



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

Committee Draft 01

20 October 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-cd-01.pdf> (authoritative)
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-cd-01.doc>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-cd-01.html>

Previous Version:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-wd-07.pdf>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-wd-07.doc>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-wd-07.html>

Latest Version:

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

Technical Committee:

OASIS Biometric Identity Assurance Services (BIAS) Integration TC

Chair(s):

Cathy Tilton, Daon

Editor(s):

Matthew Swayze and Cathy Tilton, Daon

Related Work:

This specification is related to:

- ANSI INCITS 442-2008, Biometric Identity Assurance Services (BIAS)

Declared XML Namespace(s):

<http://docs.oasis-open.org/bias/ns/bias-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 BIAS TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" 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>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/bias/>.

Notices

Copyright © OASIS® 2009. 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 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", "BIAS" are trademarks of OASIS, 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	7
1.1	Purpose/Scope	7
1.2	Overview	7
1.3	Background	7
1.4	Relationship to Other Standards	7
1.5	Terminology	8
1.6	References	9
1.6.1	Normative References	9
1.6.2	Non-Normative References	10
2	Design Concepts and Architecture (non-normative)	12
2.1	Philosophy	12
2.2	Context	12
2.3	Architecture	12
3	Data dictionary	15
3.1	Documentation Conventions	15
3.2	Common Elements	16
3.2.1	ApplicationIdentifier	16
3.2.2	ApplicationUserIdentifier	16
3.2.3	BIASBiometricDataType	16
3.2.4	BIASFaultCode	16
3.2.5	BIASFaultDetail	17
3.2.6	BIASIdentity	18
3.2.7	BIASIDType	18
3.2.8	BiographicDataItem Type	18
3.2.9	BiographicDataSet Type	19
3.2.10	BiographicDataType	19
3.2.11	BiometricDataElementType	20
3.2.12	BiometricDataList Type	20
3.2.13	CandidateList Type	21
3.2.14	CandidateType	21
3.2.15	CapabilityList Type	21
3.2.16	CapabilityName	21
3.2.17	CapabilityType	22
3.2.18	CBEFF_BIR_List Type	22
3.2.19	CBEFF_BIR_Type	23
3.2.20	Classification	24
3.2.21	ClassificationAlgorithmType	24
3.2.22	ClassificationData	24
3.2.23	EncounterList Type	24
3.2.24	FusionInformationList Type	24
3.2.25	FusionInformationType	25
3.2.26	GenericRequestParameters	25
3.2.27	InformationType	25

3.2.28	ListFilterType	26
3.2.29	MatchType.....	26
3.2.30	ProcessingOptionsType	26
3.2.31	ProductID.....	26
3.2.32	QualityData.....	26
3.2.33	ResponseStatus	27
3.2.34	ReturnCode	27
3.2.35	Score	27
3.2.36	TokenType.....	27
3.2.37	VendorIdentifier	28
3.2.38	Version	28
3.2.39	VersionType	28
4	BIAS Messages	29
4.1	Primitive Operations	29
4.1.1	AddSubjectToGallery	29
4.1.2	CheckQuality	30
4.1.3	ClassifyBiometricData	32
4.1.4	CreateSubject.....	33
4.1.5	DeleteBiographicData	34
4.1.6	DeleteBiometricData	35
4.1.7	DeleteSubject	37
4.1.8	DeleteSubjectFromGallery	38
4.1.9	GetIdentifySubjectResults	39
4.1.10	IdentifySubject.....	41
4.1.11	ListBiographicData	43
4.1.12	ListBiometricData	45
4.1.13	PerformFusion	49
4.1.14	QueryCapabilities	51
4.1.15	RetrieveBiographicInformation	53
4.1.16	RetrieveBiometricInformation	55
4.1.17	SetBiographicData	56
4.1.18	SetBiometricData	58
4.1.19	TransformBiometricData	60
4.1.20	UpdateBiographicData	61
4.1.21	UpdateBiometricData	63
4.1.22	VerifySubject	64
4.2	Aggregate Operations.....	66
4.2.1	Enroll	66
4.2.2	GetEnrollResults	67
4.2.3	GetIdentifyResults	68
4.2.4	GetVerifyResults	70
4.2.5	Identify	71
4.2.6	RetrieveInformation	72
4.2.7	Verify	74
5	Message structure and rules	76

5.1 Purpose and constraints	76
5.2 Message requirements	77
5.3 Handling binary data.....	78
5.3.1 Base64 encoding.....	78
5.3.2 Use of XOP.....	78
5.4 Discovery	78
5.5 Identifying operations.....	79
5.5.1 Operation name element.....	79
5.5.2 WS-Addressing Action	79
5.6 Security	80
5.6.1 Use of SSL 3.0 or TLS 1.0	80
5.6.2 Data Origin Authentication	81
5.6.3 Message Integrity	81
5.6.4 Message Confidentiality	81
5.6.5 CBEFF BIR security features	81
5.6.6 Security Considerations	81
5.6.7 Security of Stored Data	82
5.6.8 Key Management	82
5.7 Use with other WS* standards.....	82
5.8 Tailoring	82
6 Error handling	83
6.1 BIAS operation return codes.....	83
6.2 SOAP fault codes	83
7 Conformance	84
Annex A. XML Schema	85
Annex B. Use Cases (non-normative)	140
B.1 Verification Use Case.....	140
B.2 Asynchronous Verification.....	141
B.3 Primitive Verification.....	142
B.4 Identification Use Case	143
B.5 Biometric Enrollment	144
B.6 Primitive Enrollment	145
Annex C. Samples (non-normative)	146
C.1 Create Subject Request/Response Example.....	146
C.2 Set Biographic Data Request/Response Example	148
C.3 Set Biometric Data Request/Response Example	149
Annex D. Acknowledgements	151
Annex E. Revision History.....	152

1 Introduction

2 1.1 Purpose/Scope

3 This Organization for the Advancement of Structured Information Standards (OASIS) Biometric Identity
4 Assurance Services (BIAS) profile specifies how to use the eXtensible Markup Language (XML) [[XML10](#)]
5 defined in ANSI INCITS 442-2008 – Biometric Identity Assurance Services [[INCITS-BIAS](#)] to invoke
6 Simple Object Access Protocol (SOAP) -based services that implement BIAS operations. These SOAP-
7 based services enable an application to invoke biometric identity assurance operations remotely in a
8 Services Oriented Architecture (SOA) infrastructure.

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

12 1.2 Overview

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

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

25 1.3 Background

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

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

41 1.4 Relationship to Other Standards

42 This OASIS BIAS profile comprises a companion standard to ANSI INCITS 442-2008 – Biometric Identity
43 Assurance Services, which defines the BIAS requirements and taxonomy, specifying the identity

44 assurance operations and the associated data elements. This OASIS BIAS profile specifies the design
45 concepts and architecture, data model and data dictionary, message structure and rules, and error
46 handling necessary to invoke SOAP-based services that implement BIAS operations.
47 Together, the BIAS standard and the BIAS profile provide an open framework for deploying and remotely
48 invoking biometric-based identity assurance capabilities that can be readily accessed across an SOA
49 infrastructure.
50 This relationship allows the leveraging of the biometrics and web services expertise of the two standards
51 development organizations. Existing standards are available in both domains and many of these
52 standards will provide the foundation and underlying capabilities upon which the biometric services
53 depend.

54 **1.5 Terminology**

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

58 The following additional terms and definitions are used:

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

61
62 BIAS operation and data element names are not defined here, but in their respective sections.
63

64 **BIAS** Biometric Identity Assurance Services
65

66 **BIR** Biometric Information Record
67

68 **ESB** Enterprise Service Bus
69

70 **HTTP** HyperText Transfer Protocol
71

72 **HTTPS** HyperText Transfer Protocol over SSL or HTTP Secure
73

74 **IRI** Internationalized Resource Identifier
75

76 **SOA** Service-Oriented Architecture
77

78 **SOAP** Simple Object Access Protocol
79

80 **SSL** Secure Sockets Layer
81

82 **TLS** Transport Layer Security
83

84 **UDDI** Universal Description, Discovery, and Integration
85

86 **URI** Uniform Resource Identifier
87

88 **VPN** Virtual Private Network
89

90 **WSDL** Web Services Description Language
91

92 **WSS** Web Services Security
93

94 **XML** eXtensible Markup Language
95

96
97

98 **CBEFF**
99 Common Biometric Exchange Formats Framework - data elements and BIR formats specified in
100 ISO/IEC 19785-1
101 **BIAS implementation**
102 software entity that is capable of creating, processing, sending, and receiving BIAS messages
103 **BIAS endpoint**
104 runtime entity, identified by an endpoint URI/IRI, capable of sending and receiving BIAS
105 messages, and containing a running BIAS implementation
106 **BIAS message**
107 message that can be sent from a BIAS endpoint to another BIAS endpoint through a BIAS link
108 channel
109 **BIAS request message**
110 BIAS message conveying a request for an action to be performed by the receiving BIAS endpoint
111 **BIAS response message**
112 BIAS message conveying a response to a prior BIAS request message

113 1.6 References

114 1.6.1 Normative References

- 115 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, IETF
116 RFC 2119, March 1997.
117 <http://www.ietf.org/rfc/rfc2119.txt>
- 118 **[CBEFF]** ISO/IEC 19785, Information technology – Common Biometric Exchange Formats
119 Framework – Part 1: Data element specification & Part 3: Patron format
120 specifications
121 <http://www.iso.org>
- 122 **[DATE-TIME]** ISO 8601:2004, Data elements and interchange formats — Information
123 interchange — Representation of dates and times
124 <http://www.iso.org>
- 125 **[INCITS-BIAS]** ANSI INCITS 442-2008, Biometric Identity Assurance Services (BIAS), May 2008
126 <http://www.incits.org>
- 127 **[IRI]** M. Duerst, et al, Internationalized Resource Identifiers, W3C RFC3987, January
128 2005
129 <http://www.ietf.org/rfc/rfc3987.txt>
- 130 **[SOAP11]** Simple Object Access Protocol (SOAP) 1.1, 8 May 2000
131 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 132 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, *Uniform Resource Identifiers (URI):
133 Generic Syntax*, RFC 3986, MIT/LCS, U.C. Irvine, Xerox Corporation, January
134 2005.
135 <http://ietf.org/rfc/rfc3986>
- 136 **[UTF-8]** ISO/IEC 10646:2003, Information technology — Universal Multiple-Octet Coded
137 Character Set (UCS)
138 <http://www.iso.org>

143	[WS-Addr]	W3C Recommendation, Web Services Addressing 1.0 - Core, and Web Services Addressing 1.0 - SOAP Binding, 9 May 2006 http://www.w3.org/2002/ws/addr/
144		
145		
146	[WS-I-Basic]	Basic Profile Version 1.1, 10 April 2006 http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html
147		
148	[WS-I-Bind]	Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile Version 1.0, 24 August 2004 http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html
149		
150		
151	[WSDL11]	Web Services Description Language (WSDL) 1.1, 15 March 2001 http://www.w3.org/TR/2001/NOTE-wsdl-20010315
152		
153	[XML 10]	Extensible Markup Language (XML) 1.0, 16 August 2006 http://www.w3.org/TR/2006/REC-xml-20060816/
154		
155	[XOP]	XML-binary Optimized Packaging, W3C Recommendation, 25 January 2005 http://www.w3.org/TR/2005/REC-xop10-20050125/
156		

1.6.2 Non-Normative References

157	[BioAPI]	ISO/IEC 19784-1:2006, Information technology – Biometric Application Programming Interface – Part 1: BioAPI Specification http://www.iso.org
158		
159		
160		
161	[BIO SEC]	ISO 19092 Financial services -- Biometrics -- Security framework http://www.iso.org
162		
163	[EBTS-DOD]	Department of Defense Electronic Biometric Transmission Specification, Version 2.0, 27 March 2009 http://www.biometrics.dod.mil/CurrentInitiatives/Standards/EBTS.aspx
164		
165		
166	[EBTS-FBI]	IAFIS-DOC-01078-8.1, “Electronic Biometric Transmission Specification (EBTS)”, Version 8.1, November 19, 2008, Federal Bureau of Investigation, Criminal Justice Information Services Division http://www.frbiospecs.org
167		
168		
169		
170	[EFTS]	IAFIS-DOC-01078-7, “Electronic Fingerprint Transmission Specification (EFTS)”, Version 7.1, May 2, 2005, Federal Bureau of Investigation, Criminal Justice Information Services Division http://www.frbiospecs.org
171		
172		
173		
174	[HR-XML]	HR-XML Consortium Library, 2007 April 15 http://www.hr-xml.org
175		
176	[INT-I]	Interpol Implementation of ANSI/NIST ITL1-2000, Ver 4.22b, October 28, 2005, The Interpol AFIS Expert Group http://www.interpol.int
177		
178		
179	[NIEM]	National Information Exchange Model (NIEM), Ver 2.0, June 2007, US DOJ/DHS http://www.niem.gov
180		
181	[RFC2246]	T. Dierks & C. Allen, <i>The TLS Protocol, Version 1.0</i> , January 1999 http://www.ietf.org/rfc/rfc2246.txt
182		

- 183 [RFC2617] J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*,
184 June 1999
185 <http://www.ietf.org/rfc/rfc2617.txt>
- 186 [RFC3280] R. Housley, et al, Internet X.509 Public Key Infrastructure Certificate and
187 Certificate Revocation List (CRL) Profile, April 2002
188 <http://www.ietf.org/rfc/rfc3280.txt>
- 189 [SAML] OASIS Standard, Security Assertion Markup Language (SAML) 2.0, March 2005
190 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- 191 [SAML SEC] OASIS Standard, Security and Privacy Considerations for the OASIS Security
192 Assertion Markup Language (SAML) V2.0, March 2005
193 <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>
- 194 [SSL3] SSL 3.0 Specification
195 <http://www.freesoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM>
- 196 [WSS] OASIS Standard, Web Services Security: SOAP Message Security 1.1, (WS-
197 Security 2004), 1 February 2006
198 <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>
- 200 [X509] X.509: Information technology - Open Systems Interconnection - The Directory:
201 Public-key and attribute certificate frameworks, ITU-T, August 2005
- 202 [xNAL] OASIS Committee Specification 02, Customer Information Quality Specifications
203 Version 3.0: Name (xNL), Address (xAL), Name and Address (xNAL) and Party
204 (xPIL), 20 September 2008
205 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq#download
- 206

207 2 Design Concepts and Architecture (non-normative)

208 2.1 Philosophy

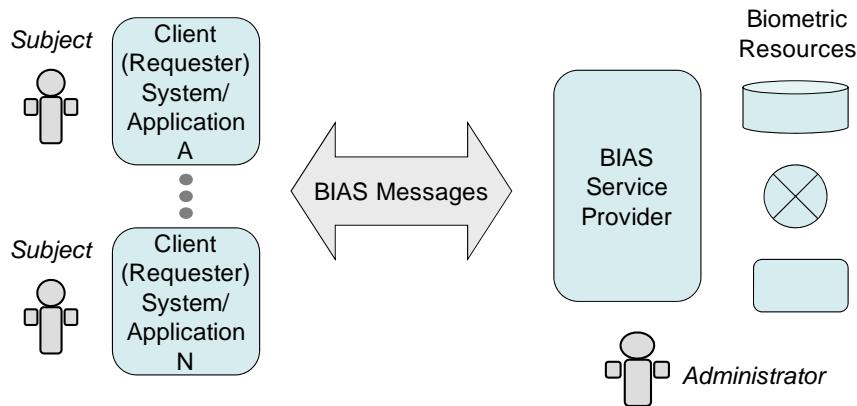
209 Rather than define a totally new and unique messaging protocol for biometric services, this specification
210 instead defines a method for using existing biometric and Web services standards to exchange biometric
211 data and perform biometric operations.

212 2.2 Context

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

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

222
223



224
225
226 **Figure 1. BIAS Context**

227
228 The clients (requesters) may use standard discovery mechanisms (i.e., UDDI directories) to discover the
229 BIAS service provider (implementation) or, particularly in closed systems, the URI/IRI and WSDL for the
230 service provider may be known *a priori* by the client BIAS application developer.

231 2.3 Architecture

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

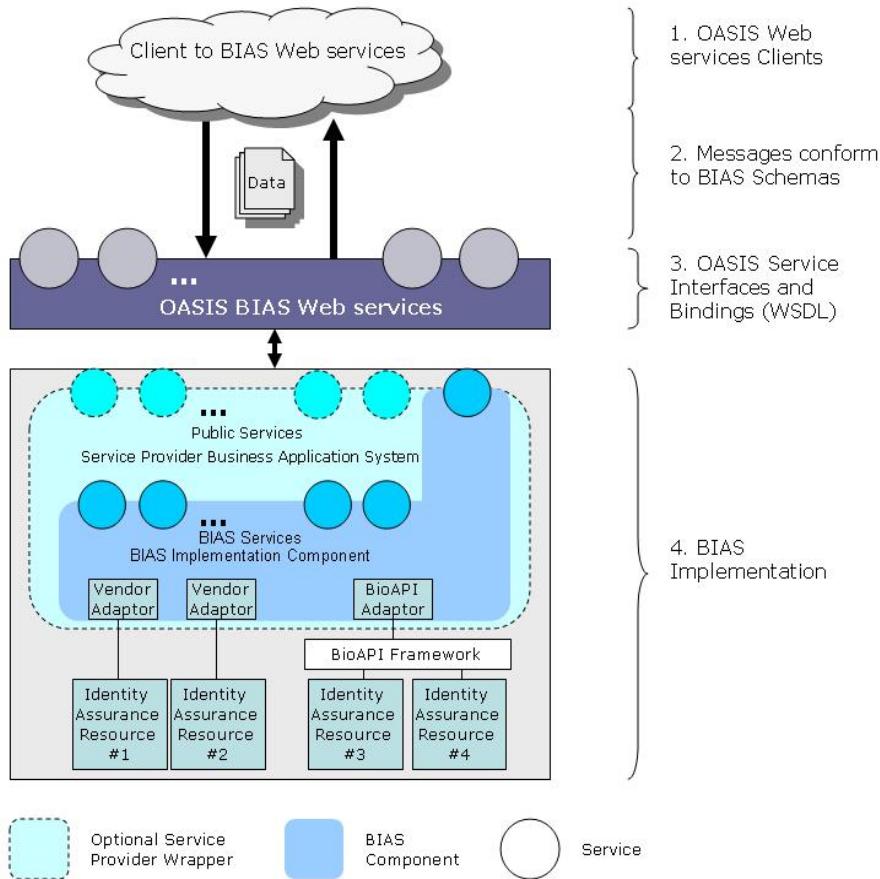
236 The BIAS Architecture as shown in Figure 2, in which:

- 237 • A Client request to the BIAS Web services may be triggered by a human interaction OR any
238 proxy system such as an ESB.

- 239 • Client sends and receives SOAP messages that conform to the BIAS schemas
 240 • The calls to the BIAS Implementation use OASIS Service Interfaces and Bindings (via
 241 WSDL)
 242 • The BIAS implementation maps the service call to the appropriate internal API or set of APIs
 243 and returns data according to the service interface.

244 Note that services are represented as circles.

245



246

247

248 **Figure 2. Representative BIAS Architecture**

249

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

252

253 At the heart of the BIAS messaging protocol are the concepts of BIAS messages and endpoints.

254 ***BIAS implementation***

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

258 ***BIAS messages***

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

261 ***BIAS endpoints***

- 262 A BIAS endpoint is a runtime entity, uniquely identified and accessed by an endpoint URI/IRI [[URI](#)] [[IRI](#)],
263 capable of sending and receiving BIAS messages.
- 264 Note that when not publicly and directly exposed, the endpoints for purposes of this specification are the
265 BIAS service provider exposing BIAS services and the component that directly interacts with that service
266 provider, e.g., the business application or ESB, rather than the ultimate end client requester.

267 3 Data dictionary

268 This section describes the BIAS data elements used within BIAS messages (as defined in Clause 4).
269 Common data elements are defined for use in one or more operations. These include common data types
270 or return codes. BIAS data elements are defined in ANSI INCITS 442-2008. The elements, complex types
271 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 for the XML schema.

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

275 3.1 Documentation Conventions

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

280

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 0..1 0..* 1..*	One occurrence Zero or one occurrence Zero or more occurrences One or more occurrences
?	Defines if the field must be present.	Y N C	Yes – is always required No – is not always required, an optional field. Conditional – requirement is dependent on system or message conditions.
Meaning	Gives a short description of the field's use		

281 **Figure 3. BIAS Message Input/Output Dictionary Table Headings**

282 **Fields Hierarchy Explained:**

283 To denote the field hierarchy the symbol ↑ is used to denote the child-of relationship.

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

286 **3.2 Common Elements**

287 **3.2.1 ApplicationIdentifier**

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

288 **3.2.2 ApplicationUserIdentifier**

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

289 **3.2.3 BIASBiometricDataType**

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

290 **3.2.4 BIASFaultCode**

Type:	String
Description:	Error code referenced in a SOAP fault.

291 **BIASFaultCode Enumeration Values**

Value	Description
UNKNOWN_ERROR	The service failed for an unknown reason.

Value	Description
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.

292

NOTES:

293

(1) See Clause 6 (Error handling) for an explanation of BIAS faults and return codes.

294

(2) Service provider MAY define additional values specific to their service implementation.

295

(3) See section 5.5 for additional information on BIAS security.

296

3.2.5 BIASFaultDetail

Field	Type	#	?	Meaning
BIASFaultDetail			Y	Defines the error information associated with a SOAP fault.
└ BIASFaultType	BIASFaultCode	1	Y	References an error code.
└ BIASFaultMessage	string	1	Y	Provides a brief explanation of the fault.
└ BIASFaultDescription	string	0..1	N	Provides detailed information about a BIAS fault, such as trace details.

297 **3.2.6 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	BIASIDType	0..1	C	A system unique identifier for a subject. Required as input to many operations.
↳ IdentityClaim	BIASIDType	0..1	N	An identifier by which a subject is known to a particular gallery or population group.
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
↳ EncounterList	EncounterListType	0..1	N	A list of encounters associated with a subject.
↳ BiographicData	BiographicDataType	0..1	N	An Identity's biographic data.
↳ BiographicDataElements	BiographicDataType	0..1	N	An Identity's biographic data elements that are stored in the implementing system.
↳ BiometricData	BIASBiometricDataType	0..1	N	An Identity's biometric data.

298 **3.2.7 BIASIDType**

Type:	string
Description:	A BIAS Identifier.

299 **3.2.8 BiographicDataItem Type**

Field	Type	#	?	Meaning
BiographicDataItemType			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.

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

302 **3.2.9 BiographicDataSetType**

Field	Type	#	?	Meaning
BiographicDataSetType			Y	Defines a set of biographic data that is formatted according to the specified format.
└ <i>unspecified</i>	any	0..*	N	Biographic data formatted according to a specific format.

303 **BiographicDataSetType Attributes**

Name	Type	Use	Meaning
name	string	REQUIRED	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	OPTIONAL	The version of the biographic data format (e.g., "7.1" for FBI-EFTS or "2.0" for NIEM).
source	string	REQUIRED	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	REQUIRED	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).

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

307

308 **3.2.10 BiographicDataType**

Field	Type	#	?	Meaning
BiographicDataType			Y	Defines a set of biographic data elements, utilizing either the BiographicDataItem to represent a list of elements or the BiographicDataSet to represent a complete, formatted set of biographic information. 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	BiographicDataItem	0..*	C	A single biographic data element.

Field	Type	#	?	Meaning
↳ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

- 309 NOTE: The implementer is given three choices for encoding biographic data:
- 310 • Encode only first and last name using the defined fields within BiographicDataType
- 311 • Define a list of biographic data elements using the BiographicDataItem Type
- 312 • Use a pre-defined set of biographic data (e.g., as specified in another standard) using the BiographicDataSetType.

314 See also INCITS 442, section 8.1 for further information.

315 3.2.11 BiometricDataElementType

Field	Type	#	?	Meaning
BiometricDataElementType			Y	Provides descriptive information about biometric data, such as the biometric type, subtype, and format, contained in the BDB of the CBEFF-BIR.
↳ BiometricType	iso-iec19785-3-7: Multiple-types	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	iso-iec19785-3-7: Subtype	0..1	N	More specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF.
↳ BDBFormatOwner	iso-iec19785-3-7: Registered-int	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	iso-iec19785-3-7: Registered-int	1	Y	Identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field.

316 NOTE: XML biometric metadata (BIR header elements) is aligned with ISO/IEC 19785-3, XML
 317 Patron Format. [\[CBEFF\]](#)

318

319 3.2.12 BiometricDataListType

Field	Type	#	?	Meaning
BiometricDataListType			Y	A list of biometric data elements.
↳ BiometricDataElement	BiometricDataElementType	0..*	N	Data structure containing information about a biometric record.

320 **3.2.13 CandidateListType**

Field	Type	#	?	Meaning
CandidateListType			Y	Defines a set of candidates, utilizing the CandidateType to represent each element in the set.
↳ Candidate	CandidateType	0..*	N	A single candidate.

321 **3.2.14 CandidateType**

Field	Type	#	?	Meaning
CandidateType			Y	Defines a single candidate as a possible match in response to a biometric identification request.
↳ Score	Score	0..1	N	The match score.
↳ BiographicData	BiographicDataType	0..1	N	Biographic data associated with the candidate match.
↳ BIRList	CBEFF_BIR_ListType	1	Y	Biometric data associated with the candidate match.

322 **CandidateType Attributes**

Name	Type	Use	Meaning
rank	integer	REQUIRED	The rank of the candidate in relation to other candidates for the same biometric identification operation.

323 **3.2.15 CapabilityListType**

Field	Type	#	?	Meaning
CapabilityListType			Y	Defines a set of capabilities.
↳ Capability	CapabilityType	0..*	N	A single capability.

324 **3.2.16 CapabilityName**

Type:	string
Description:	A list of capability items.

325 **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.

Value	Description
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.
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.

326

3.2.17 CapabilityType

Field	Type	#	?	Meaning
CapabilityType			Y	Defines a single capability supported by an implementing system.
↳ CapabilityName	CapabilityName	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.

327

3.2.18 CBEFF_BIR_ListType

Field	Type	#	?	Meaning
CBEFF_BIR_ListType			Y	A list of CBEFF-BIR elements.

Field	Type	#	?	Meaning
↳ BIR	CBEFF_BIR_Type	0..*	N	CBEFF structure containing information about a biometric sample.

328 **3.2.19 CBEFF_BIR_Type**

Field	Type	#	?	Meaning
CBEFF_BIR_Type			Y	Represents biometric information, with either a non-XML or XML representation.
↳ BIR_Information		0..1	N	Describes what is contained in a BIR.
↳ bir-info	iso-iec19785-3-7:BIR-info	0..1	N	Contains information about the CBEFF-BIR.
↳ bdb-info	iso-iec19785-3-7:BDB-info	0..1	N	Contains information about the BDB in a simple CBEFF-BIR.
↳ sb-info	iso-iec19785-3-7:SB-info	0..1	N	Contains information about the security block, if used, in a simple CBEFF-BIR.
↳ BIR		1	Y	One of the following sub-elements must be present.
↳ BinaryBIR	base64Binary	0..1	C	A non-XML CBEFF-BIR.
↳ URI_BIR	anyURI/IRI	0..1	C	A URI/IRI reference to a CBEFF-BIR.
↳ XML_BIR	iso-iec19785-3-7:BIR	0..1	C	An XML CBEFF-BIR, using the XML Patron Format as defined in ISO/IEC 19785-3:2007.

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

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

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

337 **CBEFF_BIR_Type Attributes**

Name	Type	Use	Meaning
format-owner	iso-iec19785-3-7:Registered-int	REQUIRED	Identifies the Patron format owner.
format-type	iso-iec19785-3-7:Registered-int	REQUIRED	Identifies the Patron format type.

338 NOTE:

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

343 **3.2.20 Classification**

Type:	string
Description:	The result of a classification.

344 **3.2.21 ClassificationAlgorithmType**

Type:	string
Description:	Type of classification algorithm that was used to perform the classification.

345 **3.2.22 ClassificationData**

Field	Type	#	?	Meaning
ClassificationData			Y	Contains information on classification results and the algorithm used to determine the classification.
└ Classification	Classification	1	Y	The result of the classification.
└ ClassificationAlgorithmType	ClassificationAlgorithmType	1	Y	Identifies the type of classification algorithm that was used to perform the classification.

346 **3.2.23 EncounterListType**

Field	Type	#	?	Meaning
EncounterListType			Y	Defines a set of encounters.
└ EncounterID	BIASIDType	0..*	N	The identifier of an encounter.

347 **3.2.24 FusionInformationListType**

Field	Type	#	?	Meaning
FusionInformationListType			Y	Contains at a minimum two sets of fusion input elements, as input to the PerformFusion operation.
└ FusionElement	FusionInformationType	2..*	Y	A set of fusion information.

