>
1786 Contains at a minimum two sets of fusion input
1787 elements, as input to the PerformFusion request.
1788 </xsd:documentation>
1789 </xsd:annotation>
1790 <xsd:sequence>
1791 <xsd:element name="FusionElement" type="tns:FusionInformationType" minOccurs="2"
1792 maxOccurs="unbounded">
1793 <xsd:annotation>

```

```

1794 <xsd:documentation>A set of fusion information.</xsd:documentation>
1795 </xsd:annotation>
1796 </xsd:element>
1797 </xsd:sequence>
1798 </xsd:complexType>
1799 <xsd:complexType name="FusionInformationType">
1800 <xsd:annotation>
1801 <xsd:documentation>Represents the information necessary to perform a fusion operation.</xsd:documentation>
1802 </xsd:annotation>
1803 <xsd:sequence>
1804 | <xsd:element name="BiometricType" type="iso-iec19785-3-7oasis_cbeff:MultipleTypesType" minOccurs="1"
1805 maxOccurs="1">
1806 <xsd:annotation>
1807 <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1808 CBEFF.</xsd:documentation>
1809 </xsd:annotation>
1810 </xsd:element>
1811 | <xsd:element name="BiometricSubType" type="iso-iec19785-3-7oasis_cbeff:SubtypeType" minOccurs="0"
1812 maxOccurs="1">
1813 <xsd:annotation>
1814 <xsd:documentation>More specifically defines the type of biometric data stored in the biometric
1815 record.</xsd:documentation>
1816 </xsd:annotation>
1817 </xsd:element>
1818 <xsd:element name="AlgorithmOwner" type="xsd:string" minOccurs="1" maxOccurs="1">
1819 <xsd:annotation>
1820 <xsd:documentation>The owner or vendor of the algorithm used to determine the score or
1821 decision.</xsd:documentation>
1822 </xsd:annotation>
1823 </xsd:element>
1824 <xsd:element name="AlgorithmType" type="xsd:string" minOccurs="1" maxOccurs="1">
1825 <xsd:annotation>
1826 <xsd:documentation>The Algorithm Owner's identifier for the specific algorithm product and version used to
1827 determine the score or decision.</xsd:documentation>
1828 </xsd:annotation>
1829 </xsd:element>
1830 <xsd:element name="FusionResult" type="tns:FusionResult" />
1831 </xsd:sequence>
1832 </xsd:complexType>
1833 <xsd:complexType name="FusionResult">
1834 <xsd:sequence />
1835 </xsd:complexType>

```

```

1836 <xsd:complexType name="FusionScore">
1837 <xsd:complexContent>
1838 <xsd:extension base="tns:FusionResult">
1839 <xsd:sequence>
1840 <xsd:element name="Score" type="tns:Score">
1841 <xsd:annotation>
1842 <xsd:documentation>The similarity score assigned by the matching algorithm.</xsd:documentation>
1843 </xsd:annotation>
1844 </xsd:element>
1845 </xsd:sequence>
1846 </xsd:extension>
1847 </xsd:complexContent>
1848 </xsd:complexType>
1849 <xsd:complexType name="FusionDecision">
1850 <xsd:complexContent>
1851 <xsd:extension base="tns:FusionResult">
1852 <xsd:sequence>
1853 <xsd:element name="Decision" type="xsd:string">
1854 <xsd:annotation>
1855 <xsd:documentation>The match decision assigned by the matching algorithm.</xsd:documentation>
1856 </xsd:annotation>
1857 </xsd:element>
1858 </xsd:sequence>
1859 </xsd:extension>
1860 </xsd:complexContent>
1861 </xsd:complexType>
1862 <xsd:complexType name="GenericRequestParameters">
1863 <xsd:annotation>
1864 <xsd:documentation>Common request paramters that can be used to identify the
1865 requester.</xsd:documentation>
1866 </xsd:annotation>
1867 <xsd:sequence>
1868 <xsd:element name="Application" type="tns:ApplicationIdentifier" minOccurs="0" maxOccurs="1">
1869 <xsd:annotation>
1870 <xsd:documentation>Identifies the requesting application.</xsd:documentation>
1871 </xsd:annotation>
1872 </xsd:element>
1873 <xsd:element name="ApplicationUser" type="tns:ApplicationUserIdentifier" minOccurs="0" maxOccurs="1">
1874 <xsd:annotation>
1875 <xsd:documentation>Identifers the user or instance of the requesting application.</xsd:documentation>
1876 </xsd:annotation>

```

```

1877 </xsd:element>
1878 <xsd:element name="BIASOperationName" type="xsd:string" minOccurs="0" maxOccurs="1">
1879 <xsd:annotation>
1880 <xsd:documentation>Identifiers the BIAS operation name that is being requested.</xsd:documentation>
1881 </xsd:annotation>
1882 </xsd:element>
1883 </xsd:sequence>
1884 </xsd:complexType>
1885 <xsd:complexType name="InformationType">
1886 <xsd:annotation>
1887 <xsd:documentation>
1888 Allows for an unlimited number of data element types, and it does
1889 not specify nor require any particular data element.
1890 </xsd:documentation>
1891 </xsd:annotation>
1892 <xsd:sequence>
1893 <xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
1894 </xsd:sequence>
1895 </xsd:complexType>
1896 <xsd:complexType name="ListFilterType">
1897 <xsd:annotation>
1898 <xsd:documentation>
1899 Provides a method to filter the amount of information returned in
1900 a search of biometric data.
1901 </xsd:documentation>
1902 </xsd:annotation>
1903 <xsd:sequence>
1904 <xsd:element name="BiometricTypeFilters">
1905 <xsd:complexType>
1906 <xsd:sequence>
1907 | <xsd:element name="BiometricTypeFilter" type="iso-iec19785-3-7oasis_cbeff:MultipleTypesType" minOccurs="1"
1908 maxOccurs="unbounded">
1909 <xsd:annotation>
1910 <xsd:documentation>Limits the returned information to a specific type of biometric, as defined by
1911 CBEFF.</xsd:documentation>
1912 </xsd:annotation>
1913 </xsd:element>
1914 </xsd:sequence>
1915 </xsd:complexType>
1916 </xsd:element>
1917 <xsd:element name="IncludeBiometricSubtype" type="xsd:boolean" minOccurs="1" maxOccurs="1">

```



```

1918 <xsd:annotation>
1919 <xsd:documentation>A Boolean flag indicating if biometric subtype information should be
1920 returned.</xsd:documentation>
1921 </xsd:annotation>
1922 </xsd:element>
1923 </xsd:sequence>
1924 </xsd:complexType>
1925 <xsd:simpleType name="MatchType">
1926 <xsd:annotation>
1927 <xsd:documentation>The result of a fusion method.</xsd:documentation>
1928 </xsd:annotation>
1929 <xsd:restriction base="xsd:boolean"/>
1930 </xsd:simpleType>
1931 <xsd:complexType name="ProcessingOptionsType">
1932 <xsd:annotation>
1933 <xsd:documentation>
1934 BIAS aggregate services support the ability to include various
1935 processing options which direct and possibly control the business
1936 logic for that service. The ProcessingOptionsType provides a
1937 method to represent those options. Processing options should be
1938 defined by the implementing system.
1939 </xsd:documentation>
1940 </xsd:annotation>
1941 <xsd:sequence>
1942 <xsd:element name="Option" type="xsd:string" minOccurs="0" maxOccurs="unbounded">
1943 <xsd:annotation>
1944 <xsd:documentation>An option supported by the implementing system.</xsd:documentation>
1945 </xsd:annotation>
1946 </xsd:element>
1947 </xsd:sequence>
1948 </xsd:complexType>
1949 <xsd:simpleType name="ProductID">
1950 <xsd:annotation>
1951 <xsd:documentation>The vendor's ID for a particular product.</xsd:documentation>
1952 </xsd:annotation>
1953 <xsd:restriction base="xsd:string"/>
1954 </xsd:simpleType>
1955 <xsd:complexType name="QualityData">
1956 <xsd:annotation>
1957 <xsd:documentation>Contains information about a biometric sample's quality and the algorithm used to compute
1958 the quality.</xsd:documentation>

```

```

1959 </xsd:annotation>
1960 <xsd:sequence>
1961 | <xsd:element name="QualityScore" type="iso-iec19785-3-7oasis_cbeff:QualityType" minOccurs="0"
1962   maxOccurs="1">
1963 <xsd:annotation>
1964 <xsd:documentation>The quality of a biometric sample.</xsd:documentation>
1965 </xsd:annotation>
1966 </xsd:element>
1967 <xsd:element name="AlgorithmVendor" type="tns:VendorIdentifier" minOccurs="1" maxOccurs="1">
1968 <xsd:annotation>
1969 <xsd:documentation>The vendor of the qualilty algorithm used to determine the quality
1970 score.</xsd:documentation>
1971 </xsd:annotation>
1972 </xsd:element>
1973 <xsd:element name="AlgorithmVendorProductID" type="tns:ProductID" minOccurs="1" maxOccurs="1">
1974 <xsd:annotation>
1975 <xsd:documentation>The vendor's ID for the algorithm used to determine the quality.</xsd:documentation>
1976 </xsd:annotation>
1977 </xsd:element>
1978 <xsd:element name="AlgorithmVersion" type="tns:VersionType" minOccurs="0" maxOccurs="1">
1979 <xsd:annotation>
1980 <xsd:documentation>The version of the algorithm used to determine the quality.</xsd:documentation>
1981 </xsd:annotation>
1982 </xsd:element>
1983 </xsd:sequence>
1984 </xsd:complexType>
1985 <xsd:complexType name="RequestTemplate">
1986 <xsd:annotation>
1987 <xsd:documentation>Base template for BIAS primitive service requests.</xsd:documentation>
1988 </xsd:annotation>
1989 <xsd:sequence>
1990 <xsd:element name="GenericRequestParameters" type="tns:GenericRequestParameters" minOccurs="0"
1991   maxOccurs="1"/>
1992 </xsd:sequence>
1993 </xsd:complexType>
1994 <xsd:complexType name="ResponseStatus">
1995 <xsd:sequence>
1996 <xsd:element name="Return" type="tns:ReturnCode" minOccurs="1" maxOccurs="1">
1997 <xsd:annotation>
1998 <xsd:documentation>The return code indicates the return status of the operation.</xsd:documentation>
1999 </xsd:annotation>

```

2000 </xsd:element>

2001 <xsd:element name="Message" type="xsd:string" minOccurs="0" maxOccurs="1">

2002 <xsd:annotation>

2003 <xsd:documentation>A short message corresponding to the return code.</xsd:documentation>

2004 </xsd:annotation>

2005 </xsd:element>

2006 </xsd:sequence>

2007 </xsd:complexType>

2008 <xsd:complexType name="ResponseTemplate">

2009 <xsd:annotation>

2010 <xsd:documentation>Base template for BIAS responses.</xsd:documentation>

2011 </xsd:annotation>

2012 <xsd:sequence>

2013 <xsd:element name="ResponseStatus" type="tns:ResponseStatus" minOccurs="1" maxOccurs="1">

2014 <xsd:annotation>

2015 <xsd:documentation>Returned status for the operation.</xsd:documentation>

2016 </xsd:annotation>

2017 </xsd:element>

2018 </xsd:sequence>

2019 </xsd:complexType>

2020 <xsd:simpleType name="ReturnCode">

2021 <xsd:annotation>

2022 <xsd:documentation>BIAS Operation Return Codes</xsd:documentation>

2023 </xsd:annotation>

2024 <xsd:restriction base="xsd:unsignedLong">

2025 <xsd:enumeration value="0">

2026 <xsd:annotation>

2027 <xsd:documentation>Success</xsd:documentation>

2028 </xsd:annotation>

2029 </xsd:enumeration>

2030 </xsd:restriction>

2031 </xsd:simpleType>

2032 <xsd:simpleType name="Score">

2033 <xsd:annotation>

2034 <xsd:documentation>Match result or quality score.</xsd:documentation>

2035 </xsd:annotation>

2036 <xsd:restriction base="xsd:float"/>

2037 </xsd:simpleType>

2038 <xsd:complexType name="TokenType">

2039 <xsd:annotation>

2040 <xsd:documentation>Defines a token that is returned for asynchronous processing.</xsd:documentation>

```

2041 </xsd:annotation>
2042 <xsd:sequence>
2043 <xsd:element name="TokenValue" type="xsd:string" minOccurs="1" maxOccurs="1">
2044 <xsd:annotation>
2045 <xsd:documentation>A value returned by the implementing system that is used to retrieve the results to a service
2046 at a later time.</xsd:documentation>
2047 </xsd:annotation>
2048 </xsd:element>
2049 <xsd:element name="Expiration" type="xsd:date" minOccurs="1" maxOccurs="1">
2050 <xsd:annotation>
2051 <xsd:documentation>A date and time at which point the token expires and the service results are no longer
2052 guaranteed to be available.</xsd:documentation>
2053 </xsd:annotation>
2054 </xsd:element>
2055 </xsd:sequence>
2056 </xsd:complexType>
2057 <xsd:simpleType name="VendorIdentifier">
2058 <xsd:annotation>
2059 <xsd:documentation>Identifies a vendor.</xsd:documentation>
2060 </xsd:annotation>
2061 <xsd:restriction base="xsd:string"/>
2062 </xsd:simpleType>
2063 <xsd:complexType name="Version">
2064 <xsd:annotation>
2065 <xsd:documentation>
2066 For a description or definition of each data element, see the
2067 referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
2068 </xsd:documentation>
2069 </xsd:annotation>
2070 <xsd:sequence>
2071 <xsd:element name="major" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1"/>
2072 <xsd:element name="minor" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1"/>
2073 </xsd:sequence>
2074 </xsd:complexType>
2075 <xsd:simpleType name="VersionType">
2076 <xsd:annotation>
2077 <xsd:documentation>The version of a component.</xsd:documentation>
2078 </xsd:annotation>
2079 <xsd:restriction base="xsd:string"/>
2080 </xsd:simpleType>
2081 <xsd:element name="AddSubjectToGallery">

```

```

2082 <xsd:complexType>
2083 <xsd:annotation>
2084 <xsd:documentation>
2085 Register a subject to a given gallery or population group.
2086 </xsd:documentation>
2087 </xsd:annotation>
2088 <xsd:sequence>
2089 <xsd:element name="AddSubjectToGalleryRequest" nillable="true">
2090 <xsd:complexType>
2091 <xsd:complexContent>
2092 <xsd:extension base="tns:RequestTemplate">
2093 <xsd:sequence>
2094 <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2095 <xsd:annotation>
2096 <xsd:documentation>The identifier of the gallery or population group to which the subject will be
2097 added.</xsd:documentation>
2098 </xsd:annotation>
2099 </xsd:element>
2100 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2101 <xsd:annotation>
2102 <xsd:documentation>The identity to add to the gallery.</xsd:documentation>
2103 </xsd:annotation>
2104 </xsd:element>
2105 </xsd:sequence>
2106 </xsd:extension>
2107 </xsd:complexContent>
2108 </xsd:complexType>
2109 </xsd:element>
2110 </xsd:sequence>
2111 </xsd:complexType>
2112 </xsd:element>
2113 <xsd:element name="AddSubjectToGalleryResponse">
2114 <xsd:complexType>
2115 <xsd:annotation>
2116 <xsd:documentation>
2117 The response to an AddSubjectToGallery request.
2118 </xsd:documentation>
2119 </xsd:annotation>
2120 <xsd:sequence>
2121 <xsd:element name="AddSubjectToGalleryResponsePackage" nillable="true">
2122 <xsd:complexType>

```

```

2123 <xsd:complexContent>
2124 <xsd:extension base="tns:ResponseTemplate"/>
2125 </xsd:complexContent>
2126 </xsd:complexType>
2127 </xsd:element>
2128 </xsd:sequence>
2129 </xsd:complexType>
2130 </xsd:element>
2131 <xsd:element name="CheckQuality">
2132 <xsd:complexType>
2133 <xsd:annotation>
2134 <xsd:documentation>
2135 Calculate a quality score for a given biometric.
2136 </xsd:documentation>
2137 </xsd:annotation>
2138 <xsd:sequence>
2139 <xsd:element name="CheckQualityRequest" nillable="true">
2140 <xsd:complexType>
2141 <xsd:complexContent>
2142 <xsd:extension base="tns:RequestTemplate">
2143 <xsd:sequence>
2144 <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
2145 <xsd:annotation>
2146 <xsd:documentation>Data structure containing a single biometric sample for which a quality score is to be
2147 determined.</xsd:documentation>
2148 </xsd:annotation>
2149 </xsd:element>
2150 <xsd:element name="Quality" type="tns:QualityData" minOccurs="0" maxOccurs="1">
2151 <xsd:annotation>
2152 <xsd:documentation>Specifies a particular algorithm vendor and vender product ID.</xsd:documentation>
2153 </xsd:annotation>
2154 </xsd:element>
2155 </xsd:sequence>
2156 </xsd:extension>
2157 </xsd:complexContent>
2158 </xsd:complexType>
2159 </xsd:element>
2160 </xsd:sequence>
2161 </xsd:complexType>
2162 </xsd:element>
2163 <xsd:element name="CheckQualityResponse">

```

```

2164 <xsd:complexType>
2165 <xsd:annotation>
2166 <xsd:documentation>
2167 The response to a CheckQuality request.
2168 </xsd:documentation>
2169 </xsd:annotation>
2170 <xsd:sequence>
2171 <xsd:element name="CheckQualityResponsePackage" nillable="true">
2172 <xsd:complexType>
2173 <xsd:complexContent>
2174 <xsd:extension base="tns:ResponseTemplate">
2175 <xsd:sequence>
2176 <xsd:element name="QualityInfo" type="tns:QualityData" minOccurs="1" maxOccurs="1">
2177 <xsd:annotation>
2178 <xsd:documentation>Contains the quality information for the submitted biometric sample.</xsd:documentation>
2179 </xsd:annotation>
2180 </xsd:element>
2181 </xsd:sequence>
2182 </xsd:extension>
2183 </xsd:complexContent>
2184 </xsd:complexType>
2185 </xsd:element>
2186 </xsd:sequence>
2187 </xsd:complexType>
2188 </xsd:element>
2189 <xsd:element name="ClassifyBiometricData">
2190 <xsd:complexType>
2191 <xsd:annotation>
2192 <xsd:documentation>
2193 Classifies a biometric sample.
2194 </xsd:documentation>
2195 </xsd:annotation>
2196 <xsd:sequence>
2197 <xsd:element name="ClassifyBiometricDataRequest" nillable="true">
2198 <xsd:complexType>
2199 <xsd:complexContent>
2200 <xsd:extension base="tns:RequestTemplate">
2201 <xsd:sequence>
2202 <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
2203 <xsd:annotation>

```

2204 <xsd:documentation>Data structure containing a single biometric sample for which the classification is to be  
 2205 determined.</xsd:documentation>  
 2206 </xsd:annotation>  
 2207 </xsd:element>  
 2208 </xsd:sequence>  
 2209 </xsd:extension>  
 2210 </xsd:complexContent>  
 2211 </xsd:complexType>  
 2212 </xsd:element>  
 2213 </xsd:sequence>  
 2214 </xsd:complexType>  
 2215 </xsd:element>  
 2216 <xsd:element name="ClassifyBiometricDataResponse">  
 2217 <xsd:complexType>  
 2218 <xsd:annotation>  
 2219 <xsd:documentation>  
 2220 The response to a ClassifyBiometricData request, containing  
 2221 the classification of a biometric sample.  
 2222 </xsd:documentation>  
 2223 </xsd:annotation>  
 2224 <xsd:sequence>  
 2225 <xsd:element name="ClassifyBiometricDataResponsePackage" nillable="true">  
 2226 <xsd:complexType>  
 2227 <xsd:complexContent>  
 2228 <xsd:extension base="tns:ResponseTemplate">  
 2229 <xsd:sequence>  
 2230 <xsd:element name="ClassificationData" type="tns:ClassificationData" minOccurs="1" maxOccurs="1">  
 2231 <xsd:annotation>  
 2232 <xsd:documentation>Information on the results and type of classification performed.</xsd:documentation>  
 2233 </xsd:annotation>  
 2234 </xsd:element>  
 2235 </xsd:sequence>  
 2236 </xsd:extension>  
 2237 </xsd:complexContent>  
 2238 </xsd:complexType>  
 2239 </xsd:element>  
 2240 </xsd:sequence>  
 2241 </xsd:complexType>  
 2242 </xsd:element>  
 2243 <xsd:element name="CreateSubject">  
 2244 <xsd:complexType>



```

2245 <xsd:annotation>
2246 <xsd:documentation>
2247 Create a new subject record.
2248 </xsd:documentation>
2249 </xsd:annotation>
2250 <xsd:sequence>
2251 <xsd:element name="CreateSubjectRequest" nillable="true">
2252 <xsd:complexType>
2253 <xsd:complexContent>
2254 <xsd:extension base="tns:RequestTemplate"/>
2255 </xsd:complexContent>
2256 </xsd:complexType>
2257 </xsd:element>
2258 </xsd:sequence>
2259 </xsd:complexType>
2260 </xsd:element>
2261 <xsd:element name="CreateSubjectResponse">
2262 <xsd:complexType>
2263 <xsd:annotation>
2264 <xsd:documentation>
2265 The response to a CreateSubject request, containing the subject
2266 ID of the new subject record.
2267 </xsd:documentation>
2268 </xsd:annotation>
2269 <xsd:sequence>
2270 <xsd:element name="CreateSubjectResponsePackage" nillable="true">
2271 <xsd:complexType>
2272 <xsd:complexContent>
2273 <xsd:extension base="tns:ResponseTemplate">
2274 <xsd:sequence>
2275 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2276 <xsd:annotation>
2277 <xsd:documentation>Contains the subject ID of the new subject record.</xsd:documentation>
2278 </xsd:annotation>
2279 </xsd:element>
2280 </xsd:sequence>
2281 </xsd:extension>
2282 </xsd:complexContent>
2283 </xsd:complexType>
2284 </xsd:element>
2285 </xsd:sequence>

```

```

2286 </xsd:complexType>
2287 </xsd:element>
2288 <xsd:element name="DeleteBiographicData">
2289 <xsd:complexType>
2290 <xsd:annotation>
2291 <xsd:documentation>
2292 Erase all of the biographic data associated with a given
2293 subject record or, in the encounter-centric model, with a
2294 given encounter.
2295 </xsd:documentation>
2296 </xsd:annotation>
2297 <xsd:sequence>
2298 <xsd:element name="DeleteBiographicDataRequest" nillable="true">
2299 <xsd:complexType>
2300 <xsd:complexContent>
2301 <xsd:extension base="tns:RequestTemplate">
2302 <xsd:sequence>
2303 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2304 <xsd:annotation>
2305 <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
2306 </xsd:annotation>
2307 </xsd:element>
2308 </xsd:sequence>
2309 </xsd:extension>
2310 </xsd:complexContent>
2311 </xsd:complexType>
2312 </xsd:element>
2313 </xsd:sequence>
2314 </xsd:complexType>
2315 </xsd:element>
2316 <xsd:element name="DeleteBiographicDataResponse">
2317 <xsd:complexType>
2318 <xsd:annotation>
2319 <xsd:documentation>
2320 The response to a DeleteBiographicData request.
2321 </xsd:documentation>
2322 </xsd:annotation>
2323 <xsd:sequence>
2324 <xsd:element name="DeleteBiographicDataResponsePackage" nillable="true">
2325 <xsd:complexType>
2326 <xsd:complexContent>

```

```

2327 <xsd:extension base="tns:ResponseTemplate"/>
2328 </xsd:complexContent>
2329 </xsd:complexType>
2330 </xsd:element>
2331 </xsd:sequence>
2332 </xsd:complexType>
2333 </xsd:element>
2334 <xsd:element name="DeleteBiometricData">
2335 <xsd:complexType>
2336 <xsd:annotation>
2337 <xsd:documentation>
2338 Erase all of the biometric data associated with a given
2339 subject record or, in the encounter-centric model, with a
2340 given encounter.
2341 </xsd:documentation>
2342 </xsd:annotation>
2343 <xsd:sequence>
2344 <xsd:element name="DeleteBiometricDataRequest" nillable="true">
2345 <xsd:complexType>
2346 <xsd:complexContent>
2347 <xsd:extension base="tns:RequestTemplate">
2348 <xsd:sequence>
2349 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2350 <xsd:annotation>
2351 <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
2352 </xsd:annotation>
2353 </xsd:element>
2354 </xsd:sequence>
2355 </xsd:extension>
2356 </xsd:complexContent>
2357 </xsd:complexType>
2358 </xsd:element>
2359 </xsd:sequence>
2360 </xsd:complexType>
2361 </xsd:element>
2362 <xsd:element name="DeleteBiometricDataResponse">
2363 <xsd:complexType>
2364 <xsd:annotation>
2365 <xsd:documentation>
2366 The response to a DeleteBiometricData request.
2367 </xsd:documentation>

```

