# Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0 Plus Errata 01

## OASIS Standard Incorporating Public Review Draft 01 of Errata 01

## 25 October 2012

### Specification URIs
**This version:**
http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/biasprofile-v1.0-errata01-csprd01-complete.doc (Authoritative)
http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/biasprofile-v1.0-errata01-csprd01-complete.html
http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/biasprofile-v1.0-errata01-csprd01-complete.pdf

**Previous version:**
http://docs.oasis-open.org/bias/soap-profile/v1.0/os/biasprofile-v1.0-os.doc (Authoritative)
http://docs.oasis-open.org/bias/soap-profile/v1.0/os/biasprofile-v1.0-os.html
http://docs.oasis-open.org/bias/soap-profile/v1.0/os/biasprofile-v1.0-os.pdf

**Latest version:**
http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/biasprofile-v1.0-errata01-complete.doc (Authoritative)
http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/biasprofile-v1.0-errata01-complete.html
http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/biasprofile-v1.0-errata01-complete.pdf

**Technical Committee:**
OASIS Biometric Identity Assurance Services (BIAS) Integration TC

**Chairs:**
Cathy Tilton (cathy.tilton@daon.com), Daon
Kevin Mangold (kevin.mangold@nist.gov), NIST

**Editors:**
Kevin Mangold (kevin.mangold@nist.gov), NIST
Matthew Swayze (matthew.swayze@daon.com), Daon
Cathy Tilton (cathy.tilton@daon.com), Daon

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

* *Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0 Errata 01*. 25 October 2012. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/biasprofile-v1.0-errata01-csprd01.html
* XML schema: http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/cbeff.xsd
* WSDL: http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/BIAS.wsdl

**Related work:**

This specification is related to:

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

**Declared XML namespaces:**

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

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

**Abstract:**

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

**Status:**

This document was last revised or approved by the membership of OASIS on the above date. The level of approval is also listed above. Check the "Latest 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).

**Citation format:**

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

**[BIASPROFILE]**

*Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0 Plus Errata 01*. 25 October 2012. OASIS Standard Incorporating Public Review Draft 01 of Errata 01. http://docs.oasis-open.org/bias/soap-profile/v1.0/errata01/csprd01/biasprofile-v1.0-errata01-csprd01-complete.html.

# Notices

Copyright © OASIS Open 2012. 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 name "OASIS" is a trademark 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/policies-guidelines/trademark for above guidance.

# Table of Contents

# 1 Introduction

## 1.1 Purpose/Scope

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

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

## 1.2 Overview

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

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

## 1.3 Background

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

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

## 1.4 Relationship to Other Standards

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

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

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

## 1.5 Terminology

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

The following additional terms and definitions are used:

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

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

**BIAS**
> Biometric Identity Assurance Services

**BIR**
> Biometric Information Record

**ESB**
> Enterprise Service Bus

**HTTP**
> HyperText Transfer Protocol

**HTTPS**
> HyperText Transfer Protocol over SSL or HTTP Secure

**IRI**
> Internationalized Resource Identifier

**SOA**
> Service-Oriented Architecture

**SOAP**
> Simple Object Access Protocol

**SSL**
> Secure Sockets Layer

**TLS**
> Transport Layer Security

**UDDI**
> Universal Description, Discovery, and Integration

**URI**
> Uniform Resource Identifier

**VPN**
> Virtual Private Network

**WSDL**
> Web Services Description Language

**WSS**
> Web Services Security

95 **XML**
96     eXtensible Markup Language
97
98
99 **CBEFF**
100     Common Biometric Exchange Formats Framework - data elements and BIR formats specified in
101     ISO/IEC 19785-1
102 **BIAS implementation**
103     software entity that is capable of creating, processing, sending, and receiving BIAS messages

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

107 **BIAS message**
108     message that can be sent from a BIAS endpoint to another BIAS endpoint through a BIAS link
109     channel

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

112 **BIAS response message**
113     BIAS message conveying a response to a prior BIAS requestmessage

## 114 1.6 References

## 115 1.6.1 Normative References

116 **[RFC2119]** S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119,
117     March 1997
118     http://www.ietf.org/rfc/rfc2119.txt
119

120 **[CBEFF]** ISO/IEC19785-1:2006, Information technology – Common Biometric Exchange Formats
121     Framework – Part 1: Data element specification, with Amendment 1:2010
122     http://www.iso.org
123

124 **[DATE-TIME]** ISO 8601:2004, Data elements and interchange formats — Information interchange —
125     Representation of dates and times
126     http://www.iso.org
127

128 **[INCITS-BIAS]** ANSI INCITS 442-2010, Biometric Identity Assurance Services (BIAS), July 2010
129     http://www.incits.org
130

131 **[IRI]** M. Duerst, et al, Internationalized Resouce Identifiers, RFC3987, January 2005
132     http://www.ietf.org/rfc/rfc3987.txt
133 **[SOAP11]** Simple Object Access Protocol (SOAP) 1.1, 8 May 2000
134     http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

135 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, *Uniform Resource Identifiers (URI): Generic Syntax*,
136     RFC 3986, MIT/LCS, U.C. Irvine, Xerox Corporation, January 2005.
137     http://ietf.org/rfc/rfc3986

138 **[UTF-8]** ISO/IEC 10646:2003, Information technology — Universal Multiple-Octet Coded Character
139     Set (UCS)
140     http://www.iso.org

141      **[WS-Addr]** W3C Recommendation,Web Services Addressing 1.0 - Core,  and Web Services
142                Addressing 1.0 - SOAP Binding, 9 May 2006
143                http://www.w3.org/2002/ws/addr/

144      **[WS-I-Basic]** Basic Profile Version 1.1, 10 April 2006
145                http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html

146      **[WS-I-Bind]** Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile Version
147                1.0, 24 August 2004
148                http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html

149      **[WSDL11]** Web Services Description Language (WSDL) 1.1, 15 March 2001
150                http://www.w3.org/TR/2001/NOTE-wsdl-20010315

151      **[XML 10]** Extensible Markup Language (XML) 1.0, 16 August 2006
152                http://www.w3.org/TR/2006/REC-xml-20060816/

153      **[XOP]** XML-binary Optimized Packaging, W3C Recommendation, 25 January 2005
154                http://www.w3.org/TR/2005/REC-xop10-20050125/

## 155 1.6.2 Non-Normative References

156      **[BioAPI]** ISO/IEC 19784-1:2006, Information technology – Biometric Application Programming
157                Interface – Part 1: BioAPI Specification
158                http://www.iso.org

159      **[CBEFF-3]** ISO/IEC19785-3:2007, Information technology – Common Biometric Exchange Formats
160                Framework – Part 3: Patron format specifications, with Amendment 1:2010
161                http://www.iso.org
162
163      **[BIO SEC]** ISO 19092 Financial services -- Biometrics -- Security framework
164                http://www.iso.org

165      **[EBTS-DOD]** Department of DefenseElectronic Biometric TransmissionSpecification, Version 2.0,
166                27 March 2009
167                http://www.biometrics.dod.mil/CurrentInitiatives/Standards/dodebts.aspx
168      **[EBTS-FBI]**IAFIS-DOC-01078-8.1, "Electronic Biometric Transmission Specification (EBTS)",
169                Version 8.1, November 19, 2008, Federal Bureau of Investigation, Criminal Justice
170                Information Services Division
171                https://www.fbibiospecs.org
172      **[EFTS]** IAFIS-DOC-01078-7, "Electronic Fingerprint Transmission Specification (EFTS)", Version
173                7.1, May 2, 2005, Federal Bureau of Investigation, Criminal Justice Information Services
174                Division
175                https://www.fbibiospecs.org
176      **[HR-XML]** HR-XML Consortium Library, 2007 April 15
177                http://www.hr-xml.org
178      **[INT-I]** Interpol Implementation of ANSI/NIST ITL1-2000, Ver 4.22b, October 28, 2005, The Interpol
179                AFIS Expert Group
180                http://www.interpol.int
181      **[NIEM]** National Information Exchange Model (NIEM), Ver 2.0, June 2007, US DOJ/DHS
182                http://www.niem.gov
183      **[RFC2246]** T. Dierks & C. Allen,*The TLS Protocol, Version 1.0*, January 1999
184                http://www.ietf.org/rfc/rfc2246.txt

185        **[RFC2617]** J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*, June
186              1999
187              http://www.ietf.org/rfc/rfc2617.txt

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

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

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

196        **[SSL3]** SSL 3.0 Specification
197              http://www.freesoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM

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

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

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

# 2 Design Concepts and Architecture (non-normative)

## 2.1 Philosophy

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

## 2.2 Context

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

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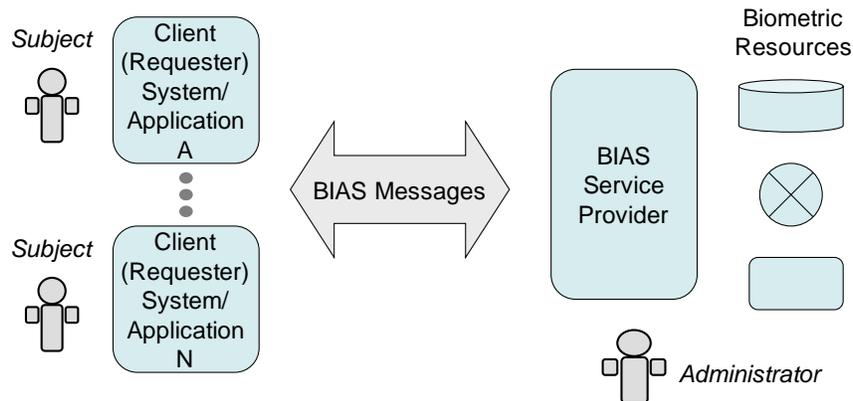


**Figure 1.  BIAS Context**

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

## 2.3 Architecture

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

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

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

241     •     Client sends and receives SOAP messages that conform to the BIAS schemas

242     •     Calls to the BIAS Implementation use OASIS Service Interfaces and Bindings (via WSDL)

243     •     The BIAS implementation maps the service call to the appropriate internal API or set of APIs

244         and returns data according to the service interface.

245 Note that services are represented as circles.

246
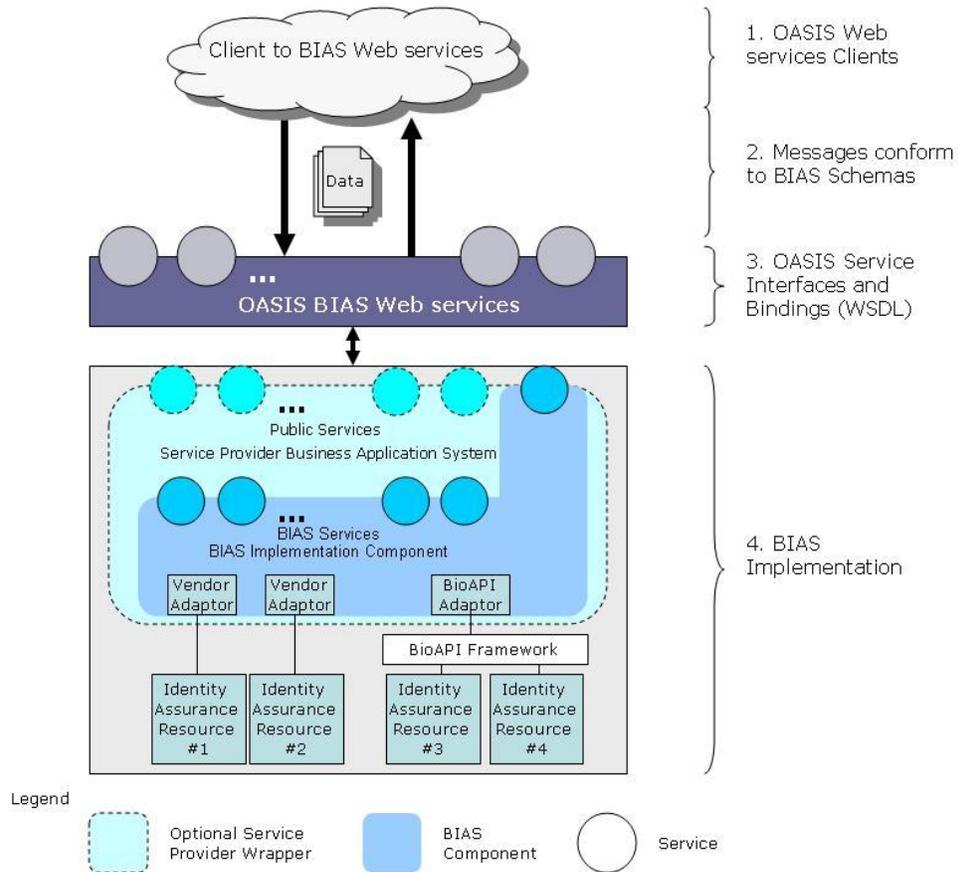


247
248
249 **Figure 2. Representative BIAS Architecture**

250

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

253

254 At the heart of the BIAS SOAP Profile are the concepts of BIAS messages and endpoints.

255

256 ***BIAS implementation***

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

260 ***BIAS messages***

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

263 ***BIAS endpoints***

264 A BIAS endpoint is a runtime entity, uniquely identified and accessed by an endpoint URI/IRI [URI] [IRI],
265 capable of sending and receiving BIAS messages.

266     NOTE: When not publicly and directly exposed, the endpoints for purposes of this specification are
267     the BIAS service provider exposing BIAS services and the component that directly interacts with that
268     service provider, e.g., the business application or ESB, rather than the ultimate end client requester.

# 3   Data dictionary

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

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

## 3.1   Documentation Conventions

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

282

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

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

284   ***Fields Hierarchy Explained:***

285   To denote the field hierarchy the symbol ⌐ is used to denote the child-of relationship.

286   All string types/elements MUST consist of ISO/IEC 10646 (Unicode) characters encoded in UTF-8 [UTF-
287   8] (see ISO/IEC 10646:2003, Annex D).