348 **3.2.25 FusionInformationType**

Field	Type	#	?	Meaning
FusionInformationType			Y	Represents the information necessary to perform a fusion operation.
↳ BiometricType	iso-iec19785-3-7: Multiple-types	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↳ BiometricSubType	iso-iec19785-3-7: Subtype	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.
↳ Score	Score	0..1	C	The similarity score assigned by the matching algorithm. Either a Score or a Decision element MUST be used.
↳ Decision	string	0..1	C	The match decision assigned by the matching algorithm. Either a Score or a Decision element MUST be used.

349 **3.2.26 GenericRequestParameters**

Field	Type	#	?	Meaning
GenericRequestParameters			Y	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	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.

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

352 **3.2.27 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.

Field	Type	#	?	Meaning
↳ <i>unspecified</i>	any	0..*	N	

353 **3.2.28 ListFilterType**

Field	Type	#	?	Meaning
ListFilterType			Y	Provides a method to filter the amount of information returned in a search of biometric data.
↳ BiometricTypeFilter	iso-iec19785-3-7: Multiple-types	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.

354 **3.2.29 MatchType**

Type:	boolean
Description:	The result of a fusion method.

355 **3.2.30 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.

356 **3.2.31 ProductID**

Type:	string
Description:	The vendor's ID for a particular product.

357 **3.2.32 QualityData**

Field	Type	#	?	Meaning
QualityData			Y	Contains information about a biometric sample's quality and the algorithm used to compute the quality.
↳ QualityScore	iso-iec19785-3-7: Quality	0..1	N	The quality of a biometric sample.

Field	Type	#	?	Meaning
└ AlgorithmVendor	VendorIdentifier	1	Y	The vendor of the quality algorithm used to determine the quality score.
└ AlgorithmVendorProductID	ProductID	1	Y	The vendor's ID for the algorithm used to determine the quality.
└ AlgorithmVersion	VersionType	0..1	N	The version of the algorithm used to determine the quality.

358 3.2.33 ResponseStatus

Field	Type	#	?	Meaning
ResponseStatus			Y	
└ Return	ReturnCode	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.

359 3.2.34 ReturnCode

Type:	unsignedLong
Description:	Return value specifying success or other condition.

360 ReturnCode Enumeration Values

Value	Description
0	Success

361 3.2.35 Score

Type:	float
Description:	Match result or quality score.

362 NOTE: Matching scores MAY be in a standardized or proprietary form in terms of value range and interpretation. Quality scores, however, follow the definition found in ISO/IEC 19785-3, section 13.

364 3.2.36 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.

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

367 **3.2.37 VendorIdentifier**

Type:	string
Description:	Identifies a vendor.

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

371 **3.2.38 Version**

Field	Type	#	?	Meaning
Version			Y	For a description or definition of each data element, see the referenced CBEFF standards in the CBEFF_BIR_Type schema.
└ major	nonNegativeInteger	1	Y	
└ minor	nonNegativeInteger	1	Y	

372 **3.2.39 VersionType**

Type:	string
Description:	The version of a component.

373 4 BIAS Messages

374 This section describes the BIAS messages implementing BIAS operations as defined in ANSI INCITS
375 442-2008. The operations are listed alphabetically, with each operation containing a request and a
376 response message. The tables follow the conventions described in section 3.1.

377 4.1 Primitive Operations

378 4.1.1 AddSubjectToGallery

379 [AddSubjectToGalleryRequest](#)

380 [AddSubjectToGalleryResponse](#)

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

387 Request Message

Field	Type	#	?	Meaning
AddSubjectToGalleryRequest			Y	Register a subject to a given gallery or population group.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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	BIASIDType	1	Y	The identifier of the gallery or population group to which the subject will be added.
└ Identity	BIASIdentity	1	Y	The identity to add to the gallery.
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

Field	Type	#	?	Meaning
↳ IdentityClaim	BIASIDType	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	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

388 **Response Message**

Field	Type	#	?	Meaning
AddSubjectToGalleryResponse			Y	The response to an AddSubjectToGallery operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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.

389 **4.1.2 CheckQuality**

390 [CheckQualityRequest](#)

391 [CheckQualityResponse](#)

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

399 **Request Message**

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

Field	Type	#	?	Meaning
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	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	BIASBiometricDataType	1	Y	Data structure containing a single biometric sample for which a quality score is to be determined.
↳ BIR	CBEFF_BIR_Type	1	Y	The biometric sample.
↳ Quality	QualityData	0..1	N	Specifies a particular algorithm vendor and vendor product ID.
↳ AlgorithmVendor	VendorIdentifier	1	Y	The vendor of the quality algorithm used to determine the quality score.
↳ AlgorithmVendorProductID	ProductID	1	Y	The vendor's ID for the algorithm used to determine the quality.

400 Response Message

Field	Type	#	?	Meaning
CheckQualityResponse			Y	The response to a CheckQuality operation.

Field	Type	#	?	Meaning
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	QualityData	1	Y	Contains the quality information for the submitted biometric sample.
↳ QualityScore	iso-iec19785-3-7: Quality	0..1	N	The quality of a biometric sample.
↳ AlgorithmVendor	VendorIdentifier	1	Y	The vendor of the quality algorithm used to determine the quality score.
↳ AlgorithmVendorProductID	ProductID	1	Y	The vendor's ID for the algorithm used to determine the quality.
↳ AlgorithmVersion	VersionType	1	Y	The version of the algorithm used to determine the quality.

401 **4.1.3 ClassifyBiometricData**

402 [ClassifyBiometricDataRequest](#)

403 [ClassifyBiometricDataResponse](#)

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

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

408 **Request Message**

Field	Type	#	?	Meaning
ClassifyBiometricDataRequest			Y	Classifies a biometric sample.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	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: "ClassifyBiometricData".
↳ BiometricData	BIASBiometricDataType	1	Y	Data structure containing a single biometric sample for which the classification is to be determined.
↳ BIR	CBEFF_BIR_Type	1	Y	The biometric sample.

409 **Response Message**

Field	Type	#	?	Meaning
ClassifyBiometricDataResponse			Y	The response to a ClassifyBiometricData operation, containing the classification of a biometric sample.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	ClassificationData	1	Y	Information on the results and type of classification performed.
↳ Classification	Classification	1	Y	The result of the classification.
↳ ClassificationAlgorithmType	ClassificationAlgorithmType	1	Y	Identifies the type of classification algorithm that was used to perform the classification.

410 **4.1.4 CreateSubject**

411 [CreateSubjectRequest](#)

412 [CreateSubjectResponse](#)

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

416 **Request Message**

Field	Type	#	?	Meaning
CreateSubjectRequest			Y	
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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".

417 **Response Message**

Field	Type	#	?	Meaning
CreateSubjectResponse			Y	The response to a CreateSubject operation, containing the subject ID of the new subject record.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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	BIASIdentity	1	Y	
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

418 **4.1.5 DeleteBiographicData**

419 [DeleteBiographicDataRequest](#)

420 [DeleteBiographicDataResponse](#)

421 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.

424 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.

427 **Request Message**

Field	Type	#	?	Meaning
DeleteBiographicDataRequest			Y	Erase all of the biographic data associated with a given subject record or, in the encounter-centric model, with a given encounter.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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	BIASIdentity	1	Y	
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
└ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

428 **Response Message**

Field	Type	#	?	Meaning
DeleteBiographicDataResponse			Y	The response to a DeleteBiographicData operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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.

429 **4.1.6 DeleteBiometricData**

430 [DeleteBiometricDataRequest](#)

431 [DeleteBiometricDataResponse](#)

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

435 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.

438 **Request Message**

Field	Type	#	?	Meaning
DeleteBiometricDataRequest			Y	Erase all of the biometric data associated with a given subject record or, in the encounter-centric model, with a given encounter.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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	BIASIdentity	1	Y	
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
└ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

439 **Response Message**

Field	Type	#	?	Meaning
DeleteBiometricDataResponse			Y	The response to a DeleteBiometricData operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
↳ Return	ReturnCode	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.

440 4.1.7 DeleteSubject

441 [DeleteSubjectRequest](#)

442 [DeleteSubjectResponse](#)

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

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

449 Request Message

Field	Type	#	?	Meaning
DeleteSubjectRequest			Y	Delete an existing subject record and, in an encounter-centric model, any associated encounter information.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentity	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	BIASIdentity	1	Y	The identity of the subject to delete.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

450 Response Message

Field	Type	#	?	Meaning
DeleteSubjectResponse			Y	The response to a DeleteSubject operation.

Field	Type	#	?	Meaning
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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 **4.1.8 DeleteSubjectFromGallery**

452 [DeleteSubjectFromGalleryRequest](#)

453 [DeleteSubjectFromGalleryResponse](#)

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

457 **Request Message**

Field	Type	#	?	Meaning
DeleteSubjectFromGalleryRequest			Y	Remove the registration of a subject from a gallery or population group.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentity	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	BIASIDType	1	Y	The identifier of the gallery or population group from which the subject will be deleted.
↳ Identity	BIASIdentity	1	Y	The identity to remove from the gallery.
↳ SubjectID	BIASIDType	0..1	C	A system unique identifier for a subject. Required if an Identity Claim is not provided.

Field	Type	#	?	Meaning
└ IdentityClaim	BIASIDType	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.

458 **Response Message**

Field	Type	#	?	Meaning
DeleteSubjectFromGalleryResponse			Y	The response to a DeleteSubjectFromGallery operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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.

459 **4.1.9 GetIdentifySubjectResults**

460 [GetIdentifySubjectResultsRequest](#)

461 [GetIdentifySubjectResultsResponse](#)

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

466 **Request Message**

Field	Type	#	?	Meaning
GetIdentifySubjectResultsRequest			Y	Retrieve the identification results for a specified token, which was returned by the IdentifySubject operation.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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	TokenType	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.

467 Response Message

Field	Type	#	?	Meaning
GetIdentifySubjectResultsResponse			Y	The response to a GetIdentifySubjectResults operation, which includes a candidate list.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	CandidateListType	1	Y	A rank-ordered list of candidates that have a likelihood of matching the input biometric sample.
↳ Candidate	CandidateType	0..*	N	A single candidate.
↳ Score	Score	0..1	N	The match score.
↳ BiographicData	BiographicDataType	0..1	N	Biographic data associated with the candidate match.
↳ BIRList	CBEFF_BIR_ListType	1	Y	Biometric data associated with the candidate match.

Field	Type	#	?	Meaning
↳ BIR	CBEFF_BIR_Type	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
IdentifySubjectRequest			Y	Perform an identification search against a given gallery for a given biometric.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	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	BIASIDType	1	Y	The identifier of the gallery or population group which will be searched.
↳ Identity	BIASIdentity	1	Y	Contains the BIR, a data structure containing the biometric sample for the search.
↳ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.

Field	Type	#	?	Meaning
↑ BIR	CBEFF_BIR_Type	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.

479 Response Message

Field	Type	#	?	Meaning
IdentifySubjectResponse			Y	The response to an IdentifySubject operation, returning a rank-ordered candidate list.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	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	CandidateListType	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	CandidateType	0..*	N	A single candidate.
↑ Score	string	0..1	N	The match score.
↑ BiographicData	BiographicDataType	0..1	N	Biographic data associated with the candidate match.
↑ BIRList	CBEFF_BIR_ListType	1	Y	Biometric data associated with the candidate match.
↑ BIR	CBEFF_BIR_Type	0..*	N	CBEFF structure containing information about a biometric sample.

Field	Type	#	?	Meaning
↳ Token	TokenType	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.

480 NOTES:

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

4.1.11 ListBiographicData

486 [ListBiographicDataRequest](#)

487 [ListBiographicDataResponse](#)

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

495 Request Message

Field	Type	#	?	Meaning
ListBiographicDataRequest			Y	Lists the biographic data elements stored for a subject.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.

Field	Type	#	?	Meaning
↳ ApplicationUser	ApplicationUserIdentity	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	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.

496 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.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	BIASIdentity	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	BiographicDataType	0..1	C	An Identity's biographic data elements that are stored in the implementing system.
↳ BiographicDataItem	BiographicDataItemType	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	EncounterListType	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.
↳ EncounterID	BIASIDType	0..*	N	The identifier of an encounter.

497 **4.1.12 ListBiometricData**

498 [ListBiometricDataRequest](#)

499 [ListBiometricDataResponse](#)

500 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

503 encounter ID MAY be specified to indicate that only the biometric data elements stored for that encounter
504 should be returned. If an encounter ID is not specified and encounter data exists for the subject, the
505 operation returns the list of encounter IDs which contain biometric data using the Encounter List output
506 parameter, and the Biometric Data List output parameter is empty.

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

512 Request Message

Field	Type	#	?	Meaning
ListBiometricDataRequest			Y	Lists the biometric data elements stored for a subject.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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".
└ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
└ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
└ ListFilterType	ListFilterType	0..1	N	Indicates what biometric information should be returned.

Field	Type	#	?	Meaning
↳ BiometricTypeFilter	iso-iec19785-3-7:Multiple-types	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.

513 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.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	BIASIdentity	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	BIASBiometricDataType	0..1	C	An Identity's biometric data.
↳ BiometricDataList	BiometricDataListType	0..1	N	A list of biometric data elements.
↳ BiometricDataElement	BiometricDataElementType	1..*	Y	Data structure containing information about a biometric record.
↳ BiometricType	iso-iec19785-3-7:Multiple-types	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	iso-iec19785-3-7:Subtype	0..1	N	More specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF.

Field	Type	#	?	Meaning
↳ BDBFormatOwner	iso-iec19785-3-7: Registered-int	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	iso-iec19785-3-7: Registered-int	1	Y	Identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field.
↳ EncounterList	EncounterListType	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	BIASIDType	1..*	Y	The identifier of an encounter.

514 **4.1.13 PerformFusion**

515 [PerformFusionRequest](#)

516 [PerformFusionResponse](#)

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

521 Request Message

Field	Type	#	?	Meaning
PerformFusionRequest			Y	Accepts either match score or match decision information and creates a fused match result.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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	FusionInformationListType	1	Y	Score or decision input information to the fusion method.
└ FusionElement	FusionInformationType	2..*	Y	A set of fusion information.
└ BiometricType	iso-iec19785-3-7:Multiple-types	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
└ BiometricSubType	iso-iec19785-3-7:Subtype	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.

Field	Type	#	?	Meaning
↑ AlgorithmType	string	1	Y	The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision.
↑ Score	Score	0..1	C	The similarity score assigned by the matching algorithm. Either a Score or a Decision element MUST be used.
↑ Decision	string	0..1	C	The match decision assigned by the matching algorithm. Either a Score or a Decision element MUST be used.

522 **Response Message**

Field	Type	#	?	Meaning
PerformFusionResponse			Y	The response to the PerformFusion operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	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	MatchType	1	1	Indicates the result of the fusion method.

523 **4.1.14 QueryCapabilities**

524 [QueryCapabilitiesRequest](#)

525 [QueryCapabilitiesResponse](#)

526 The QueryCapabilities operation returns a list of the capabilities, options, galleries, etc. that are supported
 527 by the BIAS implementation. Refer to Annex A in the INCITS BIAS standard for conformance
 528 requirements regarding which capability names an implementation must use in the QueryCapabilities
 529 operation.

530 **Request Message**

Field	Type	#	?	Meaning
QueryCapabilitiesRequest			Y	Returns a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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".

531 **Response Message**

Field	Type	#	?	Meaning
QueryCapabilitiesResponse			Y	The response to a QueryCapabilities operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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	CapabilityListType	1	Y	A list of capabilities supported by the BIAS implementation.
└ Capability	CapabilityType	0..*	N	A single capability.
└ CapabilityName	CapabilityName	1	Y	The name of the capability.

Field	Type	#	?	Meaning
↳ 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.

532 **4.1.15 RetrieveBiographicInformation**

533 [RetrieveBiographicInformationRequest](#)

534 [RetrieveBiographicInformationResponse](#)

535 The RetrieveBiographicInformation operation retrieves the biographic data associated with a subject ID.
 536 In the encounter-centric model, the encounter ID MAY be specified and the operation will return the
 537 biographic data associated with that encounter. If the encounter ID is not specified in the encounter-
 538 centric model, the operation returns the biographic information associated with the most recent
 539 encounter.

540 **Request Message**

Field	Type	#	?	Meaning
RetrieveBiographicInformationRequest			Y	Retrieves the biographic data associated with a subject ID.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	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: "RetrieveBiographicInformation".
↳ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.

541 Response Message

Field	Type	#	?	Meaning
RetrieveBiographicInformationResponse			Y	The response to a RetrieveBiographicInformation operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	BIASIdentity	1	Y	Includes the set of biographic data associated with a subject.
↳ BiographicData	BiographicDataType	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.

Field	Type	#	?	Meaning
↳ BiographicDataItem	BiographicDataItemType	0..*	C	A single biographic data element.
↳ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

542 **4.1.16 RetrieveBiometricInformation**

543 [RetrieveBiometricInformationRequest](#)

544 [RetrieveBiometricInformationResponse](#)

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

551 **Request Message**

Field	Type	#	?	Meaning
RetrieveBiometricInformationRequest			Y	Retrieves the biometric data associated with a subject ID.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	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	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.

Field	Type	#	?	Meaning
↳ BiometricType	iso-iec19785-3-7:Multiple-types	0..1	N	The type of biological or behavioral data to retrieve.

552 **Response Message**

Field	Type	#	?	Meaning
RetrieveBiometricInformationResponse			Y	The response to a RetrieveBiometricInformation operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	BIASIdentity	1	Y	Includes the biometric data associated with a subject.
↳ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↳ BIRList	CBEFF_BIR_ListType	1	Y	A list of CBEFF-BIR elements.
↳ BIR	CBEFF_BIR_Type	0..*	N	CBEFF structure containing information about a biometric sample.

553 **4.1.17 SetBiographicData**

554 [SetBiographicDataRequest](#)

555 [SetBiographicDataResponse](#)

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

563 Request Message

Field	Type	#	?	Meaning
SetBiographicDataRequest			Y	Associates biographic data to a given subject record.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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".
└ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to store.
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
└ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
└ BiographicData	BiographicDataType	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.

Field	Type	#	?	Meaning
↳ BiographicDataItem	BiographicDataItemType	0..*	C	A single biographic data element.
↳ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

564 **Response Message**

Field	Type	#	?	Meaning
SetBiographicDataResponse			Y	The response to a SetBiographicData operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	BIASIdentity	0..1	C	In an encounter-centric model, identifies the encounter ID assigned to a new encounter.
↳ EncounterID	BIASIDType	1	Y	The identifier of an encounter associated with the subject.

565 **4.1.18 SetBiometricData**

566 [SetBiometricDataRequest](#)

567 [SetBiometricDataResponse](#)

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

575 **Request Message**

Field	Type	#	?	Meaning
SetBiometricDataRequest			Y	Associates biometric data to a given subject record.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.

Field	Type	#	?	Meaning
↑ ApplicationUser	ApplicationUserIdentity	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	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biometric data to store.
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↑ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
↑ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↑ BIRList	CBEFF_BIR_ListType	1	Y	A list of CBEFF-BIR elements.
↑ BIR	CBEFF_BIR_Type	1..*	Y	CBEFF structure containing information about a biometric sample.

576 Response Message

Field	Type	#	?	Meaning
SetBiometricDataResponse			Y	The response to a SetBiometricData operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	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	BIASIdentity	0..1	C	In an encounter-centric model, identifies the encounter ID assigned to a new encounter.
↑ EncounterID	BIASIDType	1	Y	The identifier of an encounter associated with the subject.

577 **4.1.19 TransformBiometricData**

578 [TransformBiometricDataRequest](#)

579 [TransformBiometricDataResponse](#)

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

582 **Request Message**

Field	Type	#	?	Meaning
TransformBiometricDataRequest			Y	Transforms or processes a given biometric in one format into a new target format.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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	CBEFF_BIR_Type	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.

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

584 **Response Message**

Field	Type	#	?	Meaning
TransformBiometricDataResponse			Y	The response to a TransformBiometricData operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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	CBEFF_BIR_Type	0..1	N	Data structure containing the new, transformed biometric information.

585 **4.1.20 UpdateBiographicData**586 [UpdateBiographicDataRequest](#)587 [UpdateBiographicDataResponse](#)

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

591 **Request Message**

Field	Type	#	?	Meaning
UpdateBiographicDataRequest			Y	Updates the biographic data for a given subject record.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to update.
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
└ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
└ BiographicData	BiographicDataType	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	BiographicDataItemType	0..*	C	A single biographic data element.
└ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

592 Response Message

Field	Type	#	?	Meaning
UpdateBiographicDataResponse			Y	The response to an UpdateBiographicData operation.
└ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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.

593 **4.1.21 UpdateBiometricData**
 594 [UpdateBiometricDataRequest](#)
 595 [UpdateBiometricDataResponse](#)
 596 The UpdateBiometricData operation updates the biometric data for an existing subject record. The
 597 operation includes an OPTIONAL parameter indicating if the new biometric sample should be merged
 598 with the existing biometric sample. If this parameter is set to "False" or is not used in the request, the
 599 operation replaces the existing biometric sample with the new biometric sample. In the encounter-centric
 600 model, the encounter ID MUST be specified.

601 **Request Message**

Field	Type	#	?	Meaning
UpdateBiometricDataRequest			Y	Updates a single biometric sample for a given subject record.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biometric data to update.
└ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
└ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
└ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
└ BIR	CBEFF_BIR_Type	1	Y	Contains biometric information in either a non-XML or an XML representation.

Field	Type	#	?	Meaning
↳ Merge	boolean	0..1	N	Value indicating if the input biometric sample should be merged with any existing biometric information.

602 **Response Message**

Field	Type	#	?	Meaning
UpdateBiometricDataResponse			Y	The response to an UpdateBiometricData operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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.

603 **4.1.22 VerifySubject**

604 [VerifySubjectRequest](#)

605 [VerifySubjectResponse](#)

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

609 **Request Message**

Field	Type	#	?	Meaning
VerifySubjectRequest			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.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentity	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: "VerifySubject".
↳ GalleryID	BIASIDType	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	BIASIdentity	1	Y	Includes the identifying information and/or input and reference biometric samples.
↳ IdentityClaim	BIASIDType	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	BIASBiometricDataType	1	Y	An Identity's biometric data.
↳ InputBIR	CBEFF_BIR_Type	1	Y	Maps to specific INCITS BIAS elements as required by that specification.
↳ ReferenceBIR	CBEFF_BIR_Type	0..1	C	Maps to specific INCITS BIAS elements as required by that specification. Required if an Identity Claim is not provided.

610 Response Message

Field	Type	#	?	Meaning
VerifySubjectResponse			Y	The response to a VerifySubject operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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
↳ 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	Score	0..1	N	The score if the biometric information matched.

611 4.2 Aggregate Operations

612 4.2.1 Enroll

613 [EnrollRequest](#)

614 [EnrollResponse](#)

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

622 Request Message

Field	Type	#	?	Meaning
EnrollRequest			Y	Adds a new subject or, in an encounter-centric model, a new encounter to the system.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentity	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	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed.
↳ Option	string	0..*	N	An option supported by the implementing system.

Field	Type	#	?	Meaning
└ InputData	InformationType	1	Y	Contains the input data for the operation, as required by the implementing system.

623 **Response Message**

Field	Type	#	?	Meaning
EnrollResponse			Y	The response to an Enroll operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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	InformationType	0..1	N	Contains the output data for the response.

624 **4.2.2 GetEnrollResults**

625 [GetEnrollResultsRequest](#)

626 [GetEnrollResultsResponse](#)

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

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

633 **Request Message**

Field	Type	#	?	Meaning
GetEnrollResultsRequest			Y	Retrieves the enrollment results for the specified token.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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: "GetEnrollResults".
↳ Token	TokenType	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.

634 Response Message

Field	Type	#	?	Meaning
GetEnrollResultsResponse			Y	The response to a GetEnrollResults operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	InformationType	0..1	N	Contains the output data for the response.

635 4.2.3 GetIdentifyResults

636 [GetIdentifyResultsRequest](#)

637 [GetIdentifyResultsResponse](#)

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

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

644

645 Request Message

Field	Type	#	?	Meaning
GetIdentifyResultsRequest			Y	Retrieves the identification results for the specified token
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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	TokenType	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 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.

646 Response Message

Field	Type	#	?	Meaning
GetIdentifyResultsResponse			Y	The response to a GetIdentifyResults operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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	InformationType	0..1	N	Contains the output data for the response.

647 **4.2.4 GetVerifyResults**

648 [GetVerifyResultsRequest](#)

649 [GetVerifyResultsResponse](#)

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

654 If the service provider implements an asynchronous Verify_operation, then it MUST also implement the
655 GetVerifyResults operation.

656 **Request Message**

Field	Type	#	?	Meaning
GetVerifyResultsRequest			Y	Retrieves the verification results for the specified token
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentity	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	TokenType	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.

657 **Response Message**

Field	Type	#	?	Meaning
GetVerifyResultsResponse			Y	The response to a GetVerifyResults operation.
└ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
└ Return	ReturnCode	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	InformationType	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	Score	0..1	N	The score if the biometric information matched.

658 **4.2.5 Identify**

659 [IdentifyRequest](#)

660 [IdentifyResponse](#)

661 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.

667 **Request Message**

Field	Type	#	?	Meaning
IdentifyRequest			Y	Performs an identification function.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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: "Identify".
↳ ProcessingOptions	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed.
↳ Option	string	0..*	N	An option supported by the implementing system.
↳ InputData	InformationType	1	Y	Contains the input data for the aggregate services.

668 Response Message

Field	Type	#	?	Meaning
IdentifyResponse			Y	The response to an Identify operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	InformationType	0..1	N	Contains the output data for the response.

669 4.2.6 RetrieveInformation

670 [RetrieveInformationRequest](#)

671 [RetrieveInformationResponse](#)

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

677 Request Message

Field	Type	#	?	Meaning
RetrieveInformationRequest			Y	Retrieves requested information about a subject or encounter.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.

Field	Type	#	?	Meaning
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentity	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".
↳ ProcessingOptions	ProcessingOptionsType	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	BIASIdentity	1	Y	Includes the identifier of the subject or encounter.
↳ SubjectID	BIASIDType	0..1	C	A system unique identifier for a subject. Required if an Encounter ID is not provided.
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required if a Subject ID is not provided.

678 Response Message

Field	Type	#	?	Meaning
RetrieveInformationResponse			Y	Response to a RetrieveInformation operation.
↳ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	InformationType	0..1	N	Contains the output data for the response.

679 **4.2.7 Verify**

680 [VerifyRequest](#)

681 [VerifyResponse](#)

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

689 **Request Message**

Field	Type	#	?	Meaning
VerifyRequest			Y	Performs a 1:1 verification function.
└ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
└ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
└ ApplicationUser	ApplicationUserIdentifier	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	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed.
└ Option	string	0..*	N	An option supported by the implementing system.
└ InputData	InformationType	1	Y	Contains the input data for the aggregate services.
└ Identity	BIASIdentity	1	Y	Includes either the Identity Claim or Reference BIR.
└ IdentityClaim	BIASIDType	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.

Field	Type	#	?	Meaning
↳ BiometricData	BIASBiometricDataType	0..1	N	An Identity's biometric data.
↳ ReferenceBIR	CBEFF_BIR_Type	0..1	C	Maps to specific INCITS BIAS elements as required by that specification. Required if an Identity Claim is not provided.
↳ GalleryID	BIASIDType	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.

690 Response Message

Field	Type	#	?	Meaning
VerifyResponse			Y	The response to a Verify operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	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	InformationType	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	Score	0..1	N	The score if the biometric information matched.

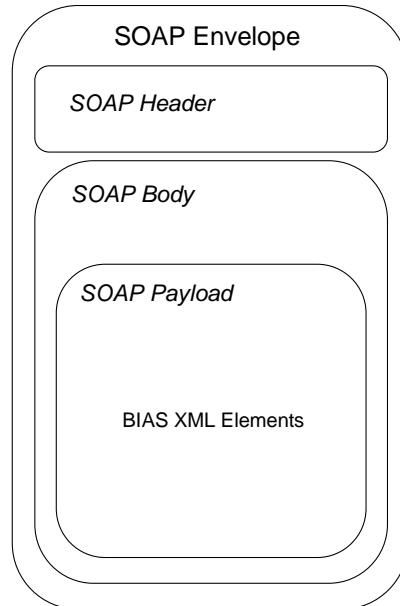
691

692 5 Message structure and rules

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

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

700



701
702 **Figure 4. BIAS SOAP Structure**
703

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

707 5.1 Purpose and constraints

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

- 711 • SOAP messages will carry BIAS XML [XML10] payloads.
- 712 • SOAP messages will follow WS-I and will deviate only when absolutely necessary.
- 713 • Message structures and interchanges will be kept as simple as possible – “nice to have”
714 features will be addressed in future revisions.
- 715 • XML schemas will be produced based on INCITS 442.
- 716 • BIAS will support a broad range of application domains.
- 717 • BIAS will allow for a variety of biometric and biographic data formats to be used
- 718 • Only the SOAP messaging will be defined – no message protocols or client/server agents
719 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:

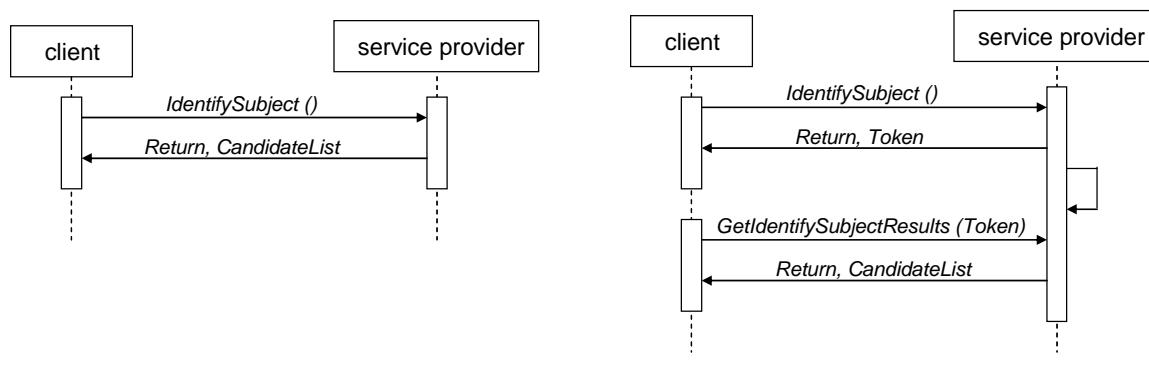


Figure 5. Example of Synchronous and Asynchronous BIAS Operations

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

758 interchange is a simple request-response pattern, adding additional items such as error
759 conditions would needlessly complicate the protocol.