```

2368 </xsd:annotation>
2369 <xsd:sequence>
2370 <xsd:element name="DeleteBiometricDataResponsePackage" nillable="true">
2371 <xsd:complexType>
2372 <xsd:complexContent>
2373 <xsd:extension base="tns:ResponseTemplate"/>
2374 </xsd:complexContent>
2375 </xsd:complexType>
2376 </xsd:element>
2377 </xsd:sequence>
2378 </xsd:complexType>
2379 </xsd:element>
2380 <xsd:element name="DeleteSubject">
2381 <xsd:complexType>
2382 <xsd:annotation>
2383 <xsd:documentation>
2384 Delete an existing subject record and, in an encounter-centric
2385 model, any associated encounter information.
2386 </xsd:documentation>
2387 </xsd:annotation>
2388 <xsd:sequence>
2389 <xsd:element name="DeleteSubjectRequest" nillable="true">
2390 <xsd:complexType>
2391 <xsd:complexContent>
2392 <xsd:extension base="tns:RequestTemplate">
2393 <xsd:sequence>
2394 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2395 <xsd:annotation>
2396 <xsd:documentation>Subject ID of the identity to delete.</xsd:documentation>
2397 </xsd:annotation>
2398 </xsd:element>
2399 </xsd:sequence>
2400 </xsd:extension>
2401 </xsd:complexContent>
2402 </xsd:complexType>
2403 </xsd:element>
2404 </xsd:sequence>
2405 </xsd:complexType>
2406 </xsd:element>
2407 <xsd:element name="DeleteSubjectResponse">
2408 <xsd:complexType>

```

```

2409 <xsd:annotation>
2410 <xsd:documentation>
2411 The response to a DeleteSubject request.
2412 </xsd:documentation>
2413 </xsd:annotation>
2414 <xsd:sequence>
2415 <xsd:element name="DeleteSubjectResponsePackage" nillable="true">
2416 <xsd:complexType>
2417 <xsd:complexContent>
2418 <xsd:extension base="tns:ResponseTemplate"/>
2419 </xsd:complexContent>
2420 </xsd:complexType>
2421 </xsd:element>
2422 </xsd:sequence>
2423 </xsd:complexType>
2424 </xsd:element>
2425 <xsd:element name="DeleteSubjectFromGallery">
2426 <xsd:complexType>
2427 <xsd:annotation>
2428 <xsd:documentation>
2429 Remove the registration of a subject from a gallery or
2430 population group.
2431 </xsd:documentation>
2432 </xsd:annotation>
2433 <xsd:sequence>
2434 <xsd:element name="DeleteSubjectFromGalleryRequest" nillable="true">
2435 <xsd:complexType>
2436 <xsd:complexContent>
2437 <xsd:extension base="tns:RequestTemplate">
2438 <xsd:sequence>
2439 <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2440 <xsd:annotation>
2441 <xsd:documentation>The identifier of the gallery or population group from which the subject will be
2442 deleted.</xsd:documentation>
2443 </xsd:annotation>
2444 </xsd:element>
2445 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2446 <xsd:annotation>
2447 <xsd:documentation>The identity to remove from the gallery.</xsd:documentation>
2448 </xsd:annotation>
2449 </xsd:element>

```

```

2450 </xsd:sequence>
2451 </xsd:extension>
2452 </xsd:complexContent>
2453 </xsd:complexType>
2454 </xsd:element>
2455 </xsd:sequence>
2456 </xsd:complexType>
2457 </xsd:element>
2458 <xsd:element name="DeleteSubjectFromGalleryResponse">
2459 <xsd:complexType>
2460 <xsd:annotation>
2461 <xsd:documentation>
2462 The response to a DeleteSubjectFromGallery request.
2463 </xsd:documentation>
2464 </xsd:annotation>
2465 <xsd:sequence>
2466 <xsd:element name="DeleteSubjectFromGalleryResponsePackage" nillable="true">
2467 <xsd:complexType>
2468 <xsd:complexContent>
2469 <xsd:extension base="tns:ResponseTemplate"/>
2470 </xsd:complexContent>
2471 </xsd:complexType>
2472 </xsd:element>
2473 </xsd:sequence>
2474 </xsd:complexType>
2475 </xsd:element>
2476 <xsd:element name="GetIdentifySubjectResults">
2477 <xsd:complexType>
2478 <xsd:annotation>
2479 <xsd:documentation>
2480 Retrieve the identification results for a specified token,
2481 which was returned by the Identify Subject service.
2482 </xsd:documentation>
2483 </xsd:annotation>
2484 <xsd:sequence>
2485 <xsd:element name="GetIdentifySubjectResultsRequest" nillable="true">
2486 <xsd:complexType>
2487 <xsd:complexContent>
2488 <xsd:extension base="tns:RequestTemplate">
2489 <xsd:sequence>
2490 <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">

```

```

2491 <xsd:annotation>
2492 <xsd:documentation>A value used to retrieve the results of an IdentifySubject request.</xsd:documentation>
2493 </xsd:annotation>
2494 </xsd:element>
2495 </xsd:sequence>
2496 </xsd:extension>
2497 </xsd:complexContent>
2498 </xsd:complexType>
2499 </xsd:element>
2500 </xsd:sequence>
2501 </xsd:complexType>
2502 </xsd:element>
2503 <xsd:element name="GetIdentifySubjectResultsResponse">
2504 <xsd:complexType>
2505 <xsd:annotation>
2506 <xsd:documentation>
2507 The response to a GetIdentifySubjectResults request, which includes a candidate list.
2508 </xsd:documentation>
2509 </xsd:annotation>
2510 <xsd:sequence>
2511 <xsd:element name="GetIdentifySubjectResultsResponsePackage" nillable="true">
2512 <xsd:complexType>
2513 <xsd:complexContent>
2514 <xsd:extension base="tns:ResponseTemplate">
2515 <xsd:sequence>
2516 <xsd:element name="CandidateList" type="tns:CandidateListType" minOccurs="1" maxOccurs="1">
2517 <xsd:annotation>
2518 <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input biometric
2519 sample.</xsd:documentation>
2520 </xsd:annotation>
2521 </xsd:element>
2522 </xsd:sequence>
2523 </xsd:extension>
2524 </xsd:complexContent>
2525 </xsd:complexType>
2526 </xsd:element>
2527 </xsd:sequence>
2528 </xsd:complexType>
2529 </xsd:element>
2530 <xsd:element name="IdentifySubject">
2531 <xsd:complexType>

```

```

2532 <xsd:annotation>
2533 <xsd:documentation>
2534 Perform an identification search against a given gallery for
2535 a given biometric.
2536 </xsd:documentation>
2537 </xsd:annotation>
2538 <xsd:sequence>
2539 <xsd:element name="IdentifySubjectRequest" nillable="true">
2540 <xsd:complexType>
2541 <xsd:complexContent>
2542 <xsd:extension base="tns:RequestTemplate">
2543 <xsd:sequence>
2544 <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2545 <xsd:annotation>
2546 <xsd:documentation>The identifier of the gallery or population group which will be
2547 searched.</xsd:documentation>
2548 </xsd:annotation>
2549 </xsd:element>
2550 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2551 <xsd:annotation>
2552 <xsd:documentation>Contains the BIR, a data structure containing the biometric sample for the
2553 search.</xsd:documentation>
2554 </xsd:annotation>
2555 </xsd:element>
2556 <xsd:element name="MaxListSize" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">
2557 <xsd:annotation>
2558 <xsd:documentation>The maximum size of the candidate list that should be returned.</xsd:documentation>
2559 </xsd:annotation>
2560 </xsd:element>
2561 </xsd:sequence>
2562 </xsd:extension>
2563 </xsd:complexContent>
2564 </xsd:complexType>
2565 </xsd:element>
2566 </xsd:sequence>
2567 </xsd:complexType>
2568 </xsd:element>
2569 <xsd:element name="IdentifySubjectResponse">
2570 <xsd:complexType>
2571 <xsd:annotation>
2572 <xsd:documentation>

```



2573 The response to an IdentifySubject request, returning a  
 2574 rank-ordered candidate list.  
 2575 </xsd:documentation>  
 2576 </xsd:annotation>  
 2577 <xsd:sequence>  
 2578 <xsd:element name="IdentifySubjectResponsePackage" nillable="true">  
 2579 <xsd:complexType>  
 2580 <xsd:complexContent>  
 2581 <xsd:extension base="tns:ResponseTemplate">  
 2582 <xsd:sequence>  
 2583 <xsd:element name="IdentifySubjectResult" type="tns:IdentifySubjectResultType" />  
 2584 </xsd:sequence>  
 2585 </xsd:extension>  
 2586 </xsd:complexContent>  
 2587 </xsd:complexType>  
 2588 </xsd:element>  
 2589 </xsd:sequence>  
 2590 </xsd:complexType>  
 2591 </xsd:element>  
 2592 <xsd:complexType name="IdentifySubjectResultType">  
 2593 <xsd:sequence />  
 2594 </xsd:complexType>  
 2595 <xsd:complexType name="CandidateListResultType">  
 2596 <xsd:complexContent>  
 2597 <xsd:extension base="tns:IdentifySubjectResultType">  
 2598 <xsd:sequence>  
 2599 <xsd:element name="CandidateList" type="tns:CandidateListType">  
 2600 <xsd:annotation>  
 2601 <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input biometric  
 2602 sample; returned with successful synchronous request processing.</xsd:documentation>  
 2603 </xsd:annotation>  
 2604 </xsd:element>  
 2605 </xsd:sequence>  
 2606 </xsd:extension>  
 2607 </xsd:complexContent>  
 2608 </xsd:complexType>  
 2609 <xsd:complexType name="TokenResultType">  
 2610 <xsd:complexContent>  
 2611 <xsd:extension base="tns:IdentifySubjectResultType">  
 2612 <xsd:sequence>  
 2613 <xsd:element name="Token" type="tns:TokenType">

```

2614 <xsd:annotation>
2615 <xsd:documentation>A token used to retrieve the results of the IdentifySubject request; returned with
2616 asynchronous request processing.</xsd:documentation>
2617 </xsd:annotation>
2618 </xsd:element>
2619 </xsd:sequence>
2620 </xsd:extension>
2621 </xsd:complexContent>
2622 </xsd:complexType>
2623 <xsd:element name="ListBiographicData">
2624 <xsd:complexType>
2625 <xsd:annotation>
2626 <xsd:documentation>
2627 Lists the biographic data elements stored for a subject.
2628 </xsd:documentation>
2629 </xsd:annotation>
2630 <xsd:sequence>
2631 <xsd:element name="ListBiographicDataRequest" nillable="true">
2632 <xsd:complexType>
2633 <xsd:complexContent>
2634 <xsd:extension base="tns:RequestTemplate">
2635 <xsd:sequence>
2636 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2637 <xsd:annotation>
2638 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2639 encounter.</xsd:documentation>
2640 </xsd:annotation>
2641 </xsd:element>
2642 </xsd:sequence>
2643 </xsd:extension>
2644 </xsd:complexContent>
2645 </xsd:complexType>
2646 </xsd:element>
2647 </xsd:sequence>
2648 </xsd:complexType>
2649 </xsd:element>
2650 <xsd:element name="ListBiographicDataResponse">
2651 <xsd:complexType>
2652 <xsd:annotation>
2653 <xsd:documentation>
2654 The response to a ListBiographicData request, containing a list

```

2655 of biographic data elements stored for a subject. In the  
 2656 encounter-centric model, the biographic data elements for a  
 2657 specific encounter are returned. If an encounter ID is not  
 2658 specified and encounter data exists for the subject, the list  
 2659 of encounter IDs which contain biographic data is returned.  
 2660 </xsd:documentation>  
 2661 </xsd:annotation>  
 2662 <xsd:sequence>  
 2663 <xsd:element name="ListBiographicDataResponsePackage" nillable="true">  
 2664 <xsd:complexType>  
 2665 <xsd:complexContent>  
 2666 <xsd:extension base="tns:ResponseTemplate">  
 2667 <xsd:sequence>  
 2668 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">  
 2669 <xsd:annotation>  
 2670 <xsd:documentation>  
 2671 Contains a list of biographic data elements associated with a  
 2672 subject or encounter; non-empty if the service was  
 2673 successful, biographic data exists, and either (a) the  
 2674 person-centric model is being used or (b) the  
 2675 encounter-centric model is being used and an encounter  
 2676 identifier was specified.  
 2677 </xsd:documentation>  
 2678 </xsd:annotation>  
 2679 </xsd:element>  
 2680 <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">  
 2681 <xsd:annotation>  
 2682 <xsd:documentation>  
 2683 A list of encounter ID's associated with a subject and  
 2684 which contain biographic data; non-empty if the service  
 2685 was successful, biographic data exists, the  
 2686 encounter-centric model is being used, and an encounter  
 2687 identifier was not specified.  
 2688 </xsd:documentation>  
 2689 </xsd:annotation>  
 2690 </xsd:element>  
 2691 </xsd:sequence>  
 2692 </xsd:extension>  
 2693 </xsd:complexContent>  
 2694 </xsd:complexType>  
 2695 </xsd:element>

```

2696 </xsd:sequence>
2697 </xsd:complexType>
2698 </xsd:element>
2699 <xsd:element name="ListBiometricData">
2700 <xsd:complexType>
2701 <xsd:annotation>
2702 <xsd:documentation>
2703 Lists the biometric data elements stored for a subject. Note
2704 that no actual biometric data is returned by this service (see
2705 the RetrieveBiometricInformation service to obtain the biometric
2706 data).
2707 </xsd:documentation>
2708 </xsd:annotation>
2709 <xsd:sequence>
2710 <xsd:element name="ListBiometricDataRequest" nillable="true">
2711 <xsd:complexType>
2712 <xsd:complexContent>
2713 <xsd:extension base="tns:RequestTemplate">
2714 <xsd:sequence>
2715 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2716 <xsd:annotation>
2717 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2718 encounter.</xsd:documentation>
2719 </xsd:annotation>
2720 </xsd:element>
2721 <xsd:element name="ListFilter" type="tns:ListFilterType" minOccurs="0" maxOccurs="1">
2722 <xsd:annotation>
2723 <xsd:documentation>Indicates what biometric information should be returned.</xsd:documentation>
2724 </xsd:annotation>
2725 </xsd:element>
2726 </xsd:sequence>
2727 </xsd:extension>
2728 </xsd:complexContent>
2729 </xsd:complexType>
2730 </xsd:element>
2731 </xsd:sequence>
2732 </xsd:complexType>
2733 </xsd:element>
2734 <xsd:element name="ListBiometricDataResponse">
2735 <xsd:complexType>
2736 <xsd:annotation>

```

2737 <xsd:documentation>  
 2738 The response to a ListBiometricData request, containing a list  
 2739 of biometric data elements stored for a subject. In the  
 2740 encounter-centric model, the biometric data elements for a  
 2741 specific encounter are returned. If an encounter ID is not  
 2742 specified and encounter data exists for the subject, the list  
 2743 of encounter IDs which contain biometric data is returned.  
 2744 </xsd:documentation>  
 2745 </xsd:annotation>  
 2746 <xsd:sequence>  
 2747 <xsd:element name="ListBiometricDataResponsePackage" nillable="true">  
 2748 <xsd:complexType>  
 2749 <xsd:complexContent>  
 2750 <xsd:extension base="tns:ResponseTemplate">  
 2751 <xsd:sequence>  
 2752 <xsd:element name="Identity" type="tns:BIASIdentity">  
 2753 <xsd:annotation>  
 2754 <xsd:documentation>  
 2755 Includes a list of biometric data elements associated  
 2756 with a subject or encounter or a list of encounter ID's  
 2757 associated with a subject and which contain biometric  
 2758 data.  
 2759 </xsd:documentation>  
 2760 </xsd:annotation>  
 2761 </xsd:element>  
 2762 </xsd:sequence>  
 2763 </xsd:extension>  
 2764 </xsd:complexContent>  
 2765 </xsd:complexType>  
 2766 </xsd:element>  
 2767 </xsd:sequence>  
 2768 </xsd:complexType>  
 2769 </xsd:element>  
 2770 <xsd:element name="PerformFusion">  
 2771 <xsd:complexType>  
 2772 <xsd:annotation>  
 2773 <xsd:documentation>  
 2774 Accepts either match score or match decision information and creates a fused match result.  
 2775 </xsd:documentation>  
 2776 </xsd:annotation>  
 2777 <xsd:sequence>

```

2778 <xsd:element name="PerformFusionRequest" nillable="true">
2779 <xsd:complexType>
2780 <xsd:complexContent>
2781 <xsd:extension base="tns:RequestTemplate">
2782 <xsd:sequence>
2783 <xsd:element name="FusionInput" type="tns:FusionInformationListType" minOccurs="1" maxOccurs="1">
2784 <xsd:annotation>
2785 <xsd:documentation>Score or decision input information to the fusion method.</xsd:documentation>
2786 </xsd:annotation>
2787 </xsd:element>
2788 </xsd:sequence>
2789 </xsd:extension>
2790 </xsd:complexContent>
2791 </xsd:complexType>
2792 </xsd:element>
2793 </xsd:sequence>
2794 </xsd:complexType>
2795 </xsd:element>
2796 <xsd:element name="PerformFusionResponse">
2797 <xsd:complexType>
2798 <xsd:annotation>
2799 <xsd:documentation>
2800 The response to the PerformFusion request.
2801 </xsd:documentation>
2802 </xsd:annotation>
2803 <xsd:sequence>
2804 <xsd:element name="PerformFusionResponsePackage" nillable="true">
2805 <xsd:complexType>
2806 <xsd:complexContent>
2807 <xsd:extension base="tns:ResponseTemplate">
2808 <xsd:sequence>
2809 <xsd:element name="Match" type="tns:MatchType" minOccurs="1" maxOccurs="1">
2810 <xsd:annotation>
2811 <xsd:documentation>Indicates the result of the fusion method</xsd:documentation>
2812 </xsd:annotation>
2813 </xsd:element>
2814 </xsd:sequence>
2815 </xsd:extension>
2816 </xsd:complexContent>
2817 </xsd:complexType>
2818 </xsd:element>

```

```

2819 </xsd:sequence>
2820 </xsd:complexType>
2821 </xsd:element>
2822 <xsd:element name="QueryCapabilities">
2823   <xsd:complexType>
2824     <xsd:annotation>
2825       <xsd:documentation>
2826         Returns a list of the capabilities, options, galleries, etc.
2827         that are supported by the BIAS implementation.
2828       </xsd:documentation>
2829     </xsd:annotation>
2830     <xsd:sequence>
2831       <xsd:element name="QueryCapabilitiesRequest" nillable="true">
2832         <xsd:complexType>
2833           <xsd:complexContent>
2834             <xsd:extension base="tns:RequestTemplate"/>
2835           </xsd:complexContent>
2836         </xsd:complexType>
2837       </xsd:element>
2838     </xsd:sequence>
2839   </xsd:complexType>
2840 </xsd:element>
2841 <xsd:element name="QueryCapabilitiesResponse">
2842   <xsd:complexType>
2843     <xsd:annotation>
2844       <xsd:documentation>
2845         The response to a QueryCapabilities request.
2846       </xsd:documentation>
2847     </xsd:annotation>
2848     <xsd:sequence>
2849       <xsd:element name="QueryCapabilitiesResponsePackage" nillable="true">
2850         <xsd:complexType>
2851           <xsd:complexContent>
2852             <xsd:extension base="tns:ResponseTemplate">
2853               <xsd:sequence>
2854                 <xsd:element name="CapabilityList" type="tns:CapabilityListType" minOccurs="1" maxOccurs="1">
2855                   <xsd:annotation>
2856                     <xsd:documentation>A list of capabilities supported by the BIAS implementation.</xsd:documentation>
2857                   </xsd:annotation>
2858                 </xsd:element>
2859               </xsd:sequence>

```

```

2860 </xsd:extension>
2861 </xsd:complexContent>
2862 </xsd:complexType>
2863 </xsd:element>
2864 </xsd:sequence>
2865 </xsd:complexType>
2866 </xsd:element>
2867 <xsd:element name="RetrieveBiographicInformation">
2868 <xsd:complexType>
2869 <xsd:annotation>
2870 <xsd:documentation>
2871 Retrieves the biographic data associated with a subject ID.
2872 </xsd:documentation>
2873 </xsd:annotation>
2874 <xsd:sequence>
2875 <xsd:element name="RetrieveBiographicInformationRequest" nillable="true">
2876 <xsd:complexType>
2877 <xsd:complexContent>
2878 <xsd:extension base="tns:RequestTemplate">
2879 <xsd:sequence>
2880 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2881 <xsd:annotation>
2882 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2883 encounter.</xsd:documentation>
2884 </xsd:annotation>
2885 </xsd:element>
2886 </xsd:sequence>
2887 </xsd:extension>
2888 </xsd:complexContent>
2889 </xsd:complexType>
2890 </xsd:element>
2891 </xsd:sequence>
2892 </xsd:complexType>
2893 </xsd:element>
2894 <xsd:element name="RetrieveBiographicInformationResponse">
2895 <xsd:complexType>
2896 <xsd:annotation>
2897 <xsd:documentation>
2898 The response to a RetrieveBiographicInformation request,
2899 containing the biographic data associated with a subject ID. In
2900 the encounter-centric model, the biographic data associated with

```



2901 a specified encounter is returned. If the encounter ID is not  
 2902 specified in the encounter-centric model, the biographic  
 2903 information associated with the most recent encounter is returned.  
 2904 </xsd:documentation>  
 2905 </xsd:annotation>  
 2906 <xsd:sequence>  
 2907 <xsd:element name="RetrieveBiographicInformationResponsePackage" nillable="true">  
 2908 <xsd:complexType>  
 2909 <xsd:complexContent>  
 2910 <xsd:extension base="tns:ResponseTemplate">  
 2911 <xsd:sequence>  
 2912 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">  
 2913 <xsd:annotation>  
 2914 <xsd:documentation>Includes the set of biographic data associated with a subject.</xsd:documentation>  
 2915 </xsd:annotation>  
 2916 </xsd:element>  
 2917 </xsd:sequence>  
 2918 </xsd:extension>  
 2919 </xsd:complexContent>  
 2920 </xsd:complexType>  
 2921 </xsd:element>  
 2922 </xsd:sequence>  
 2923 </xsd:complexType>  
 2924 </xsd:element>  
 2925 <xsd:element name="RetrieveBiometricInformation">  
 2926 <xsd:complexType>  
 2927 <xsd:annotation>  
 2928 <xsd:documentation>  
 2929 Retrieves the biometric data associated with a subject ID.  
 2930 </xsd:documentation>  
 2931 </xsd:annotation>  
 2932 <xsd:sequence>  
 2933 <xsd:element name="RetrieveBiometricInformationRequest" nillable="true">  
 2934 <xsd:complexType>  
 2935 <xsd:complexContent>  
 2936 <xsd:extension base="tns:RequestTemplate">  
 2937 <xsd:sequence>  
 2938 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">  
 2939 <xsd:annotation>  
 2940 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an  
 2941 encounter.</xsd:documentation>