## 3.2 Common Elements

### 3.2.1 ApplicationIdentifier

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

### 3.2.2 ApplicationUserIdentifier

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

### 3.2.3 BaseBIRType

| | |
|---|---|
| **Type:** | Schema complexType |
| **Description:** | Base type for all BIR subtypes; see BinaryBIR, URI_BIR, and XML_BIR for currently available types. |

### 3.2.4 BIASBiometricDataType

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| BIASBiometricDataType | | | Y | Wraps the various BIAS biometric types. The operations that use this type specify which elements are required. |
| ⌐ BIRList | 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. |

## 3.2.5 BIASFaultCode

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

**BIASFaultCode Enumeration Values**

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

NOTES:

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

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

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

## 3.2.6 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. |

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

## 3.2.7 BIASIdentity

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| BIASIdentity | | | Y | Defines a single element for encapsulating the data associated with an Identity. Includes the Identity's reference identifiers, biographic data, and biometric data. The operations that use this type specify which elements are required. |
| ⌐ SubjectID | 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. |

## 3.2.8 BIASIDType

| | |
|---|---|
| **Type:** | string |
| **Description:** | A BIAS Identifier. |

## 3.2.9 BinaryBIR

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ Binary | base64Binary | 1 | Y | BIR information in base64 binary format |

304

## 305 3.2.10 BiographicDataItemType

| 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. |

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

## 308 3.2.11 BiographicDataSetType

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

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

## 312  3.2.12 BiographicDataType

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

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

314    • Encode only first and last name using the defined fields within BiographicDataType

315    • Define a list of biographic data elements using the BiographicDataItemType

316    • Use a pre-defined set of biographic data (e.g., as specified in another standard) using the
317      BiographicDataSetType.

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

## 319  3.2.13 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 | oasis_cbeff:MultipleTypesType | 1 | Y | The type of biological or behavioral data stored in the biometric record, as defined by CBEFF. |
| ⌐ BiometricTypeCount | positiveInteger | 0..1 | N | The number of biometric records having the biometric type recorded in the biometric type field. |

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

320

### 3.2.14 BiometricDataListType

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| BiometricDataListType | | | Y | A list of biometric data elements. |
| ⌐ BiometricDataElement | **3.2.13 BiometricDataElementType** | 0..* | N | Data structure containing information about a biometric record. |

### 3.2.15 CandidateListResultType

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| CandidateListResultType | | | Y | Defines a set of candidates, utilizing the CandidateType to represent each element in the set. |
| ⌐ CandidateList | **3.2.16 CandidateListType** | 1 | Y | The candidate list. |

323

### 3.2.16 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. |

## 3.2.17 CandidateType

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

## 3.2.18 CapabilityListType

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

## 3.2.19 CapabilityName

| Type: | string |
|---|---|
| Description: | A list of capability items. |

**CapabilityName Enumeration Values**

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

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

### 3.2.20 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. |

### 3.2.21 CBEFF_BIR_ListType

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| CBEFF_BIR_ListType | | | Y | A list of CBEFF-BIR elements. |
| └ BIR | CBEFF_BIR_Type | 0..* | N | CBEFF structure containing information about a biometric sample. |

### 3.2.22 CBEFF_BIR_Type

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ BIR_Information | | 0..1 | N | Describes what is contained in a BIR. |
| ⌐ BIR_Info | oasis_cbeff:BIRInfoType | 0..1 | N | Contains information about the CBEFF-BIR. |
| ⌐ BDB_Info | oasis_cbeff:BDBInfoType | 0..1 | N | Contains information about the BDB in a simple CBEFF-BIR. |
| ⌐ SB_Info | oasis_cbeff:SBInfoType | 0..1 | N | Contains information about the security block, if used, in a simple CBEFF-BIR. |
| ⌐ BIR | BaseBIRType | 1 | Y | One of the following sub-elements must be present: BinaryBIR, URI_BIR, or XML_BIR. |

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

333 • As an XML BIR (following the XML Patron format as specified in Annex B)

334 • As a reference to a URI (from which the receiver would retrieve the actual BIR)

335 • As a complete Base64 encoded binary (non-XML) BIR.

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

340 NOTE:

341 (1) XML BIRs MUST conform to the XML patron format in Annex B; however, non-XML (binary)
342 and URI BIRs MAY implement any CBEFF patron format.

343 (2) It is RECOMMENDED that only registered CBEFF patron formats be used; however, in closed
344 systems, this may not be required.

### 345 3.2.23 Classification

| Type: | string |
|---|---|
| Description: | The result of a classification. |

### 346 3.2.24 ClassificationAlgorithmType

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

### 347 3.2.25 ClassificationData

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ClassificationData | | | Y | Contains information on classification results and the algorithm used to determine the classification. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ Classification | Classification | 1 | Y | The result of the classification. |
| ⌐ ClassificationAlgorithmTyp e | ClassificationAlgorithmType | 1 | Y | Identifies the type of classification algorithm that was used to perform the classification. |

## 3.2.26 EncounterListType

348

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

## 3.2.27 FusionDecision

349

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

350

## 3.2.28 FusionInformationListType

351

| 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. |

## 3.2.29 FusionInformationType

352

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| FusionInformationType | | | Y | Represents the information necessary to perform a fusion operation. |
| ⌐ BiometricType | oasis_cbeff:Multipl eTypesType | 1 | Y | The type of biological or behavioral data stored in the biometric record, as defined by CBEFF. |
| ⌐ BiometricSubTyp e | oasis_cbeff: SubtypeType | 0..1 | N | More specifically defines the type of biometric data stored in the biometric record. |
| ⌐ AlgorithmOwner | string | 1 | Y | The owner or vendor of the algorithm used to determine the score or decision. |
| ⌐ AlgorithmType | string | 1 | Y | The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ FusionResult | FusionResult | 0..1 | C | Either FusionScore or a FusionDecision element MUST be used. |

## 353 3.2.30 FusionResult

| Type: | complexType |
|---|---|
| Description: | The base type for any resulting types which indicate the status of a Fusion operation |

## 354 3.2.31 FusionScore

| Type: | Score |
|---|---|
| Description: | The similarity score assigned by the matching algorithm. |

355

## 356 3.2.32 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. |

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

## 359 3.2.33 IdentifySubjectResultType

| Description: | A base type for all types that could be returned from the IdentifySubject operation |
|---|---|

360

## 361 3.2.34 InformationType

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

## 362  3.2.35 ListFilterType

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

## 363  3.2.36 MatchType

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

## 364  3.2.37 ProcessingOptionsType

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

## 365  3.2.38 ProductID

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

## 366  3.2.39 QualityData

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| QualityData | | | Y | Contains information about a biometric sample's quality and the algorithm used to compute the quality. |
| ⌐ QualityScore | oasis_cbeff:QualityType | 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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ 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. |

## 3.2.40 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. |

## 3.2.41 ReturnCode

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

**ReturnCode Enumeration Values**

| Value | Description |
|---|---|
| 0 | Success |

## 3.2.42 Score

| Type: | float |
|---|---|
| Description: | Match result or quality score. |

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 Annex B.

## 3.2.43 TokenResultType

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| TokenResultType | | | Y | Defines a token that is returned for asynchronous processing. |
| ⌐ TokenType | TokenType | 1 | Y | Defines a token that is returned for asynchronous processing. |

376 ### 3.2.44 TokenType

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

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

379 ### 3.2.45 URI_BIR

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

380

381 ### 3.2.46 VendorIdentifier

| Type: | string |
|---|---|
| Description: | Identifies a vendor. |

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

385 ### 3.2.47 Version

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| Version | | | Y | For a description or definition of each data element, see the referenced CBEFF standards in the **3.2.22 CBEFF_BIR_Type**schema. |
| ⌐ major | nonNegativeInteger | 1 | Y | |
| ⌐ minor | nonNegativeInteger | 1 | Y | |

386 ### 3.2.48 VersionType

| Type: | string |
|---|---|
| Description: | The version of a component. |

387 ### 3.2.49 XML_BIR

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| XML_BIR | BaseBIRType | | Y | Defines a BIR type of Binary |

| Field | Type | # | ? | Meaning |
|-------|------|---|---|---------|
| ⌐ XML | Oasis_cbeff:BIRType | 1 | Y | BIR information in XML format |

388

# 389 4 BIAS Messages

390 This section describes the BIAS messages implementing BIAS operations as defined in ANSI INCITS
391 442-2010.  The operations are listed alphabetically, with each operation containing a request and a
392 response message.  The tables follow the conventions described in section 3.1.

## 393 4.1 Primitive Operations

### 394 4.1.1 AddSubjectToGallery

395 AddSubjectToGalleryRequest

396 AddSubjectToGalleryResponse

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

### 403 Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| AddSubjectToGallery | | | Y | Register a subject to a given gallery or population group. |
| ⌐ AddSubjectToGalleryRequest | | 1 | 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 | 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: "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. |

## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| AddSubjectToGalleryResponse | | | Y | The response to an AddSubjectToGallery operation. |
| ┴ AddSubjectToGalleryResponsePackage | | 1 | Y | |
| ┴ 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. |

## 4.1.2 CheckQuality

CheckQualityRequest

CheckQualityResponse

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

## Request Message

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ CheckQualityRequest | | 1 | 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 | 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 vender 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. |

## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| CheckQualityResponse | | | Y | The response to a CheckQuality operation. |
| ⌐ CheckQualityResponsePackage | | 1 | Y | |
| ⌐ 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 | oasis_cbeff:QualityType | 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. |

## 4.1.3 ClassifyBiometricData

ClassifyBiometricDataRequest

ClassifyBiometricDataResponse

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

To obtain the types of classification algorithms and classes, see the QueryCapabilities operation.

## Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ClassifyBiometricData | | | Y | Classifies a biometric sample. |
| ⌐ ClassifyBiometricDataRequest | | 1 | Y | |

| 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: "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. |

## 425  Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ClassifyBiometricDataResponse | | | Y | The response to a ClassifyBiometricData operation, containing the classification of a biometric sample. |
| ⌐ ClassifyBiometricDataResponsePackage | | 1 | Y | |
| ⌐ 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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ ClassificationAlgorithmType | ClassificationAlgorithmType | 1 | Y | Identifies the type of classification algorithm that was used to perform the classification. |

## 426 **4.1.4 CreateSubject**

427 CreateSubjectRequest

428 CreateSubjectResponse

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

### 432 **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| CreateSubject | | | Y | |
| ⌐ CreateSubjectRequest | | 1 | 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 | 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: "CreateSubject". |

### 433 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| CreateSubjectResponse | | | Y | The response to a CreateSubject operation, containing the subject ID of the new subject record. |
| ⌐ CreateSubjectResponsePackage | | 1 | Y | |
| ⌐ ResponseStatus | 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 | |
| ⌐ SubjectID | BIASIDType | 1 | Y | A system unique identifier for a subject. |

## 434  4.1.5 DeleteBiographicData

435  DeleteBiographicDataRequest

436  DeleteBiographicDataResponse

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

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

## 443  Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteBiographicData | | | Y | Erase all of the biographic data associated with a given subject record or, in the encounter-centric model, with a given encounter. |
| ⌐ DeleteBiographicDataRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | 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. |

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

## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteBiographicDataResponse | | | Y | The response to a DeleteBiographicData operation. |
| ⌐ DeleteBiographicDataResponsePackage | | 1 | Y | |
| ⌐ 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. |

## 4.1.6 DeleteBiometricData

DeleteBiometricDataRequest

DeleteBiometricDataResponse

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.

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

## Request Message

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ GenericRequestParameter s | 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: "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. |

## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteBiometricDataResponse | | | Y | The response to a DeleteBiometricData operation. |
| ⌐ DeleteBiometricDataResponsePackage | | 1 | Y | |
| ⌐ 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. |

## 4.1.7 DeleteSubject

DeleteSubjectRequest

DeleteSubjectResponse

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

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

## 465 Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteSubject | | | Y | Delete an existing subject record and, in an encounter-centric model, any associated encounter information. |
| ⌐ DeleteSubjectRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | 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: "DeleteSubject". |
| ⌐ Identity | BIASIdentity | 1 | Y | The identity of the subject to delete. |
| ⌐ SubjectID | BIASIDType | 1 | Y | A system unique identifier for a subject. |

## 466 Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteSubjectResponse | | | Y | The response to a DeleteSubject operation. |
| ⌐ DeleteSubjectResponsePackage | | 1 | Y | |
| ⌐ 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. |

## 467 4.1.8 DeleteSubjectFromGallery

468 DeleteSubjectFromGalleryRequest

469 DeleteSubjectFromGalleryResponse

470 The DeleteSubjectFromGallery operation removes the registration of a subject from a gallery or
471 population group. The subject is identified by either the subject ID or the claim to identity that was
472 specified in the AddSubjectToGallery operation.

## Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteSubjectFromGallery | | | Y | Remove the registration of a subject from a gallery or population group. |
| ⌐ DeleteSubjectFromGalleryRequest | | 1 | 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 | 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: "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. |
| ⌐ 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. |

**474** **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| DeleteSubjectFromGalleryResponse | | | Y | The response to a DeleteSubjectFromGallery operation. |
| ⌐ DeleteSubjectFromGalleryResponsePackage | | | | |
| ⌐ 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. |

**475** **4.1.9 GetIdentifySubjectResults**

**476** GetIdentifyResultsRequest

**477** GetIdentifySubjectResultsResponse

**478** The GetIdentifySubjectResults operation retrieves the identification results for the specified token. This
**479** opereation is used in conjunction with the IdentifySubject operation. If the IdentifySubject operation is
**480** implemented as an asynchronous service, the implementing system returns a token and the
**481** GetIdentifySubjectResults operation is used to poll for the results of the original IdentifySubject request.