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

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

767 **5.3 Handling binary data**

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

- 770 • Embedded Base64 encoding
- 771 • XOP [XOP]

772 Use of SOAP with Attachments (SWA) is deprecated.

773 **5.3.1 Base64 encoding**

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

777 As an example, the CBEFF_BIR_Type includes, as one of the BIR types, BinaryBIR of type
778 base64binary.

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

780 However, even an XML_BIR as defined within [CBEFF] Part 3, contains a biometric data block (BDB)
781 which may be entirely binary (most common),

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

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

784 **5.3.2 Use of XOP**

785 When XOP is used, the binary content is replaced with a reference (URI) to an attachment (i.e., MIME)
786 which contains that "stripped" content via an xop:include. The advantage of this method is overall
787 message size during transmission since the overhead of the embedded Base64 is not present (since the
788 MIME attachment contains the native binary format).

789 Use of XOP is generally transparent to the developer, other than in how they configure their toolset. Most
790 frameworks support this; however, there is a possibility of mismatch if the transmitter supports and uses
791 XOP but the receiver does not.

792 **5.4 Discovery**

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

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

802 **5.5 Identifying operations**

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

807

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

809 **5.5.1 Operation name element**

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

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

816

```
817 POST /bias HTTP/1.1
818 Host: www.acme.com
819 Content-Type: application/soap+xml; charset="utf-8"
820 Content-Length: nnnn
821 SOAPAction: ""
822 <?xml version="1.0"?>
823 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
824   <soap:Body>
825     <tns:CreateSubjectRequest
826       xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
827       <tns:GenericRequestParameters>
828         <tns:Application>BIAS Application</tns:Application>
829         <tns:ApplicationUser>BIAS User</tns:ApplicationUser>
830         <tns:BIASOperationName>CreateSubject</tns:BIASOperationName>
831     </tns:GenericRequestParameters>
832     <tns:Identity>
833       <tns:SubjectID>123456789</tns:SubjectID>
834     </tns:Identity>
835   </tns:CreateSubjectRequest>
836 </soap:Body>
837 </soap:Envelope>
```

839 **5.5.2 WS-Addressing Action**

840 WS-Addressing [[WS-Addr](#)] provides a mechanism for including action information inside any SOAP
841 message. The information is in the SOAP Header. The WS-Addressing 'Action' element is used to
842 indicate the intent of the message. The value is a URI/IRI identifying that intent; however, there are no
843 restrictions on the format or specificity of the URI/IRI nor a requirement that it can be resolved. Adoption
844 of this option also requires that the WS-Addressing 'To', 'ReplyTo', and 'MessageID' elements are
845 supplied, as they are mandatory elements in a request-reply message pattern as used within BIAS.

846 Response messages would also need to use WS-Addressing, requiring the ‘To’ (matching the ‘ReplyTo’
847 element in the request), ‘RelatesTo’ (matching the ‘MessageID’ element in the request), and
848 ‘RelationshipType’ (default value to “wsa:Reply”) elements.

849 Use of WS-Addressing is OPTIONAL in this profile as is this method of using the ‘Action’ field for this
850 purpose. However, when BIAS is used within an environment using WS-Addressing, it is
851 RECOMMENDED that this approach for use of the ‘Action’ field to carry the BIAS operation name is
852 employed, either alone or in combination with the BIASOperationName approach described in section
853 5.4.1.

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

855

```
856     POST /bias HTTP/1.1
857     Host: www.acme.com
858     Content-Type: application/soap+xml; charset="utf-8"
859     Content-Length: nnnn
860     SOAPAction: ""
861
862     <?xml version="1.0"?>
863     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
864             xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addr
865             essing">
866         <soap:Header>
867             <wsa:MessageID>some-ID</wsa:MessageID>
868             <wsa:ReplyTo>
869                 <wsa:Address>response-URI</wsa:Address>
870             </wsa:ReplyTo>
871             <wsa:To>destination-URI</wsa:To>
872             <wsa:Action>CreateSubject</wsa:Action>
873         </soap:Header>
874         <soap:Body>
875             <tns>CreateSubjectRequest
876             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/" />
877         </soap:Body>
878     </soap:Envelope>
```

879

880 5.6 Security

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

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

887 5.6.1 Use of SSL 3.0 or TLS 1.0

888 Unless otherwise specified, in any BIAS binding’s use of SSL 3.0 [SSL3] or TLS1.0 [RFC2246], servers
889 MUST authenticate clients using a X.509 v3 certificate [X509]. The client MUST establish server identity

890 based on contents of the certificate (typically through examination of the certificate's subject DN field,
891 subjectAltName attribute, etc.).
892 Use of transport level security in the form of SSL or TLS is OPTIONAL but highly RECOMMENDED. Use
893 of these mechanisms alone may not be sufficient for end-to-end integrity and confidentiality, however
894 (see 5.5.3 and 5.5.4 below).

895 **5.6.2 Data Origin Authentication**

896 Authentication of both the BIAS requester and the BIAS responder associated with a message is
897 OPTIONAL and depends on the environment of use: Authentication mechanisms available at the SOAP
898 message exchange layer or from the underlying substrate protocol (for example, in many bindings the
899 SSL/TLS or HTTP protocol) MAY be utilized to provide data origin authentication.
900 Transport authentication will not meet end-to-end origin authentication requirements in bindings where
901 the BIAS SOAP message passes through an intermediary – in this case, message authentication is
902 RECOMMENDED.
903 Note that SAML [[SAML](#)] MAY be used as the mechanism for parties to authenticate to one another.

904 **5.6.3 Message Integrity**

905 Message integrity of both BIAS requests and BIAS responses is OPTIONAL and depends on the
906 environment of use. The security layer in the underlying substrate protocol or a mechanism at the SOAP
907 message exchange layer MAY be used to ensure message integrity.
908 Transport integrity will not meet end-to-end integrity requirements in bindings where the BIAS SOAP
909 message passes through an intermediary – in this case, message integrity is RECOMMENDED.

910 **5.6.4 Message Confidentiality**

911 Message confidentiality of both BIAS requests and BIAS responses is OPTIONAL and depends on the
912 environment of use. The security layer in the underlying substrate protocol or a mechanism at the SOAP
913 message exchange layer MAY be used to ensure message confidentiality.
914 Transport confidentiality will not meet end-to-end confidentiality requirements in bindings where the BIAS
915 SOAP message passes through an intermediary.
916 NOTE: Biometric and biographic data is likely to contain personal information the confidentiality of
917 which SHOULD be protected accordingly. See INCITS 442, section 6.3 for further discussion.

918 **5.6.5 CBEFF BIR security features**

919 Within BIAS, biometric data is transferred within a CBEFF BIR (as defined in ISO/IEC 19785-1). CBEFF
920 provides for the optional encryption of the Biometric Data Block (BDB) of the BIR and for the integrity of
921 the entire BIR. If implemented, this is indicated in the BIR header. The BIR structure defines an optional
922 Security Block which MAY contain a digital signature (or message authentication code), encryption
923 parameters (e.g., key name, algorithm, etc.), and/or other security related data. Such protections are
924 associated with an individual BIR and are separate from any other protections provided at the message
925 level.

926 **5.6.6 Security Considerations**

927 Before deployment, each combination of authentication, message integrity, and confidentiality
928 mechanisms SHOULD be analyzed for vulnerability in the context of the specific protocol exchange and
929 the deployment environment.
930 Special care should be given to the impact of possible caching on security.
931 IETF RFC 2617 [[RFC2617](#)] describes possible attacks in the HTTP environment when basic or message
932 digest authentication schemes are used.
933 Many of the security considerations identified in [[SAML SEC](#)] MAY also apply.

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

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

943 **5.6.7 Security of Stored Data**

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

947 **5.6.8 Key Management**

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

950 **5.7 Use with other WS* standards**

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

956 **5.8 Tailoring**

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

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

- 963 • Identification of a subset of BIAS operations to be used
- 964 • Specification of security features to be implemented (e.g., SSL, CBEFF BIR encryption, etc.)
- 965 • Choice of operation name identification method
- 966 • Choice of BIR type to be used (XML, non-XML, or URI)
- 967 • Further definition of aggregate operations
- 968 • Use (or not) of the encounter model
- 969 • Use (or not) of asynchronous operations
- 970 • Process sequences
- 971 • Implementation specific values (e.g., Transform oprerations/controls)

972 6 Error handling

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

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

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

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

983 6.1 BIAS operation return codes

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

988 6.2 SOAP fault codes

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

991 Every Web service (operation) described in the BIAS WSDL may result in a fault message that will be
992 returned in the response by the service provider in the event of an error. The fault message contains a
993 FaultCode element as defined by the SOAP 1.1 specification (see section 3.2.5). The fault message
994 MUST contain a Detail element in a common format, as described by the BIASFault element (see section
995 3.2.6).

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

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

1000 NOTE: See also section 5.2 for additional information on message returns and faults.

1001 7 Conformance

1002 Implementations claiming conformance to this standard, MUST implement, at a minimum, all mandatory
1003 requirements and provisions set forth in Clauses 3, 4, 5 and 6. If such implementations claim
1004 conformance to any OPTIONAL requirements and provisions stated in Clauses 3, 4, 5 and 6, these
1005 requirements and provisions MUST be implemented as set forth in these Clauses.

1006 INCITS 442 [[INCIS-BIAS](#)] (Annex A) specifies five BIAS conformance classes. For each class, a set of
1007 mandatory BIAS operations is identified in order for implementations (BIAS service providers) to claim
1008 conformance. These categories are:

- 1009 • Class 1: Full Primitive Services Implementation
- 1010 • Class 2: Full Aggregate Services Implementation
- 1011 • Class 3: Limited Primitive Services Implementation
- 1012 • Class 4: Minimum Primitive Services Implementation
- 1013 • Class 5: Minimum Aggregate Services Implementation

1014 In addition, the minimum capability information to be returned in response to a Query Capabilities request
1015 (the only mandatory BIAS operation across all 5 classes) is specified for each class.

1016 These conformance classes and their associated requirements apply to this BIAS SOAP Profile.

1017 There are no minimum set of operations required to be implemented by BIAS requesters; however, any
1018 operations implemented must conform to the requirements of Clauses 3 and 4 and those requirements
1019 within Clause 5 that are mandatory and are not specific to BIAS responders.

1020 Annex A. XML Schema

```
1021 <?xml version="1.0" encoding="UTF-8"?>
1022
1023 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1024   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
1025   xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:iso-iec19785-3-7="urn:oid:1.1.19785.0.257.1.7.0"
1026   xmlns:tns="http://docs.oasis-open.org/bias/ns/bias-1.0/" targetNamespace="http://docs.oasis-open.org/bias/ns/bias-
1027   1.0/">
1028   <wsdl:types>
1029     <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
1030       targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">
1031       <xsd:complexType name="AggregateRequestTemplate" abstract="true">
1032         <xsd:annotation>
1033           <xsd:documentation>Base template for BIAS aggregate service requests.</xsd:documentation>
1034         </xsd:annotation>
1035         <xsd:complexContent>
1036           <xsd:extension base="RequestTemplate">
1037             <xsd:sequence>
1038               <xsd:element name="ProcessingOptions" type="ProcessingOptionsType" minOccurs="1" maxOccurs="1">
1039                 <xsd:annotation>
1040                   <xsd:documentation>Options that guide how the aggregate service request is
1041 processed.</xsd:documentation>
1042                 </xsd:annotation>
1043               </xsd:element>
1044               <xsd:element name="InputData" type="InformationType" minOccurs="1" maxOccurs="1">
1045                 <xsd:annotation>
1046                   <xsd:documentation>Contains the input data for the aggregate service request.</xsd:documentation>
1047                 </xsd:annotation>
1048               </xsd:element>
1049             </xsd:sequence>
1050           </xsd:extension>
1051         </xsd:complexContent>
1052       </xsd:complexType>
1053       <xsd:complexType name="AggregateResponseTemplate" abstract="true">
1054         <xsd:annotation>
1055           <xsd:documentation>Base template for BIAS aggregate service responses.</xsd:documentation>
1056         </xsd:annotation>
1057         <xsd:complexContent>
1058           <xsd:extension base="tns:ResponseTemplate">
1059             <xsd:sequence>
1060               <xsd:element name="ReturnData" type="tns:InformationType" minOccurs="0" maxOccurs="1">
1061                 <xsd:annotation>
1062                   <xsd:documentation>Contains the output data for the aggregate service response.</xsd:documentation>
1063                 </xsd:annotation>
1064               </xsd:element>
1065             </xsd:sequence>
1066           </xsd:extension>
1067         </xsd:complexContent>
1068       </xsd:complexType>
1069       <xsd:simpleType name="ApplicationIdentifier">
```

```

1070    <xsd:annotation>
1071        <xsd:documentation>Identifies an application.</xsd:documentation>
1072    </xsd:annotation>
1073    <xsd:restriction base="xsd:string">
1074        <xsd:minLength value="1" />
1075        <xsd:maxLength value="255" />
1076    </xsd:restriction>
1077 </xsd:simpleType>
1078 <xsd:simpleType name="ApplicationUserIdentifier">
1079    <xsd:annotation>
1080        <xsd:documentation>Identifies an application user or instance.</xsd:documentation>
1081    </xsd:annotation>
1082    <xsd:restriction base="xsd:string">
1083        <xsd:minLength value="1" />
1084        <xsd:maxLength value="255" />
1085    </xsd:restriction>
1086 </xsd:simpleType>
1087 <xsd:complexType name="BIASBiometricDataType">
1088    <xsd:annotation>
1089        <xsd:documentation>Wraps the various BIAS biometric types.</xsd:documentation>
1090    </xsd:annotation>
1091    <xsd:sequence>
1092        <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="0" maxOccurs="1">
1093            <xsd:annotation>
1094                <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1095            </xsd:annotation>
1096        </xsd:element>
1097        <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1098            <xsd:annotation>
1099                <xsd:documentation>Contains biometric information in either a non-XML and an XML
1100 representation.</xsd:documentation>
1101            </xsd:annotation>
1102        </xsd:element>
1103        <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1104            <xsd:annotation>
1105                <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
1106 specification.</xsd:documentation>
1107            </xsd:annotation>
1108        </xsd:element>
1109        <xsd:element name="ReferenceBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1110            <xsd:annotation>
1111                <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
1112 specification.</xsd:documentation>
1113            </xsd:annotation>
1114        </xsd:element>
1115        <xsd:element name="BiometricDataList" type="tns:BiometricDataListType" minOccurs="0" maxOccurs="1">
1116            <xsd:annotation>
1117                <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1118            </xsd:annotation>
1119        </xsd:element>
1120    </xsd:sequence>
1121 </xsd:complexType>
1122 <xsd:simpleType name="BIASFaultCode">
```

```

1123    <xsd:annotation>
1124        <xsd:documentation> </xsd:documentation>
1125    </xsd:annotation>
1126    <xsd:restriction base="xsd:unsignedLong">
1127        <xsd:enumeration value="UNKNOWN_ERROR">
1128            <xsd:annotation>
1129                <xsd:documentation>The service failed for an unknown reason.</xsd:documentation>
1130            </xsd:annotation>
1131        </xsd:enumeration>
1132        <xsd:enumeration value="UNSUPPORTED_CAPABILITY">
1133            <xsd:annotation>
1134                <xsd:documentation>A requested capability is not supported by the service
1135 implementation.</xsd:documentation>
1136            </xsd:annotation>
1137        </xsd:enumeration>
1138        <xsd:enumeration value="INVALID_INPUT">
1139            <xsd:annotation>
1140                <xsd:documentation>The data in a service input parameter is invalid.</xsd:documentation>
1141            </xsd:annotation>
1142        </xsd:enumeration>
1143        <xsd:enumeration value="BIR_QUALITY_ERROR">
1144            <xsd:annotation>
1145                <xsd:documentation>Biometric sample quality is too poor for the service to succeed.</xsd:documentation>
1146            </xsd:annotation>
1147        </xsd:enumeration>
1148        <xsd:enumeration value="INVALID_BIR">
1149            <xsd:annotation>
1150                <xsd:documentation>The input BIR is empty or in an invalid or unrecognized format.</xsd:documentation>
1151            </xsd:annotation>
1152        </xsd:enumeration>
1153        <xsd:enumeration value="BIR_SIGNATURE_FAILURE">
1154            <xsd:annotation>
1155                <xsd:documentation>The service could not validate the signature, if used, on the input
1156 BIR.</xsd:documentation>
1157            </xsd:annotation>
1158        </xsd:enumeration>
1159        <xsd:enumeration value="BIR_DECRYPTION_FAILURE">
1160            <xsd:annotation>
1161                <xsd:documentation>The service could not decrypt an encrypted input BIR.</xsd:documentation>
1162            </xsd:annotation>
1163        </xsd:enumeration>
1164        <xsd:enumeration value="INVALID_ENCOUNTER_ID">
1165            <xsd:annotation>
1166                <xsd:documentation>The input encounter ID is empty or in an invalid format.</xsd:documentation>
1167            </xsd:annotation>
1168        </xsd:enumeration>
1169        <xsd:enumeration value="INVALID SUBJECT_ID">
1170            <xsd:annotation>
1171                <xsd:documentation>The input subject ID is empty or in an invalid format.</xsd:documentation>
1172            </xsd:annotation>
1173        </xsd:enumeration>
1174        <xsd:enumeration value="UNKNOWN SUBJECT">
1175            <xsd:annotation>

```

```

1176      <xsd:documentation>The subject referenced by the input subject ID does not exist.</xsd:documentation>
1177      </xsd:annotation>
1178  </xsd:enumeration>
1179  <xsd:enumeration value="UNKNOWN_GALLERY">
1180      <xsd:annotation>
1181          <xsd:documentation>The gallery referenced by the input gallery ID does not exist.</xsd:documentation>
1182      </xsd:annotation>
1183  </xsd:enumeration>
1184  <xsd:enumeration value="UNKNOWN_ENCOUNTER">
1185      <xsd:annotation>
1186          <xsd:documentation>The encounter referenced by the input encounter ID does not
1187 exist.</xsd:documentation>
1188      </xsd:annotation>
1189  </xsd:enumeration>
1190  <xsd:enumeration value="UNKNOWN_BIOGRAPHIC_FORMAT">
1191      <xsd:annotation>
1192          <xsd:documentation>The biographic data format is not known or not supported.</xsd:documentation>
1193      </xsd:annotation>
1194  </xsd:enumeration>
1195  <xsd:enumeration value="UNKNOWN_IDENTITY_CLAIM">
1196      <xsd:annotation>
1197          <xsd:documentation>The identity referenced by the input identity claim does not exist.</xsd:documentation>
1198      </xsd:annotation>
1199  </xsd:enumeration>
1200  <xsd:enumeration value="INVALID_IDENTITY_CLAIM">
1201      <xsd:annotation>
1202          <xsd:documentation>The identity claim requested is already in use.</xsd:documentation>
1203      </xsd:annotation>
1204  </xsd:enumeration>
1205  <xsd:enumeration value="NONEXISTANT_DATA">
1206      <xsd:annotation>
1207          <xsd:documentation>The data requested for deletion does not exist.</xsd:documentation>
1208      </xsd:annotation>
1209  </xsd:enumeration>
1210 </xsd:restriction>
1211 </xsd:simpleType>
1212 <xsd:complexType name="BIASFaultDetail">
1213     <xsd:annotation>
1214         <xsd:documentation>Defines the error information associated with a SOAP fault.</xsd:documentation>
1215     </xsd:annotation>
1216     <xsd:sequence>
1217         <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultType" type="tns:BIASFaultCode">
1218             <xsd:annotation>
1219                 <xsd:documentation>References an error code.</xsd:documentation>
1220             </xsd:annotation>
1221         </xsd:element>
1222         <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultMessage" type="xsd:string">
1223             <xsd:annotation>
1224                 <xsd:documentation>Provides an explanation of the fault.</xsd:documentation>
1225             </xsd:annotation>
1226         </xsd:element>
1227         <xsd:element minOccurs="0" maxOccurs="1" name="BIASFaultDescription" type="xsd:string">
1228             <xsd:annotation>

```

```

1229      <xsd:documentation>Provides detailed information about a BIAS fault, such as trace
1230 details.</xsd:documentation>
1231     </xsd:annotation>
1232   </xsd:element>
1233 </xsd:sequence>
1234 </xsd:complexType>
1235 <xsd:complexType name="BIASIdentity">
1236   <xsd:annotation>
1237     <xsd:documentation>
1238       Defines a single element for encapsulating the data associated
1239       with an Identity. Includes the Identity's reference identifiers,
1240       biographic data, and biometric data.
1241     </xsd:documentation>
1242   </xsd:annotation>
1243 <xsd:sequence>
1244   <xsd:element name="SubjectID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1245     <xsd:annotation>
1246       <xsd:documentation>A system unique identifier for a subject.</xsd:documentation>
1247     </xsd:annotation>
1248   </xsd:element>
1249   <xsd:element name="IdentityClaim" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1250     <xsd:annotation>
1251       <xsd:documentation>An identifier by which a subject is known to a particular gallery or population
1252 group.</xsd:documentation>
1253   </xsd:annotation>
1254 </xsd:element>
1255   <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1256     <xsd:annotation>
1257       <xsd:documentation>The identifier of an encounter associated with the subject, required for encounter-
1258 centric models.</xsd:documentation>
1259   </xsd:annotation>
1260 </xsd:element>
1261   <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
1262     <xsd:annotation>
1263       <xsd:documentation>A list of encounters associated with a subject.</xsd:documentation>
1264     </xsd:annotation>
1265   </xsd:element>
1266   <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1267     <xsd:annotation>
1268       <xsd:documentation>An Identity's biographic data.</xsd:documentation>
1269     </xsd:annotation>
1270   </xsd:element>
1271   <xsd:element name="BiographicDataElements" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1272     <xsd:annotation>
1273       <xsd:documentation>An Identity's biographic data elements that are stored in the implementing
1274 system.</xsd:documentation>
1275   </xsd:annotation>
1276 </xsd:element>
1277   <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="0" maxOccurs="1">
1278     <xsd:annotation>
1279       <xsd:documentation>An Identity's biometric data.</xsd:documentation>
1280     </xsd:annotation>
1281   </xsd:element>
```

```

1282      </xsd:sequence>
1283  </xsd:complexType>
1284  <xsd:simpleType name="BIASIDType">
1285    <xsd:annotation>
1286      <xsd:documentation>A BIAS identifier</xsd:documentation>
1287    </xsd:annotation>
1288    <xsd:restriction base="xsd:string" />
1289  </xsd:simpleType>
1290  <xsd:complexType name="BiographicDataItemType">
1291    <xsd:annotation>
1292      <xsd:documentation>Defines a single biographic data element.</xsd:documentation>
1293    </xsd:annotation>
1294    <xsd:sequence maxOccurs="unbounded">
1295      <xsd:element name="Name" type="xsd:string" minOccurs="1" maxOccurs="1">
1296        <xsd:annotation>
1297          <xsd:documentation>The name of the biographic data item.</xsd:documentation>
1298        </xsd:annotation>
1299      </xsd:element>
1300      <xsd:element name="Type" type="xsd:string" minOccurs="1" maxOccurs="1">
1301        <xsd:annotation>
1302          <xsd:documentation>The data type for the biographic data item.</xsd:documentation>
1303        </xsd:annotation>
1304      </xsd:element>
1305      <xsd:element name="Value" type="xsd:string" minOccurs="0" maxOccurs="1">
1306        <xsd:annotation>
1307          <xsd:documentation>The value assigned to the biographic data item.</xsd:documentation>
1308        </xsd:annotation>
1309      </xsd:element>
1310    </xsd:sequence>
1311  </xsd:complexType>
1312  <xsd:complexType name="BiographicDataSetType">
1313    <xsd:annotation>
1314      <xsd:documentation>Defines a set of biographic data that is formatted according to the specified
1315 format.</xsd:documentation>
1316    </xsd:annotation>
1317    <xsd:sequence>
1318      <xsd:any namespace="##any">
1319        <xsd:annotation>
1320          <xsd:documentation>Biographic data formatted according to a specific format.</xsd:documentation>
1321        </xsd:annotation>
1322      </xsd:any>
1323    </xsd:sequence>
1324    <xsd:attribute name="name" type="xsd:string" use="required">
1325      <xsd:annotation>
1326        <xsd:documentation>The name of the biographic data format. Use these names for common formats: FBI-EFTS,
1327 FBI-EBTS, DOD-EBTS, INT-I, NIEM, xNAL, HR-XML.</xsd:documentation>
1328      </xsd:annotation>
1329    </xsd:attribute>
1330    <xsd:attribute name="version" type="xsd:string" use="optional">
1331      <xsd:annotation>
1332        <xsd:documentation>The version of the biographic data format (e.g., "7.1" for FBI-EFTS or "2.0" for
1333 NIEM).</xsd:documentation>
1334      </xsd:annotation>

```

```

1335    </xsd:attribute>
1336    <xsd:attribute name="source" type="xsd:string" use="required">
1337      <xsd:annotation>
1338        <xsd:documentation>Reference to a URI/IRI describing the biographic data format. For example: (FBI-EFTS)
1339        www.fbibiospecs.org, (DOD-EBTS) www.biometrics.dod.mil, (INT-I) www.interpol.int, (NIEM) www.niem.gov, (xNAL)
1340        www.oasis-open.org, (HR-XML) www.hr-xml.org.</xsd:documentation>
1341      </xsd:annotation>
1342    </xsd:attribute>
1343    <xsd:attribute name="type" type="xsd:string" use="required">
1344      <xsd:annotation>
1345        <xsd:documentation>The biographic data format type. Use these types for common formats: ASCII (e.g., for
1346        non-XML versions of FBI-EFTS, FBI-EBTS, DOD-EFTS, or INT-I), XML (e.g., for NIEM, xNAL, and HR-XML or future version
1347        of FBI-EBTS).</xsd:documentation>
1348      </xsd:annotation>
1349    </xsd:attribute>
1350  </xsd:complexType>
1351  <xsd:complexType name="BiographicDataType">
1352    <xsd:annotation>
1353      <xsd:documentation>
1354        Defines a set of biographic data elements, utilizing either the
1355        BiographicDataItemType to represent a list of elements or the
1356        BiographicDataSetType to represent a complete, formatted set of
1357        biographic information.
1358      </xsd:documentation>
1359    </xsd:annotation>
1360    <xsd:choice maxOccurs="unbounded">
1361      <xsd:element name="LastName" type="xsd:string" minOccurs="0" maxOccurs="1">
1362        <xsd:annotation>
1363          <xsd:documentation>The last name of a subject.</xsd:documentation>
1364        </xsd:annotation>
1365      </xsd:element>
1366      <xsd:element name="FirstName" type="xsd:string" minOccurs="0" maxOccurs="1">
1367        <xsd:annotation>
1368          <xsd:documentation>The first name of a subject.</xsd:documentation>
1369        </xsd:annotation>
1370      </xsd:element>
1371      <xsd:element name="BiographicDataItem" type="tns:BiographicDataItemType" maxOccurs="unbounded">
1372        <xsd:annotation>
1373          <xsd:documentation>A single biographic data element.</xsd:documentation>
1374        </xsd:annotation>
1375      </xsd:element>
1376      <xsd:element name="BiographicDataSet" type="tns:BiographicDataSetType" maxOccurs="1">
1377        <xsd:annotation>
1378          <xsd:documentation>A set of biographic data information.</xsd:documentation>
1379        </xsd:annotation>
1380      </xsd:element>
1381    </xsd:choice>
1382  </xsd:complexType>
1383  <xsd:complexType name="BiometricDataElementType">
1384    <xsd:annotation>
1385      <xsd:documentation>
1386        Provides descriptive information about biometric data, such as
1387        the biometric type, subtype, and format, contained in the BDB of