```

2942 </xsd:annotation>
2943 </xsd:element>
2944 | <xsd:element name="BiometricType" type="iso-iec19785-3-7oasis\_cbeff:MultipleTypesType" minOccurs="0"
2945 maxOccurs="1">
2946 <xsd:annotation>
2947 <xsd:documentation>The type of biological or behavioral data to retrieve.</xsd:documentation>
2948 </xsd:annotation>
2949 </xsd:element>
2950 </xsd:sequence>
2951 </xsd:extension>
2952 </xsd:complexContent>
2953 </xsd:complexType>
2954 </xsd:element>
2955 </xsd:sequence>
2956 </xsd:complexType>
2957 </xsd:element>
2958 <xsd:element name="RetrieveBiometricInformationResponse">
2959 <xsd:complexType>
2960 <xsd:annotation>
2961 <xsd:documentation>
2962 The response to a RetrieveBiometricInformation request,
2963 containing the biometric data associated with a subject ID. In
2964 the encounter-centric model, the biometric data associated with
2965 a specified encounter is returned. If the encounter ID is not
2966 specified in the encounter-centric model, the biometric
2967 information associated with the most recent encounter is returned.
2968 </xsd:documentation>
2969 </xsd:annotation>
2970 <xsd:sequence>
2971 <xsd:element name="RetrieveBiometricInformationResponsePackage" nillable="true">
2972 <xsd:complexType>
2973 <xsd:complexContent>
2974 <xsd:extension base="tns:ResponseTemplate">
2975 <xsd:sequence>
2976 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2977 <xsd:annotation>
2978 <xsd:documentation>Includes the biometric data associated with a subject.</xsd:documentation>
2979 </xsd:annotation>
2980 </xsd:element>
2981 </xsd:sequence>
2982 </xsd:extension>

```

```

2983 </xsd:complexContent>
2984 </xsd:complexType>
2985 </xsd:element>
2986 </xsd:sequence>
2987 </xsd:complexType>
2988 </xsd:element>
2989 <xsd:element name="SetBiographicData">
2990 <xsd:complexType>
2991 <xsd:annotation>
2992 <xsd:documentation>
2993 Associates biographic data to a given subject record.
2994 </xsd:documentation>
2995 </xsd:annotation>
2996 <xsd:sequence>
2997 <xsd:element name="SetBiographicDataRequest" nillable="true">
2998 <xsd:complexType>
2999 <xsd:complexContent>
3000 <xsd:extension base="tns:RequestTemplate">
3001 <xsd:sequence>
3002 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3003 <xsd:annotation>
3004 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3005 includes the biographic data to store.</xsd:documentation>
3006 </xsd:annotation>
3007 </xsd:element>
3008 </xsd:sequence>
3009 </xsd:extension>
3010 </xsd:complexContent>
3011 </xsd:complexType>
3012 </xsd:element>
3013 </xsd:sequence>
3014 </xsd:complexType>
3015 </xsd:element>
3016 <xsd:element name="SetBiographicDataResponse">
3017 <xsd:complexType>
3018 <xsd:annotation>
3019 <xsd:documentation>
3020 The response to a SetBiographicData request.
3021 </xsd:documentation>
3022 </xsd:annotation>
3023 <xsd:sequence>

```

```

3024 <xsd:element name="SetBiographicDataResponsePackage" nillable="true">
3025 <xsd:complexType>
3026 <xsd:complexContent>
3027 <xsd:extension base="tns:ResponseTemplate">
3028 <xsd:sequence>
3029 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
3030 <xsd:annotation>
3031 <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
3032 encounter.</xsd:documentation>
3033 </xsd:annotation>
3034 </xsd:element>
3035 </xsd:sequence>
3036 </xsd:extension>
3037 </xsd:complexContent>
3038 </xsd:complexType>
3039 </xsd:element>
3040 </xsd:sequence>
3041 </xsd:complexType>
3042 </xsd:element>
3043 <xsd:element name="SetBiometricData">
3044 <xsd:complexType>
3045 <xsd:annotation>
3046 <xsd:documentation>
3047 Associates biometric data to a given subject record.
3048 </xsd:documentation>
3049 </xsd:annotation>
3050 <xsd:sequence>
3051 <xsd:element name="SetBiometricDataRequest" nillable="true">
3052 <xsd:complexType>
3053 <xsd:complexContent>
3054 <xsd:extension base="tns:RequestTemplate">
3055 <xsd:sequence>
3056 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3057 <xsd:annotation>
3058 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3059 includes the biometric data to store.</xsd:documentation>
3060 </xsd:annotation>
3061 </xsd:element>
3062 </xsd:sequence>
3063 </xsd:extension>
3064 </xsd:complexContent>

```

```

3065 </xsd:complexType>
3066 </xsd:element>
3067 </xsd:sequence>
3068 </xsd:complexType>
3069 </xsd:element>
3070 <xsd:element name="SetBiometricDataResponse">
3071 <xsd:complexType>
3072 <xsd:annotation>
3073 <xsd:documentation>
3074 The response to a SetBiometricData request.
3075 </xsd:documentation>
3076 </xsd:annotation>
3077 <xsd:sequence>
3078 <xsd:element name="SetBiometricDataResponsePackage" nillable="true">
3079 <xsd:complexType>
3080 <xsd:complexContent>
3081 <xsd:extension base="tns:ResponseTemplate">
3082 <xsd:sequence>
3083 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
3084 <xsd:annotation>
3085 <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
3086 encounter.</xsd:documentation>
3087 </xsd:annotation>
3088 </xsd:element>
3089 </xsd:sequence>
3090 </xsd:extension>
3091 </xsd:complexContent>
3092 </xsd:complexType>
3093 </xsd:element>
3094 </xsd:sequence>
3095 </xsd:complexType>
3096 </xsd:element>
3097 <xsd:element name="TransformBiometricData">
3098 <xsd:complexType>
3099 <xsd:annotation>
3100 <xsd:documentation>
3101 Transforms or processes a given biometric in one format into a new target format.
3102 </xsd:documentation>
3103 </xsd:annotation>
3104 <xsd:sequence>
3105 <xsd:element name="TransformBiometricDataRequest" nillable="true">

```

```

3106 <xsd:complexType>
3107 <xsd:complexContent>
3108 <xsd:extension base="tns:RequestTemplate">
3109 <xsd:sequence>
3110 <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="1" maxOccurs="1">
3111 <xsd:annotation>
3112 <xsd:documentation>Data structure containing the biometric information to be
3113 transformed.</xsd:documentation>
3114 </xsd:annotation>
3115 </xsd:element>
3116 <xsd:element name="TransformOperation" type="xsd:unsignedLong" minOccurs="1" maxOccurs="1">
3117 <xsd:annotation>
3118 <xsd:documentation>Value indicating the type of transformation to perform.</xsd:documentation>
3119 </xsd:annotation>
3120 </xsd:element>
3121 <xsd:element name="TransformControl" type="xsd:string" minOccurs="0" maxOccurs="1">
3122 <xsd:annotation>
3123 <xsd:documentation> Specifies controls for the requested transform operation.</xsd:documentation>
3124 </xsd:annotation>
3125 </xsd:element>
3126 </xsd:sequence>
3127 </xsd:extension>
3128 </xsd:complexContent>
3129 </xsd:complexType>
3130 </xsd:element>
3131 </xsd:sequence>
3132 </xsd:complexType>
3133 </xsd:element>
3134 <xsd:element name="TransformBiometricDataResponse">
3135 <xsd:complexType>
3136 <xsd:annotation>
3137 <xsd:documentation>
3138 The response to a TransformBiometricData request.
3139 </xsd:documentation>
3140 </xsd:annotation>
3141 <xsd:sequence>
3142 <xsd:element name="TransformBiometricDataResponsePackage" nillable="true">
3143 <xsd:complexType>
3144 <xsd:complexContent>
3145 <xsd:extension base="tns:ResponseTemplate">
3146 <xsd:sequence>

```

```

3147 <xsd:element name="OutputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
3148 <xsd:annotation>
3149 <xsd:documentation>Data structure containing the new, transformed biometric
3150 information.</xsd:documentation>
3151 </xsd:annotation>
3152 </xsd:element>
3153 </xsd:sequence>
3154 </xsd:extension>
3155 </xsd:complexContent>
3156 </xsd:complexType>
3157 </xsd:element>
3158 </xsd:sequence>
3159 </xsd:complexType>
3160 </xsd:element>
3161 <xsd:element name="UpdateBiographicData">
3162 <xsd:complexType>
3163 <xsd:annotation>
3164 <xsd:documentation>
3165 Updates the biographic data for a given subject record.
3166 </xsd:documentation>
3167 </xsd:annotation>
3168 <xsd:sequence>
3169 <xsd:element name="UpdateBiographicDataRequest" nillable="true">
3170 <xsd:complexType>
3171 <xsd:complexContent>
3172 <xsd:extension base="tns:RequestTemplate">
3173 <xsd:sequence>
3174 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3175 <xsd:annotation>
3176 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3177 includes the biographic data to update.</xsd:documentation>
3178 </xsd:annotation>
3179 </xsd:element>
3180 </xsd:sequence>
3181 </xsd:extension>
3182 </xsd:complexContent>
3183 </xsd:complexType>
3184 </xsd:element>
3185 </xsd:sequence>
3186 </xsd:complexType>
3187 </xsd:element>

```

```

3188 <xsd:element name="UpdateBiographicDataResponse">
3189 <xsd:complexType>
3190 <xsd:annotation>
3191 <xsd:documentation>
3192 The response to an UpdateBiographicData request.
3193 </xsd:documentation>
3194 </xsd:annotation>
3195 <xsd:sequence>
3196 <xsd:element name="UpdateBiographicDataResponsePackage" nillable="true">
3197 <xsd:complexType>
3198 <xsd:complexContent>
3199 <xsd:extension base="tns:ResponseTemplate"/>
3200 </xsd:complexContent>
3201 </xsd:complexType>
3202 </xsd:element>
3203 </xsd:sequence>
3204 </xsd:complexType>
3205 </xsd:element>
3206 <xsd:element name="UpdateBiometricData">
3207 <xsd:complexType>
3208 <xsd:annotation>
3209 <xsd:documentation>
3210 Updates a single biometric sample for a given subject record.
3211 </xsd:documentation>
3212 </xsd:annotation>
3213 <xsd:sequence>
3214 <xsd:element name="UpdateBiometricDataRequest" nillable="true">
3215 <xsd:complexType>
3216 <xsd:complexContent>
3217 <xsd:extension base="tns:RequestTemplate">
3218 <xsd:sequence>
3219 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3220 <xsd:annotation>
3221 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3222 includes the biometric data to update.</xsd:documentation>
3223 </xsd:annotation>
3224 </xsd:element>
3225 <xsd:element name="Merge" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3226 <xsd:annotation>
3227 <xsd:documentation>Value indicating if the input biometric sample should be merged with any existing biometric
3228 information.</xsd:documentation>

```



```

3229     </xsd:annotation>
3230 </xsd:element>
3231 </xsd:sequence>
3232 </xsd:extension>
3233 </xsd:complexContent>
3234 </xsd:complexType>
3235 </xsd:element>
3236 </xsd:sequence>
3237 </xsd:complexType>
3238 </xsd:element>
3239 <xsd:element name="UpdateBiometricDataResponse">
3240 <xsd:complexType>
3241 <xsd:annotation>
3242 <xsd:documentation>
3243 The response to an UpdateBiometricData request.
3244 </xsd:documentation>
3245 </xsd:annotation>
3246 <xsd:sequence>
3247 <xsd:element name="UpdateBiometricDataResponsePackage" nillable="true">
3248 <xsd:complexType>
3249 <xsd:complexContent>
3250 <xsd:extension base="tns:ResponseTemplate"/>
3251 </xsd:complexContent>
3252 </xsd:complexType>
3253 </xsd:element>
3254 </xsd:sequence>
3255 </xsd:complexType>
3256 </xsd:element>
3257 <xsd:element name="VerifySubject">
3258 <xsd:complexType>
3259 <xsd:annotation>
3260 <xsd:documentation>
3261 Performs a 1:1 verification match between a given biometric and
3262 either a claim to identity in a given gallery or another given
3263 biometric.
3264 </xsd:documentation>
3265 </xsd:annotation>
3266 <xsd:sequence>
3267 <xsd:element name="VerifySubjectRequest" nillable="true">
3268 <xsd:complexType>
3269 <xsd:complexContent>

```

```

3270 <xsd:extension base="tns:RequestTemplate">
3271 <xsd:sequence>
3272 <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
3273 <xsd:annotation>
3274 <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
3275 member.</xsd:documentation>
3276 </xsd:annotation>
3277 </xsd:element>
3278 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3279 <xsd:annotation>
3280 <xsd:documentation>Includes the identifying information and/or input and reference biometric
3281 samples.</xsd:documentation>
3282 </xsd:annotation>
3283 </xsd:element>
3284 </xsd:sequence>
3285 </xsd:extension>
3286 </xsd:complexContent>
3287 </xsd:complexType>
3288 </xsd:element>
3289 </xsd:sequence>
3290 </xsd:complexType>
3291 </xsd:element>
3292 <xsd:element name="VerifySubjectResponse">
3293 <xsd:complexType>
3294 <xsd:annotation>
3295 <xsd:documentation>
3296 The response to a VerifySubject request.
3297 </xsd:documentation>
3298 </xsd:annotation>
3299 <xsd:sequence>
3300 <xsd:element name="VerifySubjectResponsePackage" nillable="true">
3301 <xsd:complexType>
3302 <xsd:complexContent>
3303 <xsd:extension base="tns:ResponseTemplate">
3304 <xsd:sequence>
3305 <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3306 <xsd:annotation>
3307 <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with the
3308 Identity Claim or the Reference BIR.</xsd:documentation>
3309 </xsd:annotation>
3310 </xsd:element>

```

```

3311 <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
3312 <xsd:annotation>
3313 <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3314 </xsd:annotation>
3315 </xsd:element>
3316 </xsd:sequence>
3317 </xsd:extension>
3318 </xsd:complexContent>
3319 </xsd:complexType>
3320 </xsd:element>
3321 </xsd:sequence>
3322 </xsd:complexType>
3323 </xsd:element>
3324 <xsd:element name="Enroll">
3325 <xsd:complexType>
3326 <xsd:annotation>
3327 <xsd:documentation>
3328 The Enroll aggregate service adds a new subject or, in an
3329 encounter-centric model, a new encounter to the system. This may
3330 be accomplished in a number of different ways according to
3331 system requirements and/or resources. If the Enroll aggregate
3332 service is implemented as a synchronous service, the
3333 implementing system immediately processes the request and
3334 returns the results in the ReturnData parameter. If the Enroll
3335 aggregate service is implemented as an asynchronous service, the
3336 implementing system returns a token in the ReturnData
3337 parameter, which is an indication that the request is being
3338 handled asynchronously. In this case, the GetEnrollResults
3339 service is used to poll for the results of the Enroll request.
3340 </xsd:documentation>
3341 </xsd:annotation>
3342 <xsd:sequence>
3343 <xsd:element name="EnrollRequest" nillable="true">
3344 <xsd:complexType>
3345 <xsd:complexContent>
3346 <xsd:extension base="tns:AggregateRequestTemplate"/>
3347 </xsd:complexContent>
3348 </xsd:complexType>
3349 </xsd:element>
3350 </xsd:sequence>
3351 </xsd:complexType>

```

```

3352 </xsd:element>
3353 <xsd:element name="EnrollResponse">
3354 <xsd:complexType>
3355 <xsd:annotation>
3356 <xsd:documentation>The response to an Enroll request.</xsd:documentation>
3357 </xsd:annotation>
3358 <xsd:sequence>
3359 <xsd:element name="EnrollResponsePackage" nillable="true">
3360 <xsd:complexType>
3361 <xsd:complexContent>
3362 <xsd:extension base="tns:AggregateResponseTemplate"/>
3363 </xsd:complexContent>
3364 </xsd:complexType>
3365 </xsd:element>
3366 </xsd:sequence>
3367 </xsd:complexType>
3368 </xsd:element>
3369 <xsd:element name="GetEnrollResults">
3370 <xsd:complexType>
3371 <xsd:annotation>
3372 <xsd:documentation>
3373 The GetEnrollResults aggregate service retrieves the enrollment
3374 results for the specified token. This service is used in
3375 conjunction with the Enroll aggregate service. If the Enroll
3376 aggregate service is implemented as an asynchronous service, the
3377 implementing system returns a token, and the GetEnrollResults
3378 service is used to poll for the results of the original Enroll
3379 request.
3380 </xsd:documentation>
3381 </xsd:annotation>
3382 <xsd:sequence>
3383 <xsd:element name="GetEnrollResultsRequest" nillable="true">
3384 <xsd:complexType>
3385 <xsd:complexContent>
3386 <xsd:extension base="tns:RequestTemplate">
3387 <xsd:sequence>
3388 <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3389 <xsd:annotation>
3390 <xsd:documentation>A value used to retrieve the results of the Enroll request.</xsd:documentation>
3391 </xsd:annotation>
3392 </xsd:element>

```

```

3393 </xsd:sequence>
3394 </xsd:extension>
3395 </xsd:complexContent>
3396 </xsd:complexType>
3397 </xsd:element>
3398 </xsd:sequence>
3399 </xsd:complexType>
3400 </xsd:element>
3401 <xsd:element name="GetEnrollResultsResponse">
3402 <xsd:complexType>
3403 <xsd:annotation>
3404 <xsd:documentation>The response to a GetEnrollResults request.</xsd:documentation>
3405 </xsd:annotation>
3406 <xsd:sequence>
3407 <xsd:element name="GetEnrollResultsResponsePackage" nillable="true">
3408 <xsd:complexType>
3409 <xsd:complexContent>
3410 <xsd:extension base="tns:AggregateResponseTemplate"/>
3411 </xsd:complexContent>
3412 </xsd:complexType>
3413 </xsd:element>
3414 </xsd:sequence>
3415 </xsd:complexType>
3416 </xsd:element>
3417 <xsd:element name="GetIdentifyResults">
3418 <xsd:complexType>
3419 <xsd:annotation>
3420 <xsd:documentation>
3421 The GetIdentifyResults aggregate service retrieves the
3422 identification results for the specified token. This service is
3423 used in conjunction with the Identify aggregate service. If the
3424 Identify aggregate service is implemented as an asynchronous
3425 service, the implementing system returns a token, and the
3426 GetIdentifyResults service is used to poll for the results of
3427 the original Identify request.
3428 </xsd:documentation>
3429 </xsd:annotation>
3430 <xsd:sequence>
3431 <xsd:element name="GetIdentifyResultsRequest" nillable="true">
3432 <xsd:complexType>
3433 <xsd:complexContent>

```

```

3434 <xsd:extension base="tns:RequestTemplate">
3435 <xsd:sequence>
3436 <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3437 <xsd:annotation>
3438 <xsd:documentation>A value used to retrieve the results of the Identify request.</xsd:documentation>
3439 </xsd:annotation>
3440 </xsd:element>
3441 </xsd:sequence>
3442 </xsd:extension>
3443 </xsd:complexContent>
3444 </xsd:complexType>
3445 </xsd:element>
3446 </xsd:sequence>
3447 </xsd:complexType>
3448 </xsd:element>
3449 <xsd:element name="GetIdentifyResultsResponse">
3450 <xsd:complexType>
3451 <xsd:annotation>
3452 <xsd:documentation>The response to a GetIdentifyResults request.</xsd:documentation>
3453 </xsd:annotation>
3454 <xsd:sequence>
3455 <xsd:element name="GetIdentifyResultsResponsePackage" nillable="true">
3456 <xsd:complexType>
3457 <xsd:complexContent>
3458 <xsd:extension base="tns:AggregateResponseTemplate"/>
3459 </xsd:complexContent>
3460 </xsd:complexType>
3461 </xsd:element>
3462 </xsd:sequence>
3463 </xsd:complexType>
3464 </xsd:element>
3465 <xsd:element name="GetVerifyResults">
3466 <xsd:complexType>
3467 <xsd:annotation>
3468 <xsd:documentation>
3469 The GetVerifyResults aggregate service retrieves the verification
3470 results for the specified token. This service is used in
3471 conjunction with the Verify aggregate service. If the Verify
3472 aggregate service is implemented as an asynchronous service, the
3473 implementing system returns a token, and the GetVerifyResults
3474 service is used to poll for the results of the original Verify

```

```

3475 request.
3476 </xsd:documentation>
3477 </xsd:annotation>
3478 <xsd:sequence>
3479 <xsd:element name="GetVerifyResultsRequest" nillable="true">
3480 <xsd:complexType>
3481 <xsd:complexContent>
3482 <xsd:extension base="tns:RequestTemplate">
3483 <xsd:sequence>
3484 <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3485 <xsd:annotation>
3486 <xsd:documentation>A value used to retrieve the results of the Verify request.</xsd:documentation>
3487 </xsd:annotation>
3488 </xsd:element>
3489 </xsd:sequence>
3490 </xsd:extension>
3491 </xsd:complexContent>
3492 </xsd:complexType>
3493 </xsd:element>
3494 </xsd:sequence>
3495 </xsd:complexType>
3496 </xsd:element>
3497 <xsd:element name="GetVerifyResultsResponse">
3498 <xsd:complexType>
3499 <xsd:annotation>
3500 <xsd:documentation>The response to a GetVerifyResults request.</xsd:documentation>
3501 </xsd:annotation>
3502 <xsd:sequence>
3503 <xsd:element name="GetVerifyResultsResponsePackage" nillable="true">
3504 <xsd:complexType>
3505 <xsd:complexContent>
3506 <xsd:extension base="tns:AggregateResponseTemplate">
3507 <xsd:sequence>
3508 <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3509 <xsd:annotation>
3510 <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with the
3511 Identity Claim or the Reference BIR.</xsd:documentation>
3512 </xsd:annotation>
3513 </xsd:element>
3514 <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
3515 <xsd:annotation>

```

```

3516 <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3517 </xsd:annotation>
3518 </xsd:element>
3519 </xsd:sequence>
3520 </xsd:extension>
3521 </xsd:complexContent>
3522 </xsd:complexType>
3523 </xsd:element>
3524 </xsd:sequence>
3525 </xsd:complexType>
3526 </xsd:element>
3527 <xsd:element name="Identify">
3528 <xsd:complexType>
3529 <xsd:annotation>
3530 <xsd:documentation>
3531 The Identify aggregate service performs an identification
3532 function according to system requirements and/or resources. If
3533 the Identify aggregate service is implemented as a synchronous
3534 service, the implementing system immediately processes the
3535 request and returns the results in the ReturnData parameter. If
3536 the Identify aggregate service is implemented as an asynchronous
3537 service, the implementing system returns a token in the
3538 ReturnData parameter, which is an indication that the request is
3539 being handled asynchronously. In this case, the
3540 GetIdentifyResults service is used to poll for the results of
3541 the Identify request.
3542 </xsd:documentation>
3543 </xsd:annotation>
3544 <xsd:sequence>
3545 <xsd:element name="IdentifyRequest" nillable="true">
3546 <xsd:complexType>
3547 <xsd:complexContent>
3548 <xsd:extension base="tns:AggregateRequestTemplate"/>
3549 </xsd:complexContent>
3550 </xsd:complexType>
3551 </xsd:element>
3552 </xsd:sequence>
3553 </xsd:complexType>
3554 </xsd:element>
3555 <xsd:element name="IdentifyResponse">
3556 <xsd:complexType>

```



```

3557 <xsd:annotation>
3558 <xsd:documentation>The response to an Identify request.</xsd:documentation>
3559 </xsd:annotation>
3560 <xsd:sequence>
3561 <xsd:element name="IdentifyResponsePackage" nillable="true">
3562 <xsd:complexType>
3563 <xsd:complexContent>
3564 <xsd:extension base="tns:AggregateResponseTemplate"/>
3565 </xsd:complexContent>
3566 </xsd:complexType>
3567 </xsd:element>
3568 </xsd:sequence>
3569 </xsd:complexType>
3570 </xsd:element>
3571 <xsd:element name="RetrieveInformation">
3572 <xsd:complexType>
3573 <xsd:annotation>
3574 <xsd:documentation>
3575 The RetrieveInformation aggregate service retrieves requested
3576 information about a subject, or in an encounter-centric model
3577 about an encounter. In a person-centric model, this aggregate
3578 service may be used to retrieve both biographic and biometric
3579 information for a subject record. In an encounter-centric model,
3580 this aggregate service may be used to retrieve biographic and/or
3581 biometric information for either a single encounter or all
3582 encounters. Either a SubjectID or EncounterID must be specified
3583 in the Identify parameter.
3584 </xsd:documentation>
3585 </xsd:annotation>
3586 <xsd:sequence>
3587 <xsd:element name="RetrieveInformationRequest" nillable="true">
3588 <xsd:complexType>
3589 <xsd:complexContent>
3590 <xsd:extension base="tns:RequestTemplate">
3591 <xsd:sequence>
3592 <xsd:element name="ProcessingOptions" type="tns:ProcessingOptionsType" minOccurs="1" maxOccurs="1">
3593 <xsd:annotation>
3594 <xsd:documentation>Options that guide how the service request is processed, and may identify what type(s) of
3595 information should be returned.</xsd:documentation>
3596 </xsd:annotation>
3597 </xsd:element>

```