**482** **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetIdentifySubjectResults | | | Y | Retrieve the identification results for a specified token, which was returned by the IdentifySubject operation. |
| ⌐ GetIdentifySubjectResultsRequest | | 1 | 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 | 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. |

### 483 Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetIdentifySubjectResultsResponse | | | Y | The response to a GetIdentifySubjectResults operation, which includes a candidate list. |
| ⌐GetIdentifySubjectResultsResponsePackage | | 1 | Y | |
| ⌐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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ↳ 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. |

## 4.1.10 IdentifySubject

IdentifySubjectRequest

IdentifySubjectResponse

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

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

## Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| IdentifySubject | | | Y | Perform an identification search against a given gallery for a given biometric. |
| ↳ IdentifySubjectRequest | | 1 | Y | |
| ↳ GenericRequestParameters | 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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐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. |
| ⌐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. |

## 495 Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| IdentifySubjectResponse | | | Y | The response to an IdentifySubject operation, returning a rank-ordered candidate list. |
| ⌐IdentifySubjectResponsePackage | | 1 | Y | |
| ⌐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 | CandidateListResultType (see IdentifySubjectResultType) | 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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ 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. |
| ⌐ Token | TokenResultType (see IdentifySubjectResultType) | 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. |

496     NOTES:

497     (1) In the event that the number of candidates exceeding the threshold exceeds the
498     MaxListSize, the system will determine which candidate is included in the last position of
499     the rank ordered candidate list (i.e., in the event of a tie).

500     (2) Requesters MAY NOT change the system thresholds.

## 4.1.11 ListBiographicData

502     ListBiographicDataRequest

503     ListBiographicDataResponse

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

### Request Message

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ ListBiographicDataRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | 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: "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. |

512 **Response Message**

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ ResponseStatus | 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 | 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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ EncounterID | BIASIDType | 0..* | N | The identifier of an encounter. |

## 513  4.1.12 ListBiometricData

514  ListBiometricDataRequest

515  ListBiometricDataResponse

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

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

## 528  Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ListBiometricData | | | Y | Lists the biometric data elements stored for a subject. |
| ⌐ ListBiometricDataRequest | | 1 | 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 | 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". |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ 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. |
| ⌐ BiometricTypeFilter | oasis_cbeff:MultipleTypesType | 1..* | Y | Limits the returned information to a specific type of biometric, as defined by CBEFF. |
| ⌐ IncludeBiometricSubType | boolean | 1 | Y | A Boolean flag indicating if biometric subtype information should be returned. |

## 529 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ListBiometricDataResponse | | | Y | The response to a ListBiometricData operation, containing a list of biometric data elements stored for a subject. In the encounter-centric model, the biometric data elements for a specific encounter are returned. If an encounter ID is not specified and encounter data exists for the subject, the list of encounter IDs which contain biometric data is returned. |
| ∟ListBiometricDataResponsePackage | | 1 | Y | |
| ∟ ResponseStatus | 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 | 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. |

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ EncounterList | 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. |

## 4.1.13 PerformFusion

PerformFusionRequest

PerformFusionResponse

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

## Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| PerformFusion | | | Y | Accepts either match score or match decision information and creates a fused match result. |
| ⌐ PerformFusionRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | 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 | 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 | oasis_cbeff:MultipleTypesType | 1 | Y | The type of biological or behavioral data stored in the biometric record, as defined by CBEFF. |
| ∟ BiometricSubType | oasis_cbeff:SubtypeType | 0..1 | N | More specifically defines the type of biometric data stored in the biometric record. |
| ∟ AlgorithmOwner | string | 1 | Y | The owner or vendor of the algorithm used to determine the score or decision. |
| ∟ AlgorithmType | string | 1 | Y | The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision. |
| ∟ FusionResult | FusionResult | 0..1 | C | Either FusionScore or a FusionDecision element MUST be used. |

## 538 Response Message

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

| 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. |
| ⌐ Match | MatchType | 1 | 1 | Indicates the result of the fusion method. |

## 4.1.14 QueryCapabilities

QueryCapabilitiesRequest

QueryCapabilitiesResponse

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

### Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| QueryCapabilities | | | Y | Returns a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation. |
| ⌐ QueryCapabilitiesRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | 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: "QueryCapabilities". |

## 547 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| QueryCapabilitiesResponse | | | Y | The response to a QueryCapabilities operation. |
| ⌐ QueryCapabilitiesResponsePackage | | 1 | Y | |
| ⌐ 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. |
| ⌐ 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. |

## 548 **4.1.15 RetrieveBiographicInformation**

549 RetrieveBiographicInformationRequest

550 RetrieveBiographicInformationResponse

551 The RetrieveBiographicInformation operation retrieves the biographic data associated with a subject ID.
552 In the encounter-centric model, the encounter ID MAY be specified and the operationwill return the
553 biographic data associated with that encounter. If the encounter ID is not specified in the encounter-

554  centric model, the operation returns the biographic information associated with the most recent
555  encounter.

## Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| RetrieveBiographicInformation | | | Y | Retrieves the biographic data associated with a subject ID. |
| ⌐ RetrieveBiographicInformationRequest | | 1 | 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 | 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: "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. |

## Response Message

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

| 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. |
| � 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. |
| � BiographicDataItem | BiographicDataItemType | 0..* | C | A single biographic data element. |
| � BiographicDataSet | BiographicDataItemType | 0..1 | C | A set of biographic data information. |

## 4.1.16 RetrieveBiometricInformation

558

559 RetrieveBiometricInformationRequest

560 RetrieveBiometricInformationResponse

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

567 **Request Message**

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ RetrieveBiometricInformationReque st | | 1 | Y | |
| ⌐ GenericRequestParameter s | 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: "RetrieveBiometricInfo rmation". |
| ⌐ 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. |
| ⌐ BiometricType | oasis_cbeff:MultipleTypesT ype | 0..1 | N | The type of biological or behavioral data to retrieve. |

568   **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| RetrieveBiometricInformationResponse | | | Y | The response to a RetrieveBiometricInformati on operation. |
| ⌐ RetrieveBiometricInformationRespon sePackage | | 1 | Y | |
| ⌐ 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 | 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. |

## 4.1.17 SetBiographicData

569

570 SetBiographicDataRequest

571 SetBiometricDataResponse

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

579 **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| SetBiographicData | | | Y | Associates biographic data to a given subject record. |
| ⌐SetBiographicDataRequest | | 1 | 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 | 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". |

| 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 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. |
| ⌐BiographicDataItem | BiographicDataItemType | 0..* | C | A single biographic data element. |
| ⌐BiographicDataSet | BiographicDataSetType | 0..1 | C | A set of biographic data information. |

580 ## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| SetBiographicDataResponse | | | Y | The response to a SetBiographicData operation. |
| ⌐ SetBiographicDataResponsePackage | | 1 | Y | |
| ⌐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 | 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. |

## 4.1.18 SetBiometricData

SetBiometricDataRequest

SetBiometricDataResponse

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

### Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| SetBiometricData | | | Y | Associates biometric data to a given subject record. |
| ⌐SetBiometricDataRequest | | 1 | Y | |
| ⌐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: "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. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐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. |

## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| SetBiometricDataResponse | | | Y | The response to a SetBiometricData operation. |
| ⌐SetBiometricDataResponsePackage | | 1 | Y | |
| ⌐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. |

## 4.1.19 TransformBiometricData

TransformBiometricDataRequest

TransformBiometricDataResponse

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

598 **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| TransformBiometricData | | | Y | Transforms or processes a given biometric in one format into a new target format. |
| ⌐ TransformBiometricDataRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | [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.<br><br>Note: This could be a compression ratio, target data format, etc. |

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

600 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| TransformBiometricDataResponse | | | Y | The response to a TransformBiometricData operation. |
| ⌐ TransformBiometricDataResponsePackage | | 1 | Y | |
| ⌐ 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. |
| └ OutputBIR | CBEFF_BIR_Type | 0..1 | N | Data structure containing the new, transformed biometric information. |

## 4.1.20 UpdateBiographicData

UpdateBiographicDataRequest

UpdateBiographicDataResponse

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

### Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| UpdateBiographicData | | | Y | Updates the biographic data for a given subject record. |
| └ UpdateBiographicDataRequest | | 1 | Y | |
| └ GenericRequestParameters | 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: "UpdateBiographic Data". |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ㄴIdentity | BIASIdentity | 1 | Y | Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to update. |
| ㄴSubjectID | BIASIDType | 1 | Y | A system unique identifier for a subject. |
| ㄴEncounterID | BIASIDType | 0..1 | C | The identifier of an encounter associated with the subject. Required for encounter-centric models. |
| ㄴBiographicData | BiographicDataType | 1 | Y | An Identity's biographic data. One of the following elements MUST be present. |
| ㄴLastName | string | 0..1 | C | The last name of a subject. |
| ㄴFirstName | string | 0..1 | C | The first name of a subject. |
| ㄴBiographicDataItem | BiographicDataItemType | 0..* | C | A single biographic data element. |
| ㄴBiographicDataSet | BiographicDataSetType | 0..1 | C | A set of biographic data information. |

608  **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| UpdateBiographicDataResponse | | | Y | The response to an UpdateBiographicData operation. |
| ㄴUpdateBiographicDataResponsePackage | | 1 | Y | |
| ㄴResponseStatus | ResponseStatus | 1 | Y | Returned status for the operation. |
| ㄴReturn | ReturnCode | 1 | Y | The return code indicates the return status of the operation. |

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

## 609   **4.1.21 UpdateBiometricData**

610   UpdateBiometricDataRequest

611   UpdateBiometricDataResponse

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

617   **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| UpdateBiometricData | | | Y | Updates a single biometric sample for a given subject record. |
| ⌐ UpdateBiometricDataRequest | | 1 | 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 | 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. |

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

618 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| UpdateBiometricDataResponse | | | Y | The response to an UpdateBiometricData operation. |
| ⌐UpdateBiometricDataResponsePackage | | 1 | Y | |
| ⌐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. |

619 ## 4.1.22 VerifySubject

620 VerifySubjectRequest

621 VerifySubjectResponse

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

625 **Request Message**

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ 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: "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. |

**Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| VerifySubjectResponse | | | Y | The response to a VerifySubject operation. |
| ꜔ VerifySubjectResponsePackage | | 1 | Y | |
| ꜔ 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 | 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. |

627 # 4.2 Aggregate Operations

628 ## 4.2.1 Enroll

629 EnrollRequest

630 EnrollResponse

631 The Enroll operation adds a new subject or, in an encounter-centric model, a new encounter to the
632 system. This may be accomplished in a number of different ways according to system requirements
633 and/or resources.If the Enroll operation is implemented as a synchronous service, the implementing
634 system immediately processes the request and returns the results in the Return Data parameter. If the
635 Enroll operation is implemented as an asynchronous service, the implementing system returns a token in
636 the Return Data parameter, which is an indication that the request is being handled asynchronously. In
637 this case, the GetEnrollResults operationis used to poll for the results of the Enroll request.

638 **Request Message**

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

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

### Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| EnrollResponse | | | Y | The response to an Enroll operation. |
| └ EnrollResponsePackage | | 1 | Y | |
| └ 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. |

## 4.2.2 GetEnrollResults

GetEnrollResultsRequest

GetEnrollResultsResponse

The GetEnrollResults operation retrieves the enrollment results for the specified token. This operation is used in conjunction with the Enroll operation. If the Enroll operation is implemented as an asynchronous service, the implementing system returns a token and the GetEnrollResults operation is used to poll for the results of the original Enroll request.

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

649 **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetEnrollResults | | | Y | Retrieves the enrollment results for the specified token. |
| ᴸ GetEnrollResultsRequest | | 1 | Y | |
| ᴸ GenericRequestParameters | GenericRequestParameters | 0..1 | N | Common request parameters that can be used to identify the requester. |
| ᴸ Application | ApplicationIdentifier | 0..1 | N | Identifies the requesting application. |
| ᴸ ApplicationUser | ApplicationUserIdentifier | 0..1 | N | Identifies the user or instance of the requesting application. |
| ᴸ BIASOperationName | string | 0..1 | N | Identifies the BIAS operation that is being requested: "GetEnrollResults". |
| ᴸ Token | TokenType | 1 | Y | A value used to retrieve the results of the Enroll request. |
| ᴸ 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. |
| ᴸ 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. |

650 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetEnrollResultsResponse | | | Y | The response to a GetEnrollResults operation. |
| ᴸ GetEnrollResultsResponsePackage | | 1 | Y | |
| ᴸ 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. |
| ⌐ ReturnData | InformationType | 0..1 | N | Contains the output data for the response. |

## 4.2.3 GetIdentifyResults

GetIdentifyResultsRequest

GetIdentifyResultsResponse

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

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


**Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetIdentifyResults | | | Y | Retrieves the identification results for the specified token |
| ⌐ GetIdentifyResultsRequest | | 1 | Y | |
| ⌐ GenericRequestParameters | 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: "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 |

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

## Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetIdentifyResultsResponse | | | Y | The response to a GetIdentifyResults operation. |
| └ GetIdentifyResultsResponsePackage | | 1 | Y | |
| └ 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. |

## 4.2.4 GetVerifyResults

GetVerifyResultsRequest

GetVerifyResultsResponse

The GetVerifyResults operation retrieves the verification results for the specified token. This operation is used in conjunction with the Verify operation. If the Verify operation is implemented as an asynchronous service, the implementing system returns a token and the GetVerifyResults operation is used to poll for the results of the original Verify request.