```

```

1388      the CBEFF-BIR.
1389      </xsd:documentation>
1390    </xsd:annotation>
1391    <xsd:sequence>
1392      <xsd:element name="BiometricType" type="iso-iec19785-3-7: Multiple-types" minOccurs="1" maxOccurs="1">
1393        <xsd:annotation>
1394          <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1395          CBEFF.</xsd:documentation>
1396        </xsd:annotation>
1397      </xsd:element>
1398      <xsd:element name="BiometricTypeCount" type="xsd:positiveInteger" minOccurs="0" maxOccurs="1">
1399        <xsd:annotation>
1400          <xsd:documentation>The number of biometric records having the biometric type recorded in the biometric
1401          type field.</xsd:documentation>
1402        </xsd:annotation>
1403      </xsd:element>
1404      <xsd:element name="BiometricSubType" type="iso-iec19785-3-7: Subtype" minOccurs="0" maxOccurs="1">
1405        <xsd:annotation>
1406          <xsd:documentation>More specifically defines the type of biometric data stored in the biometric record, as
1407          defined by CBEFF.</xsd:documentation>
1408        </xsd:annotation>
1409      </xsd:element>
1410      <xsd:element name="BDBFormatOwner" type="iso-iec19785-3-7: Registered-int" minOccurs="1"
1411      maxOccurs="1">
1412        <xsd:annotation>
1413          <xsd:documentation>Identifies the standards body, working group, industry consortium, or other CBEFF
1414          biometric organization that has defined the format for the biometric data.</xsd:documentation>
1415        </xsd:annotation>
1416      </xsd:element>
1417      <xsd:element name="BDBFormatType" type="iso-iec19785-3-7: Registered-int" minOccurs="1" maxOccurs="1">
1418        <xsd:annotation>
1419          <xsd:documentation>Identifies the specific biometric data format specified by the CBEFF biometric
1420          organization recorded in the BDB Format Owner field.</xsd:documentation>
1421        </xsd:annotation>
1422      </xsd:element>
1423    </xsd:sequence>
1424  </xsd:complexType>
1425  <xsd:complexType name="BiometricDataListType">
1426    <xsd:annotation>
1427      <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1428    </xsd:annotation>
1429    <xsd:sequence>
1430      <xsd:element name="BiometricDataElement" type="tns:BiometricDataElementType" minOccurs="0"
1431      maxOccurs="unbounded">
1432        <xsd:annotation>
1433          <xsd:documentation>Data structure containing information about a biometric record.</xsd:documentation>
1434        </xsd:annotation>
1435      </xsd:element>
1436    </xsd:sequence>
1437  </xsd:complexType>
1438  <xsd:complexType name="CandidateListType">
1439    <xsd:annotation>
1440      <xsd:documentation>
```

```

1441     Defines a set of candidates, utilizing the Candidate Type to
1442     represent each element in the set.
1443     </xsd:documentation>
1444   </xsd:annotation>
1445   <xsd:sequence>
1446     <xsd:element name="Candidate" type="tns:CandidateType" minOccurs="0" maxOccurs="unbounded">
1447       <xsd:annotation>
1448         <xsd:documentation>A single candidate.</xsd:documentation>
1449       </xsd:annotation>
1450     </xsd:element>
1451   </xsd:sequence>
1452 </xsd:complexType>
1453 <xsd:complexType name="CandidateType">
1454   <xsd:annotation>
1455     <xsd:documentation>
1456       Defines a single candidate as a possible match in response to a
1457       biometric identification request.
1458     </xsd:documentation>
1459   </xsd:annotation>
1460   <xsd:sequence>
1461     <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
1462       <xsd:annotation>
1463         <xsd:documentation>The match score.</xsd:documentation>
1464       </xsd:annotation>
1465     </xsd:element>
1466     <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1467       <xsd:annotation>
1468         <xsd:documentation>Biographic data associated with the candidate match.</xsd:documentation>
1469       </xsd:annotation>
1470     </xsd:element>
1471     <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="1" maxOccurs="1">
1472       <xsd:annotation>
1473         <xsd:documentation>Biometric data associated with the candidate match.</xsd:documentation>
1474       </xsd:annotation>
1475     </xsd:element>
1476   </xsd:sequence>
1477   <xsd:attribute name="rank" type="xsd:integer" use="required">
1478     <xsd:annotation>
1479       <xsd:documentation>The rank of the candidate in relation to other candidates for the same biometric
1480       identification operation.</xsd:documentation>
1481     </xsd:annotation>
1482   </xsd:attribute>
1483 </xsd:complexType>
1484 <xsd:complexType name="CapabilityListType">
1485   <xsd:annotation>
1486     <xsd:documentation>Defines a set of capabilities.</xsd:documentation>
1487   </xsd:annotation>
1488   <xsd:sequence>
1489     <xsd:element name="Capability" type="tns:CapabilityType" minOccurs="0" maxOccurs="unbounded">
1490       <xsd:annotation>
1491         <xsd:documentation>A single capability.</xsd:documentation>
1492       </xsd:annotation>
1493     </xsd:element>

```

```

1494    </xsd:sequence>
1495  </xsd:complexType>
1496  <xsd:simpleType name="CapabilityName">
1497    <xsd:annotation>
1498      <xsd:documentation>A list of capability items.</xsd:documentation>
1499    </xsd:annotation>
1500    <xsd:restriction base="xsd:string">
1501      <xsd:enumeration value="AggregateInputDataOptional">
1502        <xsd:annotation>
1503          <xsd:documentation>A data element accepted as optional input by the implementing system for the
1504 aggregate services.</xsd:documentation>
1505        </xsd:annotation>
1506      </xsd:enumeration>
1507      <xsd:enumeration value="AggregateInputDataRequired">
1508        <xsd:annotation>
1509          <xsd:documentation>A data element required as input by the implementing system for the aggregate
1510 services.</xsd:documentation>
1511        </xsd:annotation>
1512      </xsd:enumeration>
1513      <xsd:enumeration value="AggregateProcessingOption">
1514        <xsd:annotation>
1515          <xsd:documentation>A processing option supported by the implementing system for the aggregate
1516 services.</xsd:documentation>
1517        </xsd:annotation>
1518      </xsd:enumeration>
1519      <xsd:enumeration value="AggregateReturnData">
1520        <xsd:annotation>
1521          <xsd:documentation>A data element returned by the implementing system for the aggregate
1522 services.</xsd:documentation>
1523        </xsd:annotation>
1524      </xsd:enumeration>
1525      <xsd:enumeration value="AggregateServiceDescription">
1526        <xsd:annotation>
1527          <xsd:documentation>Describes the processing logic of an aggregate service supported by the implementing
1528 system.</xsd:documentation>
1529        </xsd:annotation>
1530      </xsd:enumeration>
1531      <xsd:enumeration value="BiographicDataSet">
1532        <xsd:annotation>
1533          <xsd:documentation>Identifies a biographic data set supported by the implementing
1534 system.</xsd:documentation>
1535        </xsd:annotation>
1536      </xsd:enumeration>
1537      <xsd:enumeration value="CBEFFPatronFormat">
1538        <xsd:annotation>
1539          <xsd:documentation>A patron format supported by the implementing system.</xsd:documentation>
1540        </xsd:annotation>
1541      </xsd:enumeration>
1542      <xsd:enumeration value="ClassificationAlgorithmType">
1543        <xsd:annotation>
1544          <xsd:documentation>A classification algorithm type supported by the implementing
1545 system.</xsd:documentation>
1546        </xsd:annotation>

```

```

1547      </xsd:enumeration>
1548      <xsd:enumeration value="ConformanceClass">
1549          <xsd:annotation>
1550              <xsd:documentation>Identifies the conformance class of the BIAS implementation.</xsd:documentation>
1551          </xsd:annotation>
1552      </xsd:enumeration>
1553      <xsd:enumeration value="Gallery">
1554          <xsd:annotation>
1555              <xsd:documentation>A gallery or population group supported by the implementing
1556 system.</xsd:documentation>
1557          </xsd:annotation>
1558      </xsd:enumeration>
1559      <xsd:enumeration value="IdentityModel">
1560          <xsd:annotation>
1561              <xsd:documentation>Identifies whether the implementing system is person-centric or encounter-centric
1562 based.</xsd:documentation>
1563          </xsd:annotation>
1564      </xsd:enumeration>
1565      <xsd:enumeration value="MatchScore">
1566          <xsd:annotation>
1567              <xsd:documentation> Identifies the use of match scores returned by the implementing
1568 system.</xsd:documentation>
1569          </xsd:annotation>
1570      </xsd:enumeration>
1571      <xsd:enumeration value="QualityAlgorithm">
1572          <xsd:annotation>
1573              <xsd:documentation>A quality algorithm vendor and algorithm vendor product ID supported by the
1574 implementing system.</xsd:documentation>
1575          </xsd:annotation>
1576      </xsd:enumeration>
1577      <xsd:enumeration value="SupportedBiometric">
1578          <xsd:annotation>
1579              <xsd:documentation>A biometric type supported by the implementing system.</xsd:documentation>
1580          </xsd:annotation>
1581      </xsd:enumeration>
1582      <xsd:enumeration value="TransformOperation">
1583          <xsd:annotation>
1584              <xsd:documentation>A transform operation type supported by the implementing
1585 system.</xsd:documentation>
1586          </xsd:annotation>
1587      </xsd:enumeration>
1588      </xsd:restriction>
1589  </xsd:simpleType>
1590  <xsd:complexType name="CapabilityType">
1591      <xsd:annotation>
1592          <xsd:documentation>Defines a single capability supported by an implementing system.</xsd:documentation>
1593      </xsd:annotation>
1594      <xsd:sequence>
1595          <xsd:element name="CapabilityName" type="tns:CapabilityName" minOccurs="1" maxOccurs="1">
1596              <xsd:annotation>
1597                  <xsd:documentation>The name of the capability.</xsd:documentation>
1598              </xsd:annotation>
1599          </xsd:element>

```

```

1600    <xsd:element name="CapabilityID" type="xsd:string" minOccurs="0" maxOccurs="1">
1601        <xsd:annotation>
1602            <xsd:documentation>An identifier assigned to the capability by the implementing
1603        system.</xsd:documentation>
1604    </xsd:annotation>
1605    </xsd:element>
1606    <xsd:element name="CapabilityDescription" type="xsd:string" minOccurs="0" maxOccurs="1">
1607        <xsd:annotation>
1608            <xsd:documentation>A description of the capability.</xsd:documentation>
1609        </xsd:annotation>
1610    </xsd:element>
1611    <xsd:element name="CapabilityValue" type="xsd:string" minOccurs="0" maxOccurs="1">
1612        <xsd:annotation>
1613            <xsd:documentation>A value assigned to the capability.</xsd:documentation>
1614        </xsd:annotation>
1615    </xsd:element>
1616    <xsd:element name="CapabilitySupportingValue" type="xsd:string" minOccurs="0" maxOccurs="1">
1617        <xsd:annotation>
1618            <xsd:documentation>A secondary value supporting the capability.</xsd:documentation>
1619        </xsd:annotation>
1620    </xsd:element>
1621    <xsd:element name="CapabilityAdditionalInfo" type="xsd:string" minOccurs="0" maxOccurs="1">
1622        <xsd:annotation>
1623            <xsd:documentation> Contains additional information for the supported capability.</xsd:documentation>
1624        </xsd:annotation>
1625    </xsd:element>
1626    </xsd:sequence>
1627 </xsd:complexType>
1628 <xsd:complexType name="CBEFF_BIR_ListType">
1629     <xsd:annotation>
1630         <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1631     </xsd:annotation>
1632     <xsd:sequence>
1633         <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="unbounded">
1634             <xsd:annotation>
1635                 <xsd:documentation>CBEFF structure containing information about a biometric sample.</xsd:documentation>
1636             </xsd:annotation>
1637         </xsd:element>
1638     </xsd:sequence>
1639 </xsd:complexType>
1640 <xsd:complexType name="CBEFF_BIR_Type">
1641     <xsd:annotation>
1642         <xsd:documentation>Represents biometric information, with either a non-XML or XML
1643 representation.</xsd:documentation>
1644     </xsd:annotation>
1645     <xsd:sequence>
1646         <xsd:element name="BIR_Information" minOccurs="0" maxOccurs="1">
1647             <xsd:complexType>
1648                 <xsd:sequence>
1649                     <xsd:element name="bir-info" type="iso-iec19785-3-7:BIR-info" minOccurs="0" maxOccurs="1"/>
1650                     <xsd:element name="bdb-info" type="iso-iec19785-3-7:BDB-info" minOccurs="0" maxOccurs="1"/>
1651                     <xsd:element name="sb-info" type="iso-iec19785-3-7:SB-info" minOccurs="0" maxOccurs="1"/>
1652                 </xsd:sequence>

```

```

1653      </xsd:complexType>
1654  </xsd:element>
1655  <xsd:element name="BIR" minOccurs="1" maxOccurs="1">
1656    <xsd:complexType>
1657      <xsd:choice minOccurs="1" maxOccurs="1">
1658        <xsd:element name="BinaryBIR" type="xs:base64Binary" />
1659        <xsd:element name="URL_BIR" type="xs:anyURI" />
1660        <xsd:element name="XML_BIR" type="iso-iec19785-3-7:BIR" />
1661      </xsd:choice>
1662    </xsd:complexType>
1663  </xsd:element>
1664 </xsd:sequence>
1665  <xsd:attribute name="format-owner" type="iso-iec19785-3-7:Registered-int" use="required" />
1666  <xsd:attribute name="format-type" type="iso-iec19785-3-7:Registered-int" use="required" />
1667 </xsd:complexType>
1668 <xsd:simpleType name="Classification">
1669   <xsd:annotation>
1670     <xsd:documentation>The result of a classification.</xsd:documentation>
1671   </xsd:annotation>
1672   <xsd:restriction base="xsd:string" />
1673 </xsd:simpleType>
1674 <xsd:simpleType name="ClassificationAlgorithmType">
1675   <xsd:annotation>
1676     <xsd:documentation>Type of classification algorithm that was used to perform the
1677 classification.</xsd:documentation>
1678   </xsd:annotation>
1679   <xsd:restriction base="xsd:string" />
1680 </xsd:simpleType>
1681 <xsd:complexType name="ClassificationData">
1682   <xsd:annotation>
1683     <xsd:documentation>Contains information on classification results and the algorithm used to determine the
1684 classification.</xsd:documentation>
1685   </xsd:annotation>
1686   <xsd:sequence>
1687     <xsd:element name="Classification" type="tns:Classification" minOccurs="1" maxOccurs="1">
1688       <xsd:annotation>
1689         <xsd:documentation>The result of the classification.</xsd:documentation>
1690       </xsd:annotation>
1691     </xsd:element>
1692     <xsd:element name="ClassificationAlgorithmType" type="tns:ClassificationAlgorithmType" minOccurs="1"
1693 maxOccurs="1">
1694       <xsd:annotation>
1695         <xsd:documentation>Identifies the type of classification algorithm that was used to perform the
1696 classification.</xsd:documentation>
1697       </xsd:annotation>
1698     </xsd:element>
1699   </xsd:sequence>
1700 </xsd:complexType>
1701 <xsd:complexType name="EncounterListType">
1702   <xsd:annotation>
1703     <xsd:documentation>Defines a set of encounters.</xsd:documentation>
1704   </xsd:annotation>
1705   <xsd:sequence>

```

```

1706    <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="unbounded">
1707        <xsd:annotation>
1708            <xsd:documentation>The identifier of an encounter.</xsd:documentation>
1709        </xsd:annotation>
1710    </xsd:element>
1711    </xsd:sequence>
1712 </xsd:complexType>
1713 <xsd:complexType name="FusionInformationListType">
1714     <xsd:annotation>
1715         <xsd:documentation>
1716             Contains at a minimum two sets of fusion input
1717             elements, as input to the PerformFusion request.
1718         </xsd:documentation>
1719     </xsd:annotation>
1720     <xsd:sequence>
1721         <xsd:element name="FusionElement" type="tns:FusionInformationType" minOccurs="2"
1722 maxOccurs="unbounded">
1723             <xsd:annotation>
1724                 <xsd:documentation>A set of fusion information.</xsd:documentation>
1725             </xsd:annotation>
1726         </xsd:element>
1727     </xsd:sequence>
1728 </xsd:complexType>
1729 <xsd:complexType name="FusionInformationType">
1730     <xsd:annotation>
1731         <xsd:documentation>Represents the information necessary to perform a fusion
1732 operation.</xsd:documentation>
1733     </xsd:annotation>
1734     <xsd:sequence>
1735         <xsd:element name="BiometricType" type="iso-iec19785-3-7:Multiple-types" minOccurs="1" maxOccurs="1">
1736             <xsd:annotation>
1737                 <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1738 CBEFF.</xsd:documentation>
1739             </xsd:annotation>
1740         </xsd:element>
1741         <xsd:element name="BiometricSubType" type="iso-iec19785-3-7:Subtype" minOccurs="0" maxOccurs="1">
1742             <xsd:annotation>
1743                 <xsd:documentation>More specifically defines the type of biometric data stored in the biometric
1744 record.</xsd:documentation>
1745             </xsd:annotation>
1746         </xsd:element>
1747         <xsd:element name="AlgorithmOwner" type="xsd:string" minOccurs="1" maxOccurs="1">
1748             <xsd:annotation>
1749                 <xsd:documentation>The owner or vendor of the algorithm used to determine the score or
1750 decision.</xsd:documentation>
1751             </xsd:annotation>
1752         </xsd:element>
1753         <xsd:element name="AlgorithmType" type="xsd:string" minOccurs="1" maxOccurs="1">
1754             <xsd:annotation>
1755                 <xsd:documentation>The Algorithm Owner's identifier for the specific algorithm product and version used to
1756 determine the score or decision.</xsd:documentation>
1757             </xsd:annotation>
1758         </xsd:element>

```

```

1759 <xsd:choice minOccurs="1" maxOccurs="1">
1760   <xsd:element name="Score" type="tns:Score">
1761     <xsd:annotation>
1762       <xsd:documentation>The similarity score assigned by the matching algorithm.</xsd:documentation>
1763     </xsd:annotation>
1764   </xsd:element>
1765   <xsd:element name="Decision" type="xsd:string">
1766     <xsd:annotation>
1767       <xsd:documentation>The match decision assigned by the matching algorithm.</xsd:documentation>
1768     </xsd:annotation>
1769   </xsd:element>
1770 </xsd:choice>
1771 </xsd:sequence>
1772 </xsd:complexType>
1773 <xsd:complexType name="GenericRequestParameters">
1774   <xsd:annotation>
1775     <xsd:documentation>Common request paramters that can be used to identify the
1776 requester.</xsd:documentation>
1777   </xsd:annotation>
1778   <xsd:sequence>
1779     <xsd:element name="Application" type="tns:ApplicationIdentifier" minOccurs="0" maxOccurs="1">
1780       <xsd:annotation>
1781         <xsd:documentation>Identifies the requesting application.</xsd:documentation>
1782       </xsd:annotation>
1783     </xsd:element>
1784     <xsd:element name="ApplicationUser" type="tns:ApplicationUserIdentity" minOccurs="0" maxOccurs="1">
1785       <xsd:annotation>
1786         <xsd:documentation>Identifiers the user or instance of the requesting application.</xsd:documentation>
1787       </xsd:annotation>
1788     </xsd:element>
1789     <xsd:element name="BIASOperationName" type="xsd:string" minOccurs="0" maxOccurs="1">
1790       <xsd:annotation>
1791         <xsd:documentation>Identifiers the BIAS operation name that is being requested.</xsd:documentation>
1792       </xsd:annotation>
1793     </xsd:element>
1794   </xsd:sequence>
1795 </xsd:complexType>
1796 <xsd:complexType name="InformationType">
1797   <xsd:annotation>
1798     <xsd:documentation>
1799       Allows for an unlimited number of data element types, and it does
1800       not specify nor require any particular data element.
1801     </xsd:documentation>
1802   </xsd:annotation>
1803   <xsd:sequence>
1804     <xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
1805   </xsd:sequence>
1806 </xsd:complexType>
1807 <xsd:complexType name="ListFilterType">
1808   <xsd:annotation>
1809     <xsd:documentation>
1810       Provides a method to filter the amount of information returned in
1811       a search of biometric data.

```

```

1812      </xsd:documentation>
1813    </xsd:annotation>
1814    <xsd:sequence>
1815      <xsd:element name="BiometricTypeFilter" type="iso-iec19785-3-7:Multiple-types" minOccurs="1"
1816      maxOccurs="unbounded">
1817        <xsd:annotation>
1818          <xsd:documentation>Limits the returned information to a specific type of biometric, as defined by
1819          CBEFF.</xsd:documentation>
1820        </xsd:annotation>
1821      </xsd:element>
1822      <xsd:element name="IncludeBiometricSubtype" type="xsd:boolean" minOccurs="1" maxOccurs="1">
1823        <xsd:annotation>
1824          <xsd:documentation>A Boolean flag indicating if biometric subtype information should be
1825          returned.</xsd:documentation>
1826        </xsd:annotation>
1827      </xsd:element>
1828    </xsd:sequence>
1829  </xsd:complexType>
1830  <xsd:simpleType name="MatchType">
1831    <xsd:annotation>
1832      <xsd:documentation>The result of a fusion method.</xsd:documentation>
1833    </xsd:annotation>
1834    <xsd:restriction base="xsd:boolean" />
1835  </xsd:simpleType>
1836  <xsd:complexType name="ProcessingOptionsType">
1837    <xsd:annotation>
1838      <xsd:documentation>
1839        BIAS aggregate services support the ability to include various
1840        processing options which direct and possibly control the business
1841        logic for that service. The ProcessingOptionsType provides a
1842        method to represent those options. Processing options should be
1843        defined by the implementing system.
1844      </xsd:documentation>
1845    </xsd:annotation>
1846    <xsd:sequence>
1847      <xsd:element name="Option" type="xsd:string" minOccurs="0" maxOccurs="unbounded">
1848        <xsd:annotation>
1849          <xsd:documentation>An option supported by the implementing system.</xsd:documentation>
1850        </xsd:annotation>
1851      </xsd:element>
1852    </xsd:sequence>
1853  </xsd:complexType>
1854  <xsd:simpleType name="ProductID">
1855    <xsd:annotation>
1856      <xsd:documentation>The vendor's ID for a particular product.</xsd:documentation>
1857    </xsd:annotation>
1858    <xsd:restriction base="xsd:string" />
1859  </xsd:simpleType>
1860  <xsd:complexType name="QualityData">
1861    <xsd:annotation>
1862      <xsd:documentation>Contains information about a biometric sample's quality and the algorithm used to
1863      compute the quality.</xsd:documentation>
1864    </xsd:annotation>

```

```

1865    <xsd:sequence>
1866        <xsd:element name="QualityScore" type="iso-iec19785-3-7:Quality" minOccurs="0" maxOccurs="1">
1867            <xsd:annotation>
1868                <xsd:documentation>The quality of a biometric sample.</xsd:documentation>
1869            </xsd:annotation>
1870        </xsd:element>
1871        <xsd:element name="AlgorithmVendor" type="tns:VendorIdentifier" minOccurs="1" maxOccurs="1">
1872            <xsd:annotation>
1873                <xsd:documentation>The vendor of the quality algorithm used to determine the quality
1874 score.</xsd:documentation>
1875            </xsd:annotation>
1876        </xsd:element>
1877        <xsd:element name="AlgorithmVendorProductID" type="tns:ProductID" minOccurs="1" maxOccurs="1">
1878            <xsd:annotation>
1879                <xsd:documentation>The vendor's ID for the algorithm used to determine the quality.</xsd:documentation>
1880            </xsd:annotation>
1881        </xsd:element>
1882        <xsd:element name="AlgorithmVersion" type="tns:VersionType" minOccurs="0" maxOccurs="1">
1883            <xsd:annotation>
1884                <xsd:documentation>The version of the algorithm used to determine the quality.</xsd:documentation>
1885            </xsd:annotation>
1886        </xsd:element>
1887    </xsd:sequence>
1888 </xsd:complexType>
1889 <xsd:complexType name="RequestTemplate" abstract="true">
1890     <xsd:annotation>
1891         <xsd:documentation>Base template for BIAS primitive service requests.</xsd:documentation>
1892     </xsd:annotation>
1893     <xsd:sequence>
1894         <xsd:element name="GenericRequestParameters" type="tns:GenericRequestParameters" minOccurs="0"
1895 maxOccurs="1" />
1896     </xsd:sequence>
1897 </xsd:complexType>
1898 <xsd:complexType name="ResponseStatus">
1899     <xsd:sequence>
1900         <xsd:element name="Return" type="tns:ReturnCode" minOccurs="1" maxOccurs="1">
1901             <xsd:annotation>
1902                 <xsd:documentation>The return code indicates the return status of the operation.</xsd:documentation>
1903             </xsd:annotation>
1904         </xsd:element>
1905         <xsd:element name="Message" type="xsd:string" minOccurs="0" maxOccurs="1">
1906             <xsd:annotation>
1907                 <xsd:documentation>A short message corresponding to the return code.</xsd:documentation>
1908             </xsd:annotation>
1909         </xsd:element>
1910     </xsd:sequence>
1911 </xsd:complexType>
1912 <xsd:complexType name="ResponseTemplate" abstract="true">
1913     <xsd:annotation>
1914         <xsd:documentation>Base template for BIAS responses.</xsd:documentation>
1915     </xsd:annotation>
1916     <xsd:sequence>
1917         <xsd:element name="ResponseStatus" type="tns:ResponseStatus" minOccurs="1" maxOccurs="1">

```

```

1918      <xsd:annotation>
1919          <xsd:documentation>Returned status for the operation.</xsd:documentation>
1920      </xsd:annotation>
1921  </xsd:element>
1922 </xsd:sequence>
1923 </xsd:complexType>
1924 <xsd:simpleType name="ReturnCode">
1925     <xsd:annotation>
1926         <xsd:documentation>BIAS Operation Return Codes</xsd:documentation>
1927     </xsd:annotation>
1928     <xsd:restriction base="xsd:unsignedLong">
1929         <xsd:enumeration value="0">
1930             <xsd:annotation>
1931                 <xsd:documentation>Success</xsd:documentation>
1932             </xsd:annotation>
1933         </xsd:enumeration>
1934     </xsd:restriction>
1935 </xsd:simpleType>
1936 <xsd:simpleType name="Score">
1937     <xsd:annotation>
1938         <xsd:documentation>Match result or quality score.</xsd:documentation>
1939     </xsd:annotation>
1940     <xsd:restriction base="xsd:float" />
1941 </xsd:simpleType>
1942 <xsd:complexType name="TokenType">
1943     <xsd:annotation>
1944         <xsd:documentation>Defines a token that is returned for asynchronous processing.</xsd:documentation>
1945     </xsd:annotation>
1946     <xsd:sequence>
1947         <xsd:element name="TokenValue" type="xsd:string" minOccurs="1" maxOccurs="1">
1948             <xsd:annotation>
1949                 <xsd:documentation>A value returned by the implementing system that is used to retrieve the results to a
1950 service at a later time.</xsd:documentation>
1951             </xsd:annotation>
1952         </xsd:element>
1953         <xsd:element name="Expiration" type="xsd:date" minOccurs="1" maxOccurs="1">
1954             <xsd:annotation>
1955                 <xsd:documentation>A date and time at which point the token expires and the service results are no longer
1956 guaranteed to be available.</xsd:documentation>
1957             </xsd:annotation>
1958         </xsd:element>
1959     </xsd:sequence>
1960 </xsd:complexType>
1961 <xsd:simpleType name="VendorIdentifier">
1962     <xsd:annotation>
1963         <xsd:documentation>Identifies a vendor.</xsd:documentation>
1964     </xsd:annotation>
1965     <xsd:restriction base="xsd:string" />
1966 </xsd:simpleType>
1967 <xsd:complexType name="Version">
1968     <xsd:annotation>
1969         <xsd:documentation>
1970             For a description or definition of each data element, see the