```

3598 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3599 <xsd:annotation>
3600 <xsd:documentation>Includes the identifier of the subject or encounter.</xsd:documentation>
3601 </xsd:annotation>
3602 </xsd:element>
3603 </xsd:sequence>
3604 </xsd:extension>
3605 </xsd:complexContent>
3606 </xsd:complexType>
3607 </xsd:element>
3608 </xsd:sequence>
3609 </xsd:complexType>
3610 </xsd:element>
3611 <xsd:element name="RetrieveInformationResponse">
3612 <xsd:complexType>
3613 <xsd:annotation>
3614 <xsd:documentation>The response to a RetrieveInformation request.</xsd:documentation>
3615 </xsd:annotation>
3616 <xsd:sequence>
3617 <xsd:element name="RetrieveInformationResponsePackage" nillable="true">
3618 <xsd:complexType>
3619 <xsd:complexContent>
3620 <xsd:extension base="tns:AggregateResponseTemplate"/>
3621 </xsd:complexContent>
3622 </xsd:complexType>
3623 </xsd:element>
3624 </xsd:sequence>
3625 </xsd:complexType>
3626 </xsd:element>
3627 <xsd:element name="Verify">
3628 <xsd:complexType>
3629 <xsd:annotation>
3630 <xsd:documentation>
3631 The Verify aggregate service performs a 1:1 verification
3632 function according to system requirements and/or resources.
3633 Either the IdentityClaim or ReferenceBIR input data elements in
3634 the Identity parameter are required. If the Verify aggregate
3635 service is implemented as a synchronous service, the
3636 implementing system immediately processes the request and returns
3637 the results in the ReturnData parameter. If the Verify aggregate
3638 service is implemented as an asynchronous service, the

```

3639 implementing system returns a token in the ReturnData parameter,  
 3640 which is an indication that the request is being handled  
 3641 asynchronously. In this case, the GetVerifyResults service is  
 3642 used to poll for the results of the Verify request.  
 3643 </xsd:documentation>  
 3644 </xsd:annotation>  
 3645 <xsd:sequence>  
 3646 <xsd:element name="VerifyRequest" nillable="true">  
 3647 <xsd:complexType>  
 3648 <xsd:complexContent>  
 3649 <xsd:extension base="tns:AggregateRequestTemplate">  
 3650 <xsd:sequence>  
 3651 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">  
 3652 <xsd:annotation>  
 3653 <xsd:documentation>Includes either the IdentityClaim or ReferenceBIR.</xsd:documentation>  
 3654 </xsd:annotation>  
 3655 </xsd:element>  
 3656 <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">  
 3657 <xsd:annotation>  
 3658 <xsd:documentation>The identifier of the gallery or population group of which the subject must be a  
 3659 member.</xsd:documentation>  
 3660 </xsd:annotation>  
 3661 </xsd:element>  
 3662 </xsd:sequence>  
 3663 </xsd:extension>  
 3664 </xsd:complexContent>  
 3665 </xsd:complexType>  
 3666 </xsd:element>  
 3667 </xsd:sequence>  
 3668 </xsd:complexType>  
 3669 </xsd:element>  
 3670 <xsd:element name="VerifyResponse">  
 3671 <xsd:complexType>  
 3672 <xsd:annotation>  
 3673 <xsd:documentation>The response to a Verify request.</xsd:documentation>  
 3674 </xsd:annotation>  
 3675 <xsd:sequence>  
 3676 <xsd:element name="VerifyResponsePackage" nillable="true">  
 3677 <xsd:complexType>  
 3678 <xsd:complexContent>  
 3679 <xsd:extension base="tns:AggregateResponseTemplate">

```

3680 <xsd:sequence>
3681 <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3682 <xsd:annotation>
3683 <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with the
3684 Identity Claim or the Reference BIR.</xsd:documentation>
3685 </xsd:annotation>
3686 </xsd:element>
3687 <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
3688 <xsd:annotation>
3689 <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3690 </xsd:annotation>
3691 </xsd:element>
3692 </xsd:sequence>
3693 </xsd:extension>
3694 </xsd:complexContent>
3695 </xsd:complexType>
3696 </xsd:element>
3697 </xsd:sequence>
3698 </xsd:complexType>
3699 </xsd:element>
3700 <xsd:element name="BIASFault" type="tns:BIASFaultDetail"/>
3701 </xsd:schema>
3702 </wsdl:types>
3703 <wsdl:message name="BIASFaultMessage">
3704 <wsdl:part name="parameters" element="tns:BIASFault"/>
3705 </wsdl:message>
3706 <wsdl:message name="AddSubjectToGalleryRequestMessage">
3707 <wsdl:part name="parameters" element="tns:AddSubjectToGallery"/>
3708 </wsdl:message>
3709 <wsdl:message name="AddSubjectToGalleryResponseMessage">
3710 <wsdl:part name="parameters" element="tns:AddSubjectToGalleryResponse"/>
3711 </wsdl:message>
3712 <wsdl:message name="CheckQualityRequestMessage">
3713 <wsdl:part name="parameters" element="tns:CheckQuality"/>
3714 </wsdl:message>
3715 <wsdl:message name="CheckQualityResponseMessage">
3716 <wsdl:part name="parameters" element="tns:CheckQualityResponse"/>
3717 </wsdl:message>
3718 <wsdl:message name="ClassifyBiometricDataRequestMessage">
3719 <wsdl:part name="parameters" element="tns:ClassifyBiometricData"/>
3720 </wsdl:message>

```

3721 <wsdl:message name="ClassifyBiometricDataResponseMessage">  
3722 <wsdl:part name="parameters" element="tns:ClassifyBiometricDataResponse"/>  
3723 </wsdl:message>  
3724 <wsdl:message name="CreateSubjectRequestMessage">  
3725 <wsdl:part name="parameters" element="tns:CreateSubject"/>  
3726 </wsdl:message>  
3727 <wsdl:message name="CreateSubjectResponseMessage">  
3728 <wsdl:part name="parameters" element="tns:CreateSubjectResponse"/>  
3729 </wsdl:message>  
3730 <wsdl:message name="DeleteBiographicDataRequestMessage">  
3731 <wsdl:part name="parameters" element="tns:DeleteBiographicData"/>  
3732 </wsdl:message>  
3733 <wsdl:message name="DeleteBiographicDataResponseMessage">  
3734 <wsdl:part name="parameters" element="tns:DeleteBiographicDataResponse"/>  
3735 </wsdl:message>  
3736 <wsdl:message name="DeleteBiometricDataRequestMessage">  
3737 <wsdl:part name="parameters" element="tns:DeleteBiometricData"/>  
3738 </wsdl:message>  
3739 <wsdl:message name="DeleteBiometricDataResponseMessage">  
3740 <wsdl:part name="parameters" element="tns:DeleteBiometricDataResponse"/>  
3741 </wsdl:message>  
3742 <wsdl:message name="DeleteSubjectRequestMessage">  
3743 <wsdl:part name="parameters" element="tns:DeleteSubject"/>  
3744 </wsdl:message>  
3745 <wsdl:message name="DeleteSubjectResponseMessage">  
3746 <wsdl:part name="parameters" element="tns:DeleteSubjectResponse"/>  
3747 </wsdl:message>  
3748 <wsdl:message name="DeleteSubjectFromGalleryRequestMessage">  
3749 <wsdl:part name="parameters" element="tns:DeleteSubjectFromGallery"/>  
3750 </wsdl:message>  
3751 <wsdl:message name="DeleteSubjectFromGalleryResponseMessage">  
3752 <wsdl:part name="parameters" element="tns:DeleteSubjectFromGalleryResponse"/>  
3753 </wsdl:message>  
3754 <wsdl:message name="GetIdentifySubjectResultsRequestMessage">  
3755 <wsdl:part name="parameters" element="tns:GetIdentifySubjectResults"/>  
3756 </wsdl:message>  
3757 <wsdl:message name="GetIdentifySubjectResultsResponseMessage">  
3758 <wsdl:part name="parameters" element="tns:GetIdentifySubjectResultsResponse"/>  
3759 </wsdl:message>  
3760 <wsdl:message name="IdentifySubjectRequestMessage">  
3761 <wsdl:part name="parameters" element="tns:IdentifySubject"/>

```

3762 </wsdl:message>
3763 <wsdl:message name="IdentifySubjectResponseMessage">
3764 <wsdl:part name="parameters" element="tns:IdentifySubjectResponse"/>
3765 </wsdl:message>
3766 <wsdl:message name="ListBiographicDataRequestMessage">
3767 <wsdl:part name="parameters" element="tns:ListBiographicData"/>
3768 </wsdl:message>
3769 <wsdl:message name="ListBiographicDataResponseMessage">
3770 <wsdl:part name="parameters" element="tns:ListBiographicDataResponse"/>
3771 </wsdl:message>
3772 <wsdl:message name="ListBiometricDataRequestMessage">
3773 <wsdl:part name="parameters" element="tns:ListBiometricData"/>
3774 </wsdl:message>
3775 <wsdl:message name="ListBiometricDataResponseMessage">
3776 <wsdl:part name="parameters" element="tns:ListBiometricDataResponse"/>
3777 </wsdl:message>
3778 <wsdl:message name="PerformFusionRequestMessage">
3779 <wsdl:part name="parameters" element="tns:PerformFusion"/>
3780 </wsdl:message>
3781 <wsdl:message name="PerformFusionResponseMessage">
3782 <wsdl:part name="parameters" element="tns:PerformFusionResponse"/>
3783 </wsdl:message>
3784 <wsdl:message name="QueryCapabilitiesRequestMessage">
3785 <wsdl:part name="parameters" element="tns:QueryCapabilities"/>
3786 </wsdl:message>
3787 <wsdl:message name="QueryCapabilitiesResponseMessage">
3788 <wsdl:part name="parameters" element="tns:QueryCapabilitiesResponse"/>
3789 </wsdl:message>
3790 <wsdl:message name="RetrieveBiographicInformationRequestMessage">
3791 <wsdl:part name="parameters" element="tns:RetrieveBiographicInformation"/>
3792 </wsdl:message>
3793 <wsdl:message name="RetrieveBiographicInformationResponseMessage">
3794 <wsdl:part name="parameters" element="tns:RetrieveBiographicInformationResponse"/>
3795 </wsdl:message>
3796 <wsdl:message name="RetrieveBiometricInformationRequestMessage">
3797 <wsdl:part name="parameters" element="tns:RetrieveBiometricInformation"/>
3798 </wsdl:message>
3799 <wsdl:message name="RetrieveBiometricInformationResponseMessage">
3800 <wsdl:part name="parameters" element="tns:RetrieveBiometricInformationResponse"/>
3801 </wsdl:message>
3802 <wsdl:message name="SetBiographicDataRequestMessage">

```

```

3803 <wsdl:part name="parameters" element="tns:SetBiographicData"/>
3804 </wsdl:message>
3805 <wsdl:message name="SetBiographicDataResponseMessage">
3806 <wsdl:part name="parameters" element="tns:SetBiographicDataResponse"/>
3807 </wsdl:message>
3808 <wsdl:message name="SetBiometricDataRequestMessage">
3809 <wsdl:part name="parameters" element="tns:SetBiometricData"/>
3810 </wsdl:message>
3811 <wsdl:message name="SetBiometricDataResponseMessage">
3812 <wsdl:part name="parameters" element="tns:SetBiometricDataResponse"/>
3813 </wsdl:message>
3814 <wsdl:message name="TransformBiometricDataRequestMessage">
3815 <wsdl:part name="parameters" element="tns:TransformBiometricData"/>
3816 </wsdl:message>
3817 <wsdl:message name="TransformBiometricDataResponseMessage">
3818 <wsdl:part name="parameters" element="tns:TransformBiometricDataResponse"/>
3819 </wsdl:message>
3820 <wsdl:message name="UpdateBiographicDataRequestMessage">
3821 <wsdl:part name="parameters" element="tns:UpdateBiographicData"/>
3822 </wsdl:message>
3823 <wsdl:message name="UpdateBiographicDataResponseMessage">
3824 <wsdl:part name="parameters" element="tns:UpdateBiographicDataResponse"/>
3825 </wsdl:message>
3826 <wsdl:message name="UpdateBiometricDataRequestMessage">
3827 <wsdl:part name="parameters" element="tns:UpdateBiometricData"/>
3828 </wsdl:message>
3829 <wsdl:message name="UpdateBiometricDataResponseMessage">
3830 <wsdl:part name="parameters" element="tns:UpdateBiometricDataResponse"/>
3831 </wsdl:message>
3832 <wsdl:message name="VerifySubjectRequestMessage">
3833 <wsdl:part name="parameters" element="tns:VerifySubject"/>
3834 </wsdl:message>
3835 <wsdl:message name="VerifySubjectResponseMessage">
3836 <wsdl:part name="parameters" element="tns:VerifySubjectResponse"/>
3837 </wsdl:message>
3838 <wsdl:message name="EnrollRequestMessage">
3839 <wsdl:part name="parameters" element="tns:Enroll"/>
3840 </wsdl:message>
3841 <wsdl:message name="EnrollResponseMessage">
3842 <wsdl:part name="parameters" element="tns:EnrollResponse"/>
3843 </wsdl:message>

```

```

3844 <wsdl:message name="GetEnrollResultsRequestMessage">
3845 <wsdl:part name="parameters" element="tns:GetEnrollResults"/>
3846 </wsdl:message>
3847 <wsdl:message name="GetEnrollResultsResponseMessage">
3848 <wsdl:part name="parameters" element="tns:GetEnrollResultsResponse"/>
3849 </wsdl:message>
3850 <wsdl:message name="GetIdentifyResultsRequestMessage">
3851 <wsdl:part name="parameters" element="tns:GetIdentifyResults"/>
3852 </wsdl:message>
3853 <wsdl:message name="GetIdentifyResultsResponseMessage">
3854 <wsdl:part name="parameters" element="tns:GetIdentifyResultsResponse"/>
3855 </wsdl:message>
3856 <wsdl:message name="GetVerifyResultsRequestMessage">
3857 <wsdl:part name="parameters" element="tns:GetVerifyResults"/>
3858 </wsdl:message>
3859 <wsdl:message name="GetVerifyResultsResponseMessage">
3860 <wsdl:part name="parameters" element="tns:GetVerifyResultsResponse"/>
3861 </wsdl:message>
3862 <wsdl:message name="IdentifyRequestMessage">
3863 <wsdl:part name="parameters" element="tns:Identify"/>
3864 </wsdl:message>
3865 <wsdl:message name="IdentifyResponseMessage">
3866 <wsdl:part name="parameters" element="tns:IdentifyResponse"/>
3867 </wsdl:message>
3868 <wsdl:message name="RetrieveInformationRequestMessage">
3869 <wsdl:part name="parameters" element="tns:RetrieveInformation"/>
3870 </wsdl:message>
3871 <wsdl:message name="RetrieveInformationResponseMessage">
3872 <wsdl:part name="parameters" element="tns:RetrieveInformationResponse"/>
3873 </wsdl:message>
3874 <wsdl:message name="VerifyRequestMessage">
3875 <wsdl:part name="parameters" element="tns:Verify"/>
3876 </wsdl:message>
3877 <wsdl:message name="VerifyResponseMessage">
3878 <wsdl:part name="parameters" element="tns:VerifyResponse"/>
3879 </wsdl:message>
3880 <wsdl:portType name="BIAS_v1">
3881 <wsdl:operation name="AddSubjectToGallery">
3882 <wsdl:input name="AddSubjectToGallery" message="tns:AddSubjectToGalleryRequestMessage"/>
3883 <wsdl:output name="AddSubjectToGalleryResponse" message="tns:AddSubjectToGalleryResponseMessage"/>
3884 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

```



```

3885 </wsdl:operation>
3886 <wsdl:operation name="CheckQuality">
3887 <wsdl:input name="CheckQuality" message="tns:CheckQualityRequestMessage"/>
3888 <wsdl:output name="CheckQualityResponse" message="tns:CheckQualityResponseMessage"/>
3889 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3890 </wsdl:operation>
3891 <wsdl:operation name="ClassifyBiometricData">
3892 <wsdl:input name="ClassifyBiometricData" message="tns:ClassifyBiometricDataRequestMessage"/>
3893 <wsdl:output name="ClassifyBiometricDataResponse" message="tns:ClassifyBiometricDataResponseMessage"/>
3894 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3895 </wsdl:operation>
3896 <wsdl:operation name="CreateSubject">
3897 <wsdl:input name="CreateSubject" message="tns:CreateSubjectRequestMessage"/>
3898 <wsdl:output name="CreateSubjectResponse" message="tns:CreateSubjectResponseMessage"/>
3899 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3900 </wsdl:operation>
3901 <wsdl:operation name="DeleteBiographicData">
3902 <wsdl:input name="DeleteBiographicData" message="tns:DeleteBiographicDataRequestMessage"/>
3903 <wsdl:output name="DeleteBiographicDataResponse" message="tns:DeleteBiographicDataResponseMessage"/>
3904 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3905 </wsdl:operation>
3906 <wsdl:operation name="DeleteBiometricData">
3907 <wsdl:input name="DeleteBiometricData" message="tns:DeleteBiometricDataRequestMessage"/>
3908 <wsdl:output name="DeleteBiometricDataResponse" message="tns:DeleteBiometricDataResponseMessage"/>
3909 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3910 </wsdl:operation>
3911 <wsdl:operation name="DeleteSubject">
3912 <wsdl:input name="DeleteSubject" message="tns:DeleteSubjectRequestMessage"/>
3913 <wsdl:output name="DeleteSubjectResponse" message="tns:DeleteSubjectResponseMessage"/>
3914 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3915 </wsdl:operation>
3916 <wsdl:operation name="DeleteSubjectFromGallery">
3917 <wsdl:input name="DeleteSubjectFromGallery" message="tns:DeleteSubjectFromGalleryRequestMessage"/>
3918 <wsdl:output name="DeleteSubjectFromGalleryResponse"
3919 message="tns:DeleteSubjectFromGalleryResponseMessage"/>
3920 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3921 </wsdl:operation>
3922 <wsdl:operation name="GetIdentifySubjectResults">
3923 <wsdl:input name="GetIdentifySubjectResults" message="tns:GetIdentifySubjectResultsRequestMessage"/>
3924 <wsdl:output name="GetIdentifySubjectResultsResponse"
3925 message="tns:GetIdentifySubjectResultsResponseMessage"/>

```

```

3926 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3927 </wsdl:operation>
3928 <wsdl:operation name="IdentifySubject">
3929 <wsdl:input name="IdentifySubject" message="tns:IdentifySubjectRequestMessage"/>
3930 <wsdl:output name="IdentifySubjectResponse" message="tns:IdentifySubjectResponseMessage"/>
3931 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3932 </wsdl:operation>
3933 <wsdl:operation name="ListBiographicData">
3934 <wsdl:input name="ListBiographicData" message="tns:ListBiographicDataRequestMessage"/>
3935 <wsdl:output name="ListBiographicDataResponse" message="tns:ListBiographicDataResponseMessage"/>
3936 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3937 </wsdl:operation>
3938 <wsdl:operation name="ListBiometricData">
3939 <wsdl:input name="ListBiometricData" message="tns:ListBiometricDataRequestMessage"/>
3940 <wsdl:output name="ListBiometricDataResponse" message="tns:ListBiometricDataResponseMessage"/>
3941 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3942 </wsdl:operation>
3943 <wsdl:operation name="PerformFusion">
3944 <wsdl:input name="PerformFusion" message="tns:PerformFusionRequestMessage"/>
3945 <wsdl:output name="PerformFusionResponse" message="tns:PerformFusionResponseMessage"/>
3946 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3947 </wsdl:operation>
3948 <wsdl:operation name="QueryCapabilities">
3949 <wsdl:input name="QueryCapabilities" message="tns:QueryCapabilitiesRequestMessage"/>
3950 <wsdl:output name="QueryCapabilitiesResponse" message="tns:QueryCapabilitiesResponseMessage"/>
3951 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3952 </wsdl:operation>
3953 <wsdl:operation name="RetrieveBiographicInformation">
3954 <wsdl:input name="RetrieveBiographicInformation"
3955 message="tns:RetrieveBiographicInformationRequestMessage"/>
3956 <wsdl:output name="RetrieveBiographicInformationResponse"
3957 message="tns:RetrieveBiographicInformationResponseMessage"/>
3958 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3959 </wsdl:operation>
3960 <wsdl:operation name="RetrieveBiometricInformation">
3961 <wsdl:input name="RetrieveBiometricInformation"
3962 message="tns:RetrieveBiometricInformationRequestMessage"/>
3963 <wsdl:output name="RetrieveBiometricInformationResponse"
3964 message="tns:RetrieveBiometricInformationResponseMessage"/>
3965 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3966 </wsdl:operation>
3967 <wsdl:operation name="SetBiographicData">

```

```

3968 <wsdl:input name="SetBiographicData" message="tns:SetBiographicDataRequestMessage"/>
3969 <wsdl:output name="SetBiographicDataResponse" message="tns:SetBiographicDataResponseMessage"/>
3970 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3971 </wsdl:operation>
3972 <wsdl:operation name="SetBiometricData">
3973 <wsdl:input name="SetBiometricData" message="tns:SetBiometricDataRequestMessage"/>
3974 <wsdl:output name="SetBiometricDataResponse" message="tns:SetBiometricDataResponseMessage"/>
3975 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3976 </wsdl:operation>
3977 <wsdl:operation name="TransformBiometricData">
3978 <wsdl:input name="TransformBiometricData" message="tns:TransformBiometricDataRequestMessage"/>
3979 <wsdl:output name="TransformBiometricDataResponse"
3980 message="tns:TransformBiometricDataResponseMessage"/>
3981 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3982 </wsdl:operation>
3983 <wsdl:operation name="UpdateBiographicData">
3984 <wsdl:input name="UpdateBiographicData" message="tns:UpdateBiographicDataRequestMessage"/>
3985 <wsdl:output name="UpdateBiographicDataResponse" message="tns:UpdateBiographicDataResponseMessage"/>
3986 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3987 </wsdl:operation>
3988 <wsdl:operation name="UpdateBiometricData">
3989 <wsdl:input name="UpdateBiometricData" message="tns:UpdateBiometricDataRequestMessage"/>
3990 <wsdl:output name="UpdateBiometricDataResponse" message="tns:UpdateBiometricDataResponseMessage"/>
3991 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3992 </wsdl:operation>
3993 <wsdl:operation name="VerifySubject">
3994 <wsdl:input name="VerifySubject" message="tns:VerifySubjectRequestMessage"/>
3995 <wsdl:output name="VerifySubjectResponse" message="tns:VerifySubjectResponseMessage"/>
3996 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3997 </wsdl:operation>
3998 <wsdl:operation name="Enroll">
3999 <wsdl:input name="Enroll" message="tns:EnrollRequestMessage"/>
4000 <wsdl:output name="EnrollResponse" message="tns:EnrollResponseMessage"/>
4001 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4002 </wsdl:operation>
4003 <wsdl:operation name="GetEnrollResults">
4004 <wsdl:input name="GetEnrollResults" message="tns:GetEnrollResultsRequestMessage"/>
4005 <wsdl:output name="GetEnrollResultsResponse" message="tns:GetEnrollResultsResponseMessage"/>
4006 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4007 </wsdl:operation>
4008 <wsdl:operation name="GetIdentifyResults">

```