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

## Request Message

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

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ GenericRequestParameter s | 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: "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. |

## 673    Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| GetVerifyResultsResponse | | | Y | The response to a GetVerifyResults operation. |
| ⌐ GetVerifyResultsResponsePackage | | 1 | Y | |
| ⌐ 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. |

| 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. |

## 4.2.5 Identify

IdentifyRequest

IdentifyResponse

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

## Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| Identify | | | Y | Performs an identification function. |
| IdentifyRequest | | 1 | Y | |
| ⌐GenericRequestParameters | 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: "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. |

## 684 Response Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| IdentifyResponse | | | Y | The response to an Identify operation. |
| ⌐IdentifyResponsePackage | | 1 | Y | |
| ⌐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. |

## 685 4.2.6 RetrieveInformation

686 RetrieveInformationRequest

687 RetrieveInformationResponse

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

## 693 Request Message

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| RetrieveInformation | | | Y | Retrieves requested information about a subject or encounter. |
| ⌐RetrieveInformationRequest | | 1 | 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 | 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: "RetrieveInformation". |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ 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. |

694 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| RetrieveInformationResponse | | | Y | Response to a RetrieveInformation operation. |
| ⌐ RetrieveInformationResponsePackage | | 1 | Y | |
| ⌐ 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. |

695 **4.2.7 Verify**

696 VerifyRequest

697 VerifyResponse

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

705 **Request Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| Verify | | | Y | Performs a 1:1 verification function. |
| ⌐ VerifyRequest | | 1 | 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 | 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. |
| ⌐ BiometricData | BIASBiometricDataType | 0..1 | N | An Identity's biometric data. |

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| ⌐ 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. |

706 **Response Message**

| Field | Type | # | ? | Meaning |
|---|---|---|---|---|
| VerifyResponse | | | Y | The response to a Verify operation. |
| ⌐ VerifyResponsePackage | | 1 | Y | |
| ⌐ 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. |

707

# 5  Message structure and rules

708

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

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

723  SOAP 1.1 messages consist of three elements: an envelope, header data, and a message body.  BIAS
724  request-response elements MUST be enclosed within the SOAP message body.  The general structure of
725  the BIAS SOAP message is shown in Figure 4, below. The data model for BIAS is addressed in Section3
726  and BIAS messages in Section 4.

727

<br>

SOAP Envelope

*SOAP Header*

*SOAP Body*

*SOAP Payload*

BIAS XML Elements

728
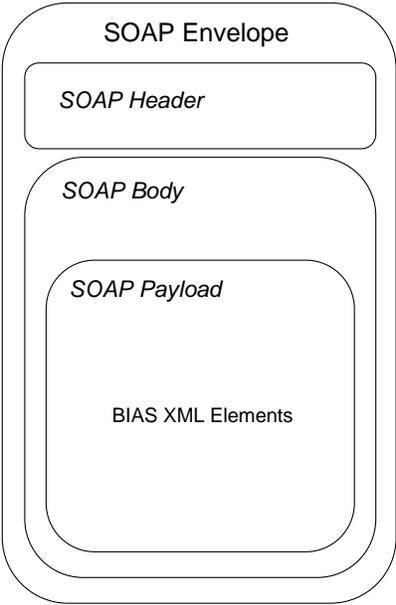
729  **Figure 4. BIAS SOAP Structure**

730

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

## 5.1  Purpose and constraints

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

738     •     SOAP messages will carry BIAS XML [XML 10] payloads.

739     •     SOAP messages will follow WS-I and will deviate only when absolutely necessary.

740     •     Message structures and interchanges will be kept as simple as possible – "nice to have"
741           features will be addressed in future revisions.

742     •     XML schemas will be produced based on INCITS 442.

743     •     BIAS will support a broad range of application domains.

744     •     BIAS will allow for a variety of biometric and biographic data formats to be used

745     •     Only the SOAP messaging will be defined – no message protocols or client/server agents
746           will be defined.

747     •     Basic usage/formatting rules (beyond WS-I) will be defined.

748     •     Existing biometric and Web services standards will be leveraged wherever possible.

749     •     Sample WSDL and use cases will be provided as an aid in implementation.

750     •     Use of basic SOAP will allow all other compatible WS* standards (and discovery
751           mechanisms) to be used in conjunction with BIAS messaging.

752     •     BIAS will support both secure (i.e., using existing security mechanisms such as WS-
753           Security, SAML, etc,) and non-secure implementations.

754     •     Generic biometric operations will be defined – use of biometrics within a Web services
755           authentication protocol is not addressed.

756     •     OASIS namespace rules will be followed, though some external schemas MAY also be
757           referenced.

## 5.2 Message requirements

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

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

769



(a) Synchronous Operation              (b) Asynchronous Operation
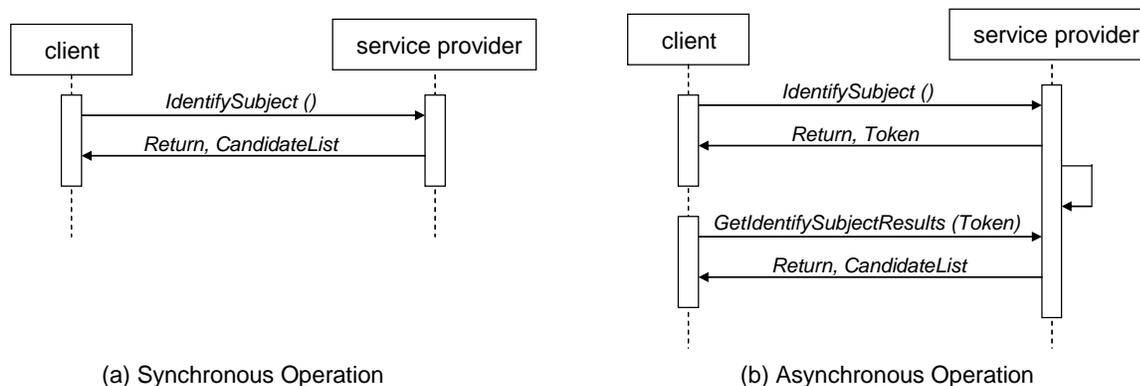
**Figure 5. Example of Synchronous and Asynchronous BIAS Operations**

772

773

774   The basic process for using SOAP for BIAS operations is:

775     1. A system entity acting as a BIAS requester transmits a BIAS request element within the body of a
776        SOAP message to a system entity acting as a BIAS responder. The BIAS requester MUST NOT
777        include more than one BIAS request per SOAP message or include any additional XML elements
778        in the SOAP body.

779     2. The BIAS responder MUST return either a BIAS response element within the body of another
780        SOAP message or generate a SOAP fault. The BIAS responder MUST NOT include more than
781        one BIAS response per SOAP message or include any additional XML elements in the SOAP
782        body. If a BIAS responder cannot, for some reason, process a BIAS request, it MUST generate a
783        SOAP fault.  (SOAP 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)

784     3. On receiving a BIAS response in a SOAP message, the BIAS requester MUST NOT send a fault
785        code or other error messages to the BIAS responder. Since the format for the message
786        interchange is a simple request-response pattern, adding additional items such as error
787        conditions would needlessly complicate the protocol.

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

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

## 5.3 Handling binary data

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

798     • Embedded Base64 encoding

799     • XOP [XOP]

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

## 5.3.1 Base64 encoding

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

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

807
808        ```
             <xsd:element name="BinaryBIR" type="xs:base64Binary" />
             ```
809

810   However, even an XML_BIR as defined within [CBEFF3], contains a biometric data block (BDB) which
811   may be entirely binary (most common),

812
813        ```
             <xs:element name="bdb" type="xs:base64Binary" minOccurs="0"/>
             ```
814

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

## 5.3.2 Use of XOP

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

819    message size during transmission since the overhead of the embedded Base64 is not present (since the
820    MIME attachment contains the native binary format).

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

## 5.4 Discovery

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

828    In addition, it is REQUIRED that the BIAS implementation include the QueryCapabilities operation to
829    provide dynamic information regarding BIAS capabilities, options, galleries, etc. that are supported.

## 5.5 Identifying operations

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

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

### 5.5.1 Operation name element

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

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

844

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

```
862              <tns:SubjectID>123456789</tns:SubjectID>
863          </tns:Identity>
864        </tns:CreateSubjectRequest>
865      </soap:Body>
866  </soap:Envelope>
```

## 5.5.2 WS-Addressing Action

WS-Addressing [WS-Addr] provides a mechanism for including action information inside any SOAP message.  The information is in the SOAP Header.  The WS-Addressing 'Action' element is used to indicate the intent of the message.  The value is a URI/IRI identifying that intent; however, there are no restrictions on the format or specificity of the URI/IRInor a requirement that it can be resolved.  Adoption of this option also requires that the WS-Addressing 'To', 'ReplyTo', and 'MessageID' elements are supplied, as they are mandatory elements in a request-reply message pattern as used within BIAS.  Response messages would also need to use WS-Addressing, requiring the 'To' (matching the 'ReplyTo' element in the request), 'RelatesTo' (matching the 'MessageID' element in the request), and 'RelationshipType' (default value to "wsa:Reply") elements.

Use of WS-Addressing is OPTIONAL in this profile as is this method of using the 'Action' field for this purpose.  However, when BIAS is used within an environment using WS-Addressing, it is RECOMMENDED that this approach for use of the 'Action' field to carry the BIAS operation name is employed, either alone or in combination with the BIASOperationName approach described in section 5.5.1.

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

```
POST /bias HTTP/1.1
Host: www.acme.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: ""
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
               xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addr
         essing">
    <soap:Header>
         <wsa:MessageID>some-ID</wsa:MessageID>
         <wsa:ReplyTo>
              <wsa:Address>response-URI</wsa:Address>
         </wsa:ReplyTo>
         <wsa:To>destination-URI</wsa:To>
         <wsa:Action>CreateSubject</wsa:Action>
    </soap:Header>
    <soap:Body>
         <tns:CreateSubjectRequest

   xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/" />
    </soap:Body>
</soap:Envelope>
```

907

## 5.6 Security

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

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

### 5.6.1 Use of SSL 3.0 or TLS 1.0

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

Use of transport level security in the form of SSL or TLS is OPTIONAL but highly RECOMMENDED.  Use of these mechanisms alone may not be sufficient for end-to-end integrity and confidentiality, however (see 5.6.3 and 5.6.4 below).

### 5.6.2 Data Origin Authentication

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

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

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

### 5.6.3 Message Integrity

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

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

### 5.6.4 Message Confidentiality

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

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

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

### 5.6.5 CBEFF BIR security features

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

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

## 5.6.6 Security Considerations

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

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

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

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

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

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

## 5.6.7 Security of Stored Data

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

## 5.6.8 Key Management

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

## 5.7 Use with other WS* standards

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

## 5.8 Tailoring

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

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

991     • Identification of a subset of BIAS operations to be used

992     • Specification of security features to be implemented (e.g., SSL, CBEFF BIR encryption, etc.)

993     • Choice of operation name identification method

994 • Choice of BIR type to be used (XML, non-XML, or URI)

995 • Further definition of aggregate operations

996 • Use (or not) of the encounter model

997 • Use (or not) of asynchronous operations

998 • Process sequences

999 • Implementation specific values (e.g., Transform operations/controls)

# 6 Error handling

1001 There are two levels of errors that can be returned in an error response: system and service errors.

1002 • System-level errors occur when the implementing system cannot service a request. They could
1003     result due to an internal logic error or because the implementing system does not support a
1004     particular request.
1005 • Service-level errors occur when there is a problem transmitting or representing the service
1006     request. They could result due to an invalid service request or because of a communications
1007     error.

1008 The INCITS BIAS standard defines the error condition codes for system-level errors.

1009 • If successful, a response message (containing a return code) will be generated.

1010 • If unsuccessful, a SOAP fault message (containing a fault code) will be generated.

## 6.1 BIAS operation return codes

1012 If a BIAS operation is successful, a response (service output) will be sent to the requester by the service
1013 provider.  Each response message contains a response status (see section 3.2.37) and return code (see
1014 section 3.2.38) along with any response data as defined for that operation, if any.  A response code of '0'
1015 indicates success.

## 6.2 SOAP fault codes

1017 If a BIAS operation is unsuccessful, no BIAS response message is sent.  Instead a SOAP fault message
1018 is returned.

1019 Every Web service (operation) described in the BIAS WSDL may result in a fault message that will be
1020 returned in the response by the service provider in the event of an error. The fault message contains a
1021 FaultCode element as defined by the SOAP 1.1 specification (see section 3.2.5).  The fault message
1022 MUST contain a Detail element in a common format, as described by the BIASFault element (see section
1023 3.2.6).

1024 The schema provided in Annex A defines "BIASFaultCode" and "BIASFaultDetail" types as well as
1025 "BIASFault", "BIASFaultType", "BIASFaultMessage" and "BIASFaultDescription" elements.

1026 The list of defined BIAS fault codes is provided in section 3.2.5.  Note that BIAS service providers MAY
1027 define additional fault codes unique to their service.

1028     NOTE:  See also section 5.2 for additional information on message returns and faults.

# 7 Conformance

1029

1030 Implementations claiming conformance to this standard, MUST implement, at a minimum, all mandatory
1031 requirements and provisions set forth in Clauses 3, 4, 5 and 6. If such implementations claim
1032 conformance to any OPTIONAL requirements and provisions stated in Clauses 3, 4, 5 and 6, these
1033 requirements and provisions MUST be implemented as set forth in these Clauses.

1034 INCITS 442 [INCITS-BIAS] (Annex A) specifies five BIAS conformance classes.  For each class, a set of
1035 mandatory BIAS operations is identified in order for implementations (BIAS service providers) to claim
1036 conformance.  These categories are:

1037 • Class 1: Full Primitive Services Implementation

1038 • Class 2: Full Aggregate Services Implementation

1039 • Class 3: Limited Primitive Services Implementation

1040 • Class 4: Minimum Primitive Services Implementation

1041 • Class 5: Minimum Aggregate Services Implementation

1042 In addition, the minimum capability information to be returned in response to a Query Capabilities request
1043 (the only mandatory BIAS operation across all 5 classes) is specified for each class.

1044 These conformance classes and their associated requirements apply to this BIAS SOAP Profile.

1045 There are no minimum set of operations required to be implemented by BIAS requesters; however, any
1046 operations implemented must conform to the requirements of Clauses 3 and 4 and those requirements
1047 within Clause 5 that are mandatory and are not specific to BIAS responders.