```

```

1971      referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
1972      </xsd:documentation>
1973      </xsd:annotation>
1974      <xsd:sequence>
1975          <xsd:element name="major" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1" />
1976          <xsd:element name="minor" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1" />
1977      </xsd:sequence>
1978  </xsd:complexType>
1979  <xsd:simpleType name="VersionType">
1980      <xsd:annotation>
1981          <xsd:documentation>The version of a component.</xsd:documentation>
1982      </xsd:annotation>
1983          <xsd:restriction base="xsd:string" />
1984  </xsd:simpleType>
1985  <xsd:element name="AddSubjectToGalleryRequest">
1986      <xsd:complexType>
1987          <xsd:annotation>
1988              <xsd:documentation>
1989                  Register a subject to a given gallery or population group.
1990              </xsd:documentation>
1991          </xsd:annotation>
1992          <xsd:complexContent>
1993              <xsd:extension base="tns:RequestTemplate">
1994                  <xsd:sequence>
1995                      <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1" />
1996                      <xsd:annotation>
1997                          <xsd:documentation>The identifier of the gallery or population group to which the subject will be
1998 added.</xsd:documentation>
1999                  </xsd:annotation>
2000          </xsd:element>
2001          <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1" />
2002          <xsd:annotation>
2003              <xsd:documentation>The identity to add to the gallery.</xsd:documentation>
2004          </xsd:annotation>
2005          </xsd:element>
2006      </xsd:sequence>
2007  </xsd:extension>
2008  </xsd:complexContent>
2009  </xsd:complexType>
2010 </xsd:element>
2011 <xsd:element name="AddSubjectToGalleryResponse">
2012     <xsd:complexType>
2013         <xsd:annotation>
2014             <xsd:documentation>
2015                 The response to an AddSubjectToGallery request.
2016             </xsd:documentation>
2017         </xsd:annotation>
2018         <xsd:complexContent>
2019             <xsd:extension base="tns:ResponseTemplate" />
2020         </xsd:complexContent>
2021     </xsd:complexType>
2022     </xsd:element>
2023 <xsd:element name="CheckQualityRequest">
```

```

2024    <xsd:complexType>
2025        <xsd:annotation>
2026            <xsd:documentation>
2027                Calculate a quality score for a given biometric.
2028            </xsd:documentation>
2029        </xsd:annotation>
2030        <xsd:complexContent>
2031            <xsd:extension base="tns:RequestTemplate">
2032                <xsd:sequence>
2033                    <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
2034                        <xsd:annotation>
2035                            <xsd:documentation>Data structure containing a single biometric sample for which a quality score is to be
2036 determined.</xsd:documentation>
2037                        </xsd:annotation>
2038                    </xsd:element>
2039                    <xsd:element name="Quality" type="tns:QualityData" minOccurs="0" maxOccurs="1">
2040                        <xsd:annotation>
2041                            <xsd:documentation>Specifies a particular algorithm vendor and vendor product ID.</xsd:documentation>
2042                        </xsd:annotation>
2043                    </xsd:element>
2044                </xsd:sequence>
2045            </xsd:extension>
2046        </xsd:complexContent>
2047    </xsd:complexType>
2048    </xsd:element>
2049    <xsd:element name="CheckQualityResponse">
2050        <xsd:complexType>
2051            <xsd:annotation>
2052                <xsd:documentation>
2053                    The response to a CheckQuality request.
2054                </xsd:documentation>
2055            </xsd:annotation>
2056            <xsd:complexContent>
2057                <xsd:extension base="tns:ResponseTemplate">
2058                    <xsd:sequence>
2059                        <xsd:element name="QualityInfo" type="tns:QualityData" minOccurs="1" maxOccurs="1">
2060                            <xsd:annotation>
2061                                <xsd:documentation>Contains the quality information for the submitted biometric
2062 sample.</xsd:documentation>
2063                            </xsd:annotation>
2064                        </xsd:element>
2065                    </xsd:sequence>
2066                </xsd:extension>
2067            </xsd:complexContent>
2068        </xsd:complexType>
2069    </xsd:element>
2070    <xsd:element name="ClassifyBiometricDataRequest">
2071        <xsd:complexType>
2072            <xsd:annotation>
2073                <xsd:documentation>
2074                    Classifies a biometric sample.
2075                </xsd:documentation>
2076            </xsd:annotation>

```

```

2077    <xsd:complexContent>
2078        <xsd:extension base="tns:RequestTemplate">
2079            <xsd:sequence>
2080                <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
2081                    <xsd:annotation>
2082                        <xsd:documentation>Data structure containing a single biometric sample for which the classification is to
2083 be determined.</xsd:documentation>
2084                    </xsd:annotation>
2085                    </xsd:element>
2086                </xsd:sequence>
2087            </xsd:extension>
2088        </xsd:complexContent>
2089    </xsd:complexType>
2090 </xsd:element>
2091 <xsd:element name="ClassifyBiometricDataResponse">
2092     <xsd:complexType>
2093         <xsd:annotation>
2094             <xsd:documentation>
2095                 The response to a ClassifyBiometricData request, containing
2096                 the classification of a biometric sample.
2097             </xsd:documentation>
2098         </xsd:annotation>
2099     <xsd:complexContent>
2100         <xsd:extension base="tns:ResponseTemplate">
2101             <xsd:sequence>
2102                 <xsd:element name="ClassificationData" type="tns:ClassificationData" minOccurs="1" maxOccurs="1">
2103                     <xsd:annotation>
2104                         <xsd:documentation>Information on the results and type of classification
2105 performed.</xsd:documentation>
2106                     </xsd:annotation>
2107                     </xsd:element>
2108                 </xsd:sequence>
2109             </xsd:extension>
2110         </xsd:complexContent>
2111     </xsd:complexType>
2112 </xsd:element>
2113 <xsd:element name="CreateSubjectRequest">
2114     <xsd:complexType>
2115         <xsd:annotation>
2116             <xsd:documentation>
2117                 Create a new subject record.
2118             </xsd:documentation>
2119         </xsd:annotation>
2120         <xsd:complexContent>
2121             <xsd:extension base="tns:RequestTemplate" />
2122         </xsd:complexContent>
2123     </xsd:complexType>
2124 </xsd:element>
2125 <xsd:element name="CreateSubjectResponse">
2126     <xsd:complexType>
2127         <xsd:annotation>
2128             <xsd:documentation>
2129                 The response to a CreateSubject request, containing the subject

```

```

2130      ID of the new subject record.
2131      </xsd:documentation>
2132      </xsd:annotation>
2133      <xsd:complexContent>
2134          <xsd:extension base="tns:ResponseTemplate">
2135              <xsd:sequence>
2136                  <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2137                      <xsd:annotation>
2138                          <xsd:documentation>Contains the subject ID of the new subject record.</xsd:documentation>
2139                      </xsd:annotation>
2140                  </xsd:element>
2141                  </xsd:sequence>
2142          </xsd:extension>
2143      </xsd:complexContent>
2144      </xsd:complexType>
2145      </xsd:element>
2146      <xsd:element name="DeleteBiographicDataRequest">
2147          <xsd:complexType>
2148              <xsd:annotation>
2149                  <xsd:documentation>
2150                      Erase all of the biographic data associated with a given
2151                      subject record or, in the encounter-centric model, with a
2152                      given encounter.
2153                  </xsd:documentation>
2154              </xsd:annotation>
2155              <xsd:complexContent>
2156                  <xsd:extension base="tns:RequestTemplate">
2157                      <xsd:sequence>
2158                          <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2159                              <xsd:annotation>
2160                                  <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
2161                              </xsd:annotation>
2162                          </xsd:element>
2163                          </xsd:sequence>
2164          </xsd:extension>
2165      </xsd:complexContent>
2166      </xsd:complexType>
2167      </xsd:element>
2168      <xsd:element name="DeleteBiographicDataResponse">
2169          <xsd:complexType>
2170              <xsd:annotation>
2171                  <xsd:documentation>
2172                      The response to a DeleteBiographicData request.
2173                  </xsd:documentation>
2174              </xsd:annotation>
2175              <xsd:complexContent>
2176                  <xsd:extension base="tns:ResponseTemplate" />
2177              </xsd:complexContent>
2178          </xsd:complexType>
2179      </xsd:element>
2180      <xsd:element name="DeleteBiometricDataRequest">
2181          <xsd:complexType>
2182              <xsd:annotation>

```

```

2183    <xsd:documentation>
2184        Erase all of the biometric data associated with a given
2185        subject record or, in the encounter-centric model, with a
2186        given encounter.
2187    </xsd:documentation>
2188    </xsd:annotation>
2189    <xsd:complexContent>
2190        <xsd:extension base="tns:RequestTemplate">
2191            <xsd:sequence>
2192                <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2193                    <xsd:annotation>
2194                        <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
2195                    </xsd:annotation>
2196                </xsd:element>
2197            </xsd:sequence>
2198        </xsd:extension>
2199    </xsd:complexContent>
2200    </xsd:complexType>
2201 </xsd:element>
2202 <xsd:element name="DeleteBiometricDataResponse">
2203     <xsd:complexType>
2204         <xsd:annotation>
2205             <xsd:documentation>
2206                 The response to a DeleteBiometricData request.
2207             </xsd:documentation>
2208         </xsd:annotation>
2209         <xsd:complexContent>
2210             <xsd:extension base="tns:ResponseTemplate" />
2211         </xsd:complexContent>
2212     </xsd:complexType>
2213 </xsd:element>
2214 <xsd:element name="DeleteSubjectRequest">
2215     <xsd:complexType>
2216         <xsd:annotation>
2217             <xsd:documentation>
2218                 Delete an existing subject record and, in an encounter-centric
2219                 model, any associated encounter information.
2220             </xsd:documentation>
2221         </xsd:annotation>
2222         <xsd:complexContent>
2223             <xsd:extension base="tns:RequestTemplate">
2224                 <xsd:sequence>
2225                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2226                         <xsd:annotation>
2227                             <xsd:documentation>Subject ID of the identity to delete.</xsd:documentation>
2228                         </xsd:annotation>
2229                     </xsd:element>
2230                 </xsd:sequence>
2231             </xsd:extension>
2232         </xsd:complexContent>
2233     </xsd:complexType>
2234 </xsd:element>
2235 <xsd:element name="DeleteSubjectResponse">

```

```

2236    <xsd:complexType>
2237        <xsd:annotation>
2238            <xsd:documentation>
2239                The response to a DeleteSubject request.
2240            </xsd:documentation>
2241        </xsd:annotation>
2242        <xsd:complexContent>
2243            <xsd:extension base="tns:ResponseTemplate" />
2244        </xsd:complexContent>
2245    </xsd:complexType>
2246</xsd:element>
2247<xsd:element name="DeleteSubjectFromGalleryRequest">
2248    <xsd:complexType>
2249        <xsd:annotation>
2250            <xsd:documentation>
2251                Remove the registration of a subject from a gallery or
2252                population group.
2253            </xsd:documentation>
2254        </xsd:annotation>
2255        <xsd:complexContent>
2256            <xsd:extension base="tns:RequestTemplate">
2257                <xsd:sequence>
2258                    <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2259                        <xsd:annotation>
2260                            <xsd:documentation>The identifier of the gallery or population group from which the subject will be
2261 deleted.</xsd:documentation>
2262                        </xsd:annotation>
2263                    </xsd:element>
2264                    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2265                        <xsd:annotation>
2266                            <xsd:documentation>The identity to remove from the gallery.</xsd:documentation>
2267                        </xsd:annotation>
2268                    </xsd:element>
2269                </xsd:sequence>
2270            </xsd:extension>
2271        </xsd:complexContent>
2272    </xsd:complexType>
2273</xsd:element>
2274<xsd:element name="DeleteSubjectFromGalleryResponse">
2275    <xsd:complexType>
2276        <xsd:annotation>
2277            <xsd:documentation>
2278                The response to a DeleteSubjectFromGallery request.
2279            </xsd:documentation>
2280        </xsd:annotation>
2281        <xsd:complexContent>
2282            <xsd:extension base="tns:ResponseTemplate" />
2283        </xsd:complexContent>
2284    </xsd:complexType>
2285</xsd:element>
2286<xsd:element name="GetIdentifySubjectResultsRequest">
2287    <xsd:complexType>
2288        <xsd:annotation>

```

```

2289     <xsd:documentation>
2290         Retrieve the identification results for a specified token,
2291         which was returned by the Identify Subject service.
2292     </xsd:documentation>
2293     </xsd:annotation>
2294     <xsd:complexContent>
2295         <xsd:extension base="tns:RequestTemplate">
2296             <xsd:sequence>
2297                 <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
2298                     <xsd:annotation>
2299                         <xsd:documentation>A value used to retrieve the results of an IdentifySubject
2300 request.</xsd:documentation>
2301                 </xsd:annotation>
2302             </xsd:element>
2303         </xsd:sequence>
2304     </xsd:extension>
2305     </xsd:complexContent>
2306     </xsd:complexType>
2307     </xsd:element>
2308     <xsd:element name="GetIdentifySubjectResultsResponse">
2309         <xsd:complexType>
2310             <xsd:annotation>
2311                 <xsd:documentation>
2312                     The response to a GetIdentifySubjectResults request, which includes a candidate list.
2313                 </xsd:documentation>
2314             </xsd:annotation>
2315             <xsd:complexContent>
2316                 <xsd:extension base="tns:ResponseTemplate">
2317                     <xsd:sequence>
2318                         <xsd:element name="CandidateList" type="tns:CandidateListType" minOccurs="1" maxOccurs="1">
2319                             <xsd:annotation>
2320                                 <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input
2321 biometric sample.</xsd:documentation>
2322                         </xsd:annotation>
2323                     </xsd:element>
2324                 </xsd:sequence>
2325             </xsd:extension>
2326             </xsd:complexContent>
2327         </xsd:complexType>
2328     </xsd:element>
2329     <xsd:element name="IdentifySubjectRequest">
2330         <xsd:complexType>
2331             <xsd:annotation>
2332                 <xsd:documentation>
2333                     Perform an identification search against a given gallery for
2334                     a given biometric.
2335                 </xsd:documentation>
2336             </xsd:annotation>
2337             <xsd:complexContent>
2338                 <xsd:extension base="tns:RequestTemplate">
2339                     <xsd:sequence>
2340                         <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2341                         <xsd:annotation>
```

```

2342      <xsd:documentation>The identifier of the gallery or population group which will be
2343      searched.</xsd:documentation>
2344      </xsd:annotation>
2345      </xsd:element>
2346      <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2347          <xsd:annotation>
2348              <xsd:documentation>Contains the BIR, a data structure containing the biometric sample for the
2349      search.</xsd:documentation>
2350          </xsd:annotation>
2351          </xsd:element>
2352          <xsd:element name="MaxListSize" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">
2353              <xsd:annotation>
2354                  <xsd:documentation>The maximum size of the candidate list that should be
2355      returned.</xsd:documentation>
2356          </xsd:annotation>
2357          </xsd:element>
2358          </xsd:sequence>
2359          </xsd:extension>
2360          </xsd:complexContent>
2361          </xsd:complexType>
2362      </xsd:element>
2363      <xsd:element name="IdentifySubjectResponse">
2364          <xsd:complexType>
2365              <xsd:annotation>
2366                  <xsd:documentation>
2367                      The response to an IdentifySubject request, returning a
2368                      rank-ordered candidate list.
2369                  </xsd:documentation>
2370              </xsd:annotation>
2371              <xsd:complexContent>
2372                  <xsd:extension base="tns:ResponseTemplate">
2373                      <xsd:choice minOccurs="1" maxOccurs="1">
2374                          <xsd:element name="CandidateList" type="tns:CandidateListType">
2375                              <xsd:annotation>
2376                                  <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input
2377      biometric sample; returned with successful synchronous request processing.</xsd:documentation>
2378          </xsd:annotation>
2379          </xsd:element>
2380          <xsd:element name="Token" type="tns:TokenType">
2381              <xsd:annotation>
2382                  <xsd:documentation>A token used to retrieve the results of the IdentifySubject request; returned with
2383      asynchronous request processing.</xsd:documentation>
2384          </xsd:annotation>
2385          </xsd:element>
2386          </xsd:choice>
2387          </xsd:extension>
2388          </xsd:complexContent>
2389      </xsd:complexType>
2390      </xsd:element>
2391      <xsd:element name="ListBiographicDataRequest">
2392          <xsd:complexType>
2393              <xsd:annotation>
2394                  <xsd:documentation>
```

```

2395      Lists the biographic data elements stored for a subject.
2396      </xsd:documentation>
2397      </xsd:annotation>
2398      <xsd:complexContent>
2399          <xsd:extension base="tns:RequestTemplate">
2400              <xsd:sequence>
2401                  <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2402                      <xsd:annotation>
2403                          <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2404 encounter.</xsd:documentation>
2405                  </xsd:annotation>
2406                  </xsd:element>
2407              </xsd:sequence>
2408          </xsd:extension>
2409      </xsd:complexContent>
2410      </xsd:complexType>
2411  </xsd:element>
2412  <xsd:element name="ListBiographicDataResponse">
2413      <xsd:complexType>
2414          <xsd:annotation>
2415              <xsd:documentation>
2416                  The response to a ListBiographicData request, containing a list
2417                  of biographic data elements stored for a subject. In the
2418                  encounter-centric model, the biographic data elements for a
2419                  specific encounter are returned. If an encounter ID is not
2420                  specified and encounter data exists for the subject, the list
2421                  of encounter IDs which contain biographic data is returned.
2422          </xsd:documentation>
2423      </xsd:annotation>
2424      <xsd:complexContent>
2425          <xsd:extension base="tns:ResponseTemplate">
2426              <xsd:sequence>
2427                  <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2428                      <xsd:annotation>
2429                          <xsd:documentation>
2430                              Contains a list of biographic data elements associated with a
2431                              subject or encounter; non-empty if the service was
2432                              successful, biographic data exists, and either (a) the
2433                              person-centric model is being used or (b) the
2434                              encounter-centric model is being used and an encounter
2435                              identifier was specified.
2436                      </xsd:documentation>
2437                  </xsd:annotation>
2438              </xsd:element>
2439              <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
2440                  <xsd:annotation>
2441                      <xsd:documentation>
2442                          A list of encounter ID's associated with a subject and
2443                          which contain biographic data; non-empty if the service
2444                          was successful, biographic data exists, the
2445                          encounter-centric model is being used, and an encounter
2446                          identifier was not specified.
2447                  </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:element name="ListBiometricDataRequest">
2456          <xsd:complexType>
2457              <xsd:annotation>
2458                  <xsd:documentation>
2459                      Lists the biometric data elements stored for a subject. Note
2460                      that no actual biometric data is returned by this service (see
2461                      the RetrieveBiometricInformation service to obtain the biometric
2462                      data).
2463              </xsd:documentation>
2464          </xsd:annotation>
2465          <xsd:complexContent>
2466              <xsd:extension base="tns:RequestTemplate">
2467                  <xsd:sequence>
2468                      <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2469                          <xsd:annotation>
2470                              <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2471 encounter.</xsd:documentation>
2472                          </xsd:annotation>
2473                      </xsd:element>
2474                      <xsd:element name="ListFilter" type="tns>ListFilterType" minOccurs="0" maxOccurs="1">
2475                          <xsd:annotation>
2476                              <xsd:documentation>Indicates what biometric information should be returned.</xsd:documentation>
2477                          </xsd:annotation>
2478                      </xsd:element>
2479                  </xsd:sequence>
2480              </xsd:extension>
2481          </xsd:complexContent>
2482      </xsd:complexType>
2483  </xsd:element>
2484  <xsd:element name="ListBiometricDataResponse">
2485      <xsd:complexType>
2486          <xsd:annotation>
2487              <xsd:documentation>
2488                  The response to a ListBiometricData request, containing a list
2489                  of biometric data elements stored for a subject. In the
2490                  encounter-centric model, the biometric data elements for a
2491                  specific encounter are returned. If an encounter ID is not
2492                  specified and encounter data exists for the subject, the list
2493                  of encounter IDs which contain biometric data is returned.
2494              </xsd:documentation>
2495          </xsd:annotation>
2496          <xsd:complexContent>
2497              <xsd:extension base="tns:ResponseTemplate">
2498                  <xsd:sequence>
2499                      <xsd:element name="Identity" type="tns:BIASIdentity">
2500                          <xsd:annotation>
```

```

2501      <xsd:documentation>
2502          Includes a list of biometric data elements associated
2503          with a subject or encounter or a list of encounter ID's
2504          associated with a subject and which contain biometric
2505          data.
2506      </xsd:documentation>
2507      </xsd:annotation>
2508      </xsd:element>
2509      </xsd:sequence>
2510      </xsd:extension>
2511      </xsd:complexContent>
2512      </xsd:complexType>
2513  </xsd:element>
2514  <xsd:element name="PerformFusionRequest">
2515      <xsd:complexType>
2516          <xsd:annotation>
2517              <xsd:documentation>
2518                  Accepts either match score or match decision information and creates a fused match result.
2519              </xsd:documentation>
2520          </xsd:annotation>
2521          <xsd:complexContent>
2522              <xsd:extension base="tns:RequestTemplate">
2523                  <xsd:sequence>
2524                      <xsd:element name="FusionInput" type="tns:FusionInformationListType" minOccurs="1" maxOccurs="1">
2525                          <xsd:annotation>
2526                              <xsd:documentation>Score or decision input information to the fusion method.</xsd:documentation>
2527                          </xsd:annotation>
2528                      </xsd:element>
2529                  </xsd:sequence>
2530                  </xsd:extension>
2531              </xsd:complexContent>
2532          </xsd:complexType>
2533  </xsd:element>
2534  <xsd:element name="PerformFusionResponse">
2535      <xsd:complexType>
2536          <xsd:annotation>
2537              <xsd:documentation>
2538                  The response to the PerformFusion request.
2539              </xsd:documentation>
2540          </xsd:annotation>
2541          <xsd:complexContent>
2542              <xsd:extension base="tns:ResponseTemplate">
2543                  <xsd:sequence>
2544                      <xsd:element name="Match" type="tns:MatchType" minOccurs="1" maxOccurs="1">
2545                          <xsd:annotation>
2546                              <xsd:documentation>Indicates the result of the fusion method</xsd:documentation>
2547                          </xsd:annotation>
2548                      </xsd:element>
2549                  </xsd:sequence>
2550                  </xsd:extension>
2551              </xsd:complexContent>
2552          </xsd:complexType>
2553  </xsd:element>

```

```

2554 <xsd:element name="QueryCapabilitiesRequest">
2555   <xsd:complexType>
2556     <xsd:annotation>
2557       <xsd:documentation>
2558         Returns a list of the capabilities, options, galleries, etc.
2559         that are supported by the BIAS implementation.
2560       </xsd:documentation>
2561     </xsd:annotation>
2562     <xsd:complexContent>
2563       <xsd:extension base="tns:RequestTemplate" />
2564     </xsd:complexContent>
2565   </xsd:complexType>
2566 </xsd:element>
2567 <xsd:element name="QueryCapabilitiesResponse">
2568   <xsd:complexType>
2569     <xsd:annotation>
2570       <xsd:documentation>
2571         The response to a QueryCapabilities request.
2572       </xsd:documentation>
2573     </xsd:annotation>
2574     <xsd:complexContent>
2575       <xsd:extension base="tns:ResponseTemplate">
2576         <xsd:sequence>
2577           <xsd:element name="CapabilityList" type="tns:CapabilityListType" minOccurs="1" maxOccurs="1">
2578             <xsd:annotation>
2579               <xsd:documentation>A list of capabilities supported by the BIAS implementation.</xsd:documentation>
2580             </xsd:annotation>
2581           </xsd:element>
2582         </xsd:sequence>
2583       </xsd:extension>
2584     </xsd:complexContent>
2585   </xsd:complexType>
2586 </xsd:element>
2587 <xsd:element name="RetrieveBiographicInformationRequest">
2588   <xsd:complexType>
2589     <xsd:annotation>
2590       <xsd:documentation>
2591         Retrieves the biographic data associated with a subject ID.
2592       </xsd:documentation>
2593     </xsd:annotation>
2594     <xsd:complexContent>
2595       <xsd:extension base="tns:RequestTemplate">
2596         <xsd:sequence>
2597           <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2598             <xsd:annotation>
2599               <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2600               encounter.</xsd:documentation>
2601             </xsd:annotation>
2602           </xsd:element>
2603         </xsd:sequence>
2604       </xsd:extension>
2605     </xsd:complexContent>
2606   </xsd:complexType>

```

```

2607    </xsd:element>
2608    <xsd:element name="RetrieveBiographicInformationResponse">
2609        <xsd:complexType>
2610            <xsd:annotation>
2611                <xsd:documentation>
2612                    The response to a RetrieveBiographicInformation request,
2613                    containing the biographic data associated with a subject ID. In
2614                    the encounter-centric model, the biographic data associated with
2615                    a specified encounter is returned. If the encounter ID is not
2616                    specified in the encounter-centric model, the biographic
2617                    information associated with the most recent encounter is returned.
2618                </xsd:documentation>
2619            </xsd:annotation>
2620            <xsd:complexContent>
2621                <xsd:extension base="tns:ResponseTemplate">
2622                    <xsd:sequence>
2623                        <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2624                            <xsd:annotation>
2625                                <xsd:documentation>Includes the set of biographic data associated with a subject.</xsd:documentation>
2626                            </xsd:annotation>
2627                        </xsd:element>
2628                    </xsd:sequence>
2629                </xsd:extension>
2630            </xsd:complexContent>
2631        </xsd:complexType>
2632    </xsd:element>
2633    <xsd:element name="RetrieveBiometricInformationRequest">
2634        <xsd:complexType>
2635            <xsd:annotation>
2636                <xsd:documentation>
2637                    Retrieves the biometric data associated with a subject ID.
2638                </xsd:documentation>
2639            </xsd:annotation>
2640            <xsd:complexContent>
2641                <xsd:extension base="tns:RequestTemplate">
2642                    <xsd:sequence>
2643                        <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2644                            <xsd:annotation>
2645                                <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2646 encounter.</xsd:documentation>
2647                            </xsd:annotation>
2648                        </xsd:element>
2649                        <xsd:element name="BiometricType" type="iso-iec19785-3-7:Multiple-types" minOccurs="0"
2650 maxOccurs="1">
2651                            <xsd:annotation>
2652                                <xsd:documentation>The type of biological or behavioral data to retrieve.</xsd:documentation>
2653                            </xsd:annotation>
2654                        </xsd:element>
2655                    </xsd:sequence>
2656                </xsd:extension>
2657            </xsd:complexContent>
2658        </xsd:complexType>
2659    </xsd:element>

```

```

2660    <xsd:element name="RetrieveBiometricInformationResponse">
2661        <xsd:complexType>
2662            <xsd:annotation>
2663                <xsd:documentation>
2664                    The response to a RetrieveBiometricInformation request,
2665                    containing the biometric data associated with a subject ID. In
2666                    the encounter-centric model, the biometric data associated with
2667                    a specified encounter is returned. If the encounter ID is not
2668                    specified in the encounter-centric model, the biometric
2669                    information associated with the most recent encounter is returned.
2670                </xsd:documentation>
2671            </xsd:annotation>
2672            <xsd:complexContent>
2673                <xsd:extension base="tns:ResponseTemplate">
2674                    <xsd:sequence>
2675                        <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2676                            <xsd:annotation>
2677                                <xsd:documentation>Includes the biometric data associated with a subject.</xsd:documentation>
2678                            </xsd:annotation>
2679                        </xsd:element>
2680                    </xsd:sequence>
2681                </xsd:extension>
2682            </xsd:complexContent>
2683        </xsd:complexType>
2684    </xsd:element>
2685    <xsd:element name="SetBiographicDataRequest">
2686        <xsd:complexType>
2687            <xsd:annotation>
2688                <xsd:documentation>
2689                    Associates biographic data to a given subject record.
2690                </xsd:documentation>
2691            </xsd:annotation>
2692            <xsd:complexContent>
2693                <xsd:extension base="tns:RequestTemplate">
2694                    <xsd:sequence>
2695                        <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2696                            <xsd:annotation>
2697                                <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2698                                and includes the biographic data to store.</xsd:documentation>
2699                            </xsd:annotation>
2700                        </xsd:element>
2701                    </xsd:sequence>
2702                </xsd:extension>
2703            </xsd:complexContent>
2704        </xsd:complexType>
2705    </xsd:element>
2706    <xsd:element name="SetBiographicDataResponse">
2707        <xsd:complexType>
2708            <xsd:annotation>
2709                <xsd:documentation>
2710                    The response to a SetBiographicData request.
2711                </xsd:documentation>
2712            </xsd:annotation>

```

```

2713    <xsd:complexContent>
2714      <xsd:extension base="tns:ResponseTemplate">
2715        <xsd:sequence>
2716          <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2717            <xsd:annotation>
2718              <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
2719 encounter.</xsd:documentation>
2720            </xsd:annotation>
2721            </xsd:element>
2722          </xsd:sequence>>
2723        </xsd:extension>
2724      </xsd:complexContent>
2725    </xsd:complexType>
2726  </xsd:element>
2727  <xsd:element name="SetBiometricDataRequest">
2728    <xsd:complexType>
2729      <xsd:annotation>
2730        <xsd:documentation>
2731          Associates biometric data to a given subject record.
2732        </xsd:documentation>
2733      </xsd:annotation>
2734    <xsd:complexContent>
2735      <xsd:extension base="tns:RequestTemplate">
2736        <xsd:sequence>
2737          <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2738            <xsd:annotation>
2739              <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2740 and includes the biometric data to store.</xsd:documentation>
2741            </xsd:annotation>
2742            </xsd:element>
2743          </xsd:sequence>
2744        </xsd:extension>
2745      </xsd:complexContent>
2746    </xsd:complexType>
2747  </xsd:element>
2748  <xsd:element name="SetBiometricDataResponse">
2749    <xsd:complexType>
2750      <xsd:annotation>
2751        <xsd:documentation>
2752          The response to a SetBiometricData request.
2753        </xsd:documentation>
2754      </xsd:annotation>
2755    <xsd:complexContent>
2756      <xsd:extension base="tns:ResponseTemplate">
2757        <xsd:sequence>
2758          <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2759            <xsd:annotation>
2760              <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
2761 encounter.</xsd:documentation>
2762            </xsd:annotation>
2763            </xsd:element>
2764          </xsd:sequence>>
2765        </xsd:extension>