```

4009 <wsdl:input name="GetIdentifyResults" message="tns:GetIdentifyResultsRequestMessage"/>
4010 <wsdl:output name="GetIdentifyResultsResponse" message="tns:GetIdentifyResultsResponseMessage"/>
4011 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4012 </wsdl:operation>
4013 <wsdl:operation name="GetVerifyResults">
4014 <wsdl:input name="GetVerifyResults" message="tns:GetVerifyResultsRequestMessage"/>
4015 <wsdl:output name="GetVerifyResultsResponse" message="tns:GetVerifyResultsResponseMessage"/>
4016 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4017 </wsdl:operation>
4018 <wsdl:operation name="Identify">
4019 <wsdl:input name="Identify" message="tns:IdentifyRequestMessage"/>
4020 <wsdl:output name="IdentifyResponse" message="tns:IdentifyResponseMessage"/>
4021 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4022 </wsdl:operation>
4023 <wsdl:operation name="RetrieveInformation">
4024 <wsdl:input name="RetrieveInformation" message="tns:RetrieveInformationRequestMessage"/>
4025 <wsdl:output name="RetrieveInformationResponse" message="tns:RetrieveInformationResponseMessage"/>
4026 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4027 </wsdl:operation>
4028 <wsdl:operation name="Verify">
4029 <wsdl:input name="Verify" message="tns:VerifyRequestMessage"/>
4030 <wsdl:output name="VerifyResponse" message="tns:VerifyResponseMessage"/>
4031 <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4032 </wsdl:operation>
4033 </wsdl:portType>
4034 <wsdl:binding name="BIAS_v1HttpBinding" type="tns:BIAS_v1">
4035 <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
4036 <wsdl:operation name="AddSubjectToGallery">
4037 <soap:operation soapAction="AddSubjectToGallery"/>
4038 <wsdl:input name="AddSubjectToGallery">
4039 <soap:body use="literal"/>
4040 </wsdl:input>
4041 <wsdl:output name="AddSubjectToGalleryResponse">
4042 <soap:body use="literal"/>
4043 </wsdl:output>
4044 <wsdl:fault name="BIASFault">
4045 <soap:fault name="BIASFault" use="literal"/>
4046 </wsdl:fault>
4047 </wsdl:operation>
4048 <wsdl:operation name="CheckQuality">
4049 <soap:operation soapAction="CheckQuality"/>

```

```

4050 <wsdl:input name="CheckQuality">
4051 <soap:body use="literal"/>
4052 </wsdl:input>
4053 <wsdl:output name="CheckQualityResponse">
4054 <soap:body use="literal"/>
4055 </wsdl:output>
4056 <wsdl:fault name="BIASFault">
4057 <soap:fault name="BIASFault" use="literal"/>
4058 </wsdl:fault>
4059 </wsdl:operation>
4060 <wsdl:operation name="ClassifyBiometricData">
4061 <soap:operation soapAction="ClassifyBiometricData"/>
4062 <wsdl:input name="ClassifyBiometricData">
4063 <soap:body use="literal"/>
4064 </wsdl:input>
4065 <wsdl:output name="ClassifyBiometricDataResponse">
4066 <soap:body use="literal"/>
4067 </wsdl:output>
4068 <wsdl:fault name="BIASFault">
4069 <soap:fault name="BIASFault" use="literal"/>
4070 </wsdl:fault>
4071 </wsdl:operation>
4072 <wsdl:operation name="CreateSubject">
4073 <soap:operation soapAction="CreateSubject"/>
4074 <wsdl:input name="CreateSubject">
4075 <soap:body use="literal"/>
4076 </wsdl:input>
4077 <wsdl:output name="CreateSubjectResponse">
4078 <soap:body use="literal"/>
4079 </wsdl:output>
4080 <wsdl:fault name="BIASFault">
4081 <soap:fault name="BIASFault" use="literal"/>
4082 </wsdl:fault>
4083 </wsdl:operation>
4084 <wsdl:operation name="DeleteBiographicData">
4085 <soap:operation soapAction="DeleteBiographicData"/>
4086 <wsdl:input name="DeleteBiographicData">
4087 <soap:body use="literal"/>
4088 </wsdl:input>
4089 <wsdl:output name="DeleteBiographicDataResponse">
4090 <soap:body use="literal"/>

```

```

4091 </wsdl:output>
4092 <wsdl:fault name="BIASFault">
4093 <soap:fault name="BIASFault" use="literal"/>
4094 </wsdl:fault>
4095 </wsdl:operation>
4096 <wsdl:operation name="DeleteBiometricData">
4097 <soap:operation soapAction="DeleteBiometricData"/>
4098 <wsdl:input name="DeleteBiometricData">
4099 <soap:body use="literal"/>
4100 </wsdl:input>
4101 <wsdl:output name="DeleteBiometricDataResponse">
4102 <soap:body use="literal"/>
4103 </wsdl:output>
4104 <wsdl:fault name="BIASFault">
4105 <soap:fault name="BIASFault" use="literal"/>
4106 </wsdl:fault>
4107 </wsdl:operation>
4108 <wsdl:operation name="DeleteSubject">
4109 <soap:operation soapAction="DeleteSubject"/>
4110 <wsdl:input name="DeleteSubject">
4111 <soap:body use="literal"/>
4112 </wsdl:input>
4113 <wsdl:output name="DeleteSubjectResponse">
4114 <soap:body use="literal"/>
4115 </wsdl:output>
4116 <wsdl:fault name="BIASFault">
4117 <soap:fault name="BIASFault" use="literal"/>
4118 </wsdl:fault>
4119 </wsdl:operation>
4120 <wsdl:operation name="DeleteSubjectFromGallery">
4121 <soap:operation soapAction="DeleteSubjectFromGallery"/>
4122 <wsdl:input name="DeleteSubjectFromGallery">
4123 <soap:body use="literal"/>
4124 </wsdl:input>
4125 <wsdl:output name="DeleteSubjectFromGalleryResponse">
4126 <soap:body use="literal"/>
4127 </wsdl:output>
4128 <wsdl:fault name="BIASFault">
4129 <soap:fault name="BIASFault" use="literal"/>
4130 </wsdl:fault>
4131 </wsdl:operation>

```

```

4132 <wsdl:operation name="GetIdentifySubjectResults">
4133 <soap:operation soapAction="GetIdentifySubjectResults"/>
4134 <wsdl:input name="GetIdentifySubjectResults">
4135 <soap:body use="literal"/>
4136 </wsdl:input>
4137 <wsdl:output name="GetIdentifySubjectResultsResponse">
4138 <soap:body use="literal"/>
4139 </wsdl:output>
4140 <wsdl:fault name="BIASFault">
4141 <soap:fault name="BIASFault" use="literal"/>
4142 </wsdl:fault>
4143 </wsdl:operation>
4144 <wsdl:operation name="IdentifySubject">
4145 <soap:operation soapAction="IdentifySubject"/>
4146 <wsdl:input name="IdentifySubject">
4147 <soap:body use="literal"/>
4148 </wsdl:input>
4149 <wsdl:output name="IdentifySubjectResponse">
4150 <soap:body use="literal"/>
4151 </wsdl:output>
4152 <wsdl:fault name="BIASFault">
4153 <soap:fault name="BIASFault" use="literal"/>
4154 </wsdl:fault>
4155 </wsdl:operation>
4156 <wsdl:operation name="ListBiographicData">
4157 <soap:operation soapAction="ListBiographicData"/>
4158 <wsdl:input name="ListBiographicData">
4159 <soap:body use="literal"/>
4160 </wsdl:input>
4161 <wsdl:output name="ListBiographicDataResponse">
4162 <soap:body use="literal"/>
4163 </wsdl:output>
4164 <wsdl:fault name="BIASFault">
4165 <soap:fault name="BIASFault" use="literal"/>
4166 </wsdl:fault>
4167 </wsdl:operation>
4168 <wsdl:operation name="ListBiometricData">
4169 <soap:operation soapAction="ListBiometricData"/>
4170 <wsdl:input name="ListBiometricData">
4171 <soap:body use="literal"/>
4172 </wsdl:input>

```

```

4173 <wsdl:output name="ListBiometricDataResponse">
4174 <soap:body use="literal"/>
4175 </wsdl:output>
4176 <wsdl:fault name="BIASFault">
4177 <soap:fault name="BIASFault" use="literal"/>
4178 </wsdl:fault>
4179 </wsdl:operation>
4180 <wsdl:operation name="PerformFusion">
4181 <soap:operation soapAction="PerformFusion"/>
4182 <wsdl:input name="PerformFusion">
4183 <soap:body use="literal"/>
4184 </wsdl:input>
4185 <wsdl:output name="PerformFusionResponse">
4186 <soap:body use="literal"/>
4187 </wsdl:output>
4188 <wsdl:fault name="BIASFault">
4189 <soap:fault name="BIASFault" use="literal"/>
4190 </wsdl:fault>
4191 </wsdl:operation>
4192 <wsdl:operation name="QueryCapabilities">
4193 <soap:operation soapAction="QueryCapabilities"/>
4194 <wsdl:input name="QueryCapabilities">
4195 <soap:body use="literal"/>
4196 </wsdl:input>
4197 <wsdl:output name="QueryCapabilitiesResponse">
4198 <soap:body use="literal"/>
4199 </wsdl:output>
4200 <wsdl:fault name="BIASFault">
4201 <soap:fault name="BIASFault" use="literal"/>
4202 </wsdl:fault>
4203 </wsdl:operation>
4204 <wsdl:operation name="RetrieveBiographicInformation">
4205 <soap:operation soapAction="RetrieveBiographicInformation"/>
4206 <wsdl:input name="RetrieveBiographicInformation">
4207 <soap:body use="literal"/>
4208 </wsdl:input>
4209 <wsdl:output name="RetrieveBiographicInformationResponse">
4210 <soap:body use="literal"/>
4211 </wsdl:output>
4212 <wsdl:fault name="BIASFault">
4213 <soap:fault name="BIASFault" use="literal"/>

```



```

4214 </wsdl:fault>
4215 </wsdl:operation>
4216 <wsdl:operation name="RetrieveBiometricInformation">
4217 <soap:operation soapAction="RetrieveBiometricInformation"/>
4218 <wsdl:input name="RetrieveBiometricInformation">
4219 <soap:body use="literal"/>
4220 </wsdl:input>
4221 <wsdl:output name="RetrieveBiometricInformationResponse">
4222 <soap:body use="literal"/>
4223 </wsdl:output>
4224 <wsdl:fault name="BIASFault">
4225 <soap:fault name="BIASFault" use="literal"/>
4226 </wsdl:fault>
4227 </wsdl:operation>
4228 <wsdl:operation name="SetBiographicData">
4229 <soap:operation soapAction="SetBiographicData"/>
4230 <wsdl:input name="SetBiographicData">
4231 <soap:body use="literal"/>
4232 </wsdl:input>
4233 <wsdl:output name="SetBiographicDataResponse">
4234 <soap:body use="literal"/>
4235 </wsdl:output>
4236 <wsdl:fault name="BIASFault">
4237 <soap:fault name="BIASFault" use="literal"/>
4238 </wsdl:fault>
4239 </wsdl:operation>
4240 <wsdl:operation name="SetBiometricData">
4241 <soap:operation soapAction="SetBiometricData"/>
4242 <wsdl:input name="SetBiometricData">
4243 <soap:body use="literal"/>
4244 </wsdl:input>
4245 <wsdl:output name="SetBiometricDataResponse">
4246 <soap:body use="literal"/>
4247 </wsdl:output>
4248 <wsdl:fault name="BIASFault">
4249 <soap:fault name="BIASFault" use="literal"/>
4250 </wsdl:fault>
4251 </wsdl:operation>
4252 <wsdl:operation name="TransformBiometricData">
4253 <soap:operation soapAction="TransformBiometricData"/>
4254 <wsdl:input name="TransformBiometricData">

```

```

4255 <soap:body use="literal"/>
4256 </wsdl:input>
4257 <wsdl:output name="TransformBiometricDataResponse">
4258 <soap:body use="literal"/>
4259 </wsdl:output>
4260 <wsdl:fault name="BIASFault">
4261 <soap:fault name="BIASFault" use="literal"/>
4262 </wsdl:fault>
4263 </wsdl:operation>
4264 <wsdl:operation name="UpdateBiographicData">
4265 <soap:operation soapAction="UpdateBiographicData"/>
4266 <wsdl:input name="UpdateBiographicData">
4267 <soap:body use="literal"/>
4268 </wsdl:input>
4269 <wsdl:output name="UpdateBiographicDataResponse">
4270 <soap:body use="literal"/>
4271 </wsdl:output>
4272 <wsdl:fault name="BIASFault">
4273 <soap:fault name="BIASFault" use="literal"/>
4274 </wsdl:fault>
4275 </wsdl:operation>
4276 <wsdl:operation name="UpdateBiometricData">
4277 <soap:operation soapAction="UpdateBiometricData"/>
4278 <wsdl:input name="UpdateBiometricData">
4279 <soap:body use="literal"/>
4280 </wsdl:input>
4281 <wsdl:output name="UpdateBiometricDataResponse">
4282 <soap:body use="literal"/>
4283 </wsdl:output>
4284 <wsdl:fault name="BIASFault">
4285 <soap:fault name="BIASFault" use="literal"/>
4286 </wsdl:fault>
4287 </wsdl:operation>
4288 <wsdl:operation name="VerifySubject">
4289 <soap:operation soapAction="VerifySubject"/>
4290 <wsdl:input name="VerifySubject">
4291 <soap:body use="literal"/>
4292 </wsdl:input>
4293 <wsdl:output name="VerifySubjectResponse">
4294 <soap:body use="literal"/>
4295 </wsdl:output>

```

```

4296 <wsdl:fault name="BIASFault">
4297 <soap:fault name="BIASFault" use="literal"/>
4298 </wsdl:fault>
4299 </wsdl:operation>
4300 <wsdl:operation name="Enroll">
4301 <soap:operation soapAction="Enroll"/>
4302 <wsdl:input name="Enroll">
4303 <soap:body use="literal"/>
4304 </wsdl:input>
4305 <wsdl:output name="EnrollResponse">
4306 <soap:body use="literal"/>
4307 </wsdl:output>
4308 <wsdl:fault name="BIASFault">
4309 <soap:fault name="BIASFault" use="literal"/>
4310 </wsdl:fault>
4311 </wsdl:operation>
4312 <wsdl:operation name="GetEnrollResults">
4313 <soap:operation soapAction="GetEnrollResults"/>
4314 <wsdl:input name="GetEnrollResults">
4315 <soap:body use="literal"/>
4316 </wsdl:input>
4317 <wsdl:output name="GetEnrollResultsResponse">
4318 <soap:body use="literal"/>
4319 </wsdl:output>
4320 <wsdl:fault name="BIASFault">
4321 <soap:fault name="BIASFault" use="literal"/>
4322 </wsdl:fault>
4323 </wsdl:operation>
4324 <wsdl:operation name="GetIdentifyResults">
4325 <soap:operation soapAction="GetIdentifyResults"/>
4326 <wsdl:input name="GetIdentifyResults">
4327 <soap:body use="literal"/>
4328 </wsdl:input>
4329 <wsdl:output name="GetIdentifyResultsResponse">
4330 <soap:body use="literal"/>
4331 </wsdl:output>
4332 <wsdl:fault name="BIASFault">
4333 <soap:fault name="BIASFault" use="literal"/>
4334 </wsdl:fault>
4335 </wsdl:operation>
4336 <wsdl:operation name="GetVerifyResults">

```

```

4337 <soap:operation soapAction="GetVerifyResults"/>
4338 <wsdl:input name="GetVerifyResults">
4339 <soap:body use="literal"/>
4340 </wsdl:input>
4341 <wsdl:output name="GetVerifyResultsResponse">
4342 <soap:body use="literal"/>
4343 </wsdl:output>
4344 <wsdl:fault name="BIASFault">
4345 <soap:fault name="BIASFault" use="literal"/>
4346 </wsdl:fault>
4347 </wsdl:operation>
4348 <wsdl:operation name="Identify">
4349 <soap:operation soapAction="Identify"/>
4350 <wsdl:input name="Identify">
4351 <soap:body use="literal"/>
4352 </wsdl:input>
4353 <wsdl:output name="IdentifyResponse">
4354 <soap:body use="literal"/>
4355 </wsdl:output>
4356 <wsdl:fault name="BIASFault">
4357 <soap:fault name="BIASFault" use="literal"/>
4358 </wsdl:fault>
4359 </wsdl:operation>
4360 <wsdl:operation name="RetrieveInformation">
4361 <soap:operation soapAction="RetrieveInformation"/>
4362 <wsdl:input name="RetrieveInformation">
4363 <soap:body use="literal"/>
4364 </wsdl:input>
4365 <wsdl:output name="RetrieveInformationResponse">
4366 <soap:body use="literal"/>
4367 </wsdl:output>
4368 <wsdl:fault name="BIASFault">
4369 <soap:fault name="BIASFault" use="literal"/>
4370 </wsdl:fault>
4371 </wsdl:operation>
4372 <wsdl:operation name="Verify">
4373 <soap:operation soapAction="Verify"/>
4374 <wsdl:input name="Verify">
4375 <soap:body use="literal"/>
4376 </wsdl:input>
4377 <wsdl:output name="VerifyResponse">

```