# Annex A.    XML Schema

1048

1049    <?xml version="1.0" encoding="UTF-8"?>

1050    <!--

1051    Document   : BIAS.xml

1052    Created on : December 17, 2009, 9:31 AM

1053    Last Edit  : December 6, 2010 by Kevin Mangold, National Institute of Standards and Technology

1054    Description:

1055    This document represents the WSDL definitions of BIAS Profile.

1056    -->

1057    <wsdl:definitions

1058    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

1059    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

1060    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"

1061    xmlns:xsd="http://www.w3.org/2001/XMLSchema"

1062    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"

1063    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

1064    xmlns:oasis_cbeff=" http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/"

1065    xmlns:tns="http://docs.oasis-open.org/bias/ns/bias-1.0/"

1066    targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">

1067

1068    <wsdl:types>

1069    <xsd:schema elementFormDefault="qualified" targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">

1070    <xsd:import namespace=" http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/" schemaLocation="cbeff.xsd"

1071    />

1072    <xsd:complexType name="AggregateRequestTemplate">

1073    <xsd:annotation>

1074    <xsd:documentation>Base template for BIAS aggregate service requests.</xsd:documentation>

1075    </xsd:annotation>

1076    <xsd:complexContent>

1077    <xsd:extension base="tns:RequestTemplate">

1078    <xsd:sequence>

1079    <xsd:element name="ProcessingOptions" type="tns:ProcessingOptionsType" minOccurs="1" maxOccurs="1">

1080    <xsd:annotation>

1081    <xsd:documentation>Options that guide how the aggregate service request is processed.</xsd:documentation>

1082    </xsd:annotation>

1083    </xsd:element>

1084    <xsd:element name="InputData" type="tns:InformationType" minOccurs="1" maxOccurs="1">

1085    <xsd:annotation>

1086    <xsd:documentation>Contains the input data for the aggregate service request.</xsd:documentation>

```xml
1087    </xsd:annotation>
1088    </xsd:element>
1089    </xsd:sequence>
1090    </xsd:extension>
1091    </xsd:complexContent>
1092    </xsd:complexType>
1093    <xsd:complexType name="AggregateResponseTemplate">
1094    <xsd:annotation>
1095    <xsd:documentation>Base template for BIAS aggregate service responses.</xsd:documentation>
1096    </xsd:annotation>
1097    <xsd:complexContent>
1098    <xsd:extension base="tns:ResponseTemplate">
1099    <xsd:sequence>
1100    <xsd:element name="ReturnData" type="tns:InformationType" minOccurs="0" maxOccurs="1">
1101    <xsd:annotation>
1102    <xsd:documentation>Contains the output data for the aggregate service response.</xsd:documentation>
1103    </xsd:annotation>
1104    </xsd:element>
1105    </xsd:sequence>
1106    </xsd:extension>
1107    </xsd:complexContent>
1108    </xsd:complexType>
1109    <xsd:simpleType name="ApplicationIdentifier">
1110    <xsd:annotation>
1111    <xsd:documentation>Identifies an application.</xsd:documentation>
1112    </xsd:annotation>
1113    <xsd:restriction base="xsd:string">
1114    <xsd:minLength value="1"/>
1115    <xsd:maxLength value="255"/>
1116    </xsd:restriction>
1117    </xsd:simpleType>
1118    <xsd:simpleType name="ApplicationUserIdentifier">
1119    <xsd:annotation>
1120    <xsd:documentation>Identifies an application user or instance.</xsd:documentation>
1121    </xsd:annotation>
1122    <xsd:restriction base="xsd:string">
1123    <xsd:minLength value="1"/>
1124    <xsd:maxLength value="255"/>
1125    </xsd:restriction>
1126    </xsd:simpleType>
1127    <xsd:complexType name="BIASBiometricDataType">
```

```
1128    <xsd:annotation>
1129    <xsd:documentation>Wraps the various BIAS biometric types.</xsd:documentation>
1130    </xsd:annotation>
1131    <xsd:sequence>
1132    <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="0" maxOccurs="1">
1133    <xsd:annotation>
1134    <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1135    </xsd:annotation>
1136    </xsd:element>
1137    <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1138    <xsd:annotation>
1139    <xsd:documentation>Contains biometric information in either a non-XML and an XML
1140    representation.</xsd:documentation>
1141    </xsd:annotation>
1142    </xsd:element>
1143    <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1144    <xsd:annotation>
1145    <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
1146    specification.</xsd:documentation>
1147    </xsd:annotation>
1148    </xsd:element>
1149    <xsd:element name="ReferenceBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
1150    <xsd:annotation>
1151    <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
1152    specification.</xsd:documentation>
1153    </xsd:annotation>
1154    </xsd:element>
1155    <xsd:element name="BiometricDataList" type="tns:BiometricDataListType" minOccurs="0" maxOccurs="1">
1156    <xsd:annotation>
1157    <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1158    </xsd:annotation>
1159    </xsd:element>
1160    </xsd:sequence>
1161    </xsd:complexType>
1162    <xsd:simpleType name="BIASFaultCode">
1163    <xsd:annotation>
1164    <xsd:documentation></xsd:documentation>
1165    </xsd:annotation>
1166    <xsd:restriction base="xsd:string">
1167    <xsd:enumeration value="UNKNOWN_ERROR">
1168    <xsd:annotation>
```

1169    <xsd:documentation>The service failed for an unknown reason.</xsd:documentation>

1170    </xsd:annotation>

1171    </xsd:enumeration>

1172    <xsd:enumeration value="UNSUPPORTED_CAPABILITY">

1173    <xsd:annotation>

1174    <xsd:documentation>A requested capability is not supported by the service

1175    implementation.</xsd:documentation>

1176    </xsd:annotation>

1177    </xsd:enumeration>

1178    <xsd:enumeration value="INVALID_INPUT">

1179    <xsd:annotation>

1180    <xsd:documentation>The data in a service input parameter is invalid.</xsd:documentation>

1181    </xsd:annotation>

1182    </xsd:enumeration>

1183    <xsd:enumeration value="BIR_QUALITY_ERROR">

1184    <xsd:annotation>

1185    <xsd:documentation>Biometric sample quality is too poor for the service to succeed.</xsd:documentation>

1186    </xsd:annotation>

1187    </xsd:enumeration>

1188    <xsd:enumeration value="INVALID_BIR">

1189    <xsd:annotation>

1190    <xsd:documentation>The input BIR is empty or in an invalid or unrecognized format.</xsd:documentation>

1191    </xsd:annotation>

1192    </xsd:enumeration>

1193    <xsd:enumeration value="BIR_SIGNATURE_FAILURE">

1194    <xsd:annotation>

1195    <xsd:documentation>The service could not validate the signature, if used, on the input BIR.</xsd:documentation>

1196    </xsd:annotation>

1197    </xsd:enumeration>

1198    <xsd:enumeration value="BIR_DECRYPTION_FAILURE">

1199    <xsd:annotation>

1200    <xsd:documentation>The service could not decrypt an encrypted input BIR.</xsd:documentation>

1201    </xsd:annotation>

1202    </xsd:enumeration>

1203    <xsd:enumeration value="INVALID_ENCOUNTER_ID">

1204    <xsd:annotation>

1205    <xsd:documentation>The input encounter ID is empty or in an invalid format.</xsd:documentation>

1206    </xsd:annotation>

1207    </xsd:enumeration>

1208    <xsd:enumeration value="INVALID_SUBJECT_ID">

1209    <xsd:annotation>

```
1210    <xsd:documentation>The input subject ID is empty or in an invalid format.</xsd:documentation>
1211    </xsd:annotation>
1212    </xsd:enumeration>
1213    <xsd:enumeration value="UNKNOWN_SUBJECT">
1214    <xsd:annotation>
1215    <xsd:documentation>The subject referenced by the input subject ID does not exist.</xsd:documentation>
1216    </xsd:annotation>
1217    </xsd:enumeration>
1218    <xsd:enumeration value="UNKNOWN_GALLERY">
1219    <xsd:annotation>
1220    <xsd:documentation>The gallery referenced by the input gallery ID does not exist.</xsd:documentation>
1221    </xsd:annotation>
1222    </xsd:enumeration>
1223    <xsd:enumeration value="UNKNOWN_ENCOUNTER">
1224    <xsd:annotation>
1225    <xsd:documentation>The encounter referenced by the input encounter ID does not exist.</xsd:documentation>
1226    </xsd:annotation>
1227    </xsd:enumeration>
1228    <xsd:enumeration value="UNKNOWN_BIOGRAPHIC_FORMAT">
1229    <xsd:annotation>
1230    <xsd:documentation>The biographic data format is not known or not supported.</xsd:documentation>
1231    </xsd:annotation>
1232    </xsd:enumeration>
1233    <xsd:enumeration value="UNKNOWN_IDENTITY_CLAIM">
1234    <xsd:annotation>
1235    <xsd:documentation>The identity referenced by the input identity claim does not exist.</xsd:documentation>
1236    </xsd:annotation>
1237    </xsd:enumeration>
1238    <xsd:enumeration value="INVALID_IDENTITY_CLAIM">
1239    <xsd:annotation>
1240    <xsd:documentation>The identity claim requested is already in use.</xsd:documentation>
1241    </xsd:annotation>
1242    </xsd:enumeration>
1243    <xsd:enumeration value="NONEXISTANT_DATA">
1244    <xsd:annotation>
1245    <xsd:documentation>The data requested for deletion does not exist.</xsd:documentation>
1246    </xsd:annotation>
1247    </xsd:enumeration>
1248    </xsd:restriction>
1249    </xsd:simpleType>
1250    <xsd:complexType name="BIASFaultDetail">
```

```
1251    <xsd:annotation>
1252    <xsd:documentation>Defines the error information associated with a SOAP fault.</xsd:documentation>
1253    </xsd:annotation>
1254    <xsd:sequence>
1255    <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultType" type="tns:BIASFaultCode">
1256    <xsd:annotation>
1257    <xsd:documentation>References an error code.</xsd:documentation>
1258    </xsd:annotation>
1259    </xsd:element>
1260    <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultMessage" type="xsd:string">
1261    <xsd:annotation>
1262    <xsd:documentation>Provides an explanation of the fault.</xsd:documentation>
1263    </xsd:annotation>
1264    </xsd:element>
1265    <xsd:element minOccurs="0" maxOccurs="1" name="BIASFaultDescription" type="xsd:string">
1266    <xsd:annotation>
1267    <xsd:documentation>Provides detailed information about a BIAS fault, such as trace details.</xsd:documentation>
1268    </xsd:annotation>
1269    </xsd:element>
1270    </xsd:sequence>
1271    </xsd:complexType>
1272    <xsd:complexType name="BIASIdentity">
1273    <xsd:annotation>
1274    <xsd:documentation>
1275    Defines a single element for encapsulating the data associated
1276    with an Identity.  Includes the Identity's reference identifiers,
1277    biographic data, and biometric data.
1278    </xsd:documentation>
1279    </xsd:annotation>
1280    <xsd:sequence>
1281    <xsd:element name="SubjectID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1282    <xsd:annotation>
1283    <xsd:documentation>A system unique identifier for a subject.</xsd:documentation>
1284    </xsd:annotation>
1285    </xsd:element>
1286    <xsd:element name="IdentityClaim" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1287    <xsd:annotation>
1288    <xsd:documentation>An identifier by which a subject is known to a particular gallery or population
1289    group.</xsd:documentation>
1290    </xsd:annotation>
1291    </xsd:element>
```

```xsd
1292    <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1293    <xsd:annotation>
1294    <xsd:documentation>The identifier of an encounter associated with the subject, required for encounter-centric
1295    models.</xsd:documentation>
1296    </xsd:annotation>
1297    </xsd:element>
1298    <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
1299    <xsd:annotation>
1300    <xsd:documentation>A list of encounters associated with a subject.</xsd:documentation>
1301    </xsd:annotation>
1302    </xsd:element>
1303    <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1304    <xsd:annotation>
1305    <xsd:documentation>An Identity's biographic data.</xsd:documentation>
1306    </xsd:annotation>
1307    </xsd:element>
1308    <xsd:element name="BiographicDataElements" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1309    <xsd:annotation>
1310    <xsd:documentation>An Identity's biographic data elements that are stored in the implementing
1311    system.</xsd:documentation>
1312    </xsd:annotation>
1313    </xsd:element>
1314    <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="0" maxOccurs="1">
1315    <xsd:annotation>
1316    <xsd:documentation>An Identity's biometric data.</xsd:documentation>
1317    </xsd:annotation>
1318    </xsd:element>
1319    </xsd:sequence>
1320    </xsd:complexType>
1321    <xsd:simpleType name="BIASIDType">
1322    <xsd:annotation>
1323    <xsd:documentation>A BIAS identifier</xsd:documentation>
1324    </xsd:annotation>
1325    <xsd:restriction base="xsd:string"/>
1326    </xsd:simpleType>
1327    <xsd:complexType name="BiographicDataItemType">
1328    <xsd:annotation>
1329    <xsd:documentation>Defines a single biographic data element.</xsd:documentation>
1330    </xsd:annotation>
1331    <xsd:sequence>
1332    <xsd:element name="Name" type="xsd:string" minOccurs="1" maxOccurs="1">
```

1333    <xsd:annotation>

1334    <xsd:documentation>The name of the biographic data item.</xsd:documentation>

1335    </xsd:annotation>

1336    </xsd:element>

1337    <xsd:element name="Type" type="xsd:string" minOccurs="1" maxOccurs="1">

1338    <xsd:annotation>

1339    <xsd:documentation>The data type for the biographic data item.</xsd:documentation>

1340    </xsd:annotation>

1341    </xsd:element>

1342    <xsd:element name="Value" type="xsd:string" minOccurs="0" maxOccurs="1">

1343    <xsd:annotation>

1344    <xsd:documentation>The value assigned to the biographic data item.</xsd:documentation>