```

```

2766      </xsd:complexContent>
2767      </xsd:complexType>
2768      </xsd:element>
2769      <xsd:element name="TransformBiometricDataRequest">
2770          <xsd:complexType>
2771              <xsd:annotation>
2772                  <xsd:documentation>
2773                      Transforms or processes a given biometric in one format into a new target format.
2774                  </xsd:documentation>
2775              </xsd:annotation>
2776              <xsd:complexContent>
2777                  <xsd:extension base="tns:RequestTemplate">
2778                      <xsd:sequence>
2779                          <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="1" maxOccurs="1">
2780                              <xsd:annotation>
2781                                  <xsd:documentation>Data structure containing the biometric information to be
2782 transformed.</xsd:documentation>
2783                              </xsd:annotation>
2784                      </xsd:element>
2785                      <xsd:element name="TransformOperation" type="xsd:unsignedLong" minOccurs="1" maxOccurs="1">
2786                          <xsd:annotation>
2787                              <xsd:documentation>Value indicating the type of transformation to perform.</xsd:documentation>
2788                          </xsd:annotation>
2789                      </xsd:element>
2790                      <xsd:element name="TransformControl" type="xsd:string" minOccurs="0" maxOccurs="1">
2791                          <xsd:annotation>
2792                              <xsd:documentation> Specifies controls for the requested transform operation.</xsd:documentation>
2793                          </xsd:annotation>
2794                      </xsd:element>
2795                      </xsd:sequence>
2796                  </xsd:extension>
2797              </xsd:complexContent>
2798          </xsd:complexType>
2799      </xsd:element>
2800      <xsd:element name="TransformBiometricDataResponse">
2801          <xsd:complexType>
2802              <xsd:annotation>
2803                  <xsd:documentation>
2804                      The response to a TransformBiometricData request.
2805                  </xsd:documentation>
2806              </xsd:annotation>
2807              <xsd:complexContent>
2808                  <xsd:extension base="tns:ResponseTemplate">
2809                      <xsd:sequence>
2810                          <xsd:element name="OutputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
2811                              <xsd:annotation>
2812                                  <xsd:documentation>Data structure containing the new, transformed biometric
2813 information.</xsd:documentation>
2814                              </xsd:annotation>
2815                      </xsd:element>
2816                      </xsd:sequence>
2817                  </xsd:extension>
2818              </xsd:complexContent>

```

```

2819    </xsd:complexType>
2820  </xsd:element>
2821  <xsd:element name="UpdateBiographicDataRequest">
2822    <xsd:complexType>
2823      <xsd:annotation>
2824        <xsd:documentation>
2825          Updates the biographic data for a given subject record.
2826        </xsd:documentation>
2827      </xsd:annotation>
2828      <xsd:complexContent>
2829        <xsd:extension base="tns:RequestTemplate">
2830          <xsd:sequence>
2831            <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2832              <xsd:annotation>
2833                <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2834 and includes the biographic data to update.</xsd:documentation>
2835              </xsd:annotation>
2836            </xsd:element>
2837          </xsd:sequence>
2838        </xsd:extension>
2839      </xsd:complexContent>
2840    </xsd:complexType>
2841  </xsd:element>
2842  <xsd:element name="UpdateBiographicDataResponse">
2843    <xsd:complexType>
2844      <xsd:annotation>
2845        <xsd:documentation>
2846          The response to an UpdateBiographicData request.
2847        </xsd:documentation>
2848      </xsd:annotation>
2849      <xsd:complexContent>
2850        <xsd:extension base="tns:ResponseTemplate" />
2851      </xsd:complexContent>
2852    </xsd:complexType>
2853  </xsd:element>
2854  <xsd:element name="UpdateBiometricDataRequest">
2855    <xsd:complexType>
2856      <xsd:annotation>
2857        <xsd:documentation>
2858          Updates a single biometric sample for a given subject record.
2859        </xsd:documentation>
2860      </xsd:annotation>
2861      <xsd:complexContent>
2862        <xsd:extension base="tns:RequestTemplate">
2863          <xsd:sequence>
2864            <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2865              <xsd:annotation>
2866                <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2867 and includes the biometric data to update.</xsd:documentation>
2868              </xsd:annotation>
2869            </xsd:element>
2870            <xsd:element name="Merge" type="xsd:boolean" minOccurs="0" maxOccurs="1">
2871              <xsd:annotation>

```

```

2872      <xsd:documentation>Value indicating if the input biometric sample should be merged with any existing
2873  biometric information.</xsd:documentation>
2874      </xsd:annotation>
2875      </xsd:element>
2876      </xsd:sequence>
2877      </xsd:extension>
2878      </xsd:complexContent>
2879      </xsd:complexType>
2880  </xsd:element>
2881  <xsd:element name="UpdateBiometricDataResponse">
2882      <xsd:complexType>
2883      <xsd:annotation>
2884          <xsd:documentation>
2885              The response to an UpdateBiometricData request.
2886          </xsd:documentation>
2887      </xsd:annotation>
2888      <xsd:complexContent>
2889          <xsd:extension base="tns:ResponseTemplate" />
2890      </xsd:complexContent>
2891  </xsd:complexType>
2892  </xsd:element>
2893  <xsd:element name="VerifySubjectRequest">
2894      <xsd:complexType>
2895      <xsd:annotation>
2896          <xsd:documentation>
2897              Performs a 1:1 verification match between a given biometric and
2898              either a claim to identity in a given gallery or another given
2899              biometric.
2900          </xsd:documentation>
2901      </xsd:annotation>
2902      <xsd:complexContent>
2903          <xsd:extension base="tns:RequestTemplate">
2904              <xsd:sequence>
2905                  <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
2906                      <xsd:annotation>
2907                          <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
2908 member.</xsd:documentation>
2909                      </xsd:annotation>
2910                  </xsd:element>
2911                  <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2912                      <xsd:annotation>
2913                          <xsd:documentation>Includes the identifying information and/or input and reference biometric
2914 samples.</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:element name="VerifySubjectResponse">
2923      <xsd:complexType>
2924      <xsd:annotation>
```

```

2925      <xsd:documentation>
2926          The response to a VerifySubject request.
2927      </xsd:documentation>
2928  </xsd:annotation>
2929  <xsd:complexContent>
2930      <xsd:extension base="tns:ResponseTemplate">
2931          <xsd:sequence>
2932              <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
2933                  <xsd:annotation>
2934                      <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with
2935 the Identity Claim or the Reference BIR.</xsd:documentation>
2936                  </xsd:annotation>
2937          </xsd:element>
2938          <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
2939              <xsd:annotation>
2940                  <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
2941              </xsd:annotation>
2942          </xsd:element>
2943          </xsd:sequence>
2944      </xsd:extension>
2945  </xsd:complexContent>
2946  </xsd:complexType>
2947 </xsd:element>
2948 <xsd:element name="EnrollRequest">
2949     <xsd:complexType>
2950         <xsd:annotation>
2951             <xsd:documentation>
2952                 The Enroll aggregate service adds a new subject or, in an
2953                 encounter-centric model, a new encounter to the system. This may
2954                 be accomplished in a number of different ways according to
2955                 system requirements and/or resources. If the Enroll aggregate
2956                 service is implemented as a synchronous service, the
2957                 implementing system immediately processes the request and
2958                 returns the results in the ReturnData parameter. If the Enroll
2959                 aggregate service is implemented as an asynchronous service, the
2960                 implementing system returns a token in the ReturnData
2961                 parameter, which is an indication that the request is being
2962                 handled asynchronously. In this case, the GetEnrollResults
2963                 service is used to poll for the results of the Enroll request.
2964             </xsd:documentation>
2965         </xsd:annotation>
2966         <xsd:complexContent>
2967             <xsd:extension base="tns:AggregateRequestTemplate" />
2968         </xsd:complexContent>
2969     </xsd:complexType>
2970 </xsd:element>
2971 <xsd:element name="EnrollResponse">
2972     <xsd:complexType>
2973         <xsd:annotation>
2974             <xsd:documentation>The response to an Enroll request.</xsd:documentation>
2975         </xsd:annotation>
2976         <xsd:complexContent>
2977             <xsd:extension base="tns:AggregateResponseTemplate" />

```

```

2978      </xsd:complexContent>
2979      </xsd:complexType>
2980      </xsd:element>
2981      <xsd:element name="GetEnrollResultsRequest">
2982          <xsd:complexType>
2983              <xsd:annotation>
2984                  <xsd:documentation>
2985                      The GetEnrollResults aggregate service retrieves the enrollment
2986                      results for the specified token. This service is used in
2987                      conjunction with the Enroll aggregate service. If the Enroll
2988                      aggregate service is implemented as an asynchronous service, the
2989                      implementing system returns a token, and the GetEnrollResults
2990                      service is used to poll for the results of the original Enroll
2991                      request.
2992                  </xsd:documentation>
2993              </xsd:annotation>
2994              <xsd:complexContent>
2995                  <xsd:extension base="tns:RequestTemplate">
2996                      <xsd:sequence>
2997                          <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
2998                              <xsd:annotation>
2999                                  <xsd:documentation>A value used to retrieve the results of the Enroll request.</xsd:documentation>
3000                              </xsd:annotation>
3001                      </xsd:element>
3002                  </xsd:sequence>
3003              </xsd:extension>
3004          </xsd:complexContent>
3005      </xsd:complexType>
3006      </xsd:element>
3007      <xsd:element name="GetEnrollResultsResponse">
3008          <xsd:complexType>
3009              <xsd:annotation>
3010                  <xsd:documentation>The response to a GetEnrollResults request.</xsd:documentation>
3011              </xsd:annotation>
3012              <xsd:complexContent>
3013                  <xsd:extension base="tns:AggregateResponseTemplate" />
3014              </xsd:complexContent>
3015          </xsd:complexType>
3016      </xsd:element>
3017      <xsd:element name="GetIdentifyResultsRequest">
3018          <xsd:complexType>
3019              <xsd:annotation>
3020                  <xsd:documentation>
3021                      The GetIdentifyResults aggregate service retrieves the
3022                      identification results for the specified token. This service is
3023                      used in conjunction with the Identify aggregate service. If the
3024                      Identify aggregate service is implemented as an asynchronous
3025                      service, the implementing system returns a token, and the
3026                      GetIdentifyResults service is used to poll for the results of
3027                      the original Identify request.
3028                  </xsd:documentation>
3029              </xsd:annotation>
3030          <xsd:complexContent>

```

```

3031    <xsd:extension base="tns:RequestTemplate">
3032        <xsd:sequence>
3033            <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3034                <xsd:annotation>
3035                    <xsd:documentation>A value used to retrieve the results of the Identify request.</xsd:documentation>
3036                </xsd:annotation>
3037            </xsd:element>
3038        </xsd:sequence>
3039    </xsd:extension>
3040    </xsd:complexContent>
3041 </xsd:complexType>
3042 </xsd:element>
3043 <xsd:element name="GetIdentifyResultsResponse">
3044     <xsd:complexType>
3045         <xsd:annotation>
3046             <xsd:documentation>The response to a GetIdentifyResults request.</xsd:documentation>
3047         </xsd:annotation>
3048         <xsd:complexContent>
3049             <xsd:extension base="tns:AggregateResponseTemplate" />
3050         </xsd:complexContent>
3051     </xsd:complexType>
3052 </xsd:element>
3053 <xsd:element name="GetVerifyResultsRequest">
3054     <xsd:complexType>
3055         <xsd:annotation>
3056             <xsd:documentation>
3057                 The GetVerifyResults aggregate service retrieves the verification
3058                 results for the specified token. This service is used in
3059                 conjunction with the Verify aggregate service. If the Verify
3060                 aggregate service is implemented as an asynchronous service, the
3061                 implementing system returns a token, and the GetVerifyResults
3062                 service is used to poll for the results of the original Verify
3063                 request.
3064             </xsd:documentation>
3065         </xsd:annotation>
3066         <xsd:complexContent>
3067             <xsd:extension base="tns:RequestTemplate">
3068                 <xsd:sequence>
3069                     <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3070                         <xsd:annotation>
3071                             <xsd:documentation>A value used to retrieve the results of the Verify request.</xsd:documentation>
3072                         </xsd:annotation>
3073                     </xsd:element>
3074                 </xsd:sequence>
3075             </xsd:extension>
3076         </xsd:complexContent>
3077     </xsd:complexType>
3078 </xsd:element>
3079 <xsd:element name="GetVerifyResultsResponse">
3080     <xsd:complexType>
3081         <xsd:annotation>
3082             <xsd:documentation>The response to a GetVerifyResults request.</xsd:documentation>
3083         </xsd:annotation>

```

```

3084      <xsd:complexContent>
3085          <xsd:extension base="tns:AggregateResponseTemplate">
3086              <xsd:sequence>
3087                  <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3088                      <xsd:annotation>
3089                          <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with
3090 the Identity Claim or the Reference BIR.</xsd:documentation>
3091                      </xsd:annotation>
3092                  </xsd:element>
3093                  <xsd:element name="Score" type="Score" minOccurs="0" maxOccurs="1">
3094                      <xsd:annotation>
3095                          <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3096                      </xsd:annotation>
3097                  </xsd:element>
3098              </xsd:sequence>
3099          </xsd:extension>
3100      </xsd:complexContent>
3101  </xsd:complexType>
3102 </xsd:element>
3103 <xsd:element name="IdentifyRequest">
3104     <xsd:complexType>
3105         <xsd:annotation>
3106             <xsd:documentation>
3107                 The Identify aggregate service performs an identification
3108                 function according to system requirements and/or resources. If
3109                 the Identify aggregate service is implemented as a synchronous
3110                 service, the implementing system immediately processes the
3111                 request and returns the results in the ReturnData parameter. If
3112                 the Identify aggregate service is implemented as an asynchronous
3113                 service, the implementing system returns a token in the
3114                 ReturnData parameter, which is an indication that the request is
3115                 being handled asynchronously. In this case, the
3116                 GetIdentifyResults service is used to poll for the results of
3117                 the Identify request.
3118             </xsd:documentation>
3119         </xsd:annotation>
3120         <xsd:complexContent>
3121             <xsd:extension base="tns:AggregateRequestTemplate" />
3122         </xsd:complexContent>
3123     </xsd:complexType>
3124 </xsd:element>
3125 <xsd:element name="IdentifyResponse">
3126     <xsd:complexType>
3127         <xsd:annotation>
3128             <xsd:documentation>The response to an Identify request.</xsd:documentation>
3129         </xsd:annotation>
3130         <xsd:complexContent>
3131             <xsd:extension base="tns:AggregateResponseTemplate" />
3132         </xsd:complexContent>
3133     </xsd:complexType>
3134 </xsd:element>
3135 <xsd:element name="RetrieveInformationRequest">
3136     <xsd:complexType>

```

```

3137 <xsd:annotation>
3138   <xsd:documentation>
3139     The RetrieveInformation aggregate service retrieves requested
3140     information about a subject, or in an encounter-centric model
3141     about an encounter. In a person-centric model, this aggregate
3142     service may be used to retrieve both biographic and biometric
3143     information for a subject record. In an encounter-centric model,
3144     this aggregate service may be used to retrieve biographic and/or
3145     biometric information for either a single encounter or all
3146     encounters. Either a SubjectID or EncounterID must be specified
3147     in the Identify parameter.
3148   </xsd:documentation>
3149 </xsd:annotation>
3150 <xsd:complexContent>
3151   <xsd:extension base="tns:RequestTemplate">
3152     <xsd:sequence>
3153       <xsd:element name="ProcessingOptions" type="tns:ProcessingOptionsType" minOccurs="1" maxOccurs="1">
3154         <xsd:annotation>
3155           <xsd:documentation>Options that guide how the service request is processed, and may identify what
3156 type(s) of information should be returned.</xsd:documentation>
3157         </xsd:annotation>
3158       </xsd:element>
3159       <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3160         <xsd:annotation>
3161           <xsd:documentation>Includes the identifier of the subject or encounter.</xsd:documentation>
3162         </xsd:annotation>
3163       </xsd:element>
3164     </xsd:sequence>
3165   </xsd:extension>
3166   </xsd:complexContent>
3167 </xsd:complexType>
3168 </xsd:element>
3169 <xsd:element name="RetrieveInformationResponse">
3170   <xsd:complexType>
3171     <xsd:annotation>
3172       <xsd:documentation>The response to a RetrieveInformation request.</xsd:documentation>
3173     </xsd:annotation>
3174     <xsd:complexContent>
3175       <xsd:extension base="tns:AggregateResponseTemplate" />
3176     </xsd:complexContent>
3177   </xsd:complexType>
3178 </xsd:element>
3179 <xsd:element name="VerifyRequest">
3180   <xsd:complexType>
3181     <xsd:annotation>
3182       <xsd:documentation>
3183         The Verify aggregate service performs a 1:1 verification
3184         function according to system requirements and/or resources.
3185         Either the IdentityClaim or ReferenceBIR input data elements in
3186         the Identity parameter are required. If the Verify aggregate
3187         service is implemented as a synchronous service, the
3188         implementing system immediately processes the request and returns
3189         the results in the ReturnData parameter. If the Verify aggregate

```

```

3190     service is implemented as an asynchronous service, the
3191     implementing system returns a token in the ReturnData parameter,
3192     which is an indication that the request is being handled
3193     asynchronously. In this case, the GetVerifyResults service is
3194     used to poll for the results of the Verify request.
3195     </xsd:documentation>
3196   </xsd:annotation>
3197   <xsd:complexContent>
3198     <xsd:extension base="tns:AggregateRequestTemplate">
3199       <xsd:sequence>
3200         <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3201           <xsd:annotation>
3202             <xsd:documentation>Includes either the IdentityClaim or ReferenceBIR.</xsd:documentation>
3203           </xsd:annotation>
3204         </xsd:element>
3205         <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
3206           <xsd:annotation>
3207             <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
3208 member.</xsd:documentation>
3209           </xsd:annotation>
3210         </xsd:element>
3211       </xsd:sequence>
3212     </xsd:extension>
3213     </xsd:complexContent>
3214   </xsd:complexType>
3215 </xsd:element>
3216 <xsd:element name="VerifyResponse">
3217   <xsd:complexType>
3218     <xsd:annotation>
3219       <xsd:documentation>The response to a Verify request.</xsd:documentation>
3220     </xsd:annotation>
3221     <xsd:complexContent>
3222       <xsd:extension base="tns:AggregateResponseTypeTemplate">
3223         <xsd:sequence>
3224           <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3225             <xsd:annotation>
3226               <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with
3227 the Identity Claim or the Reference BIR.</xsd:documentation>
3228             </xsd:annotation>
3229           </xsd:element>
3230           <xsd:element name="Score" type="Score" minOccurs="0" maxOccurs="1">
3231             <xsd:annotation>
3232               <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3233             </xsd:annotation>
3234           </xsd:element>
3235         </xsd:sequence>
3236       </xsd:extension>
3237     </xsd:complexContent>
3238   </xsd:complexType>
3239 </xsd:element>
3240   <xsd:element name="BIASFault" type="tns:BIASFaultDetail" />
3241 </xsd:schema>
3242 </wsdl:types>

```

```

3243 <wsdl:message name="AddSubjectToGalleryRequestMessage">
3244   <wsdl:part name="AddSubjectToGalleryRequest" element="AddSubjectToGalleryRequest" />
3245 </wsdl:message>
3246 <wsdl:message name="AddSubjectToGalleryResponseMessage">
3247   <wsdl:part name="AddSubjectToGalleryResponse" element="AddSubjectToGalleryResponse" />
3248 </wsdl:message>
3249 <wsdl:message name="CheckQualityRequestMessage">
3250   <wsdl:part name="CheckQualityRequest" element="CheckQualityRequest" />
3251 </wsdl:message>
3252 <wsdl:message name="CheckQualityResponseMessage">
3253   <wsdl:part name="CheckQualityResponse" element="CheckQualityResponse" />
3254 </wsdl:message>
3255 <wsdl:message name="ClassifyBiometricDataRequestMessage">
3256   <wsdl:part name="ClassifyBiometricDataRequest" element="ClassifyBiometricDataRequest" />
3257 </wsdl:message>
3258 <wsdl:message name="ClassifyBiometricDataResponseMessage">
3259   <wsdl:part name="ClassifyBiometricDataResponse" element="ClassifyBiometricDataResponse" />
3260 </wsdl:message>
3261 <wsdl:message name="CreateSubjectRequestMessage">
3262   <wsdl:part name="CreateSubjectRequest" element="CreateSubjectRequest" />
3263 </wsdl:message>
3264 <wsdl:message name="CreateSubjectResponseMessage">
3265   <wsdl:part name="CreateSubjectResponse" element="CreateSubjectResponse" />
3266 </wsdl:message>
3267 <wsdl:message name="DeleteBiographicDataRequestMessage">
3268   <wsdl:part name="DeleteBiographicDataRequest" element="DeleteBiographicDataRequest" />
3269 </wsdl:message>
3270 <wsdl:message name="DeleteBiographicDataResponseMessage">
3271   <wsdl:part name="DeleteBiographicDataResponse" element="DeleteBiographicDataResponse" />
3272 </wsdl:message>
3273 <wsdl:message name="DeleteBiometricDataRequestMessage">
3274   <wsdl:part name="DeleteBiometricDataRequest" element="DeleteBiometricDataRequest" />
3275 </wsdl:message>
3276 <wsdl:message name="DeleteBiometricDataResponseMessage">
3277   <wsdl:part name="DeleteBiometricDataResponse" element="DeleteBiometricDataResponse" />
3278 </wsdl:message>
3279 <wsdl:message name="DeleteSubjectRequestMessage">
3280   <wsdl:part name="DeleteSubjectRequest" element="DeleteSubjectRequest" />
3281 </wsdl:message>
3282 <wsdl:message name="DeleteSubjectResponseMessage">
3283   <wsdl:part name="DeleteSubjectResponse" element="DeleteSubjectResponse" />
3284 </wsdl:message>
3285 <wsdl:message name="DeleteSubjectFromGalleryRequestMessage">
3286   <wsdl:part name="DeleteSubjectFromGalleryRequest" element="DeleteSubjectFromGalleryRequest" />
3287 </wsdl:message>
3288 <wsdl:message name="DeleteSubjectFromGalleryResponseMessage">
3289   <wsdl:part name="DeleteSubjectFromGalleryResponse" element="DeleteSubjectFromGalleryResponse" />
3290 </wsdl:message>
3291 <wsdl:message name="GetIdentifySubjectResultsRequestMessage">
3292   <wsdl:part name="GetIdentifySubjectResultsRequest" element="GetIdentifySubjectResultsRequest" />
3293 </wsdl:message>
3294 <wsdl:message name="GetIdentifySubjectResultsResponseMessage">
3295   <wsdl:part name="GetIdentifySubjectResultsResponse" element="GetIdentifySubjectResultsResponse" />

```

```

3296 </wsdl:message>
3297 <wsdl:message name="IdentifySubjectRequestMessage">
3298   <wsdl:part name="IdentifySubjectRequest" element="IdentifySubjectRequest" />
3299 </wsdl:message>
3300 <wsdl:message name="IdentifySubjectResponseMessage">
3301   <wsdl:part name="IdentifySubjectResponse" element="IdentifySubjectResponse" />
3302 </wsdl:message>
3303 <wsdl:message name="ListBiographicDataRequestMessage">
3304   <wsdl:part name="ListBiographicDataRequest" element="ListBiographicDataRequest" />
3305 </wsdl:message>
3306 <wsdl:message name="ListBiographicDataResponseMessage">
3307   <wsdl:part name="ListBiographicDataResponse" element="ListBiographicDataResponse" />
3308 </wsdl:message>
3309 <wsdl:message name="ListBiometricDataRequestMessage">
3310   <wsdl:part name="ListBiometricDataRequest" element="ListBiometricDataRequest" />
3311 </wsdl:message>
3312 <wsdl:message name="ListBiometricDataResponseMessage">
3313   <wsdl:part name="ListBiometricDataResponse" element="ListBiometricDataResponse" />
3314 </wsdl:message>
3315 <wsdl:message name="PerformFusionRequestMessage">
3316   <wsdl:part name="PerformFusionRequest" element="PerformFusionRequest" />
3317 </wsdl:message>
3318 <wsdl:message name="PerformFusionResponseMessage">
3319   <wsdl:part name="PerformFusionResponse" element="PerformFusionResponse" />
3320 </wsdl:message>
3321 <wsdl:message name="QueryCapabilitiesRequestMessage">
3322   <wsdl:part name="QueryCapabilitiesRequest" element="QueryCapabilitiesRequest" />
3323 </wsdl:message>
3324 <wsdl:message name="QueryCapabilitiesResponseMessage">
3325   <wsdl:part name="QueryCapabilitiesResponse" element="QueryCapabilitiesResponse" />
3326 </wsdl:message>
3327 <wsdl:message name="RetrieveBiographicInformationRequestMessage">
3328   <wsdl:part name="RetrieveBiographicInformationRequest" element="RetrieveBiographicInformationRequest" />
3329 </wsdl:message>
3330 <wsdl:message name="RetrieveBiographicInformationResponseMessage">
3331   <wsdl:part name="RetrieveBiographicInformationResponse" element="RetrieveBiographicInformationResponse" />
3332 </wsdl:message>
3333 <wsdl:message name="RetrieveBiometricInformationRequestMessage">
3334   <wsdl:part name="RetrieveBiometricInformationRequest" element="RetrieveBiometricInformationRequest" />
3335 </wsdl:message>
3336 <wsdl:message name="RetrieveBiometricInformationResponseMessage">
3337   <wsdl:part name="RetrieveBiometricInformationResponse" element="RetrieveBiometricInformationResponse" />
3338 </wsdl:message>
3339 <wsdl:message name="SetBiographicDataRequestMessage">
3340   <wsdl:part name="SetBiographicDataRequest" element="SetBiographicDataRequest" />
3341 </wsdl:message>
3342 <wsdl:message name="SetBiographicDataResponseMessage">
3343   <wsdl:part name="SetBiographicDataResponse" element="SetBiographicDataResponse" />
3344 </wsdl:message>
3345 <wsdl:message name="SetBiometricDataRequestMessage">
3346   <wsdl:part name="SetBiometricDataRequest" element="SetBiometricDataRequest" />
3347 </wsdl:message>
3348 <wsdl:message name="SetBiometricDataResponseMessage">

```

```

3349 <wsdl:part name="SetBiometricDataResponse" element="SetBiometricDataResponse" />
3350 </wsdl:message>
3351 <wsdl:message name="TransformBiometricDataRequestMessage">
3352   <wsdl:part name="TransformBiometricDataRequest" element="TransformBiometricDataRequest" />
3353 </wsdl:message>
3354 <wsdl:message name="TransformBiometricDataResponseMessage">
3355   <wsdl:part name="TransformBiometricDataResponse" element="TransformBiometricDataResponse" />
3356 </wsdl:message>
3357 <wsdl:message name="UpdateBiographicDataRequestMessage">
3358   <wsdl:part name="UpdateBiographicDataRequest" element="UpdateBiographicDataRequest" />
3359 </wsdl:message>
3360 <wsdl:message name="UpdateBiographicDataResponseMessage">
3361   <wsdl:part name="UpdateBiographicDataResponse" element="UpdateBiographicDataResponse" />
3362 </wsdl:message>
3363 <wsdl:message name="UpdateBiometricDataRequestMessage">
3364   <wsdl:part name="UpdateBiometricDataRequest" element="UpdateBiometricDataRequest" />
3365 </wsdl:message>
3366 <wsdl:message name="UpdateBiometricDataResponseMessage">
3367   <wsdl:part name="UpdateBiometricDataResponse" element="UpdateBiometricDataResponse" />
3368 </wsdl:message>
3369 <wsdl:message name="VerifySubjectRequestMessage">
3370   <wsdl:part name="VerifySubjectRequest" element="VerifySubjectRequest" />
3371 </wsdl:message>
3372 <wsdl:message name="VerifySubjectResponseMessage">
3373   <wsdl:part name="VerifySubjectResponse" element="VerifySubjectResponse" />
3374 </wsdl:message>
3375 <wsdl:message name="EnrollRequestMessage">
3376   <wsdl:part name="EnrollRequest" element="EnrollRequest" />
3377 </wsdl:message>
3378 <wsdl:message name="EnrollResponseMessage">
3379   <wsdl:part name="EnrollResponse" element="EnrollResponse" />
3380 </wsdl:message>
3381 <wsdl:message name="GetEnrollResultsRequestMessage">
3382   <wsdl:part name="GetEnrollResultsRequest" element="GetEnrollResultsRequest" />
3383 </wsdl:message>
3384 <wsdl:message name="GetEnrollResultsResponseMessage">
3385   <wsdl:part name="GetEnrollResultsResponse" element="GetEnrollResultsResponse" />
3386 </wsdl:message>
3387 <wsdl:message name="GetIdentifyResultsRequestMessage">
3388   <wsdl:part name="GetIdentifyResultsRequest" element="GetIdentifyResultsRequest" />
3389 </wsdl:message>
3390 <wsdl:message name="GetIdentifyResultsResponseMessage">
3391   <wsdl:part name="GetIdentifyResultsResponse" element="GetIdentifyResultsResponse" />
3392 </wsdl:message>
3393 <wsdl:message name="GetVerifyResultsRequestMessage">
3394   <wsdl:part name="GetVerifyResultsRequest" element="GetVerifyResultsRequest" />
3395 </wsdl:message>
3396 <wsdl:message name="GetVerifyResultsResponseMessage">
3397   <wsdl:part name="GetVerifyResultsResponse" element="GetVerifyResultsResponse" />
3398 </wsdl:message>
3399 <wsdl:message name="IdentifyRequestMessage">
3400   <wsdl:part name="IdentifyRequest" element="IdentifyRequest" />
3401 </wsdl:message>