```
4378 <soap:body use="literal"/>
4379 </wsdl:output>
4380 <wsdl:fault name="BIASFault">
4381 <soap:fault name="BIASFault" use="literal"/>
4382 </wsdl:fault>
4383 </wsdl:operation>
4384 </wsdl:binding>
4385 <wsdl:service name="BIAS_v1">
4386 <wsdl:port name="BIAS_v1HttpBindingPort" binding="tns:BIAS_v1HttpBinding">
4387 <soap:address location="http://your_URI/IRI_to_your_BIAS_implementation"/>
4388 </wsdl:port>
4389 </wsdl:service>
4390 </wsdl:definitions>
```

## **Annex B: BIAS Patron format specification**

The BIAS SOAP Profile defines an XML CBEFF Patron Format based on, but tailored from, Clause 13/15 of ISO/IEC 19785-3 [CBEFF3] as specified below.

### **B.1 Patron**

Organization for the Advancement of Structured Information Standards (OASIS)

### **B.2 Patron identifier**

82 (0052 Hex).

This has been allocated by the Registration Authority for ISO/IEC 19785-2.

### **B.3 Patron format name**

OASIS BIAS CBEFF XML Patron Format

### **B.4 Patron format identifier**

01 (0001 Hex).

This has been registered in accordance with ISO/IEC 19785-2.

### **B.5 ASN.1 object identifier for this patron format**

No ASN.1 object identifiers are assigned to this patron format

### **B.6 Domain of use**

This clause specifies a patron format based on XML that is designed to be friendly with code generation tools. It defines a CBEFF structure that allows for the creation of simple, complex, and multi-modal BIRs for use within BIAS transactions.

### **B.7 Version identifier**

This patron format specification has a version identifier of (major 1, minor 0).

### **B.8 CBEFF version**

This specification conforms to CBEFF version (major 2, minor 0).

## **B.9 General**

**B.9.1** This patron format is based on W3C XML 1.0. It supports all the mandatory and optional data elements specified in ISO/IEC 19785-1. It can support either a simple BIR or a complex BIR structure where each intermediate node or leaf of the structure is itself a BIR (called a "child BIR").

**B.9.2** Most fields in this patron format are optional. Some mandatory and optional fields are represented by XML elements, others are represented by attributes of XML elements. The presence of an optional field in a BIR is signaled by simply including the corresponding element or attribute, and its absence is signaled by simply omitting the corresponding element or attribute.

**B.9.3** Special encodings are specified for integers (see B.17), octet strings (see B.18), and date and time-of-the-day abstract values (see B.19).

**B.9.4** An instance of a BIR or child BIR contains either a BDB or one or more BIR children, but never contains both.

**B.9.5** An extension mechanism is specified, which enables the inclusion of application-specific data (not standardized) within a BIR or child BIR (see B.11.1.6).

## **B.10 Specification**

**B.10.1** In the rest of this clause, the terms "element" and "attribute" are used with the meaning of "XML element" and "XML attribute", respectively.

**B.10.2** The namespace with the name "<http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/>" is called the patron format namespace of this patron format.

**B.10.3** All elements defined in this patron format have the patron format namespace name. All attribute names are unqualified.

**B.10.4** An instance of a BIR shall be represented as a <BIR> element (see B.11).

**B.10.5** The <BIR> element may be the root of an XML document, but this is not required.

**B.10.6** The portion of the XML document consisting of the <BIR> element and its whole content shall be valid according to the XML schema provided in B.22.

NOTE 1 – Validity according to that XML schema does not imply that the <BIR> element satisfies all the requirements in the normative text of this specification, as there are some requirements that cannot be (or are not) formally expressed in the XML schema.

NOTE 2 – When the <BIR> element is the root of an XML document, the UTF-8 character encoding is recommended for the XML document, because it will usually produce a smaller encoding.

**B.10.7** The abstract value NO VALUE AVAILABLE, for any CBEFF data element that supports this abstract value, shall be encoded as the omission of the corresponding element or attribute both in the <BIR> element and in all of its ancestor <BIR> elements.

NOTE – The inheritance mechanism specified in B.14.2.1, B.15.2.1 and B.16.2.1 causes a data element of a BIR to inherit an abstract value (different from NO VALUE AVAILABLE) from its closest ancestor <BIR> element that contains that element or attribute when the <BIR> element in question does not contain it. If any <BIR> element in a hierarchy of <BIR> elements specifies an abstract value for a given data element, that abstract value can be overridden by a different abstract value in any of its descendant <BIR> elements, but the overriding abstract value can never be NO VALUE AVAILABLE.

## **B.11 Element <BIR>**

### **B.11.1 Syntax**

**B.11.1.1** This element shall have no attributes, and shall have a content consisting of the following (in order):

- a) an optional <Version> element (see B.12);
- b) an optional <CBEFFVersion> element (see B.13);
- c) zero or more application-specific elements;
- d) a mandatory <BIRInfo> element (see B.14);
- e) an optional <BDBInfo> element (see B.15);
- f) an optional <SBInfo> element (see B.16);
- g) zero or more <BIR> elements (see B.11);
- h) either an optional <BDB> element that shall contain a valid representation of an octet string (see B.18), or an optional <bdbx> element that shall contain a valid XML string;
- i) an optional <SB> element – the content of this element shall be a valid representation of an octet string.

**B.11.1.2** The <BDB> or <bdbx> element shall not be present if one or more child <BIR> elements are present, and shall be present if no child <BIR> elements are present.

**B.11.1.3** The <SB> element shall be absent unless its presence is required by F.14.2.2 or permitted by F.15.2.3.

**B.11.1.4** If the <BDB> or <bdbx> element is present, then the <BDBInfo> element shall also be present.

**B.11.1.5** If the <SB> element is present, then the <SBInfo> element shall also be present.

**B.11.1.6** The number of application-specific elements and their name, namespace name, attributes, and content are not defined in this patron format specification. However, the namespace name of those elements shall be different from the patron format namespace name (see B.10.2).

### **B.11.2 Semantics**

**B.11.2.1** This element is either a complex or a simple BIR, depending on which child elements are present. If a child <BDB> or <bdbx> element is present, this element is a simple BIR. If one or more child <BIR> elements are present, this element is a complex BIR.

**B.11.2.2** The elements <Version>, <CBEFFVersion>, <BIRInfo>, <BDBInfo>, and <SBInfo> and their content form the standard biometric header of the BIR.

**B.11.2.3** The <Version> element (if present) carries the major and minor version number of this patron format.

**B.11.2.4** The <CBEFFVersion> element (if present) carries the major and minor version number of the CBEFF standard.

**B.11.2.5** Each <BIR> element is a whole BIR (of the same patron format) that is a child BIR of the BIR.



**B.11.2.6** The **<BDB>** or **<bdbx>** element (if present) carries the biometric data block (BDB) of the BIR.

NOTE – A **<BDB>** or **<bdbx>** element and a **<BIR>** element cannot coexist as children of the same **<BIR>** element (see B.11.1.2).

**B.11.2.7** The **<SB>** element (if present) carries the security block (SB) of the BIR.

NOTE – A **<SB>** element can coexist with either a **<BIR>** element or a **<BDB>** or **<bdbx>** element that is a child of the same **<BIR>** element.

**B.11.2.8** The **<BIRInfo>** element carries information about both the BIR and (possibly) about its descendant BIRs (if the **<BIR>** element has one or more child **<BIR>** elements), as specified in B.14.2.1.

**B.11.2.9** The **<BDBInfo>** element (if present) carries information about either the BDB of the BIR (if the **<BIR>** element has a child **<BDB>** or **<bdbx>** element) or about the BDBs of the descendant BIRs that have a child **<BDB>** or **<bdbx>** element (if the **<BIR>** element has one or more child **<BIR>** elements), as specified in B.15.2.1.

**B.11.2.10** The **<SBInfo>** element (if present) carries information about either the SB of the BIR (if the **<BIR>** element has a child **<SB>** element) or about the SBs of the descendant BIRs that have a child **<SB>** element (if the **<BIR>** element has one or more child **<BIR>** elements but no child **<SB>** element), as specified in B.16.2.1.

## **B.12Element <Version>**

### **B.12.1 Syntax**

This element shall have contents consisting of the following (in order):

a) a required **<Major>** element – the value of this element shall be a valid representation of a non-negative integer.

b) a required **<Minor>** element – the value of this element shall be a valid representation of a non-negative integer.

### **B.12.2 Semantics**

**B.12.2.1** This element represents the data element CBEFF\_patron\_header\_version, and carries the (major and minor) version number of the patron format. The number assigned to this version of the patron format is major 1, minor 0.

**B.12.2.2** The **<Major>** element represents the major version number (1 in this version).

**B.12.2.3** The **<Minor>** element represents the minor version number (0 in this version).

**B.12.2.4** If this element is not present, the values **Major="1"** **Minor="0"** are implied.

**B.12.2.5** A child **<BIR>** element shall have the same (major and minor) version number as its parent **<BIR>** element.

NOTE – This implies that the **<Version>** element, if present in a child **<BIR>** element, has to carry the same values as the **<Version>** element in the parent **<BIR>** element. This is equivalent to omitting the **<Version>** element. Therefore, this element is normally omitted in child **<BIR>** elements.

## **B.13 Element <CBEFFVersion>**

### **B.13.1 Syntax**

This element shall have content consisting of the following (in order):

- a) a required <Major> element – the value of this element shall be a valid representation of a non-negative integer (see B.17);
- b) a required <Minor> element – the value of this element shall be a valid representation of a non-negative integer.

### **B.13.2 Semantics**

**B.13.2.1** This element represents the data element CBEFF version, and carries the version number of the CBEFF standard supported by this patron format. The number assigned to the version of CBEFF supported by this patron format is Major=2, Minor=0.

**B.13.2.2** The <Major> element represents the major version number (2 in this version).

**B.13.2.3** The <Minor> element represents the minor version number (0 in this version).

**B.13.2.4** If this element is not present, the values Major="2" Minor="0" are implied.

**B.13.2.5** A child <BIR> element shall have the same CBEFF version number (major and minor) as its parent <BIR> element.

NOTE – Thus, the <CBEFFVersion> element is normally omitted from all child <BIR> elements, as it would be redundant.

## **B.14 Element <BIRInfo>**

### **B.14.1 Syntax**

**B.14.1.1** This element shall have a content consisting of the following (in order):

- a) an optional <Creator> element – the content of this element shall be a string of ISO/IEC 10646 characters;
- b) an optional <Index> element – the content of this element shall be a valid representation of a universally unique identifier (see B.20), and shall not inherit its value from any other level BIR;
- c) an optional <Payload> element – the content of this element shall be a valid representation of an octet string, and shall not inherit its value from any other level BIR.
- d) a required <Integrity> element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.1;
- e) an optional <CreationDate> element – the value of this element shall be a valid representation of a date and time of the day (see B.19);
- f) an optional <NotValidBefore> element – the value of this element shall be a valid representation of a date and time of the day;
- g) an optional <NotValidAfter> element – the value of this element shall be a valid representation of a date and time of the day.

## B.14.2 Semantics

**B.14.2.1** The <BIRInfo> element carries information about the BIR. In addition, if the BIR has one or more child BIRs (the <BIR> element has one or more child <BIR> elements), the information carried by the attributes and child elements of the <BIRInfo> element is inherited by those child BIRs except where overridden by a corresponding attribute or child element of the <BIRInfo> element of a child BIR. The information inherited by a BIR applies to that BIR, and (if the BIR has itself child BIRs) is further inherited by its child BIRs in the same way (and so on recursively).

**NOTE** – Since the **Integrity** element is required and the <BIRInfo> element is mandatory in all <BIR> elements, inheritance of the **Integrity** element can never occur.

**B.14.2.2** The **Integrity** element indicates whether integrity information about this BIR is provided within the security block (SB) of the BIR (the child <SB> element of the parent <BIR> element of this <BIRInfo> element).

**NOTE** – This information may consist of a digital signature or MAC, a reference to a key or certificate, an encrypted key (with or without a reference to the key used to encrypt that key), or other parameters of the digital signing (or MAC) process.

**B.14.2.3** If the value of the <Integrity> element is "true", then the parent <BIR> element of this <BIRInfo> element shall have a child <SB> element.

**B.14.2.4** Table B.1 specifies the correspondence between the attributes and child elements of this element and CBEFF data elements, and specifies the supported abstract values and their encodings (see also B.10.7).

**NOTE** - This element represents all CBEFF data elements whose name begins with "CBEFF\_BIR\_".

**I) Table B.1 – BIR information**

<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
<u>CBEFF_BIR_creator</u>	<u>&lt;Creator&gt;</u>	All ISO/IEC 10646 character strings are supported. The character string shall be encoded as the string itself.	
<u>CBEFF_BIR_index</u>	<u>&lt;Index&gt;</u>	All well-formed UUIDs are supported. The UUIDs shall be encoded as specified in B.20. Shall not inherit its value from any other BIR level.	
<u>CBEFF_BIR_payload</u>	<u>&lt;Payload&gt;</u>	All octet strings are supported. The octet strings shall be encoded as specified in B.18. Shall not inherit its value from any other BIR level.	
<u>CBEFF_BIR_integrity_options</u>	<u>&lt;Integrity&gt;</u>	The following abstract values are supported. The abstract values shall be encoded as shown below. <b>NO INTEGRITY:</b>	

<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
		<u>"false"</u> <u>INTEGRITY:</u> <u>"true"</u>	
<u>CBEFF_BIR_creation_date</u>	<u>&lt;CreationDate&gt;</u>	<u>All date and time-of-the-day abstract values permitted by CBEFF are supported.</u> <u>The abstract values shall be encoded as specified in B.19.</u>	
<u>CBEFF_BIR_validity_period (lower end)</u>	<u>&lt;NotValidBefore&gt;</u>	<u>All date and time-of-the-day abstract values permitted by CBEFF are supported.</u> <u>The abstract values shall be encoded as specified in B.19.</u>	
<u>CBEFF_BIR_validity_period (upper end)</u>	<u>&lt;NotValidAfter&gt;</u>	<u>All date and time-of-the-day abstract values permitted by CBEFF are supported.</u> <u>The abstract values shall be encoded as specified in B.19.</u>	

## **B.15 Element <BDBInfo>**

### **B.15.1 Syntax**

**B.15.1.1** This element shall have a content consisting of the following (in order):

- a) an optional <ChallengeResponse> element – the content of this element shall be a valid representation of an octet string (see B.18);
- b) an optional <Index> element – the content of this element shall be a valid representation of a universally unique identifier (see B.20).
- c) an optional <FormatOwner> element – the value of this element shall be a valid representation of an integer in the range 1 to 65535 (see B.17);
- d) an optional <FormatType> element – the value of this element shall be a valid representation of an integer in the range 1 to 65535;
- e) an optional <Encryption> element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.2;
- f) an optional <CreationDate> element – the value of this element shall be a valid representation of a date and time of the day (see B.19);
- g) an optional <NotValidBefore> element – the value of this element shall be a valid representation of a date and time of the day;
- h) an optional <NotValidAfter> element – the value of this element shall be a valid representation of a date and time of the day;

- i) an optional **<Type>** element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.2;
- j) an optional **<Subtype>** element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.2;
- k) an optional **<Level>** element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.2;
- l) an optional **<ProductOwner>** element – the value of this element shall be a valid representation of an integer in the range 1..65535 (see B.17);
- m) an optional **<ProductType>** element – the value of this element shall be a valid representation of an integer in the range 1..65535;
- n) an optional **<CaptureDeviceOwner>** element – the value of this element shall be a valid representation of an integer in the range 1..65535 (see B.17);
- o) an optional **<CaptureDeviceType>** element – the value of this element shall be a valid representation of an integer in the range 1..65535;
- p) an optional **<FeatureExtractionAlgorithmOwner>** element – the value of this element shall be a valid representation of an integer in the range 1..65535 (see B.17);
- q) an optional **<FeatureExtractionAlgorithmType>** element – the value of this element shall be a valid representation of an integer in the range 1..65535;
- r) an optional **<ComparisonAlgorithmOwner>** element – the value of this element shall be a valid representation of an integer in the range 1..65535 (see B.17);
- s) an optional **<ComparisonAlgorithmType>** element – the value of this element shall be a valid representation of an integer in the range 1..65535;
- t) an optional **<QualityAlgorithmOwner>** element – the value of this element shall be a valid representation of an integer in the range 1..65535 (see B.17);
- u) an optional **<QualityAlgorithmType>** element – the value of this element shall be a valid representation of an integer in the range 1..65535;
- v) an optional **<CompressionAlgorithmOwner>** element – the value of this element shall be a valid representation of an integer in the range 1..65535 (see B.17);
- w) an optional **<CompressionAlgorithmType>** element – the value of this element shall be a valid representation of an integer in the range 1..65535;
- x) an optional **<Purpose>** element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.2;
- y) an optional **<Quality>** element – the value of this element shall be a valid representation of an integer in the range –2..100 (see B.17), as specified in the third cell of the corresponding row of Table B.2.

**B.15.1.3** If the parent **<BIR>** element has a child **<BDB>** element, then the **<Encryption>** element shall be present in this **<BDBInfo>** element unless it is present in the child **<BDBInfo>** element of an ancestor **<BIR>** element (see also B.11.1.4).

**B.15.1.4** If the parent <BIR> element has a child <BDB> element, then the <FormatOwner> element shall be present in this <BDBInfo> element unless it is present in the child <BDBInfo> element of an ancestor <BIR> element (see also B.11.1.4).

**B.15.1.5** If the parent <BIR> element has a child <BDB> element, then the <FormatType> element shall be present in this <BDBInfo> element unless it is present in the child <BDBInfo> element of an ancestor <BIR> element (see also B.11.1.4).

**NOTE** – The ancestor <BIR> elements mentioned in the last three subclauses above need not be the same.

## **B.15.2 Semantics**

**B.15.2.1** If the BIR has a BDB (the <BIR> element has a child <BDB> element), then the <BDBInfo> element carries information about that BDB. Otherwise, the information carried by the attributes and child elements of the <BDBInfo> element is inherited by all the BIRs that are children of the BIR except where overridden by a corresponding attribute or child element of the <BDBInfo> element of a child BIR. The information inherited by a BIR with a BDB applies to that BDB, whereas the information inherited by a BIR that has itself child BIRs is further inherited by all the BIRs that are children of the BIR in the same way (and so on recursively).

**B.15.2.2** If the BIR has a BDB and encryption is applied to that BDB (either by including the **encryption** attribute with the value "true" in the <BDBInfo> element or by having the BIR inherit that attribute value from its parent BIR), then the BDB in the <BDB> element shall be encrypted.

**B.15.2.3** If the BDB of a BIR is encrypted, information about the encryption process may be provided within the security block (SB) of that BIR (the child <SB> element of the parent <BIR> element of this <BIRInfo> element).

**NOTE** – This information may consist of a reference to an encryption key, an encrypted key (with or without a reference to the key used to encrypt that key), or other parameters of the encryption process.

**B.15.2.4** Table B.2 specifies the correspondence between the attributes and child elements of this element and CBEFF data elements, and specifies the supported abstract values and their encodings (see also F.10.7).

**NOTE** – This element represents all CBEFF data elements whose name begins with "CBEFF\_BDB\_".

**II) Table B.2 – BDB information**

<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
<u>CBEFF_BDB_format_owner</u>	<u>&lt;FormatOwner&gt;</u>	All integers in the range 1 to 65535 are supported. The integers shall be encoded as specified in B.17.	
<u>CBEFF_BDB_format_type</u>	<u>&lt;FormatType&gt;</u>	All integers in the range 1 to 65535 are supported. The integers shall be encoded as specified in B.17.	
<u>CBEFF_BDB_encryption_options</u>	<u>&lt;Encryption&gt;</u>	The following abstract values are supported. The abstract values shall be encoded as shown below. <b>NO ENCRYPTION:</b> <b>"false"</b>	

<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
		<u>ENCRYPTION:</u> <u>"true"</u>	
<u>CBEFF_BDB_creation_date</u>	<u>&lt;CreationDate&gt;</u>	<u>All date and time-of-the-day abstract values permitted by CBEFF are supported.</u> <u>The abstract values shall be encoded as specified in B.19.</u>	
<u>CBEFF_BDB_validity_period (lower end)</u>	<u>&lt;NotValidBefore&gt;</u>	<u>All date and time-of-the-day abstract values permitted by CBEFF are supported.</u> <u>The abstract values shall be encoded as specified in B.19.</u>	
<u>CBEFF_BDB_challenge_response</u>	<u>&lt;ChallengeResponse&gt;</u>	<u>All octet strings are supported.</u> <u>The octet strings shall be encoded as specified in B.18.</u> <u>Shall appear only in BIRs that have a BDB.</u>	
<u>CBEFF_BDB_index</u>	<u>&lt;Index&gt;</u>	<u>All well-formed UUIDs are supported.</u> <u>The UUIDs shall be encoded as specified in B.20</u> <u>Shall appear only in BIRs that have a BDB.</u>	
<u>CBEFF_BDB_validity_period (upper end)</u>	<u>&lt;NotValidAfter&gt;</u>	<u>All date and time-of-the-day abstract values permitted by CBEFF are supported.</u> <u>The abstract values shall be encoded as specified in B.19.</u>	
<u>CBEFF_BDB_biometric_type</u>	<u>&lt;Type&gt;</u>	<u>The following abstract values and all their unordered combinations are supported.</u> <u>A single abstract value shall be encoded as the corresponding string shown below. A combination of two or more abstract values shall be encoded as the concatenation of the corresponding strings, using a single space as separator.</u> <u>SCENT:</u> <u>"Scent"</u> <u>DNA:</u> <u>"DNA"</u> <u>EAR:</u> <u>"Ear"</u> <u>FACE:</u> <u>"Face"</u> <u>FINGER:</u> <u>"Finger"</u> <u>FOOT:</u> <u>"Foot"</u> <u>VEIN:</u>	

	<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
			<u>"Vein"</u> <u>HAND GEOMETRY:</u> <u>"HandGeometry"</u> <u>IRIS:</u> <u>"Iris"</u> <u>RETINA:</u> <u>"Retina"</u> <u>VOICE:</u> <u>"Voice"</u> <u>GAIT:</u> <u>"Gait"</u> <u>KEYSTROKE:</u> <u>"Keystroke"</u> <u>LIP MOVEMENT:</u> <u>"LipMovement"</u> <u>SIGNATURE OR SIGN:</u> <u>"SignatureSign"</u>	
	<u>CBEFF_BDB_biometric_subtype</u>	<u>&lt;Subtype&gt;</u>	<u>The following abstract values are supported.</u> <u>The abstract values shall be encoded as shown below. A combination of two or more abstract values shall be encoded as the concatenation of the corresponding strings, using a single space as separator.</u> <u>LEFT:</u> <u>"Left"</u> <u>RIGHT:</u> <u>"Right"</u> <u>THUMB:</u> <u>"Thumb"</u> <u>INDEX FINGER:</u> <u>"IndexFinger"</u> <u>MIDDLE FINGER:</u> <u>"MiddleFinger"</u> <u>RING FINGER:</u> <u>"RingFinger"</u> <u>LITTLE FINGER:</u> <u>"LittleFinger"</u>	
	<u>CBEFF_BDB_processed_level</u>	<u>&lt;Level&gt;</u>	<u>The following abstract values are supported.</u> <u>The abstract values shall be encoded as shown below.</u> <u>RAW:</u> <u>"Raw"</u> <u>INTERMEDIATE:</u> <u>"Intermediate"</u> <u>PROCESSED:</u> <u>"Processed"</u>	



<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
<u>CBEFF_BDB_product_owner</u>	<u>&lt;ProductOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_product_type</u>	<u>&lt;ProductType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_capture_device_owner</u>	<u>&lt;CaptureDeviceOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_capture_device_type</u>	<u>&lt;CaptureDeviceType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_feature_extraction_algorithm_owner</u>	<u>&lt;FeatureExtractionAlgorithmOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_feature_extraction_algorithm_type</u>	<u>&lt;FeatureExtractionAlgorithmType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_comparison_algorithm_owner</u>	<u>&lt;ComparisonAlgorithmOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_comparison_algorithm_type</u>	<u>&lt;ComparisonAlgorithmType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_quality_algorithm_owner</u>	<u>&lt;QualityAlgorithmOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_quality_algorithm_type</u>	<u>&lt;QualityAlgorithmType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_compression_algorithm_owner</u>	<u>&lt;CompressionAlgorithmOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_BDB_compression_algorithm_type</u>	<u>&lt;CompressionAlgorithmType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	

	<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
	<u>CBEFF_BDB_purpose</u>	<u>&lt;Purpose&gt;</u>	<u>The following abstract values are supported.</u> <u>The abstract values shall be encoded as shown below.</u> <u>VERIFY:</u> <u>"Verify"</u> <u>IDENTIFY:</u> <u>"Identify"</u> <u>ENROLL:</u> <u>"Enroll"</u> <u>ENROLL FOR VERIFICATION ONLY:</u> <u>"EnrollVerify"</u> <u>ENROLL FOR IDENTIFICATION ONLY:</u> <u>"EnrollIdentify"</u> <u>AUDIT:</u> <u>"Audit"</u>	
	<u>CBEFF_BDB_quality</u>	<u>&lt;Quality&gt;</u>	<u>The following abstract values are supported.</u> <u>The integers shall be encoded as specified in B.17. The other abstract values shall be encoded as shown below.</u> <u>INTEGER</u> <u>QUALITY NOT SUPPORTED BY BDB CREATOR:</u> <u>"-2"</u> <u>QUALITY SUPPORTED BY BDB CREATOR BUT NOT SET:</u> <u>"-1"</u>	

## **B.16 Element <SBInfo>**

### **B.16.1 Syntax**

**B.16.1.1** This element shall have content consisting of the following (in order):

a) an optional <FormatOwner> element – the value of this element shall be a valid representation of an integer in the range 1 to 65535 (see B.17);

b) an optional <FormatType> element – the value of this element shall be a valid representation of an integer in the range 1 to 65535

**B.16.1.2** If the parent <BIR> element has a child <SB> element, then the <FormatOwner> element shall be present in this <SBInfo> element unless it is present in the child <SBInfo> element of an ancestor <BIR> element (see also B.11.1.5).

**B.16.1.3** If the parent <BIR> element has a child <SB> element, then the <FormatType> element shall be present in this <SBInfo> element unless it is present in the child <SBInfo> element of an ancestor <BIR> element (see also B.11.1.5).

NOTE 1 – The ancestor <BIR> elements mentioned in the last two subclauses above need not be the same.

NOTE 2 – When the parent <BIR> element has a child <SB> element and one omits both children of the <SBInfo> element, the <SBInfo> element will have no attributes and an empty content. Omission of the <SBInfo> element is not allowed in this case (see B.11.1.5).

## **B.16.2 Semantics**

**B.16.2.1** If the BIR has an SB (the <BIR> element has a child <SB> element), then the <SBInfo> element carries information about that SB. In addition, if the BIR has one or more child BIRs (the <BIR> element has one or more child <BIR> elements), the information carried by the child element of the <SBInfo> element is inherited by those child BIRs except where overridden by a corresponding child element of the <SBInfo> element of a child BIR. The information inherited by a BIR with an SB applies to that SB, and (if the BIR has itself child BIRs) is further inherited by its child BIRs in the same way (and so on recursively).

**B.16.2.2** Table B.3 specifies the correspondence between the attributes and child elements of this element and CBEFF data elements, and specifies the supported abstract values and their encodings (see also B.10.7).

NOTE – This element represents all CBEFF data elements whose name begins with "CBEFF\_SB\_".

**///) Table B.3 – SB information**

<u>CBEFF data element name</u>	<u>XML element</u>	<u>Supported abstract values and encodings</u>	<u>Reference</u>
<u>CBEFF_SB_format_owner</u>	<u>&lt;FormatOwner&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	
<u>CBEFF_SB_format_type</u>	<u>&lt;FormatType&gt;</u>	<u>All integers in the range 1 to 65535 are supported.</u> <u>The integers shall be encoded as specified in B.17.</u>	

## **B.17 Representation of Integers**

**B.17.1** A non-negative integer shall be represented as a string of one or more ISO/IEC 10646 characters in the range DIGIT ZERO to DIGIT NINE ("0" to "9") in decimal notation.

**B.17.2** A negative integer shall be represented as the corresponding positive integer, preceded by a HYPHEN-MINUS character ("-").

**B.17.3** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the encoding.

## **B.18 Representation of Octet Strings**

**B.18.1** An octet string shall be represented as a string of the following ISO/IEC 10646 characters:

a) LATIN CAPITAL LETTER A to LATIN CAPITAL LETTER Z;

b) LATIN SMALL LETTER A to LATIN SMALL LETTER Z;

4745 c) DIGIT ZERO to DIGIT NINE:

4746 d) PLUS SIGN:

4747 e) SOLIDUS:

4748 f) EQUALS SIGN.

4749 forming the Base64 encoding of the octet string (see IETF RFC 2045), with all whitespace removed.

4750 **B.18.2** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the  
4751 encoding.

4752

## 4753 **B.19 Representation of Date and Time of the Day**

4754 **B.19.1** A date and time of the day shall be represented as a string of ISO/IEC 10646 characters in the  
4755 following format, which conforms to ISO 8601.

4756 **B.19.2** The encoding shall be the concatenation of all the following components (in order):

4757 a) the "year" component, consisting of the year encoded in four digits ("2000" to "2999");

4758 b) the hyphen character "-";

4759 c) the "month" component, consisting of the month encoded in two digits ("01" to "12");

4760 d) the hyphen character "-";

4761 e) the "day" component, consisting of the day encoded in two digits ("01" to "31");

4762 f) the letter "T";

4763 g) the "hour" component, consisting of the hour encoded in two digits ("00" to "23");

4764 h) the colon character ":";

4765 i) the "minute" component, consisting of the minute encoded in two digits ("00" to "59");

4766 j) the colon character ":";

4767 k) the "second" component, consisting of the second encoded in two digits ("00" to "59");

4768 l) the letter "Z".

4769 **B.19.3** The "year", "month", "day", "hour", "minute", and "second" components shall be present.

4770 **B.19.4** The letter "T" shall be present.

4771 **B.19.5** The letter "Z" shall be present whether or not the "hour" component is present.

4772 NOTE This letter indicates that the date and time of the day are UTC.

4773 **B.19.6** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the  
4774 encoding.

4775

## **B.20 Representation of Universally Unique Identifiers**

NOTE: The following subclauses describe the same representation of a UUID as is specified in ISO/IEC 9834-8, clause 8. An example of such a representation is: f81d4fae-7dec-11d0-a765-00a0c91e6bf6

**B.20.1** A universally unique identifier (UUID) shall be represented as a string of ISO/IEC 10646 characters. Each string shall contain exactly 36 characters from the union of the following sets:

a) DIGIT ZERO to DIGIT NINE ("0" to "9"), each representing a hexadecimal digit 0 through 9;

b) LATIN CAPITAL LETTER A to LATIN CAPITAL LETTER F ("A" to "F"), each representing a hexadecimal digit A through F;

c) LATIN SMALL LETTER A to LATIN SMALL LETTER F ("a" to "f"), each representing a hexadecimal digit A through F; and

d) HYPHEN-MINUS ("-").

**B.20.2** Each of the positions 9, 14, 19, and 24 of an encoding shall contain a character from set (d). The other 32 positions shall contain characters from sets (a) through (c).

**B.20.3** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the encoding.

## **B.21 Patron format conformance statement**

### **B.21.1 Identifying information**

<u>Required Information</u>	<u>Patron format reference</u>
<u>Patron name</u>	<u>See B.1</u>
<u>Patron identifier</u>	<u>See B.2</u>
<u>Patron format name</u>	<u>See B.3</u>
<u>Patron format identifier</u>	<u>See B.4</u>
<u>Patron format ASN.1 object identifier</u>	<u>See B.5</u>
<u>Domain of use description</u>	<u>See B.6</u>
<u>Patron format version</u>	<u>See B.7</u>
<u>CBEFF version</u>	<u>See B.8</u>

### **B.21.2 ISO/IEC 19785-1:2006/Amd 1:2010 to Patron Format Mapping**

<u>CBEFF data element name</u>	<u>Mandatory/ optional</u>	<u>Patron format field name</u>	<u>Abstract values specified?</u>	<u>Encodings specified?</u>
<u>CBEFF BDB format owner</u>	<u>Mandatory (specified or inherited) if a BDB is present</u>	<u>&lt;FormatOwner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
<u>CBEFF BDB format type</u>	<u>Mandatory (specified or inherited)</u>	<u>&lt;FormatType&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>

	<u>CBEFF data element name</u>	<u>Mandatory/ optional</u>	<u>Patron format field name</u>	<u>Abstract values specified?</u>	<u>Encodings specified?</u>
		<u>if a BDB is present</u>			
	<u>CBEFF_BDB_encryption_options</u>	<u>Mandatory (specified or inherited) if a BDB is present</u>	<u>&lt;Encryption&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BIR_integrity_options</u>	<u>Mandatory</u>	<u>&lt;Integrity&gt; child of &lt;BIRInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_subheader_count</u>	<u>Mandatory</u>	<u>implied in the number of occurrences of the child &lt;BIR&gt; element</u>	<u>No (implied)</u>	<u>No (implied)</u>
	<u>CBEFF_BDB_biometric_type</u>	<u>Optional</u>	<u>&lt;Type&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_biometric_subtype</u>	<u>Optional</u>	<u>&lt;Subtype&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_challenge_response</u>	<u>Optional</u>	<u>&lt;ChallengeResponse&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_creation_date</u>	<u>Optional</u>	<u>&lt;CreationDate&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_index</u>	<u>Optional</u>	<u>&lt;Index&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_product_owner</u>	<u>Optional</u>	<u>&lt;ProductOwner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_product_type</u>	<u>Optional</u>	<u>&lt;ProductType&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_capture_device_owner</u>	<u>Optional</u>	<u>&lt;CaptureDeviceOwner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_capture_device_type</u>	<u>Optional</u>	<u>&lt;CaptureDeviceType&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_feature_extraction_algorithm_owner</u>	<u>Optional</u>	<u>&lt;FeatureExtractionAlgorithm Owner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_feature_extraction_algorithm_type</u>	<u>Optional</u>	<u>&lt;FeatureExtractionAlgorithm Type&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_comparison_algorithm_owner</u>	<u>Optional</u>	<u>&lt;ComparisonAlgorithmOwner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_comparison_algorithm_type</u>	<u>Optional</u>	<u>&lt;ComparisonAlgorithmType&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_quality_algorithm_owner</u>	<u>Optional</u>	<u>&lt;QualityAlgorithmOwner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_quality_algorithm_type</u>	<u>Optional</u>	<u>&lt;QualityAlgorithmType&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_compression_algorithm_owner</u>	<u>Optional</u>	<u>&lt;CompressionAlgorithmOwner&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_compression_algorithm_type</u>	<u>Optional</u>	<u>&lt;CompressionAlgorithmType&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_processed_level</u>	<u>Optional</u>	<u>&lt;Level&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_purpose</u>	<u>Optional</u>	<u>&lt;Purpose&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>

	<u>CBEFF data element name</u>	<u>Mandatory/ optional</u>	<u>Patron format field name</u>	<u>Abstract values specified?</u>	<u>Encodings specified?</u>
	<u>CBEFF_BDB_quality</u>	<u>Optional</u>	<u>&lt;Quality&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BDB_validity_period</u>	<u>Optional</u>	<u>&lt;NotValidBefore&gt; and &lt;NotValidAfter&gt; children of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BIR_creation_date</u>	<u>Optional</u>	<u>&lt;CreationDate&gt; child of &lt;BIRInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BIR_creator</u>	<u>Optional</u>	<u>&lt;Creator&gt; child of &lt;BIRInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BIR_index</u>	<u>Optional</u>	<u>&lt;Index&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BIR_patron_format_owner</u>	<u>N/A</u>		<u>No</u>	<u>No</u>
	<u>CBEFF_BIR_patron_format_type</u>	<u>N/A</u>		<u>No</u>	<u>No</u>
	<u>CBEFF_BIR_payload</u>	<u>Optional</u>	<u>&lt;Payload&gt; child of &lt;BDBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_SB_format_owner</u>	<u>Optional</u>	<u>&lt;FormatOwner&gt; child of &lt;SBInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_SB_format_type</u>	<u>Optional</u>	<u>&lt;FormatType&gt; child of &lt;sb- info&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_BIR_validity_period</u>	<u>Optional</u>	<u>&lt;NotValidBefore&gt; and &lt;NotValidAfter&gt; attributes of &lt;BIRInfo&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>patron_header_version</u>	<u>Optional</u>	<u>&lt;Major&gt; and &lt;Minor&gt; children of &lt;Version&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>CBEFF_version</u>	<u>Optional</u>	<u>&lt;Major&gt; and &lt;Minor&gt; children of &lt;CBEFFVersion&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>BDB</u>	<u>Optional</u>	<u>&lt;BDB&gt;</u>	<u>Yes</u>	<u>Yes</u>
	<u>SB</u>	<u>Optional</u>	<u>&lt;SB&gt;</u>	<u>Yes</u>	<u>Yes</u>

4797

4798

## B.22 XML schema of the BIAS patron format

4799

4800

4801

4802

4803

4804

4805

4806

4807

4808

4809

4810

4811

4812

4813

4814

4815

4816

4817

4818

4819

4820

4821

4822