1345    </xsd:annotation>

1346    </xsd:element>

1347    </xsd:sequence>

1348    </xsd:complexType>

1349    <xsd:complexType name="BiographicDataSetType">

1350    <xsd:annotation>

1351    <xsd:documentation>Defines a set of biographic data that is formatted according to the specified
1352    format.</xsd:documentation>

1353    </xsd:annotation>

1354    <xsd:sequence>

1355    <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1">

1356    <xsd:annotation>

1357    <xsd:documentation>The name of the biographic data format. Use these names for common formats: FBI-EFTS,
1358    FBI-EBTS, DOD-EBTS, INT-I, NIEM, xNAL, HR-XML.</xsd:documentation>

1359    </xsd:annotation>

1360    </xsd:element>

1361    <xsd:element name="version" type="xsd:string" minOccurs="0" maxOccurs="1">

1362    <xsd:annotation>

1363    <xsd:documentation>The version of the biographic data format (e.g., "7.1" for FBI-EFTS or "2.0" for
1364    NIEM).</xsd:documentation>

1365    </xsd:annotation>

1366    </xsd:element>

1367    <xsd:element name="source" type="xsd:string" minOccurs="1" maxOccurs="1">

1368    <xsd:annotation>

1369    <xsd:documentation>Reference to a URI/IRI describing the biographic data format. For example: (FBI-EFTS)
1370    www.fbibiospecs.org, (DOD-EBTS) www.biometrics.dod.mil, (INT-I) www.interpol.int, (NIEM) www.niem.gov,
1371    (xNAL) www.oasis-open.org, (HR-XML) www.hr-xml.org.</xsd:documentation>

1372    </xsd:annotation>

1373    </xsd:element>

1374    <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1">

1375    &lt;xsd:annotation&gt;

1376    &lt;xsd:documentation&gt;The biographic data format type. Use these types for common formats: ASCII (e.g., for non-

1377    XML versions of FBI-EFTS, FBI-EBTS, DOD-EFTS, or INT-I), XML (e.g., for NIEM, xNAL, and HR-XML or future version

1378    of FBI-EBTS).&lt;/xsd:documentation&gt;

1379

1380    &lt;/xsd:annotation&gt;

1381    &lt;/xsd:element&gt;

1382    &lt;xsd:any namespace="##any"&gt;

1383    &lt;xsd:annotation&gt;

1384    &lt;xsd:documentation&gt;Biographic data formatted according to a specific format.&lt;/xsd:documentation&gt;

1385    &lt;/xsd:annotation&gt;

1386    &lt;/xsd:any&gt;

1387    &lt;/xsd:sequence&gt;

1388    &lt;/xsd:complexType&gt;

1389    &lt;xsd:complexType name="BiographicDataType"&gt;

1390    &lt;xsd:annotation&gt;

1391    &lt;xsd:documentation&gt;

1392    Defines a set of biographic data elements, utilizing either the

1393    BiographicDataItemType to represent a list of elements or the

1394    BiographicDataSetType to represent a complete, formatted set of

1395    biographic information.

1396    &lt;/xsd:documentation&gt;

1397    &lt;/xsd:annotation&gt;

1398    &lt;xsd:sequence&gt;

1399    &lt;xsd:element name="LastName" type="xsd:string" minOccurs="0" maxOccurs="1"&gt;

1400    &lt;xsd:annotation&gt;

1401    &lt;xsd:documentation&gt;The last name of a subject.&lt;/xsd:documentation&gt;

1402    &lt;/xsd:annotation&gt;

1403    &lt;/xsd:element&gt;

1404    &lt;xsd:element name="FirstName" type="xsd:string" minOccurs="0" maxOccurs="1"&gt;

1405    &lt;xsd:annotation&gt;

1406    &lt;xsd:documentation&gt;The first name of a subject.&lt;/xsd:documentation&gt;

1407    &lt;/xsd:annotation&gt;

1408    &lt;/xsd:element&gt;

1409    &lt;xsd:element name="BiographicDataItems" minOccurs="0"&gt;

1410    &lt;xsd:complexType&gt;

1411    &lt;xsd:sequence&gt;

1412    &lt;xsd:element name="BiographicDataItem" type="tns:BiographicDataItemType" maxOccurs="unbounded"&gt;

1413    &lt;xsd:annotation&gt;

1414    &lt;xsd:documentation&gt;A single biographic data element.&lt;/xsd:documentation&gt;

1415    &lt;/xsd:annotation&gt;

```
1416   </xsd:element>
1417   </xsd:sequence>
1418   </xsd:complexType>
1419   </xsd:element>
1420   <xsd:element name="BiographicDataSet" type="tns:BiographicDataSetType" minOccurs="0" maxOccurs="1">
1421   <xsd:annotation>
1422   <xsd:documentation>A set of biographic data information.</xsd:documentation>
1423   </xsd:annotation>
1424   </xsd:element>
1425   </xsd:sequence>
1426   </xsd:complexType>
1427   <xsd:complexType name="BiometricDataElementType">
1428   <xsd:annotation>
1429   <xsd:documentation>
1430   Provides descriptive information about biometric data, such as
1431   the biometric type, subtype, and format, contained in the BDB of
1432   the CBEFF-BIR.
1433   </xsd:documentation>
1434   </xsd:annotation>
1435   <xsd:sequence>
1436   <xsd:element name="BiometricType" type="oasis_cbeff:MultipleTypesType" minOccurs="1" maxOccurs="1">
1437   <xsd:annotation>
1438   <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1439   CBEFF.</xsd:documentation>
1440   </xsd:annotation>
1441   </xsd:element>
1442   <xsd:element name="BiometricTypeCount" type="xsd:positiveInteger" minOccurs="0" maxOccurs="1">
1443   <xsd:annotation>
1444   <xsd:documentation>The number of biometric records having the biometric type recorded in the biometric type
1445   field.</xsd:documentation>
1446   </xsd:annotation>
1447   </xsd:element>
1448   <xsd:element name="BiometricSubType" type="oasis_cbeff:SubtypeType" minOccurs="0" maxOccurs="1">
1449   <xsd:annotation>
1450   <xsd:documentation>More specifically defines the type of biometric data stored in the biometric record, as
1451   defined by CBEFF.</xsd:documentation>
1452   </xsd:annotation>
1453   </xsd:element>
1454   <xsd:element name="BDBFormatOwner" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">
1455   <xsd:annotation>
1456   <xsd:documentation>Identifies the standards body, working group, industry consortium, or other CBEFF biometric
1457   organization that has defined the format for the biometric data.</xsd:documentation>
```

```
1458    </xsd:annotation>
1459    </xsd:element>
1460    <xsd:element name="BDBFormatType" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">
1461    <xsd:annotation>
1462    <xsd:documentation>Identifies the specific biometric data format specified by the CBEFF biometric organization
1463    recorded in the BDB Format Owner field.</xsd:documentation>
1464    </xsd:annotation>
1465    </xsd:element>
1466    </xsd:sequence>
1467    </xsd:complexType>
1468    <xsd:complexType name="BiometricDataListType">
1469    <xsd:annotation>
1470    <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1471    </xsd:annotation>
1472    <xsd:sequence>
1473    <xsd:element name="BiometricDataElement" type="tns:BiometricDataElementType" minOccurs="0"
1474    maxOccurs="unbounded">
1475    <xsd:annotation>
1476    <xsd:documentation>Data structure containing information about a biometric record.</xsd:documentation>
1477    </xsd:annotation>
1478    </xsd:element>
1479    </xsd:sequence>
1480    </xsd:complexType>
1481    <xsd:complexType name="CandidateListType">
1482    <xsd:annotation>
1483    <xsd:documentation>
1484    Defines a set of candidates, utilizing the Candidate Type to
1485    represent each element in the set.
1486    </xsd:documentation>
1487    </xsd:annotation>
1488    <xsd:sequence>
1489    <xsd:element name="Candidate" type="tns:CandidateType" minOccurs="0" maxOccurs="unbounded">
1490    <xsd:annotation>
1491    <xsd:documentation>A single candidate.</xsd:documentation>
1492    </xsd:annotation>
1493    </xsd:element>
1494    </xsd:sequence>
1495    </xsd:complexType>
1496    <xsd:complexType name="CandidateType">
1497    <xsd:annotation>
1498    <xsd:documentation>
```

1499     Defines a single candidate as a possible match in response to a

1500     biometric identification request.

1501     </xsd:documentation>

1502     </xsd:annotation>

1503     <xsd:sequence>

1504     <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">

1505     <xsd:annotation>

1506     <xsd:documentation>The match score.</xsd:documentation>

1507     </xsd:annotation>

1508     </xsd:element>

1509     <xsd:element name="Rank" type="xsd:integer" minOccurs="1" maxOccurs="1">

1510     <xsd:annotation>

1511     <xsd:documentation>The rank of the candidate in relation to other candidates for the same biometric
1512     identification operation.</xsd:documentation>

1513     </xsd:annotation>

1514     </xsd:element>

1515     <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">

1516     <xsd:annotation>

1517     <xsd:documentation>Biographic data associated with the candidate match.</xsd:documentation>

1518     </xsd:annotation>

1519     </xsd:element>

1520     <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="1" maxOccurs="1">

1521     <xsd:annotation>

1522     <xsd:documentation>Biometric data associated with the candidate match.</xsd:documentation>

1523     </xsd:annotation>

1524     </xsd:element>

1525     </xsd:sequence>

1526     </xsd:complexType>

1527     <xsd:complexType name="CapabilityListType">

1528     <xsd:annotation>

1529     <xsd:documentation>Defines a set of capabilities.</xsd:documentation>

1530     </xsd:annotation>

1531     <xsd:sequence>

1532     <xsd:element name="Capability" type="tns:CapabilityType" minOccurs="0" maxOccurs="unbounded">