```

```

3402 <wsdl:message name="IdentifyResponseMessage">
3403   <wsdl:part name="IdentifyResponse" element="IdentifyResponse" />
3404 </wsdl:message>
3405 <wsdl:message name="RetrieveInformationRequestMessage">
3406   <wsdl:part name="RetrieveInformationRequest" element="RetrieveInformationRequest" />
3407 </wsdl:message>
3408 <wsdl:message name="RetrieveInformationResponseMessage">
3409   <wsdl:part name="RetrieveInformationResponse" element="RetrieveInformationResponse" />
3410 </wsdl:message>
3411 <wsdl:message name="VerifyRequestMessage">
3412   <wsdl:part name="VerifyRequest" element="VerifyRequest" />
3413 </wsdl:message>
3414 <wsdl:message name="VerifyResponseMessage">
3415   <wsdl:part name="VerifyResponse" element="VerifyResponse" />
3416 </wsdl:message>
3417 <wsdl:portType name="BIAS_v1">
3418   <wsdl:operation name="AddSubjectToGallery">
3419     <wsdl:input message="tns:AddSubjectToGalleryRequestMessage" name="AddSubjectToGalleryRequest" />
3420     <wsdl:output message="tns:AddSubjectToGalleryResponseMessage" name="AddSubjectToGalleryResponse" />
3421   </wsdl:operation>
3422   <wsdl:operation name="CheckQuality">
3423     <wsdl:input message="tns:CheckQualityRequestMessage" name="CheckQualityRequest" />
3424     <wsdl:output message="tns:CheckQualityResponseMessage" name="CheckQualityResponse" />
3425   </wsdl:operation>
3426   <wsdl:operation name="ClassifyBiometricData">
3427     <wsdl:input message="tns:ClassifyBiometricDataRequestMessage" name="ClassifyBiometricDataRequest" />
3428     <wsdl:output message="tns:ClassifyBiometricDataResponseMessage" name="ClassifyBiometricDataResponse" />
3429   </wsdl:operation>
3430   <wsdl:operation name="CreateSubject">
3431     <wsdl:input message="tns>CreateSubjectRequestMessage" name="CreateSubjectRequest" />
3432     <wsdl:output message="tns>CreateSubjectResponseMessage" name="CreateSubjectResponse" />
3433   </wsdl:operation>
3434   <wsdl:operation name="DeleteBiographicData">
3435     <wsdl:input message="tns>DeleteBiographicDataRequestMessage" name="DeleteBiographicDataRequest" />
3436     <wsdl:output message="tns>DeleteBiographicDataResponseMessage" name="DeleteBiographicDataResponse" />
3437   </wsdl:operation>
3438   <wsdl:operation name="DeleteBiometricData">
3439     <wsdl:input message="tns>DeleteBiometricDataRequestMessage" name="DeleteBiometricDataRequest" />
3440     <wsdl:output message="tns>DeleteBiometricDataResponseMessage" name="DeleteBiometricDataResponse" />
3441   </wsdl:operation>
3442   <wsdl:operation name="DeleteSubject">
3443     <wsdl:input message="tns>DeleteSubjectRequestMessage" name="DeleteSubjectRequest" />
3444     <wsdl:output message="tns>DeleteSubjectResponseMessage" name="DeleteSubjectResponse" />
3445   </wsdl:operation>
3446   <wsdl:operation name="DeleteSubjectFromGallery">
3447     <wsdl:input message="tns>DeleteSubjectFromGalleryRequestMessage" name="DeleteSubjectFromGalleryRequest" />
3448     <wsdl:output message="tns>DeleteSubjectFromGalleryResponseMessage" name="DeleteSubjectFromGalleryResponse" />
3449   </wsdl:operation>
3450   <wsdl:operation name="GetIdentifySubjectResults">
3451     <wsdl:input message="tns>GetIdentifySubjectResultsRequestMessage" name="GetIdentifySubjectResultsRequest" />

```

```

3455      <wsdl:output message="tns:GetIdentifySubjectResultsResponseMessage"
3456      name="GetIdentifySubjectResultsResponse" />
3457      </wsdl:operation>
3458      <wsdl:operation name="IdentifySubject">
3459          <wsdl:input message="tns:IdentifySubjectRequestMessage" name="IdentifySubjectRequest" />
3460          <wsdl:output message="tns:IdentifySubjectResponseMessage" name="IdentifySubjectResponse" />
3461      </wsdl:operation>
3462      <wsdl:operation name="ListBiographicData">
3463          <wsdl:input message="tns>ListBiographicDataRequestMessage" name="ListBiographicDataRequest" />
3464          <wsdl:output message="tns>ListBiographicDataResponseMessage" name="ListBiographicDataResponse" />
3465      </wsdl:operation>
3466      <wsdl:operation name="ListBiometricData">
3467          <wsdl:input message="tns>ListBiometricDataRequestMessage" name="ListBiometricDataRequest" />
3468          <wsdl:output message="tns>ListBiometricDataResponseMessage" name="ListBiometricDataResponse" />
3469      </wsdl:operation>
3470      <wsdl:operation name="PerformFusion">
3471          <wsdl:input message="tns:PerformFusionRequestMessage" name="PerformFusionRequest" />
3472          <wsdl:output message="tns:PerformFusionResponseMessage" name="PerformFusionResponse" />
3473      </wsdl:operation>
3474      <wsdl:operation name="QueryCapabilities">
3475          <wsdl:input message="tns:QueryCapabilitiesRequestMessage" name="QueryCapabilitiesRequest" />
3476          <wsdl:output message="tns:QueryCapabilitiesResponseMessage" name="QueryCapabilitiesResponse" />
3477      </wsdl:operation>
3478      <wsdl:operation name="RetrieveBiographicInformation">
3479          <wsdl:input message="tns:RetrieveBiographicInformationRequestMessage" name="RetrieveBiographicInformationRequest" />
3480      <wsdl:output message="tns:RetrieveBiographicInformationResponseMessage" name="RetrieveBiographicInformationResponse" />
3481      </wsdl:operation>
3482      <wsdl:operation name="RetrieveBiometricInformation">
3483          <wsdl:input message="tns:RetrieveBiometricInformationRequestMessage" name="RetrieveBiometricInformationRequest" />
3484          <wsdl:output message="tns:RetrieveBiometricInformationResponseMessage" name="RetrieveBiometricInformationResponse" />
3485      </wsdl:operation>
3486      <wsdl:operation name="SetBiographicData">
3487          <wsdl:input message="tns:SetBiographicDataRequestMessage" name="SetBiographicDataRequest" />
3488          <wsdl:output message="tns:SetBiographicDataResponseMessage" name="SetBiographicDataResponse" />
3489      </wsdl:operation>
3490      <wsdl:operation name="SetBiometricData">
3491          <wsdl:input message="tns:SetBiometricDataRequestMessage" name="SetBiometricDataRequest" />
3492          <wsdl:output message="tns:SetBiometricDataResponseMessage" name="SetBiometricDataResponse" />
3493      </wsdl:operation>
3494      <wsdl:operation name="TransformBiometricData">
3495          <wsdl:input message="tns:TransformBiometricDataRequestMessage" name="TransformBiometricDataRequest" />
3496          <wsdl:output message="tns:TransformBiometricDataResponseMessage" name="TransformBiometricDataResponse" />
3497      </wsdl:operation>
3498      <wsdl:operation name="UpdateBiographicData">
3499          <wsdl:input message="tns:UpdateBiographicDataRequestMessage" name="UpdateBiographicDataRequest" />
3500          <wsdl:output message="tns:UpdateBiographicDataResponseMessage" name="UpdateBiographicDataResponse" />
3501      </wsdl:operation>
3502      <wsdl:operation name="UpdateBiometricData">

```

```

3508 <wsdl:input message="tns:UpdateBiometricDataRequestMessage" name="UpdateBiometricDataRequest" />
3509 <wsdl:output message="tns:UpdateBiometricDataResponseMessage" name="UpdateBiometricDataResponse" />
3510 </wsdl:operation>
3511 <wsdl:operation name="VerifySubject">
3512   <wsdl:input message="tns:VerifySubjectRequestMessage" name="VerifySubjectRequest" />
3513   <wsdl:output message="tns:VerifySubjectResponseMessage" name="VerifySubjectResponse" />
3514 </wsdl:operation>
3515 <wsdl:operation name="Enroll">
3516   <wsdl:input message="tns:EnrollRequestMessage" name="EnrollRequest" />
3517   <wsdl:output message="tns:EnrollResponseMessage" name="EnrollResponse" />
3518 </wsdl:operation>
3519 <wsdl:operation name="GetEnrollResults">
3520   <wsdl:input message="tns:GetEnrollResultsRequestMessage" name="GetEnrollResultsRequest" />
3521   <wsdl:output message="tns:GetEnrollResultsResponseMessage" name="GetEnrollResultsResponse" />
3522 </wsdl:operation>
3523 <wsdl:operation name="GetIdentifyResults">
3524   <wsdl:input message="tns:GetIdentifyResultsRequestMessage" name="GetIdentifyResultsRequest" />
3525   <wsdl:output message="tns:GetIdentifyResultsResponseMessage" name="GetIdentifyResultsResponse" />
3526 </wsdl:operation>
3527 <wsdl:operation name="GetVerifyResults">
3528   <wsdl:input message="tns:GetVerifyResultsRequestMessage" name="GetVerifyResultsRequest" />
3529   <wsdl:output message="tns:GetVerifyResultsResponseMessage" name="GetVerifyResultsResponse" />
3530 </wsdl:operation>
3531 <wsdl:operation name="Identify">
3532   <wsdl:input message="tns:IdentifyRequestMessage" name="IdentifyRequest" />
3533   <wsdl:output message="tns:IdentifyResponseMessage" name="IdentifyResponse" />
3534 </wsdl:operation>
3535 <wsdl:operation name="RetrieveInformation">
3536   <wsdl:input message="tns:RetrieveInformationRequestMessage" name="RetrieveInformationRequest" />
3537   <wsdl:output message="tns:RetrieveInformationResponseMessage" name="RetrieveInformationResponse" />
3538 </wsdl:operation>
3539 <wsdl:operation name="Verify">
3540   <wsdl:input message="tns:VerifyRequestMessage" name="VerifyRequest" />
3541   <wsdl:output message="tns:VerifyResponseMessage" name="VerifyResponse" />
3542 </wsdl:operation>
3543 </wsdl:portType>
3544 <wsdl:binding type="tns:BIAS_v1" name="BIAS_v1HttpBinding">
3545   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
3546   <wsdl:operation name="AddSubjectToGallery">
3547     <soap:operation soapAction="AddSubjectToGallery" />
3548     <wsdl:input name="AddSubjectToGalleryRequest">
3549       <soap:body use="literal" />
3550     </wsdl:input>
3551     <wsdl:output name="AddSubjectToGalleryResponse">
3552       <soap:body use="literal" />
3553     </wsdl:output>
3554     <wsdl:fault name="BIASFault">
3555       <soap:fault name="BIASFault" use="literal" />
3556     </wsdl:fault>
3557   </wsdl:operation>
3558   <wsdl:operation name="CheckQuality">
3559     <soap:operation soapAction="CheckQuality" />
3560     <wsdl:input name="CheckQualityRequest">

```

```

3561      <soap:body use="literal" />
3562    </wsdl:input>
3563    <wsdl:output name="CheckQualityResponse">
3564      <soap:body use="literal" />
3565    </wsdl:output>
3566    <wsdl:fault name="BIASFault">
3567      <soap:fault name="BIASFault" use="literal" />
3568    </wsdl:fault>
3569  </wsdl:operation>
3570  <wsdl:operation name="ClassifyBiometricData">
3571    <soap:operation soapAction="ClassifyBiometricData" />
3572    <wsdl:input name="ClassifyBiometricDataRequest">
3573      <soap:body use="literal" />
3574    </wsdl:input>
3575    <wsdl:output name="ClassifyBiometricDataResponse">
3576      <soap:body use="literal" />
3577    </wsdl:output>
3578    <wsdl:fault name="BIASFault">
3579      <soap:fault name="BIASFault" use="literal" />
3580    </wsdl:fault>
3581  </wsdl:operation>
3582  <wsdl:operation name="CreateSubject">
3583    <soap:operation soapAction="CreateSubject" />
3584    <wsdl:input name="CreateSubjectRequest">
3585      <soap:body use="literal" />
3586    </wsdl:input>
3587    <wsdl:output name="CreateSubjectResponse">
3588      <soap:body use="literal" />
3589    </wsdl:output>
3590    <wsdl:fault name="BIASFault">
3591      <soap:fault name="BIASFault" use="literal" />
3592    </wsdl:fault>
3593  </wsdl:operation>
3594  <wsdl:operation name="DeleteBiographicData">
3595    <soap:operation soapAction="DeleteBiographicData" />
3596    <wsdl:input name="DeleteBiographicDataRequest">
3597      <soap:body use="literal" />
3598    </wsdl:input>
3599    <wsdl:output name="DeleteBiographicDataResponse">
3600      <soap:body use="literal" />
3601    </wsdl:output>
3602    <wsdl:fault name="BIASFault">
3603      <soap:fault name="BIASFault" use="literal" />
3604    </wsdl:fault>
3605  </wsdl:operation>
3606  <wsdl:operation name="DeleteBiometricData">
3607    <soap:operation soapAction="DeleteBiometricData" />
3608    <wsdl:input name="DeleteBiometricDataRequest">
3609      <soap:body use="literal" />
3610    </wsdl:input>
3611    <wsdl:output name="DeleteBiometricDataResponse">
3612      <soap:body use="literal" />
3613    </wsdl:output>

```

```

3614 <wsdl:fault name="BIASFault">
3615   <soap:fault name="BIASFault" use="literal" />
3616 </wsdl:fault>
3617 </wsdl:operation>
3618 <wsdl:operation name="DeleteSubject">
3619   <soap:operation soapAction="DeleteSubject" />
3620   <wsdl:input name="DeleteSubjectRequest">
3621     <soap:body use="literal" />
3622   </wsdl:input>
3623   <wsdl:output name="DeleteSubjectResponse">
3624     <soap:body use="literal" />
3625   </wsdl:output>
3626   <wsdl:fault name="BIASFault">
3627     <soap:fault name="BIASFault" use="literal" />
3628   </wsdl:fault>
3629 </wsdl:operation>
3630 <wsdl:operation name="DeleteSubjectFromGallery">
3631   <soap:operation soapAction="DeleteSubjectFromGallery" />
3632   <wsdl:input name="DeleteSubjectFromGalleryRequest">
3633     <soap:body use="literal" />
3634   </wsdl:input>
3635   <wsdl:output name="DeleteSubjectFromGalleryResponse">
3636     <soap:body use="literal" />
3637   </wsdl:output>
3638   <wsdl:fault name="BIASFault">
3639     <soap:fault name="BIASFault" use="literal" />
3640   </wsdl:fault>
3641 </wsdl:operation>
3642 <wsdl:operation name="GetIdentifySubjectResults">
3643   <soap:operation soapAction="GetIdentifySubjectResults" />
3644   <wsdl:input name="GetIdentifySubjectResultsRequest">
3645     <soap:body use="literal" />
3646   </wsdl:input>
3647   <wsdl:output name="GetIdentifySubjectResultsResponse">
3648     <soap:body use="literal" />
3649   </wsdl:output>
3650   <wsdl:fault name="BIASFault">
3651     <soap:fault name="BIASFault" use="literal" />
3652   </wsdl:fault>
3653 </wsdl:operation>
3654 <wsdl:operation name="IdentifySubject">
3655   <soap:operation soapAction="IdentifySubject" />
3656   <wsdl:input name="IdentifySubjectRequest">
3657     <soap:body use="literal" />
3658   </wsdl:input>
3659   <wsdl:output name="IdentifySubjectResponse">
3660     <soap:body use="literal" />
3661   </wsdl:output>
3662   <wsdl:fault name="BIASFault">
3663     <soap:fault name="BIASFault" use="literal" />
3664   </wsdl:fault>
3665 </wsdl:operation>
3666 <wsdl:operation name="ListBiographicData">

```

```

3667 <soap:operation soapAction="ListBiographicData" />
3668 <wsdl:input name="ListBiographicDataRequest">
3669   <soap:body use="literal" />
3670 </wsdl:input>
3671 <wsdl:output name="ListBiographicDataResponse">
3672   <soap:body use="literal" />
3673 </wsdl:output>
3674 <wsdl:fault name="BIASFault">
3675   <soap:fault name="BIASFault" use="literal" />
3676 </wsdl:fault>
3677 </wsdl:operation>
3678 <wsdl:operation name="ListBiometricData">
3679   <soap:operation soapAction="ListBiometricData" />
3680 <wsdl:input name="ListBiometricDataRequest">
3681   <soap:body use="literal" />
3682 </wsdl:input>
3683 <wsdl:output name="ListBiometricDataResponse">
3684   <soap:body use="literal" />
3685 </wsdl:output>
3686 <wsdl:fault name="BIASFault">
3687   <soap:fault name="BIASFault" use="literal" />
3688 </wsdl:fault>
3689 </wsdl:operation>
3690 <wsdl:operation name="PerformFusion">
3691   <soap:operation soapAction="PerformFusion" />
3692 <wsdl:input name="PerformFusionRequest">
3693   <soap:body use="literal" />
3694 </wsdl:input>
3695 <wsdl:output name="PerformFusionResponse">
3696   <soap:body use="literal" />
3697 </wsdl:output>
3698 <wsdl:fault name="BIASFault">
3699   <soap:fault name="BIASFault" use="literal" />
3700 </wsdl:fault>
3701 </wsdl:operation>
3702 <wsdl:operation name="QueryCapabilities">
3703   <soap:operation soapAction="QueryCapabilities" />
3704 <wsdl:input name="QueryCapabilitiesRequest">
3705   <soap:body use="literal" />
3706 </wsdl:input>
3707 <wsdl:output name="QueryCapabilitiesResponse">
3708   <soap:body use="literal" />
3709 </wsdl:output>
3710 <wsdl:fault name="BIASFault">
3711   <soap:fault name="BIASFault" use="literal" />
3712 </wsdl:fault>
3713 </wsdl:operation>
3714 <wsdl:operation name="RetrieveBiographicInformation">
3715   <soap:operation soapAction="RetrieveBiographicInformation" />
3716 <wsdl:input name="RetrieveBiographicInformationRequest">
3717   <soap:body use="literal" />
3718 </wsdl:input>
3719 <wsdl:output name="RetrieveBiographicInformationResponse">

```

```

3720      <soap:body use="literal" />
3721    </wsdl:output>
3722    <wsdl:fault name="BIASFault">
3723      <soap:fault name="BIASFault" use="literal" />
3724    </wsdl:fault>
3725  </wsdl:operation>
3726  <wsdl:operation name="RetrieveBiometricInformation">
3727    <soap:operation soapAction="RetrieveBiometricInformation" />
3728    <wsdl:input name="RetrieveBiometricInformationRequest">
3729      <soap:body use="literal" />
3730    </wsdl:input>
3731    <wsdl:output name="RetrieveBiometricInformationResponse">
3732      <soap:body use="literal" />
3733    </wsdl:output>
3734    <wsdl:fault name="BIASFault">
3735      <soap:fault name="BIASFault" use="literal" />
3736    </wsdl:fault>
3737  </wsdl:operation>
3738  <wsdl:operation name="SetBiographicData">
3739    <soap:operation soapAction="SetBiographicData" />
3740    <wsdl:input name="SetBiographicDataRequest">
3741      <soap:body use="literal" />
3742    </wsdl:input>
3743    <wsdl:output name="SetBiographicDataResponse">
3744      <soap:body use="literal" />
3745    </wsdl:output>
3746    <wsdl:fault name="BIASFault">
3747      <soap:fault name="BIASFault" use="literal" />
3748    </wsdl:fault>
3749  </wsdl:operation>
3750  <wsdl:operation name="SetBiometricData">
3751    <soap:operation soapAction="SetBiometricData" />
3752    <wsdl:input name="SetBiometricDataRequest">
3753      <soap:body use="literal" />
3754    </wsdl:input>
3755    <wsdl:output name="SetBiometricDataResponse">
3756      <soap:body use="literal" />
3757    </wsdl:output>
3758    <wsdl:fault name="BIASFault">
3759      <soap:fault name="BIASFault" use="literal" />
3760    </wsdl:fault>
3761  </wsdl:operation>
3762  <wsdl:operation name="TransformBiometricData">
3763    <soap:operation soapAction="TransformBiometricData" />
3764    <wsdl:input name="TransformBiometricDataRequest">
3765      <soap:body use="literal" />
3766    </wsdl:input>
3767    <wsdl:output name="TransformBiometricDataResponse">
3768      <soap:body use="literal" />
3769    </wsdl:output>
3770    <wsdl:fault name="BIASFault">
3771      <soap:fault name="BIASFault" use="literal" />
3772    </wsdl:fault>

```

```

3773    </wsdl:operation>
3774    <wsdl:operation name="UpdateBiographicData">
3775      <soap:operation soapAction="UpdateBiographicData" />
3776      <wsdl:input name="UpdateBiographicDataRequest">
3777        <soap:body use="literal" />
3778      </wsdl:input>
3779      <wsdl:output name="UpdateBiographicDataResponse">
3780        <soap:body use="literal" />
3781      </wsdl:output>
3782      <wsdl:fault name="BIASFault">
3783        <soap:fault name="BIASFault" use="literal" />
3784      </wsdl:fault>
3785    </wsdl:operation>
3786    <wsdl:operation name="UpdateBiometricData">
3787      <soap:operation soapAction="UpdateBiometricData" />
3788      <wsdl:input name="UpdateBiometricDataRequest">
3789        <soap:body use="literal" />
3790      </wsdl:input>
3791      <wsdl:output name="UpdateBiometricDataResponse">
3792        <soap:body use="literal" />
3793      </wsdl:output>
3794      <wsdl:fault name="BIASFault">
3795        <soap:fault name="BIASFault" use="literal" />
3796      </wsdl:fault>
3797    </wsdl:operation>
3798    <wsdl:operation name="VerifySubject">
3799      <soap:operation soapAction="VerifySubject" />
3800      <wsdl:input name="VerifySubjectRequest">
3801        <soap:body use="literal" />
3802      </wsdl:input>
3803      <wsdl:output name="VerifySubjectResponse">
3804        <soap:body use="literal" />
3805      </wsdl:output>
3806      <wsdl:fault name="BIASFault">
3807        <soap:fault name="BIASFault" use="literal" />
3808      </wsdl:fault>
3809    </wsdl:operation>
3810    <wsdl:operation name="Enroll">
3811      <soap:operation soapAction="Enroll" />
3812      <wsdl:input name="EnrollRequest">
3813        <soap:body use="literal" />
3814      </wsdl:input>
3815      <wsdl:output name="EnrollResponse">
3816        <soap:body use="literal" />
3817      </wsdl:output>
3818      <wsdl:fault name="BIASFault">
3819        <soap:fault name="BIASFault" use="literal" />
3820      </wsdl:fault>
3821    </wsdl:operation>
3822    <wsdl:operation name="GetEnrollResults">
3823      <soap:operation soapAction="GetEnrollResults" />
3824      <wsdl:input name="GetEnrollResultsRequest">
3825        <soap:body use="literal" />

```

```

3826    </wsdl:input>
3827    <wsdl:output name="GetEnrollResultsResponse">
3828      <soap:body use="literal" />
3829    </wsdl:output>
3830    <wsdl:fault name="BIASFault">
3831      <soap:fault name="BIASFault" use="literal" />
3832    </wsdl:fault>
3833  </wsdl:operation>
3834  <wsdl:operation name="GetIdentifyResults">
3835    <soap:operation soapAction="GetIdentifyResults" />
3836    <wsdl:input name="GetIdentifyResultsRequest">
3837      <soap:body use="literal" />
3838    </wsdl:input>
3839    <wsdl:output name="GetIdentifyResultsResponse">
3840      <soap:body use="literal" />
3841    </wsdl:output>
3842    <wsdl:fault name="BIASFault">
3843      <soap:fault name="BIASFault" use="literal" />
3844    </wsdl:fault>
3845  </wsdl:operation>
3846  <wsdl:operation name="GetVerifyResults">
3847    <soap:operation soapAction="GetVerifyResults" />
3848    <wsdl:input name="GetVerifyResultsRequest">
3849      <soap:body use="literal" />
3850    </wsdl:input>
3851    <wsdl:output name="GetVerifyResultsResponse">
3852      <soap:body use="literal" />
3853    </wsdl:output>
3854    <wsdl:fault name="BIASFault">
3855      <soap:fault name="BIASFault" use="literal" />
3856    </wsdl:fault>
3857  </wsdl:operation>
3858  <wsdl:operation name="Identify">
3859    <soap:operation soapAction="Identify" />
3860    <wsdl:input name="IdentifyRequest">
3861      <soap:body use="literal" />
3862    </wsdl:input>
3863    <wsdl:output name="IdentifyResponse">
3864      <soap:body use="literal" />
3865    </wsdl:output>
3866    <wsdl:fault name="BIASFault">
3867      <soap:fault name="BIASFault" use="literal" />
3868    </wsdl:fault>
3869  </wsdl:operation>
3870  <wsdl:operation name="RetrieveInformation">
3871    <soap:operation soapAction="RetrieveInformation" />
3872    <wsdl:input name="RetrieveInformationRequest">
3873      <soap:body use="literal" />
3874    </wsdl:input>
3875    <wsdl:output name="RetrieveInformationResponse">
3876      <soap:body use="literal" />
3877    </wsdl:output>
3878    <wsdl:fault name="BIASFault">

```

```
3879      <soap:fault name="BIASFault" use="literal" />
3880    </wsdl:fault>
3881  </wsdl:operation>
3882  <wsdl:operation name="Verify">
3883    <soap:operation soapAction="Verify" />
3884    <wsdl:input name="VerifyRequest">
3885      <soap:body use="literal" />
3886    </wsdl:input>
3887    <wsdl:output name="VerifyResponse">
3888      <soap:body use="literal" />
3889    </wsdl:output>
3890    <wsdl:fault name="BIASFault">
3891      <soap:fault name="BIASFault" use="literal" />
3892    </wsdl:fault>
3893  </wsdl:operation>
3894 </wsdl:binding>
3895 <wsdl:service name="BIAS_v1">
3896   <wsdl:port name="BIAS_v1HttpBindingPort" binding="tns:BIAS_v1HttpBinding">
3897     <soap:address location="http://your_URI/IRI_to_your_BIAS_implementation" />
3898   </wsdl:port>
3899 </wsdl:service>
3900 </wsdl:definitions>
3901
```

3902 Annex B. Use Cases (non-normative)

3903 The intent of this annex is to provide operational sequence diagrams / flow charts that show how the
3904 higher level usage scenarios within [INCITS-BIAS] could be implemented using the BIAS SOAP profile.
3905

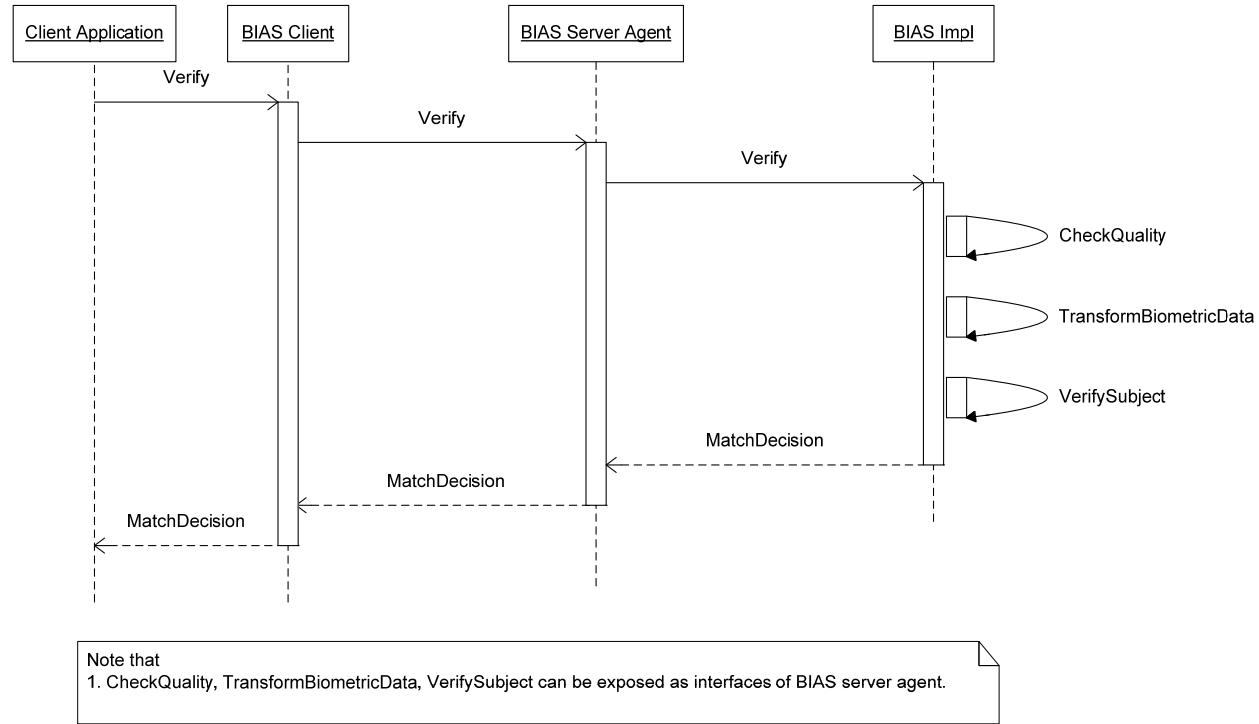
The following use cases are given:

- 3906 • Verification (synchronous/aggregate)
- 3907 • Verification (asynchronous/aggregate)
- 3908 • Verification (primitive)
- 3909 • Identification (primitive)
- 3910 • Enrollment (aggregate)
- 3911 • Enrollment (primitive)

3912 B.1 Verification Use Case

3913 This use case uses the aggregate Verify operation in which a single request results in some set of
3914 operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service
3915 provider.