```

<?xml version='1.0' encoding="utf-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/"
  targetNamespace="http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/"
  elementFormDefault="qualified">
  <xs:element name="BIR" type="BIRType"/>
  <xs:complexType name="BIRType">
    <xs:sequence>
      <xs:element name="Version" type="VersionType" minOccurs="0"/>
      <xs:element name="CBEFFVersion" type="VersionType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="BIRInfo" type="BIRInfoType"/>
      <xs:element name="BDBInfo" type="BDBInfoType" minOccurs="0"/>
      <xs:element name="SBInfo" type="SBInfoType" minOccurs="0"/>
      <xs:element name="BIR" type="BIRType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="BDB" type="xs:base64Binary" minOccurs="0"/>
      <xs:element name="SB" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="VersionType">

```

```

4823 <xs:sequence>
4824 <xs:element name="Major" type="xs:nonNegativeInteger" />
4825 <xs:element name="Minor" type="xs:nonNegativeInteger" />
4826 </xs:sequence>
4827 </xs:complexType>
4828
4829 <xs:complexType name="BIRInfoType">
4830 <xs:sequence>
4831 <xs:element name="Creator" type="xs:string" minOccurs="0"/>
4832 <xs:element name="Index" type="xs:UIDType" minOccurs="0"/>
4833 <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
4834 <xs:element name="Integrity" type="xs:boolean" />
4835 <xs:element name="CreationDate" type="xs:dateTime" minOccurs="0" />
4836 <xs:element name="NotValidBefore" type="xs:dateTime" minOccurs="0" />
4837 <xs:element name="NotValidAfter" type="xs:dateTime" minOccurs="0" />
4838 </xs:sequence>
4839 </xs:complexType>
4840
4841 <xs:complexType name="BDBInfoType">
4842 <xs:sequence>
4843 <xs:element name="ChallengeResponse" type="xs:base64Binary" minOccurs="0"/>
4844 <xs:element name="Index" type="xs:UIDType" minOccurs="0"/>
4845 <xs:element name="FormatOwner" type="xs:positiveInteger" minOccurs="0" />
4846 <xs:element name="FormatType" type="xs:positiveInteger" minOccurs="0" />
4847 <xs:element name="Encryption" type="xs:boolean" minOccurs="0" />
4848 <xs:element name="CreationDate" type="xs:dateTime" minOccurs="0" />
4849 <xs:element name="NotValidBefore" type="xs:dateTime" minOccurs="0" />
4850 <xs:element name="NotValidAfter" type="xs:dateTime" minOccurs="0" />
4851 <xs:element name="Type" type="MultipleTypesType" minOccurs="0" />
4852 <xs:element name="Subtype" type="SubtypeType" minOccurs="0" />
4853 <xs:element name="Level" type="ProcessedLevelType" minOccurs="0" />
4854 <xs:element name="ProductOwner" type="xs:positiveInteger" minOccurs="0" />
4855 <xs:element name="ProductType" type="xs:positiveInteger" minOccurs="0" />
4856 <xs:element name="CaptureDeviceOwner" type="xs:positiveInteger" minOccurs="0" />
4857 <xs:element name="CaptureDeviceType" type="xs:positiveInteger" minOccurs="0" />
4858 <xs:element name="FeatureAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4859 <xs:element name="FeatureAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4860 <xs:element name="ComparisonAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4861 <xs:element name="ComparisonAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4862 <xs:element name="QualityAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4863 <xs:element name="QualityAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4864 <xs:element name="CompressionAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4865 <xs:element name="CompressionAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4866 <xs:element name="Purpose" type="PurposeType" minOccurs="0" />
4867 <xs:element name="Quality" type="QualityType" minOccurs="0" />
4868 </xs:sequence>
4869 </xs:complexType>
4870
4871 <xs:complexType name="SBInfoType">
4872 <xs:sequence>
4873 <xs:element name="FormatOwner" type="xs:positiveInteger" minOccurs="0" />
4874 <xs:element name="FormatType" type="xs:positiveInteger" minOccurs="0" />
4875 </xs:sequence>
4876 </xs:complexType>
4877
4878 <xs:simpleType name="QualityType">
4879 <xs:restriction base="xs:integer">
4880 <xs:minInclusive value="-2"/>
4881 <xs:maxInclusive value="100"/>
4882 </xs:restriction>
4883 </xs:simpleType>
4884
4885 <xs:simpleType name="SingleTypeType">
4886 <xs:restriction base="xs:string">
4887 <xs:enumeration value="Scent"/>
4888 <xs:enumeration value="DNA"/>
4889 <xs:enumeration value="Ear" />
4890 <xs:enumeration value="Face"/>
4891 <xs:enumeration value="Finger"/>
4892 <xs:enumeration value="Foot"/>
4893 <xs:enumeration value="HandGeometry"/>

```



```

4894     <xs:enumeration value="Vein"/>
4895     <xs:enumeration value="Iris"/>
4896     <xs:enumeration value="Retina"/>
4897     <xs:enumeration value="Voice"/>
4898     <xs:enumeration value="Gait"/>
4899     <xs:enumeration value="Keystroke"/>
4900     <xs:enumeration value="LipMovement"/>
4901     <xs:enumeration value="SignatureSign"/>
4902 </xs:restriction>
4903 </xs:simpleType>
4904
4905     <xs:simpleType name="MultipleTypesType">
4906     <xs:list itemType="SingleTypeType"/>
4907 </xs:simpleType>
4908
4909     <xs:simpleType name="SingleAnySubtypeType">
4910     <xs:restriction base="xs:string">
4911     <xs:enumeration value="Left"/>
4912     <xs:enumeration value="Right"/>
4913     <xs:enumeration value="Thumb"/>
4914     <xs:enumeration value="IndexFinger"/>
4915     <xs:enumeration value="MiddleFinger"/>
4916     <xs:enumeration value="RingFinger"/>
4917     <xs:enumeration value="LittleFinger"/>
4918 </xs:restriction>
4919 </xs:simpleType>
4920
4921     <xs:simpleType name="SingleVeinOnlySubtypeType">
4922     <xs:restriction base="xs:string">
4923     <xs:enumeration value="LeftVein"/>
4924     <xs:enumeration value="RightVein"/>
4925     <xs:enumeration value="Palm"/>
4926     <xs:enumeration value="BackOfHand"/>
4927     <xs:enumeration value="Wrist"/>
4928     <xs:enumeration value="Reserved1"/>
4929     <xs:enumeration value="Reserved2"/>
4930 </xs:restriction>
4931 </xs:simpleType>
4932
4933     <xs:simpleType name="MultipleAnySubtypesType">
4934     <xs:list itemType="SingleAnySubtypeType"/>
4935 </xs:simpleType>
4936
4937     <xs:simpleType name="MultipleVeinOnlySubtypesType">
4938     <xs:list itemType="SingleVeinOnlySubtypeType"/>
4939 </xs:simpleType>
4940
4941     <xs:simpleType name="SubtypeType">
4942     <xs:union memberTypes="MultipleAnySubtypesType MultipleVeinOnlySubtypesType"/>
4943 </xs:simpleType>
4944
4945     <xs:simpleType name="ProcessedLevelType">
4946     <xs:restriction base="xs:string">
4947     <xs:enumeration value="Raw"/>
4948     <xs:enumeration value="Intermediate"/>
4949     <xs:enumeration value="Processed"/>
4950 </xs:restriction>
4951 </xs:simpleType>
4952
4953     <xs:simpleType name="PurposeType">
4954     <xs:restriction base="xs:string">
4955     <xs:enumeration value="Verify"/>
4956     <xs:enumeration value="Identify"/>
4957     <xs:enumeration value="Enroll"/>
4958     <xs:enumeration value="EnrollVerify"/>
4959     <xs:enumeration value="EnrollIdentify"/>
4960     <xs:enumeration value="Audit"/>
4961 </xs:restriction>
4962 </xs:simpleType>
4963
4964     <xs:simpleType name="UUIDType">

```

```

4965     <xs:restriction base="xs:string">
4966     <xs:pattern value="[a-fA-F0-9]{8}\-[a-fA-F0-9]{4}\-[a-fA-F0-9]{3}[a-fA-F0-9]{12}"/>
4967     </xs:restriction>
4968     </xs:simpleType>
4969   </xs:schema>
4970

```

NOTE NO VALUE AVAILABLE is encoded by the absence of optional fields in the XML encoding. There is little value in, for example, having the following string appear in a record: <level> no value available <level>.

## B.23 Sample BIR encoding

An example of a simple BIR in XML encoding (complying with the XSD schema and the normative textual description) follows.