1533     <xsd:annotation>

1534     <xsd:documentation>A single capability.</xsd:documentation>

1535     </xsd:annotation>

1536     </xsd:element>

1537     </xsd:sequence>

1538     </xsd:complexType>

1539     <xsd:simpleType name="CapabilityName">

```
1540    <xsd:annotation>

1541    <xsd:documentation>A list of capability items.</xsd:documentation>

1542    </xsd:annotation>

1543    <xsd:restriction base="xsd:string">

1544    <xsd:enumeration value="AggregateInputDataOptional">

1545    <xsd:annotation>

1546    <xsd:documentation>A data element accepted as optional input by the implementing system for the aggregate
1547    services.</xsd:documentation>

1548    </xsd:annotation>

1549    </xsd:enumeration>

1550    <xsd:enumeration value="AggregateInputDataRequired">

1551    <xsd:annotation>

1552    <xsd:documentation>A data element required as input by the implementing system for the aggregate
1553    services.</xsd:documentation>

1554    </xsd:annotation>

1555    </xsd:enumeration>

1556    <xsd:enumeration value="AggregateProcessingOption">

1557    <xsd:annotation>

1558    <xsd:documentation>A processing option supported by the implementing system for the aggregate
1559    services.</xsd:documentation>

1560    </xsd:annotation>

1561    </xsd:enumeration>

1562    <xsd:enumeration value="AggregateReturnData">

1563    <xsd:annotation>

1564    <xsd:documentation>A data element returned by the implementing system for the aggregate
1565    services.</xsd:documentation>

1566    </xsd:annotation>

1567    </xsd:enumeration>

1568    <xsd:enumeration value="AggregateServiceDescription">

1569    <xsd:annotation>

1570    <xsd:documentation>Describes the processing logic of an aggregate service supported by the implementing
1571    system.</xsd:documentation>

1572    </xsd:annotation>

1573    </xsd:enumeration>

1574    <xsd:enumeration value="BiographicDataSet">

1575    <xsd:annotation>

1576    <xsd:documentation>Identifies a biographic data set supported by the implementing
1577    system.</xsd:documentation>

1578    </xsd:annotation>

1579    </xsd:enumeration>

1580    <xsd:enumeration value="CBEFFPatronFormat">

1581    <xsd:annotation>
```

```xml
1582    <xsd:documentation>A patron format supported by the implementing system.</xsd:documentation>
1583    </xsd:annotation>
1584    </xsd:enumeration>
1585    <xsd:enumeration value="ClassificationAlgorithmType">
1586    <xsd:annotation>
1587    <xsd:documentation>A classification algorithm type supported by the implementing system.</xsd:documentation>
1588    </xsd:annotation>
1589    </xsd:enumeration>
1590    <xsd:enumeration value="ConformanceClass">
1591    <xsd:annotation>
1592    <xsd:documentation>Identifies the conformance class of the BIAS implementation.</xsd:documentation>
1593    </xsd:annotation>
1594    </xsd:enumeration>
1595    <xsd:enumeration value="Gallery">
1596    <xsd:annotation>
1597    <xsd:documentation>A gallery or population group supported by the implementing system.</xsd:documentation>
1598    </xsd:annotation>
1599    </xsd:enumeration>
1600    <xsd:enumeration value="IdentityModel">
1601    <xsd:annotation>
1602    <xsd:documentation>Identifies whether the implementing system is person-centric or encounter-centric
1603    based.</xsd:documentation>
1604    </xsd:annotation>
1605    </xsd:enumeration>
1606    <xsd:enumeration value="MatchScore">
1607    <xsd:annotation>
1608    <xsd:documentation> Identifies the use of match scores returned by the implementing
1609    system.</xsd:documentation>
1610    </xsd:annotation>
1611    </xsd:enumeration>
1612    <xsd:enumeration value="QualityAlgorithm">
1613    <xsd:annotation>
1614    <xsd:documentation>A quality algorithm vendor and algorithm vendor product ID supported by the implementing
1615    system.</xsd:documentation>
1616    </xsd:annotation>
1617    </xsd:enumeration>
1618    <xsd:enumeration value="SupportedBiometric">
1619    <xsd:annotation>
1620    <xsd:documentation>A biometric type supported by the implementing system.</xsd:documentation>
1621    </xsd:annotation>
1622    </xsd:enumeration>
```

```xml
1623    <xsd:enumeration value="TransformOperation">
1624    <xsd:annotation>
1625    <xsd:documentation>A transform operation type supported by the implementing system.</xsd:documentation>
1626    </xsd:annotation>
1627    </xsd:enumeration>
1628    </xsd:restriction>
1629    </xsd:simpleType>
1630    <xsd:complexType name="CapabilityType">
1631    <xsd:annotation>
1632    <xsd:documentation>Defines a single capability supported by an implementing system.</xsd:documentation>
1633    </xsd:annotation>
1634    <xsd:sequence>
1635    <xsd:element name="CapabilityName" type="tns:CapabilityName" minOccurs="1" maxOccurs="1">
1636    <xsd:annotation>
1637    <xsd:documentation>The name of the capability.</xsd:documentation>
1638    </xsd:annotation>
1639    </xsd:element>
1640    <xsd:element name="CapabilityID" type="xsd:string" minOccurs="0" maxOccurs="1">
1641    <xsd:annotation>
1642    <xsd:documentation>An identifier assigned to the capability by the implementing system.</xsd:documentation>
1643    </xsd:annotation>
1644    </xsd:element>
1645    <xsd:element name="CapabilityDescription" type="xsd:string" minOccurs="0" maxOccurs="1">
1646    <xsd:annotation>
1647    <xsd:documentation>A description of the capability.</xsd:documentation>
1648    </xsd:annotation>
1649    </xsd:element>
1650    <xsd:element name="CapabilityValue" type="xsd:string" minOccurs="0" maxOccurs="1">
1651    <xsd:annotation>
1652    <xsd:documentation>A value assigned to the capability.</xsd:documentation>
1653    </xsd:annotation>
1654    </xsd:element>
1655    <xsd:element name="CapabilitySupportingValue" type="xsd:string" minOccurs="0" maxOccurs="1">
1656    <xsd:annotation>
1657    <xsd:documentation>A secondary value supporting the capability.</xsd:documentation>
1658    </xsd:annotation>
1659    </xsd:element>
1660    <xsd:element name="CapabilityAdditionalInfo" type="xsd:string" minOccurs="0" maxOccurs="1">
1661    <xsd:annotation>
1662    <xsd:documentation> Contains additional information for the supported capability.</xsd:documentation>
1663    </xsd:annotation>
```

```
1664    </xsd:element>
1665    </xsd:sequence>
1666    </xsd:complexType>
1667    <xsd:complexType name="CBEFF_BIR_ListType">
1668    <xsd:annotation>
1669    <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1670    </xsd:annotation>
1671    <xsd:sequence>
1672    <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="unbounded">
1673    <xsd:annotation>
1674    <xsd:documentation>CBEFF structure containing information about a biometric sample.</xsd:documentation>
1675    </xsd:annotation>
1676    </xsd:element>
1677    </xsd:sequence>
1678    </xsd:complexType>
1679    <xsd:complexType name="CBEFF_BIR_Type">
1680    <xsd:annotation>
1681    <xsd:documentation>Represents biometric information, with either a non-XML or XML
1682    representation.</xsd:documentation>
1683    </xsd:annotation>
1684    <xsd:sequence>
1685    <xsd:element name="FormatOwner" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
1686    <xsd:element name="FormatType" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1" />
1687    <xsd:element name="BIR_Information" minOccurs="0" maxOccurs="1">
1688    <xsd:complexType>
1689    <xsd:sequence>
1690    <xsd:element name="BIR_Info" type="oasis_cbeff:BIRInfoType" minOccurs="0" maxOccurs="1"/>
1691    <xsd:element name="BDB_Info" type="oasis_cbeff:BDBInfoType" minOccurs="0" maxOccurs="1"/>
1692    <xsd:element name="SB_Info" type="oasis_cbeff:SBInfoType" minOccurs="0" maxOccurs="1"/>
1693    </xsd:sequence>
1694    </xsd:complexType>
1695    </xsd:element>
1696    <xsd:element name="BIR" type="tns:BaseBIRType" />
1697    </xsd:sequence>
1698    </xsd:complexType>
1699    <xsd:complexType name="BaseBIRType">
1700    </xsd:complexType>
1701    <xsd:complexType name="BinaryBIR">
1702    <xsd:complexContent>
1703    <xsd:extension base="tns:BaseBIRType">
1704    <xsd:sequence>
```

```
1705    <xsd:element name="Binary" type="xsd:base64Binary" />
1706    </xsd:sequence>
1707    </xsd:extension>
1708    </xsd:complexContent>
1709    </xsd:complexType>
1710    <xsd:complexType name="URI_BIR">
1711    <xsd:complexContent>
1712    <xsd:extension base="tns:BaseBIRType">
1713    <xsd:sequence>
1714    <xsd:element name="URI" type="xsd:anyURI" />
1715    </xsd:sequence>
1716    </xsd:extension>
1717    </xsd:complexContent>
1718    </xsd:complexType>
1719    <xsd:complexType name="XML_BIR">
1720    <xsd:complexContent>
1721    <xsd:extension base="tns:BaseBIRType">
1722    <xsd:sequence>
1723    <xsd:element name="XML" type="oasis_cbeff:BIRType" />
1724    </xsd:sequence>
1725    </xsd:extension>
1726    </xsd:complexContent>
1727    </xsd:complexType>
1728    <xsd:simpleType name="Classification">
1729    <xsd:annotation>
1730    <xsd:documentation>The result of a classification.</xsd:documentation>
1731    </xsd:annotation>
1732    <xsd:restriction base="xsd:string"/>
1733    </xsd:simpleType>
1734    <xsd:simpleType name="ClassificationAlgorithmType">
1735    <xsd:annotation>
1736    <xsd:documentation>Type of classification algorithm that was used to perform the
1737    classification.</xsd:documentation>
1738    </xsd:annotation>
1739    <xsd:restriction base="xsd:string"/>
1740    </xsd:simpleType>
1741    <xsd:complexType name="ClassificationData">
1742    <xsd:annotation>
1743    <xsd:documentation>Contains information on classification results and the algorithm used to determine the
1744    classification.</xsd:documentation>
1745    </xsd:annotation>
```

```xsd
1746    <xsd:sequence>
1747    <xsd:element name="Classification" type="tns:Classification" minOccurs="1" maxOccurs="1">
1748    <xsd:annotation>
1749    <xsd:documentation>The result of the classification.</xsd:documentation>
1750    </xsd:annotation>
1751    </xsd:element>
1752    <xsd:element name="ClassificationAlgorithmType" type="tns:ClassificationAlgorithmType" minOccurs="1"
1753    maxOccurs="1">
1754    <xsd:annotation>
1755    <xsd:documentation>Identifies the type of classification algorithm that was used to perform the
1756    classification.</xsd:documentation>
1757    </xsd:annotation>
1758    </xsd:element>
1759    </xsd:sequence>
1760    </xsd:complexType>
1761    <xsd:complexType name="EncounterListType">
1762    <xsd:annotation>
1763    <xsd:documentation>Defines a set of encounters.</xsd:documentation>
1764    </xsd:annotation>
1765    <xsd:sequence>
1766    <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="unbounded">
1767    <xsd:annotation>
1768    <xsd:documentation>The identifier of an encounter.</xsd:documentation>
1769    </xsd:annotation>
1770    </xsd:element>
1771    </xsd:sequence>
1772    </xsd:complexType>
1773    <xsd:complexType name="FusionInformationListType">
1774    <xsd:annotation>
1775    <xsd:documentation>
1776    Contains at a minimum two sets of fusion input
1777    elements, as input to the PerformFusion request.
1778    </xsd:documentation>
1779    </xsd:annotation>
1780    <xsd:sequence>
1781    <xsd:element name="FusionElement" type="tns:FusionInformationType" minOccurs="2"
1782    maxOccurs="unbounded">
1783    <xsd:annotation>
1784    <xsd:documentation>A set of fusion information.</xsd:documentation>
1785    </xsd:annotation>
1786    </xsd:element>
```

```
1787    </xsd:sequence>
1788    </xsd:complexType>
1789    <xsd:complexType name="FusionInformationType">
1790    <xsd:annotation>
1791    <xsd:documentation>Represents the information necessary to perform a fusion operation.</xsd:documentation>
1792    </xsd:annotation>
1793    <xsd:sequence>
1794    <xsd:element name="BiometricType" type="oasis_cbeff:MultipleTypesType" minOccurs="1" maxOccurs="1">
1795    <xsd:annotation>
1796    <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1797    CBEFF.</xsd:documentation>
1798    </xsd:annotation>
1799    </xsd:element>
1800    <xsd:element name="BiometricSubType" type="oasis_cbeff:SubtypeType" minOccurs="0" maxOccurs="1">
1801    <xsd:annotation>
1802    <xsd:documentation>More specifically defines the type of biometric data stored in the biometric
1803    record.</xsd:documentation>
1804    </xsd:annotation>
1805    </xsd:element>
1806    <xsd:element name="AlgorithmOwner" type="xsd:string" minOccurs="1" maxOccurs="1">
1807    <xsd:annotation>
1808    <xsd:documentation>The owner or vendor of the algorithm used to determine the score or
1809    decision.</xsd:documentation>
1810    </xsd:annotation>
1811    </xsd:element>
1812    <xsd:element name="AlgorithmType" type="xsd:string" minOccurs="1" maxOccurs="1">
1813    <xsd:annotation>
1814    <xsd:documentation>The Algorithm Owner's identifier for the specific algorithm product and version used to
1815    determine the score or decision.</xsd:documentation>
1816    </xsd:annotation>
1817    </xsd:element>
1818    <xsd:element name="FusionResult" type="tns:FusionResult" />
1819    </xsd:sequence>
1820    </xsd:complexType>
1821    <xsd:complexType name="FusionResult">
1822    <xsd:sequence />
1823    </xsd:complexType>
1824    <xsd:complexType name="FusionScore">
1825    <xsd:complexContent>
1826    <xsd:extension base="tns:FusionResult">
1827    <xsd:sequence>
1828    <xsd:element name="Score" type="tns:Score">
```

```
1829    <xsd:annotation>
1830    <xsd:documentation>The similarity score assigned by the matching algorithm.</xsd:documentation>
1831    </xsd:annotation>
1832    </xsd:element>
1833    </xsd:sequence>
1834    </xsd:extension>
1835    </xsd:complexContent>
1836    </xsd:complexType>
1837    <xsd:complexType name="FusionDecision">
1838    <xsd:complexContent>
1839    <xsd:extension base="tns:FusionResult">
1840    <xsd:sequence>
1841    <xsd:element name="Decision" type="xsd:string">
1842    <xsd:annotation>
1843    <xsd:documentation>The match decision assigned by the matching algorithm.</xsd:documentation>
1844    </xsd:annotation>
1845    </xsd:element>
1846    </xsd:sequence>
1847    </xsd:extension>
1848    </xsd:complexContent>
1849    </xsd:complexType>
1850    <xsd:complexType name="GenericRequestParameters">
1851    <xsd:annotation>
1852    <xsd:documentation>Common request paramters that can be used to identify the
1853    requester.</xsd:documentation>
1854    </xsd:annotation>
1855    <xsd:sequence>
1856    <xsd:element name="Application" type="tns:ApplicationIdentifier" minOccurs="0" maxOccurs="1">
1857    <xsd:annotation>
1858    <xsd:documentation>Identifies the requesting application.</xsd:documentation>
1859    </xsd:annotation>
1860    </xsd:element>
1861    <xsd:element name="ApplicationUser" type="tns:ApplicationUserIdentifier" minOccurs="0" maxOccurs="1">
1862    <xsd:annotation>
1863    <xsd:documentation>Identifers the user or instance of the requesting application.</xsd:documentation>
1864    </xsd:annotation>
1865    </xsd:element>
1866    <xsd:element name="BIASOperationName" type="xsd:string" minOccurs="0" maxOccurs="1">