3916



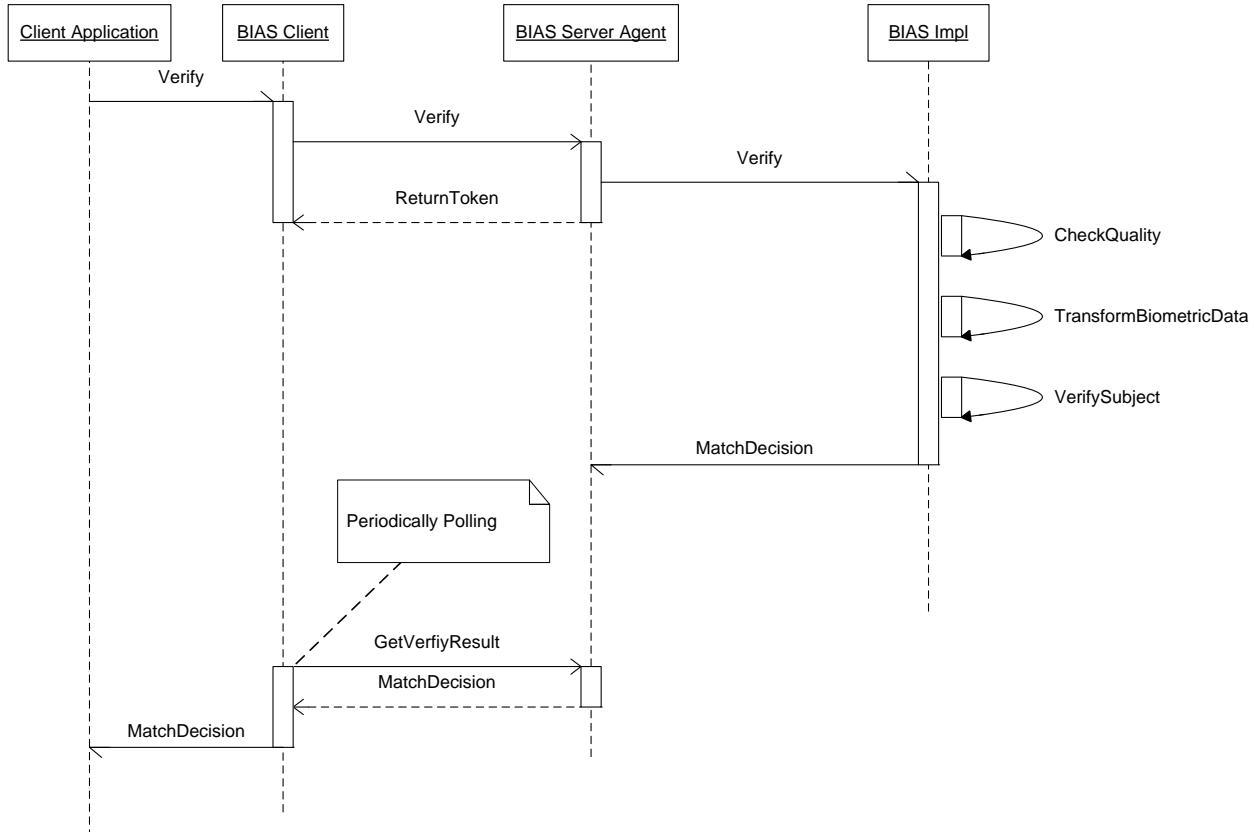
3917

3918

3919 **B.2 Asynchronous Verification**

3920 In this use case, the requester issues two requests – the BIAS Verify request to initiate the operation
3921 followed by a BIAS GetVerifyResult request to retrieve the results of that operation.

3922



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

3923

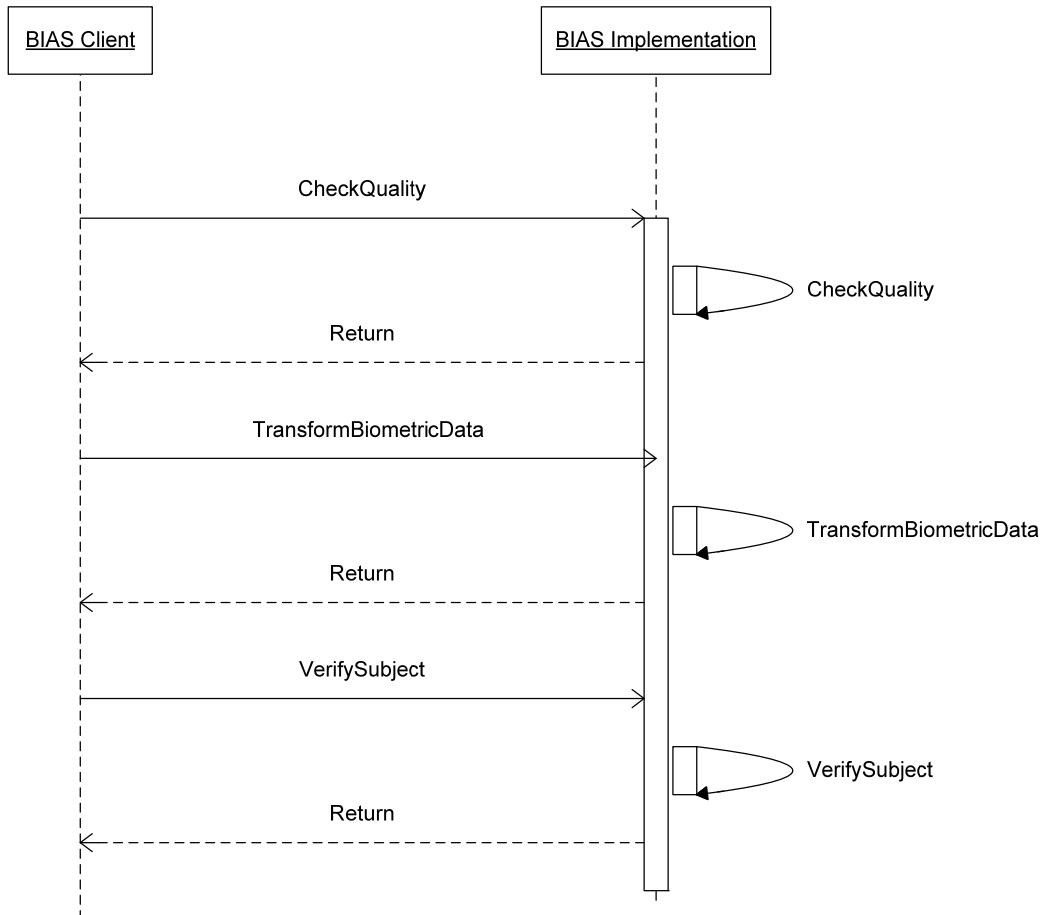
3924

3925

3926 **B.3 Primitive Verification**

3927 In this use case, the verification operation is performed as a series of requests using the BIAS primitive
3928 operations. In this case, the client rather than the service provider controls the workflow of the higher
3929 level operation.

3930



3931

3932

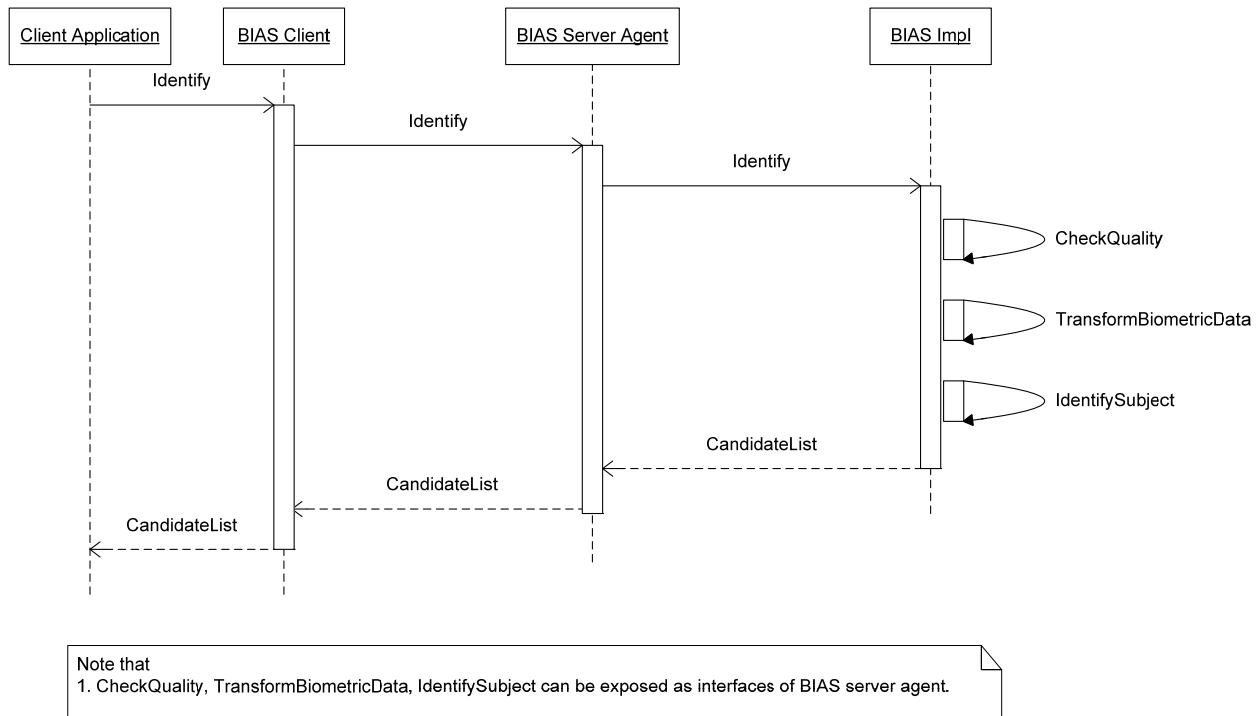
3933 **B.4 Identification Use Case**

3934 This use case uses the aggregate Identify operation in which a single request results in some set of
3935 operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service
3936 provider.

3937

3938

3939



3940

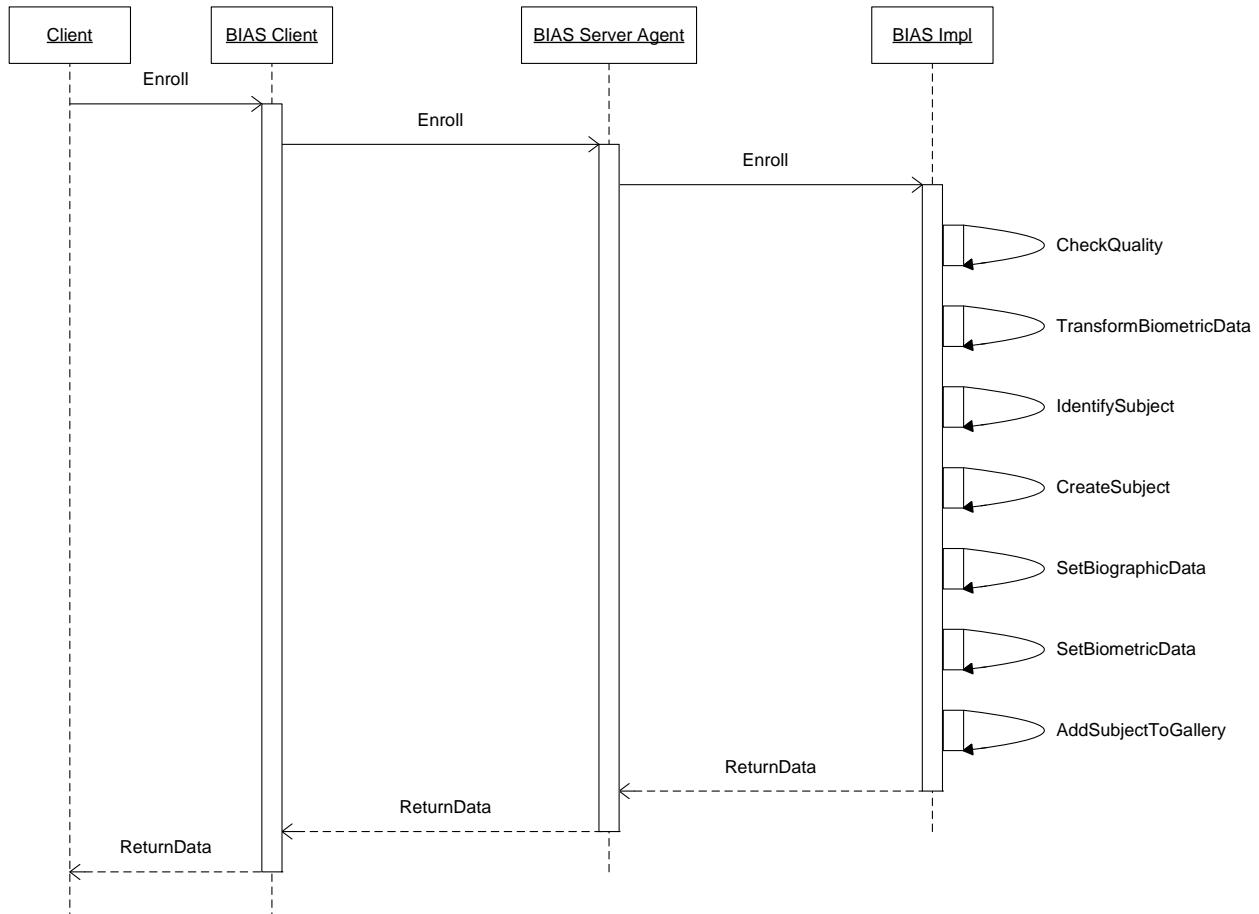
3941

3942 **B.5 Biometric Enrollment**

3943 This use case uses the aggregate Enroll operation in which a single request results in some set of
3944 operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service
3945 provider.

3946 Here, if the result of the IdentifySubject is no matches found, then the subject is added to the gallery. If a
3947 match had been found then other logic may have been applied (e.g., return candidate list, add encounter
3948 for existing subject, etc.).

3949



3950

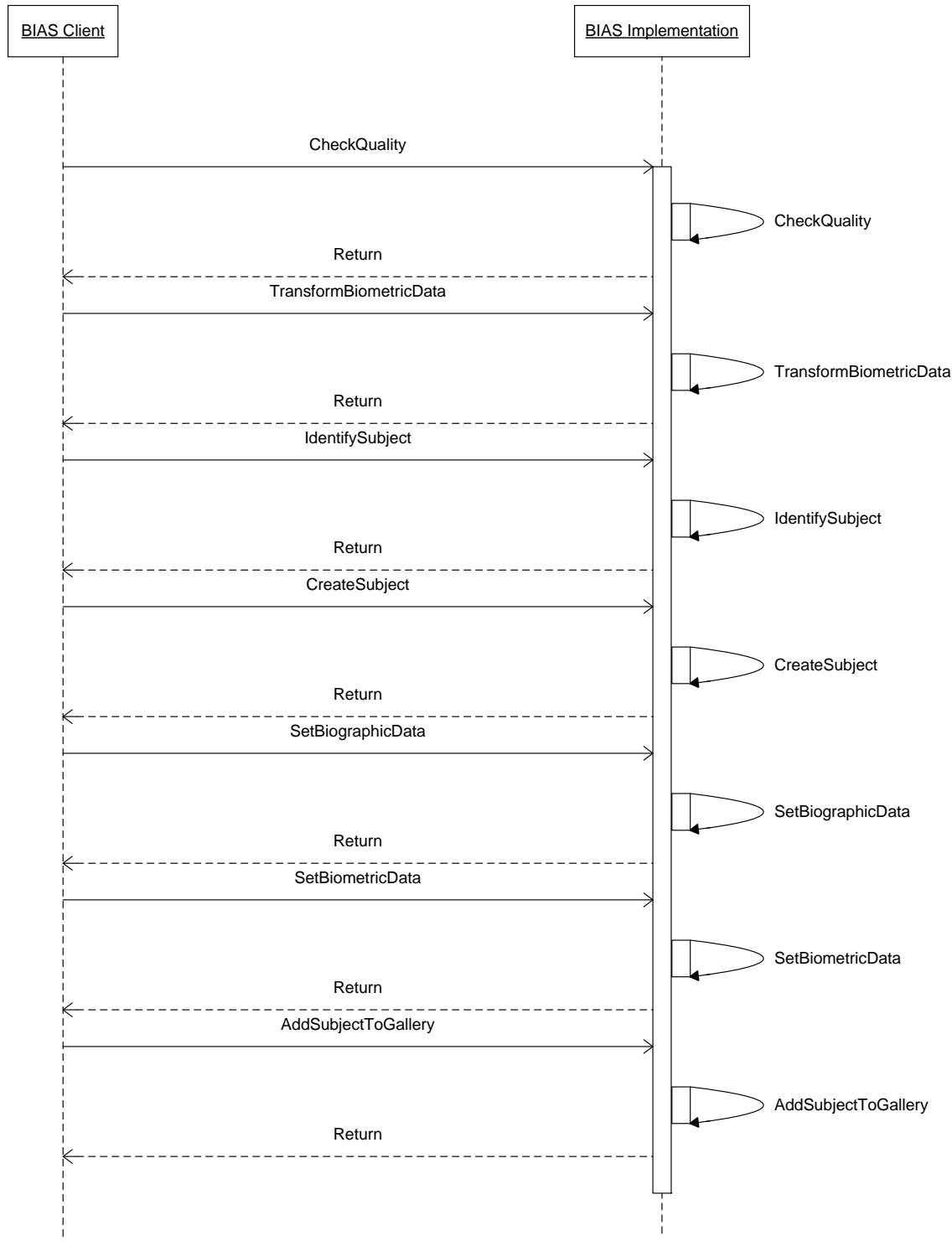
3951

3952

3953 **B.6 Primitive Enrollment**

3954 In this use case, the enrollment operation is performed as a series of requests using the BIAS primitive
3955 operations. In this case, the client rather than the service provider controls the workflow of the higher
3956 level operation.

3957



3958

3959 **Annex C. Samples (non-normative)**

3960 **C.1 Create Subject Request/Response Example**

3961 *INCITS BIAS Specification*

```
3962 <interface name="CreateSubject">
3963     <parameter name="SubjectID" type="xs:string" direction="inout"
3964     use="optional" />
3965     <parameter name="Return" type="xs:unsignedLong" direction="out" />
3966 </interface>
```

3967 *OASIS BIAS Examples*

3968 **Simple Create Subject Request:**

```
3969 POST /bias HTTP/1.1
3970 Host: www.acme.com
3971 Content-Type: application/soap+xml; charset="utf-8"
3972 Content-Length: nnnn
3973 SOAPAction: "CreateSubject"
3974 <?xml version="1.0"?>
3975 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3976     <soap:Body>
3977         <tns:CreateSubjectRequest
3978             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3979         />
3980     </soap:Body>
3981 </soap:Envelope>
```

3982 **Create Subject Request with SubjectID Parameter:**

```
3983 POST /bias HTTP/1.1
3984 Host: www.acme.com
3985 Content-Type: application/soap+xml; charset="utf-8"
3986 Content-Length: nnnn
3987 SOAPAction: "CreateSubject"
3988 <?xml version="1.0"?>
3989 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3990     <soap:Body>
3991         <tns:CreateSubjectRequest
3992             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3993             <tns:Identity>
3994                 <tns:SubjectID>123456789</tns:SubjectID>
3995             </tns:Identity>
3996         </tns:CreateSubjectRequest>
3997     </soap:Body>
```

```

3998 </soap:Envelope>

3999 Create Subject Request with Optional OASIS BIAS Content:
4000 POST /bias HTTP/1.1
4001 Host: www.acme.com
4002 Content-Type: application/soap+xml; charset="utf-8"
4003 Content-Length: nnnn
4004 SOAPAction: "CreateSubject"
4005 <?xml version="1.0"?>
4006 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4007     <soap:Body>
4008         <tns:CreateSubjectRequest
4009             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4010             <tns:GenericRequestParameters>
4011                 <tns:Application>BIAS Application</tns:Application>
4012                 <tns:ApplicationUser>BIAS User</tns:ApplicationUser>
4013             </tns:GenericRequestParameters>
4014             <tns:Identity>
4015                 <tns:SubjectID>123456789</tns:SubjectID>
4016             </tns:Identity>
4017         </tns:CreateSubjectRequest>
4018     </soap:Body>
4019 </soap:Envelope>

4020 Simple Create Subject Response:
4021 HTTP/1.1 200 OK
4022 Content-Type: application/soap+xml; charset="utf-8"
4023 Content-Length: nnnn
4024 <?xml version="1.0"?>
4025 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4026     <soap:Body>
4027         <tns:CreateSubjectResponse
4028             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4029             <tns:ResponseStatus>
4030                 <tns:Return>0</tns:Return>
4031             </tns:ResponseStatus>
4032             <tns:Identity>
4033                 <tns:SubjectID>123456789</tns:SubjectID>
4034             </tns:Identity>
4035         </tns:CreateSubjectResponse>
4036     </soap:Body>
4037 </soap:Envelope>

4038 Create Subject Response with Optional OASIS BIAS Content:

```

```

4039 HTTP/1.1 200 OK
4040 Content-Type: application/soap+xml; charset="utf-8"
4041 Content-Length: nnnn
4042 <?xml version="1.0"?>
4043 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4044     <soap:Body>
4045         <tns:CreateSubjectResponse
4046             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4047                 <tns:ResponseStatus>
4048                     <tns:Return>0</tns:Return>
4049                     <tns:Message>Subject ID 123456789 successfully
4050 created.</tns:Message>
4051                 </tns:ResponseStatus>
4052                 <tns:Identity>
4053                     <tns:SubjectID>123456789</tns:SubjectID>
4054                 </tns:Identity>
4055             </tns:CreateSubjectResponse>
4056         </soap:Body>
4057     </soap:Envelope>
4058

```

4059 C.2 Set Biographic Data Request/Response Example

4060 *INCITS BIAS Specification*

```

4061 <interface name="SetBiographicData">
4062     <parameter name="SubjectID" type="xs:string" direction="in" />
4063     <parameter name="IdentityModel" type="IdentityModelType" direction="in" />
4064
4065     <parameter name="EncounterID" type="xs:string" direction="inout" use="optional" />
4066     <parameter name="BiographicData" type="BiographicDataType" direction="in" />
4067
4068     <parameter name="Return" type="xs:unsignedLong" direction="out" />
4069
4070 </interface>

```

4071 *OASIS BIAS Examples*

4072 Set Biographic Data Request:

```

4073 POST /bias HTTP/1.1
4074 Host: www.acme.com
4075 Content-Type: application/soap+xml; charset="utf-8"
4076 Content-Length: nnnn
4077 SOAPAction: "SetBiographicData"
4078 <?xml version="1.0"?>
4079 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4080     <soap:Body>

```

```

4081     <tns:SetBiographicDataRequest
4082             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4083         <tns:Identity>
4084             <tns:SubjectID>123456789</tns:SubjectID>
4085             <tns:BiographicData>
4086                 <tns:BiographicDataItem>
4087                     <tns:Name>Last</tns:Name>
4088                     <tns>Type>string</tns>Type>
4089                     <tns:Value>Doe</tns:Value>
4090                 </tns:BiographicDataItem>
4091             </tns:BiographicData>
4092         </tns:Identity>
4093         <tns:IdentityModel>person</tns:IdentityModel>
4094     </tns:SetBiographicDataRequest>
4095     </soap:Body>
4096 </soap:Envelope>
```

4097 Set Biographic Data Response:

```

4098 HTTP/1.1 200 OK
4099 Content-Type: application/soap+xml; charset="utf-8"
4100 Content-Length: nnnn
4101 <?xml version="1.0"?>
4102 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4103     <soap:Body>
4104         <tns:SetBiographicDataResponse
4105             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4106             <tns:ResponseStatus>
4107                 <tns:Return>0</tns:Return>
4108             </tns:ResponseStatus>
4109         </tns:SetBiographicDataResponse>
4110     </soap:Body>
4111 </soap:Envelope>
4112
```

4113 C.3 Set Biometric Data Request/Response Example

4114 *INCITS BIAS Specification*

```

4115 <interface name="SetBiometricData">
4116     <parameter name="SubjectID" type="xs:string" direction="in" />
4117     <parameter name="IdentityModel" type="IdentityModelType" direction="in" />
4118     <parameter name="EncounterID" type="xs:string" direction="inout" use="optional" />
4119     <parameter name="BIRList" type="CBEFF_BIR_ListType" direction="in" />
```

```

4122      <parameter name="Return" type="xs:unsignedLong" direction="out" />
4123  </interface>

4124  OASIS BIAS Examples

4125  Set Biometric Data Request:

4126  POST /bias HTTP/1.1
4127  Host: www.acme.com
4128  Content-Type: application/soap+xml; charset="utf-8"
4129  Content-Length: nnnn
4130  SOAPAction: "SetBiometricData"
4131  <?xml version="1.0"?>
4132  <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4133      <soap:Body>
4134          <tns:SetBiometricDataRequest
4135              xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4136              <tns:Identity>
4137                  <tns:SubjectID>123456789</tns:SubjectID>
4138                  <tns:BiometricData>
4139                      <tns:BIRList>
4140                          <tns:BIR>biometric data</tns:BIR>
4141                      </tns:BIRList>
4142                  </tns:BiometricData>
4143              </tns:Identity>
4144              <tns:IdentityModel>person</tns:IdentityModel>
4145          </tns:SetBiometricDataRequest>
4146      </soap:Body>
4147  </soap:Envelope>

4148  Set Biometric Data Response:

4149  HTTP/1.1 200 OK
4150  Content-Type: application/soap+xml; charset="utf-8"
4151  Content-Length: nnnn
4152  <?xml version="1.0"?>
4153  <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
4154      <soap:Body>
4155          <tns:SetBiometricDataResponse
4156              xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
4157              <tns:ResponseStatus>
4158                  <tns:Return>0</tns:Return>
4159              </tns:ResponseStatus>
4160          </tns:SetBiometricDataResponse>
4161      </soap:Body>
4162  </soap:Envelope>
```

4163 Annex D. Acknowledgements

4164 The following individuals have participated in the creation of this specification and are gratefully
4165 acknowledged:

4166

4167 **Participants:**

4168

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. 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
Mr. Brad Wing	NIST (formerly DHS)
Mr. Michael Wittman	Raytheon
Mr. Gregory Zektser	Booz Allen Hamilton

4169

4170 * Though no longer members of the BIAS TC, these individuals contributed in the early stages of the
4171 development of this standard.

4172 In addition, the inputs from the INCITS technical committee M1 are also gratefully appreciated.

4173

Annex E. Revision History

4174

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: - Breaking Clause 3 into 2 clauses for data elements and operations - Specification of URI & IRI - Clarifications and formatting
0.10	2009-10-19	Tilton/Swayze	Expansion of conformance clause
0.11	2009-11-16	Tilton/Swayze	Miscellaneous edits and clarifications

4175