```

4976   <?xml version="1.0" encoding="utf-8"?>
4977   <BIR xmlns="http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/">
4978     <Version>
4979       <Major>1</Major>
4980       <Minor>0</Minor>
4981     </Version>
4982     <CBEFFVersion>
4983       <Major>2</Major>
4984       <Minor>0</Minor>
4985     </CBEFFVersion>
4986     <BIRInfo>
4987       <Creator>ABCDE</Creator>
4988       <Index>86CA3100-43F3-0D23-A941-7871E519A00E</Index>
4989       <Payload>a2V2aW4ubWFuZ29sZEBuaXN0Lmdvdq==</Payload>
4990       <Integrity>true</Integrity>
4991       <CreationDate>2004-03-02T15:03:15Z</CreationDate>
4992       <NotValidBefore>2004-03-02T15:00:00Z</NotValidBefore>
4993       <NotValidAfter>2004-03-03T15:00:00Z</NotValidAfter>
4994     </BIRInfo>
4995     <BDBInfo>
4996       <ChallengeResponse>VmlzaXQgaHR0cDovL2J3cy5uaXN0LmdvdiBmb3lqc29tZSBhd2Vzb21lIGJpb21ldH
5000       JpY3Mvd2ViIHNIcnZpY2UgcHJvamVjdHMh</ChallengeResponse>
5001       <Index>86CA3100-43F3-0D23-A941-7871E519A00E</Index>
5002       <FormatOwner>51</FormatOwner>
5003       <FormatType>88</FormatType>
5004       <Encryption>true</Encryption>
5005       <CreationDate>2004-03-02T15:00:00Z</CreationDate>
5006       <NotValidBefore>2004-03-02T15:00:00Z</NotValidBefore>
5007       <NotValidAfter>2004-03-02T15:00:00Z</NotValidAfter>
5008       <Type>Iris</Type>
5009       <Subtype>Left</Subtype>
5010       <Level>Processed</Level>
5011       <ProductOwner>16</ProductOwner>
5012       <ProductType>2</ProductType>
5013       <Purpose>Verify</Purpose>
5014       <Quality>100</Quality>
5015

```

5016	<u>&lt;/BDBInfo&gt;</u>
5017	<u>&lt;SBInfo&gt;</u>
5018	<u>&lt;FormatOwner&gt;51&lt;/FormatOwner&gt;</u>
5019	<u>&lt;FormatType&gt;99&lt;/FormatType&gt;</u>
5020	<u>&lt;/SBInfo&gt;</u>
5021	<u>&lt;BDB&gt;a2V2aW4ubWFuZ29sZEBuaXN0Lmdvdg==&lt;/BDB&gt;</u>
5022	<u>&lt;SB&gt;TmF0aW9uYWwgSW5zdGI0dXRlIG9mlFN0YW5kYXJkcyBhbmQqVGVjaG5vbG9neQ==&lt;/SB&gt;</u>
5023	<u>&lt;/BIR&gt;</u>

## Annex C. Use Cases (non-normative)

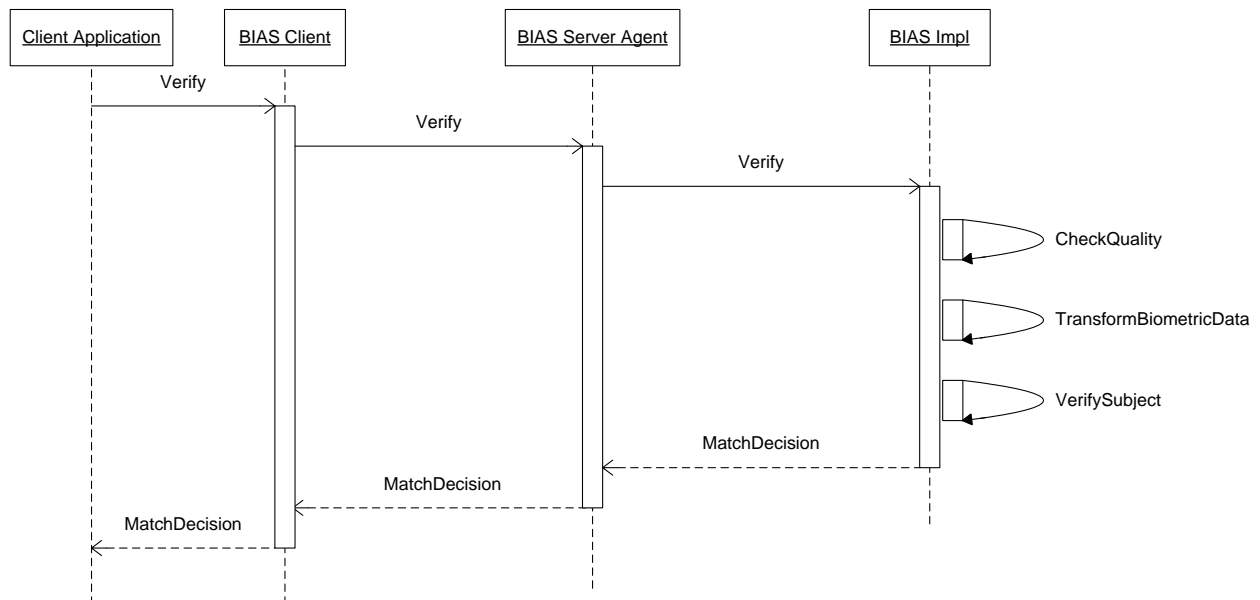
The intent of this annex is to provide operational sequence diagrams / flow charts that show how the higher level usage scenarios within [INCITS-BIAS] could be implemented using the BIAS SOAP profile.

The following use cases are given:

- Verification (synchronous/aggregate)
- Verification (asynchronous/aggregate)
- Verification (primitive)
- Identification (primitive)
- Enrollment (aggregate)
- Enrollment (primitive)

### BC.1 Verification Use Case

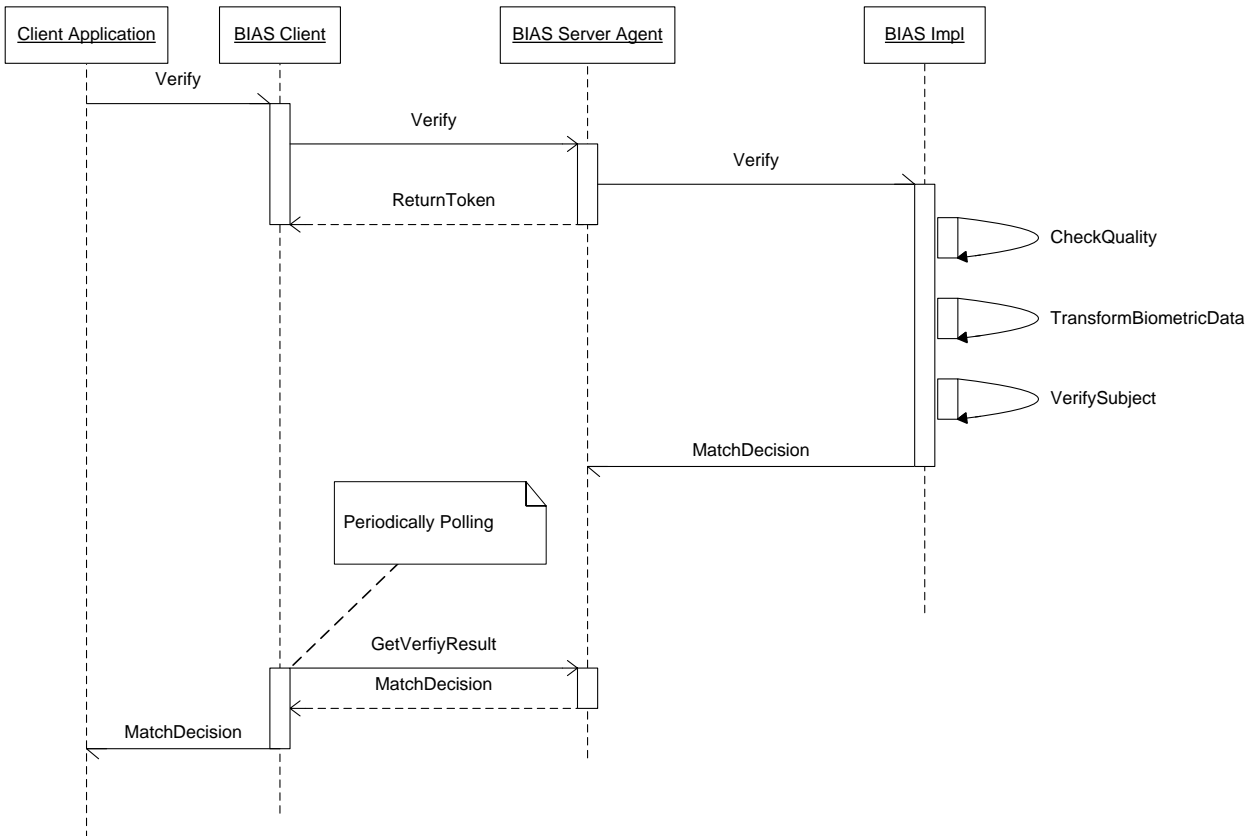
This use case uses the aggregate Verify operation in which a single request results in some set of operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service provider.



Note that  
1. CheckQuality, TransformBiometricData, VerifySubject can be exposed as interfaces of BIAS server agent.

## BC.2 Asynchronous Verification Use Case

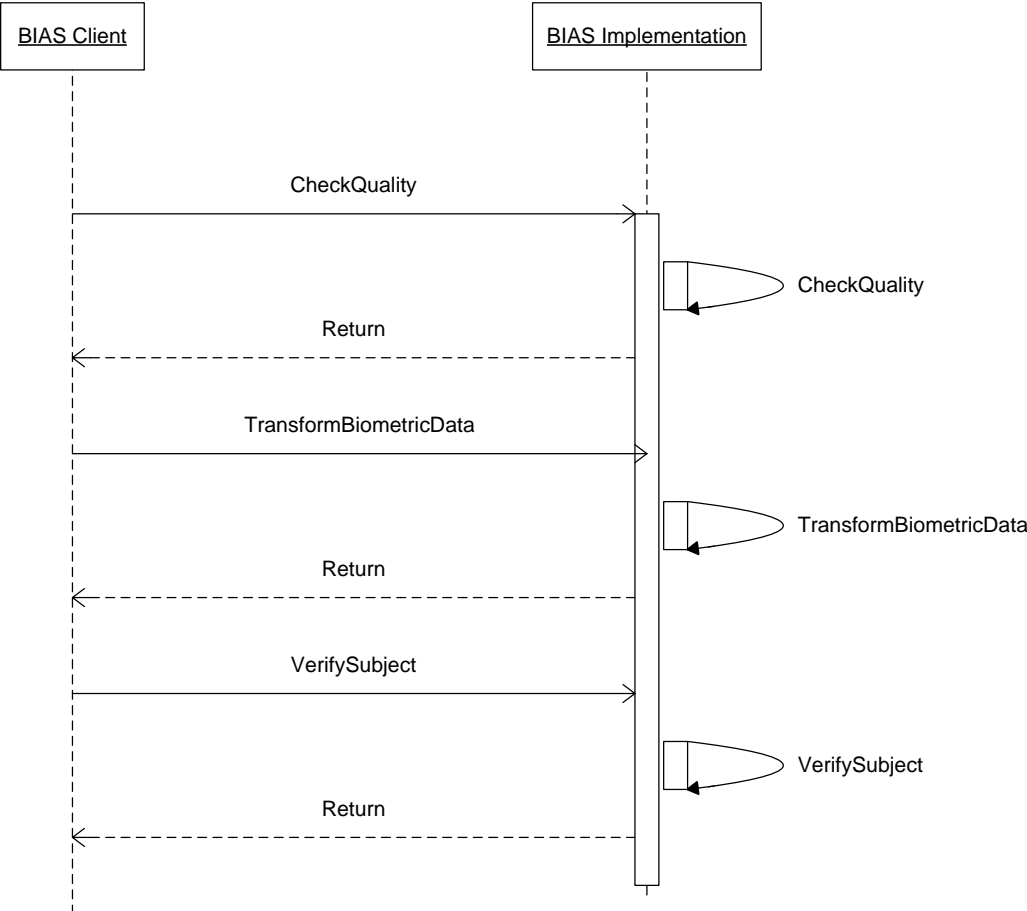
In this use case, the requester issues two requests – the BIAS Verify request to initiate the operation followed by a BIAS GetVerifyResult request to retrieve the results of that operation.



Note that  
1. CheckQuality, TransformBiometricData, VerifySubject can be exposed as interfaces of BIAS server agent.

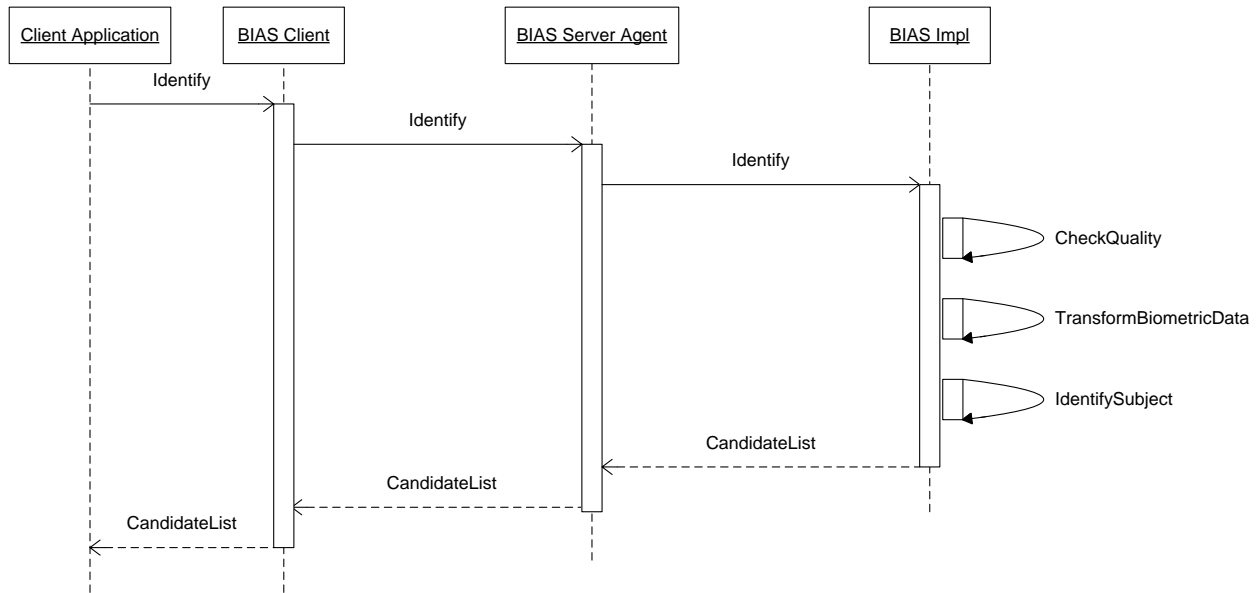
**BC.3 Primitive Verification Use Case**

In this use case, the verification operation is performed as a series of requests using the BIAS primitive operations. In this case, the client rather than the service provider controls the workflow of the higher level operation.



## BC.4 Identification Use Case

This use case uses the aggregate Identify operation in which a single request results in some set of operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service provider.

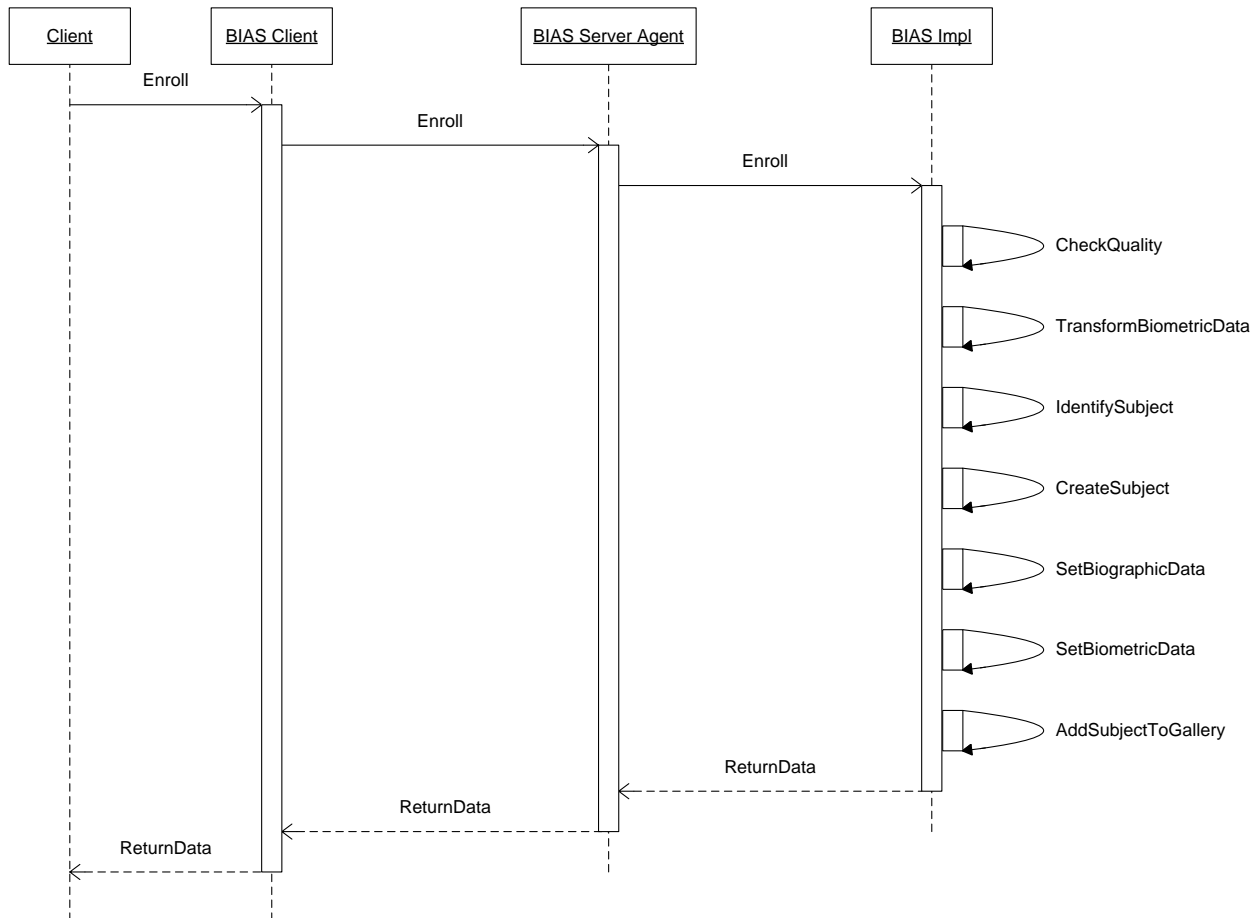


Note that  
1. CheckQuality, TransformBiometricData, IdentifySubject can be exposed as interfaces of BIAS server agent.

## BC.5 Biometric Enrollment Use Case

This use case uses the aggregate Enroll operation in which a single request results in some set of operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service provider.

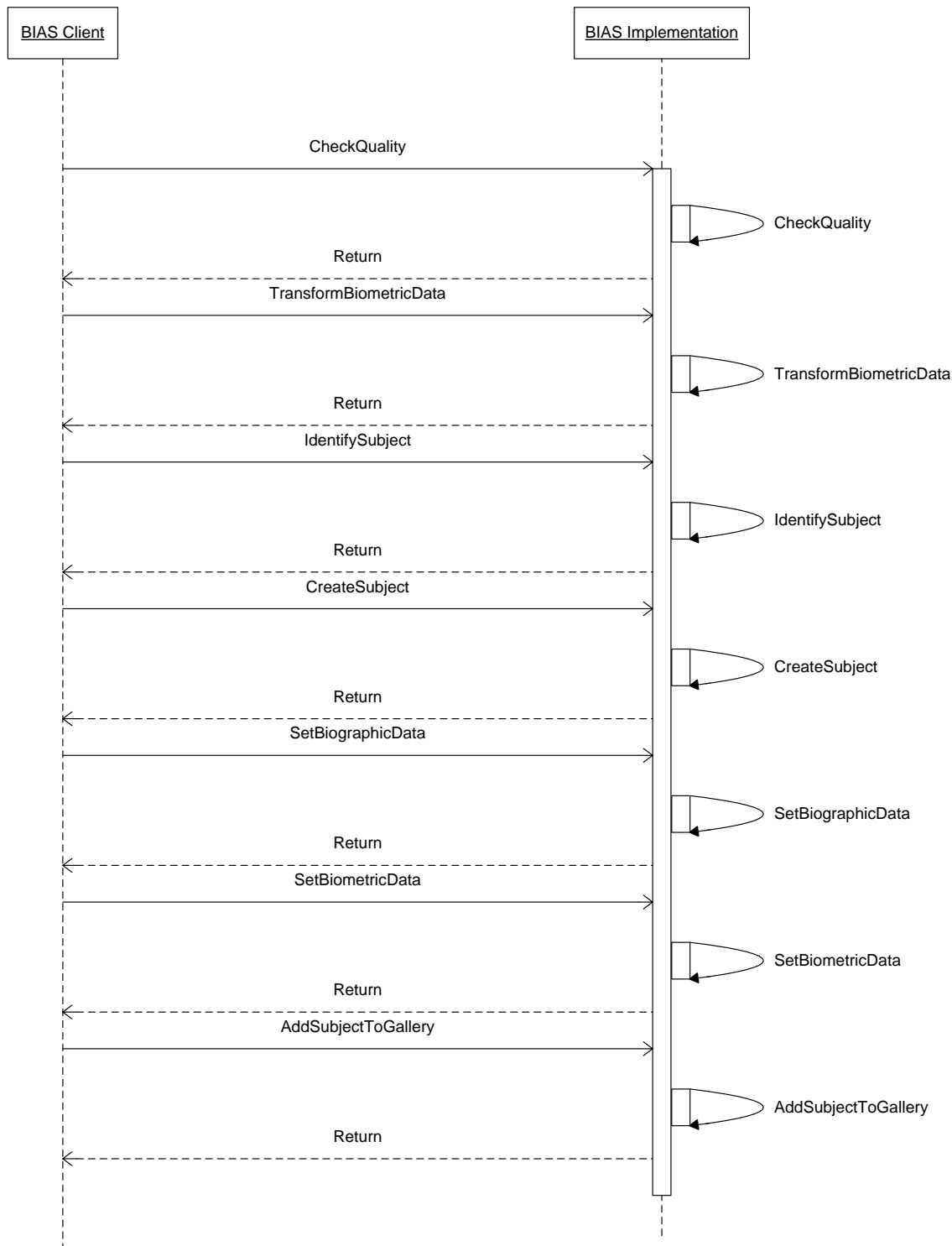
Here, if the result of the IdentifySubject is no matches found, then the subject is added to the gallery. If a match had been found then other logic may have been applied (e.g., return candidate list, add encounter for existing subject, etc.).





**BC.6 Primitive Enrollment Use Case**

In this use case, the enrollment operation is performed as a series of requests using the BIAS primitive operations. In this case, the client rather than the service provider controls the workflow of the higher level operation.



---

## Annex ~~C.D.~~ Samples (non-normative)

### ~~C.D.~~1 Create Subject Request/Response Example

#### *INCITS BIAS Specification*

```
<interface name="CreateSubject">
  <parameter name="SubjectID" type="xs:string"
direction="inout" use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### *OASIS BIAS Examples*

##### **Simple Create Subject Request:**

```
POST /bias HTTP/1.1
Host: www.acme.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "CreateSubject"
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <tns:CreateSubject
                                xmlns:tns="http://docs.oasis-open.org/bias/bias-
1.0/" />
  </soap:Body>
</soap:Envelope>
```

##### **Create Subject Request with SubjectID Parameter:**

```
POST /bias HTTP/1.1
Host: www.acme.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "CreateSubject"
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <tns:CreateSubject xmlns:tns="http://docs.oasis-
open.org/bias/bias-1.0/">
      <tns:Identity>
        <tns:SubjectID>123456789</tns:SubjectID>
      </tns:Identity>
    </tns:CreateSubject>
  </soap:Body>
```

```

5120 </soap:Envelope>

5121 Create Subject Request with Optional OASIS BIAS Content:
5122 POST /bias HTTP/1.1
5123 Host: www.acme.com
5124 Content-Type: application/soap+xml; charset="utf-8"
5125 Content-Length: nnnn
5126 SOAPAction: "CreateSubject"
5127 <?xml version="1.0"?>
5128 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5129     <soap:Body>
5130         <tns:CreateSubject xmlns:tns="http://docs.oasis-
5131 open.org/bias/bias-1.0/">
5132             <tns:GenericRequestParameters>
5133                 <tns:Application>BIAS Application</tns:Application>
5134                 <tns:ApplicationUser>BIAS User</tns:ApplicationUser>
5135             </tns:GenericRequestParameters>
5136             <tns:Identity>
5137                 <tns:SubjectID>123456789</tns:SubjectID>
5138             </tns:Identity>
5139         </tns:CreateSubject>
5140     </soap:Body>
5141 </soap:Envelope>

5142 Simple Create Subject Response:
5143 HTTP/1.1 200 OK
5144 Content-Type: application/soap+xml; charset="utf-8"
5145 Content-Length: nnnn
5146 <?xml version="1.0"?>
5147 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5148     <soap:Body>
5149         <tns:CreateSubjectResponse xmlns:tns="http://docs.oasis-
5150 open.org/bias/bias-1.0/">
5151             <tns:ResponseStatus>
5152                 <tns:Return>0</tns:Return>
5153             </tns:ResponseStatus>
5154             <tns:Identity>
5155                 <tns:SubjectID>123456789</tns:SubjectID>
5156             </tns:Identity>
5157         </tns:CreateSubjectResponse>
5158     </soap:Body>
5159 </soap:Envelope>

5160 Create Subject Response with Optional OASIS BIAS Content:

```

```

5161 HTTP/1.1 200 OK
5162 Content-Type: application/soap+xml; charset="utf-8"
5163 Content-Length: nnnn
5164 <?xml version="1.0"?>
5165 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5166     <soap:Body>
5167         <tns:CreateSubjectResponse xmlns:tns="http://docs.oasis-
5168 open.org/bias/bias-1.0/">
5169             <tns:ResponseStatus>
5170                 <tns:Return>0</tns:Return>
5171                 <tns:Message>Subject ID 123456789 successfully
5172 created.</tns:Message>
5173             </tns:ResponseStatus>
5174             <tns:Identity>
5175                 <tns:SubjectID>123456789</tns:SubjectID>
5176             </tns:Identity>
5177         </tns:CreateSubjectResponse>
5178     </soap:Body>
5179 </soap:Envelope>
5180

```

## CD.2 Set Biographic Data Request/Response Example

### INCITS BIAS Specification

```

5183 <interface name="SetBiographicData">
5184     <parameter name="SubjectID" type="xs:string" direction="in" />
5185     <parameter name="IdentityModel" type="IdentityModelType" direction="in"
5186 />
5187     <parameter name="EncounterID" type="xs:string" direction="inout"
5188 use="optional" />
5189     <parameter name="BiographicData"
5190 type="BiographicDataType" direction="in" />
5191     <parameter name="Return" type="xs:unsignedLong" direction="out" />
5192 </interface>

```

### OASIS BIAS Examples

#### Set Biographic Data Request:

```

5195 POST /bias HTTP/1.1
5196 Host: www.acme.com
5197 Content-Type: application/soap+xml; charset="utf-8"
5198 Content-Length: nnnn
5199 SOAPAction: "SetBiographicData"
5200 <?xml version="1.0"?>
5201 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5202     <soap:Body>

```

```

5203         <tns:SetBiographicData
5204             xmlns:tns="http://docs.oasis-open.org/bias/bias-
5205 1.0/">
5206             <tns:Identity>
5207                 <tns:SubjectID>123456789</tns:SubjectID>
5208                 <tns:BiographicData>
5209                     <tns:BiographicDataItem>
5210                         <tns:Name>Last</tns:Name>
5211                         <tns:Type>string</tns:Type>
5212                         <tns:Value>Doe</tns:Value>
5213                     </tns:BiographicDataItem>
5214                 </tns:BiographicData>
5215             </tns:Identity>
5216             <tns:IdentityModel>person</tns:IdentityModel>
5217         </tns:SetBiographicData>
5218     </soap:Body>
5219 </soap:Envelope>

```

#### 5220 **Set Biographic Data Response:**

```

5221 HTTP/1.1 200 OK
5222 Content-Type: application/soap+xml; charset="utf-8"
5223 Content-Length: nnnn
5224 <?xml version="1.0"?>
5225 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5226     <soap:Body>
5227         <tns:SetBiographicDataResponse
5228             xmlns:tns="http://docs.oasis-open.org/bias/bias-
5229 1.0/">
5230             <tns:ResponseStatus>
5231                 <tns:Return>0</tns:Return>
5232             </tns:ResponseStatus>
5233         </tns:SetBiographicDataResponse>
5234     </soap:Body>
5235 </soap:Envelope>
5236

```

### 5237 **CD.3 Set Biometric Data Request/Response Example**

#### 5238 *INCITS BIAS Specification*

```

5239 <interface name="SetBiometricData">
5240     <parameter name="SubjectID" type="xs:string" direction="in" />
5241     <parameter name="IdentityModel" type="IdentityModelType" direction="in"
5242 />
5243     <parameter name="EncounterID" type="xs:string" direction="inout"
5244 use="optional" />

```

```

5245         <parameter name="BIRList" type="CBEFF_BIR_ListType" direction="in" />
5246         <parameter name="Return" type="xs:unsignedLong" direction="out" />
5247     </interface>

5248     OASIS BIAS Examples

5249     Set Biometric Data Request:

5250     POST /bias HTTP/1.1
5251     Host: www.acme.com
5252     Content-Type: application/soap+xml; charset="utf-8"
5253     Content-Length: nnnn
5254     SOAPAction: "SetBiometricData"

5255     <?xml version="1.0"?>
5256     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5257         <soap:Body>
5258             <tns:SetBiometricData
5259                                     xmlns:tns="http://docs.oasis-open.org/bias/bias-
5260 1.0/">
5261                 <tns:Identity>
5262                     <tns:SubjectID>123456789</tns:SubjectID>
5263                     <tns:BiometricData>
5264                         <tns:BIRList>
5265                             <tns:BIR>biometric data</tns:BIR>
5266                         </tns:BIRList>
5267                     </tns:BiometricData>
5268                 </tns:Identity>
5269                 <tns:IdentityModel>person</tns:IdentityModel>
5270             </tns:SetBiometricData>
5271         </soap:Body>
5272     </soap:Envelope>

5273     Set Biometric Data Response:

5274     HTTP/1.1 200 OK
5275     Content-Type: application/soap+xml; charset="utf-8"
5276     Content-Length: nnnn
5277     <?xml version="1.0"?>
5278     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5279         <soap:Body>
5280             <tns:SetBiometricDataResponse
5281                                     xmlns:tns="http://docs.oasis-open.org/bias/bias-
5282 1.0/">
5283                 <tns:ResponseStatus>
5284                     <tns:Return>0</tns:Return>
5285                 </tns:ResponseStatus>
5286             </tns:SetBiometricDataResponse>

```

5287           </soap:Body>

5288   </soap:Envelope>

## Annex D.E. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

### Participants:

Name	Affiliation
Mr. Young Bang	Booz Allen Hamilton
Mr. Ed. Clay	Sun
Mr. Murty Gurajada *	Raining Data Corporation
Mr. Dale Hapeman	US Department of Defense
Dr. Charles Li	Raytheon
Mr. Kevin Mangold	NIST
Mr. John Mayer-Splain	US Department of Homeland Security
Dr. Ross Michaels	NIST
Mr. Ramesh Nagappan	Sun
Mr. Ash Parikh *	Raining Data Corporation
Mr. Matthew Swayze	Daon
Mr. Guy Swope*	Raytheon
Mrs. Catherine Tilton	Daon
Mr. Alessandro Triglia*	OSS Nokalva
<u>Mr. Matthew Young</u>	<u>US Department of Defense</u>
Mr. Brad Wing	NIST (formerly DHS)
Mr. Michael Wittman*	Raytheon
Mr. Gregory Zektser	Booz Allen Hamilton

\* Though no longer members of the BIAS TC at time of publication, these individuals contributed in the early stages of the development of this standard.

In addition, the inputs from the INCITS technical committee M1 are also gratefully appreciated.



5299

5300

## Annex ~~E.F.~~ Revision History

Revision	Date	Editor	Changes Made
0.01	2008-05-23	TBD	Initial draft
0.02	2008-07-23	TBD	Inserted data dictionary Added normative references Updated sec 3 & 5 + Annex B
0.03	2008-08-19	TBD	WSDL updated
0.04	2008-09-11	TBD	Updated references Added security requirements Corrected Fig. 3
0.05	2008-09-29	TBD	SSL/TLS requirement clarified Reordered material in 5.3 & App C/D Updated references 2 new use cases added (App C) Updated examples in App D
0.06	2008-11-17	TBD	Added BIAS operation name methods (new 5.3 + 4.2.27 & App B)
0.06a	2008-11-20	TBD	Updated references
0.07	2008-11-27	TBD	Revised fault structures and error handling
0.08	2009-06-22	TBD	Incorporated comments from informal public review.
0.09	2009-07-24	Tilton/Swayze	Incorporated comments from June review/meeting. Major changes included: <ul style="list-style-type: none"> <li>- Breaking Clause 3 into 2 clauses for data elements and operations</li> <li>- Specification of URI &amp; IRI</li> <li>- Clarifications and formatting</li> </ul>
0.10	2009-10-19	Tilton/Swayze	Expansion of conformance clause
0.11	2009-11-16	Tilton/Swayze	Miscellaneous edits and clarifications [Also published as CD01]
0.12	2010-11-04	<u>Mangold</u> /Tilton/Swayze	Incorporation of public review comments Update WSDL
0.13	2011-01-03	Tilton/Mangold	Clarification regarding xsd:any Updated WSDL
<u>0.14</u>	<u>2011-06-15</u>	<u>Mangold/Tilton</u>	<u>Inserted new Annex B – CBEFF Patron</u> <u>Format miscellaneous editorial changes</u>

5301

<u>0.15</u>	<u>2011-07-18</u>	<u>Mangold/Tilton</u>	<u>Updated namespace for CBEFF Patron Format + corrected finger subtype name in schema</u>
<u>0.16</u>	<u>2011-08-02</u>	<u>Mangold/Tilton</u>	<u>Changed BIAS CBEFF XML Patron Format Identifier to 0x0052 (line 4377).</u>