1867    <xsd:annotation>
1868    <xsd:documentation>Identifers the BIAS operation name that is being requested.</xsd:documentation>
1869    </xsd:annotation>
```

```xml
1870        </xsd:element>
1871        </xsd:sequence>
1872        </xsd:complexType>
1873        <xsd:complexType name="InformationType">
1874        <xsd:annotation>
1875        <xsd:documentation>
1876        Allows for an unlimited number of data element types, and it does
1877        not specify nor require any particular data element.
1878        </xsd:documentation>
1879        </xsd:annotation>
1880        <xsd:sequence>
1881        <xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
1882        </xsd:sequence>
1883        </xsd:complexType>
1884        <xsd:complexType name="ListFilterType">
1885        <xsd:annotation>
1886        <xsd:documentation>
1887        Provides a method to filter the amount of information returned in
1888        a search of biometric data.
1889        </xsd:documentation>
1890        </xsd:annotation>
1891        <xsd:sequence>
1892        <xsd:element name="BiometricTypeFilters">
1893        <xsd:complexType>
1894        <xsd:sequence>
1895        <xsd:element name="BiometricTypeFilter" type="oasis_cbeff:MultipleTypesType" minOccurs="1"
1896        maxOccurs="unbounded">
1897        <xsd:annotation>
1898        <xsd:documentation>Limits the returned information to a specific type of biometric, as defined by
1899        CBEFF.</xsd:documentation>
1900        </xsd:annotation>
1901        </xsd:element>
1902        </xsd:sequence>
1903        </xsd:complexType>
1904        </xsd:element>
1905        <xsd:element name="IncludeBiometricSubtype" type="xsd:boolean" minOccurs="1" maxOccurs="1">
1906        <xsd:annotation>
1907        <xsd:documentation>A Boolean flag indicating if biometric subtype information should be
1908        returned.</xsd:documentation>
1909        </xsd:annotation>
1910        </xsd:element>
```

```
1911    </xsd:sequence>
1912    </xsd:complexType>
1913    <xsd:simpleType name="MatchType">
1914    <xsd:annotation>
1915    <xsd:documentation>The result of a fusion method.</xsd:documentation>
1916    </xsd:annotation>
1917    <xsd:restriction base="xsd:boolean"/>
1918    </xsd:simpleType>
1919    <xsd:complexType name="ProcessingOptionsType">
1920    <xsd:annotation>
1921    <xsd:documentation>
1922    BIAS aggregate services support the ability to include various
1923    processing options which direct and possibly control the business
1924    logic for that service. The ProcessingOptionsType provides a
1925    method to represent those options. Processing options should be
1926    defined by the implementing system.
1927    </xsd:documentation>
1928    </xsd:annotation>
1929    <xsd:sequence>
1930    <xsd:element name="Option" type="xsd:string" minOccurs="0" maxOccurs="unbounded">
1931    <xsd:annotation>
1932    <xsd:documentation>An option supported by the implementing system.</xsd:documentation>
1933    </xsd:annotation>
1934    </xsd:element>
1935    </xsd:sequence>
1936    </xsd:complexType>
1937    <xsd:simpleType name="ProductID">
1938    <xsd:annotation>
1939    <xsd:documentation>The vendor's ID for a particular product.</xsd:documentation>
1940    </xsd:annotation>
1941    <xsd:restriction base="xsd:string"/>
1942    </xsd:simpleType>
1943    <xsd:complexType name="QualityData">
1944    <xsd:annotation>
1945    <xsd:documentation>Contains information about a biometric sample's quality and the algorithm used to compute
1946    the quality.</xsd:documentation>
1947    </xsd:annotation>
1948    <xsd:sequence>
1949    <xsd:element name="QualityScore" type="oasis_cbeff:QualityType" minOccurs="0" maxOccurs="1">
1950    <xsd:annotation>
1951    <xsd:documentation>The quality of a biometric sample.</xsd:documentation>
```

```xml
1952    </xsd:annotation>
1953    </xsd:element>
1954    <xsd:element name="AlgorithmVendor" type="tns:VendorIdentifier" minOccurs="1" maxOccurs="1">
1955    <xsd:annotation>
1956    <xsd:documentation>The vendor of the qualilty algorithm used to determine the quality
1957    score.</xsd:documentation>
1958    </xsd:annotation>
1959    </xsd:element>
1960    <xsd:element name="AlgorithmVendorProductID" type="tns:ProductID" minOccurs="1" maxOccurs="1">
1961    <xsd:annotation>
1962    <xsd:documentation>The vendor's ID for the algorithm used to determine the quality.</xsd:documentation>
1963    </xsd:annotation>
1964    </xsd:element>
1965    <xsd:element name="AlgorithmVersion" type="tns:VersionType" minOccurs="0" maxOccurs="1">
1966    <xsd:annotation>
1967    <xsd:documentation>The version of the algorithm used to determine the quality.</xsd:documentation>
1968    </xsd:annotation>
1969    </xsd:element>
1970    </xsd:sequence>
1971    </xsd:complexType>
1972    <xsd:complexType name="RequestTemplate">
1973    <xsd:annotation>
1974    <xsd:documentation>Base template for BIAS primitive service requests.</xsd:documentation>
1975    </xsd:annotation>
1976    <xsd:sequence>
1977    <xsd:element name="GenericRequestParameters" type="tns:GenericRequestParameters" minOccurs="0"
1978    maxOccurs="1"/>
1979    </xsd:sequence>
1980    </xsd:complexType>
1981    <xsd:complexType name="ResponseStatus">
1982    <xsd:sequence>
1983    <xsd:element name="Return" type="tns:ReturnCode" minOccurs="1" maxOccurs="1">
1984    <xsd:annotation>
1985    <xsd:documentation>The return code indicates the return status of the operation.</xsd:documentation>
1986    </xsd:annotation>
1987    </xsd:element>
1988    <xsd:element name="Message" type="xsd:string" minOccurs="0" maxOccurs="1">
1989    <xsd:annotation>
1990    <xsd:documentation>A short message corresponding to the return code.</xsd:documentation>
1991    </xsd:annotation>
1992    </xsd:element>
```

```
1993    </xsd:sequence>
1994    </xsd:complexType>
1995    <xsd:complexType name="ResponseTemplate">
1996    <xsd:annotation>
1997    <xsd:documentation>Base template for BIAS responses.</xsd:documentation>
1998    </xsd:annotation>
1999    <xsd:sequence>
2000    <xsd:element name="ResponseStatus" type="tns:ResponseStatus" minOccurs="1" maxOccurs="1">
2001    <xsd:annotation>
2002    <xsd:documentation>Returned status for the operation.</xsd:documentation>
2003    </xsd:annotation>
2004    </xsd:element>
2005    </xsd:sequence>
2006    </xsd:complexType>
2007    <xsd:simpleType name="ReturnCode">
2008    <xsd:annotation>
2009    <xsd:documentation>BIAS Operation Return Codes</xsd:documentation>
2010    </xsd:annotation>
2011    <xsd:restriction base="xsd:unsignedLong">
2012    <xsd:enumeration value="0">
2013    <xsd:annotation>
2014    <xsd:documentation>Success</xsd:documentation>
2015    </xsd:annotation>
2016    </xsd:enumeration>
2017    </xsd:restriction>
2018    </xsd:simpleType>
2019    <xsd:simpleType name="Score">
2020    <xsd:annotation>
2021    <xsd:documentation>Match result or quality score.</xsd:documentation>
2022    </xsd:annotation>
2023    <xsd:restriction base="xsd:float"/>
2024    </xsd:simpleType>
2025    <xsd:complexType name="TokenType">
2026    <xsd:annotation>
2027    <xsd:documentation>Defines a token that is returned for asynchronous processing.</xsd:documentation>
2028    </xsd:annotation>
2029    <xsd:sequence>
2030    <xsd:element name="TokenValue" type="xsd:string" minOccurs="1" maxOccurs="1">
2031    <xsd:annotation>
2032    <xsd:documentation>A value returned by the implementing system that is used to retrieve the results to a service
2033    at a later time.</xsd:documentation>
```

```
2034    </xsd:annotation>
2035    </xsd:element>
2036    <xsd:element name="Expiration" type="xsd:date" minOccurs="1" maxOccurs="1">
2037    <xsd:annotation>
2038    <xsd:documentation>A date and time at which point the token expires and the service results are no longer
2039    guaranteed to be available.</xsd:documentation>
2040    </xsd:annotation>
2041    </xsd:element>
2042    </xsd:sequence>
2043    </xsd:complexType>
2044    <xsd:simpleType name="VendorIdentifier">
2045    <xsd:annotation>
2046    <xsd:documentation>Identifies a vendor.</xsd:documentation>
2047    </xsd:annotation>
2048    <xsd:restriction base="xsd:string"/>
2049    </xsd:simpleType>
2050    <xsd:complexType name="Version">
2051    <xsd:annotation>
2052    <xsd:documentation>
2053    For a description or definition of each data element, see the
2054    referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
2055    </xsd:documentation>
2056    </xsd:annotation>
2057    <xsd:sequence>
2058    <xsd:element name="major" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1"/>
2059    <xsd:element name="minor" type="xsd:nonNegativeInteger" minOccurs="1" maxOccurs="1"/>
2060    </xsd:sequence>
2061    </xsd:complexType>
2062    <xsd:simpleType name="VersionType">
2063    <xsd:annotation>
2064    <xsd:documentation>The version of a component.</xsd:documentation>
2065    </xsd:annotation>
2066    <xsd:restriction base="xsd:string"/>
2067    </xsd:simpleType>
2068    <xsd:element name="AddSubjectToGallery">
2069    <xsd:complexType>
2070    <xsd:annotation>
2071    <xsd:documentation>
2072    Register a subject to a given gallery or population group.
2073    </xsd:documentation>
2074    </xsd:annotation>
```

```
2075    <xsd:sequence>
2076    <xsd:element name="AddSubjectToGalleryRequest" nillable="true">
2077    <xsd:complexType>
2078    <xsd:complexContent>
2079    <xsd:extension base="tns:RequestTemplate">
2080    <xsd:sequence>
2081    <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2082    <xsd:annotation>
2083    <xsd:documentation>The identifier of the gallery or population group to which the subject will be
2084    added.</xsd:documentation>
2085    </xsd:annotation>
2086    </xsd:element>
2087    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2088    <xsd:annotation>
2089    <xsd:documentation>The identity to add to the gallery.</xsd:documentation>
2090    </xsd:annotation>
2091    </xsd:element>
2092    </xsd:sequence>
2093    </xsd:extension>
2094    </xsd:complexContent>
2095    </xsd:complexType>
2096    </xsd:element>
2097    </xsd:sequence>
2098    </xsd:complexType>
2099    </xsd:element>
2100    <xsd:element name="AddSubjectToGalleryResponse">
2101    <xsd:complexType>
2102    <xsd:annotation>
2103    <xsd:documentation>
2104    The response to an AddSubjectToGallery request.
2105    </xsd:documentation>
2106    </xsd:annotation>
2107    <xsd:sequence>
2108    <xsd:element name="AddSubjectToGalleryResponsePackage" nillable="true">
2109    <xsd:complexType>
2110    <xsd:complexContent>
2111    <xsd:extension base="tns:ResponseTemplate"/>
2112    </xsd:complexContent>
2113    </xsd:complexType>
2114    </xsd:element>
2115    </xsd:sequence>
```

```
2116    </xsd:complexType>
2117    </xsd:element>
2118    <xsd:element name="CheckQuality">
2119    <xsd:complexType>
2120    <xsd:annotation>
2121    <xsd:documentation>
2122    Calculate a quality score for a given biometric.
2123    </xsd:documentation>
2124    </xsd:annotation>
2125    <xsd:sequence>
2126    <xsd:element name="CheckQualityRequest" nillable="true">
2127    <xsd:complexType>
2128    <xsd:complexContent>
2129    <xsd:extension base="tns:RequestTemplate">
2130    <xsd:sequence>
2131    <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
2132    <xsd:annotation>
2133    <xsd:documentation>Data structure containing a single biometric sample for which a quality score is to be
2134    determined.</xsd:documentation>
2135    </xsd:annotation>
2136    </xsd:element>
2137    <xsd:element name="Quality" type="tns:QualityData" minOccurs="0" maxOccurs="1">
2138    <xsd:annotation>
2139    <xsd:documentation>Specifies a particular algorithm vendor and vender product ID.</xsd:documentation>
2140    </xsd:annotation>
2141    </xsd:element>
2142    </xsd:sequence>
2143    </xsd:extension>
2144    </xsd:complexContent>
2145    </xsd:complexType>
2146    </xsd:element>
2147    </xsd:sequence>
2148    </xsd:complexType>
2149    </xsd:element>
2150    <xsd:element name="CheckQualityResponse">
2151    <xsd:complexType>
2152    <xsd:annotation>
2153    <xsd:documentation>
2154    The response to a CheckQuality request.
2155    </xsd:documentation>
2156    </xsd:annotation>
```

```
2157    <xsd:sequence>
2158    <xsd:element name="CheckQualityResponsePackage" nillable="true">
2159    <xsd:complexType>
2160    <xsd:complexContent>
2161    <xsd:extension base="tns:ResponseTemplate">
2162    <xsd:sequence>
2163    <xsd:element name="QualityInfo" type="tns:QualityData" minOccurs="1" maxOccurs="1">
2164    <xsd:annotation>
2165    <xsd:documentation>Contains the quality information for the submitted biometric sample.</xsd:documentation>
2166    </xsd:annotation>
2167    </xsd:element>
2168    </xsd:sequence>
2169    </xsd:extension>
2170    </xsd:complexContent>
2171    </xsd:complexType>
2172    </xsd:element>
2173    </xsd:sequence>
2174    </xsd:complexType>
2175    </xsd:element>
2176    <xsd:element name="ClassifyBiometricData">
2177    <xsd:complexType>
2178    <xsd:annotation>
2179    <xsd:documentation>
2180    Classifies a biometric sample.
2181    </xsd:documentation>
2182    </xsd:annotation>
2183    <xsd:sequence>
2184    <xsd:element name="ClassifyBiometricDataRequest" nillable="true">
2185    <xsd:complexType>
2186    <xsd:complexContent>
2187    <xsd:extension base="tns:RequestTemplate">
2188    <xsd:sequence>
2189    <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
2190    <xsd:annotation>
2191    <xsd:documentation>Data structure containing a single biometric sample for which the classification is to be
2192    determined.</xsd:documentation>
2193    </xsd:annotation>
2194    </xsd:element>
2195    </xsd:sequence>
2196    </xsd:extension>
2197    </xsd:complexContent>
```

```
2198    </xsd:complexType>
2199    </xsd:element>
2200    </xsd:sequence>
2201    </xsd:complexType>
2202    </xsd:element>
2203    <xsd:element name="ClassifyBiometricDataResponse">
2204    <xsd:complexType>
2205    <xsd:annotation>
2206    <xsd:documentation>
2207    The response to a ClassifyBiometricData request, containing
2208    the classification of a biometric sample.
2209    </xsd:documentation>
2210    </xsd:annotation>
2211    <xsd:sequence>
2212    <xsd:element name="ClassifyBiometricDataResponsePackage" nillable="true">
2213    <xsd:complexType>
2214    <xsd:complexContent>
2215    <xsd:extension base="tns:ResponseTemplate">
2216    <xsd:sequence>
2217    <xsd:element name="ClassificationData" type="tns:ClassificationData" minOccurs="1" maxOccurs="1">
2218    <xsd:annotation>
2219    <xsd:documentation>Information on the results and type of classification performed.</xsd:documentation>
2220    </xsd:annotation>
2221    </xsd:element>
2222    </xsd:sequence>
2223    </xsd:extension>
2224    </xsd:complexContent>
2225    </xsd:complexType>
2226    </xsd:element>
2227    </xsd:sequence>
2228    </xsd:complexType>
2229    </xsd:element>
2230    <xsd:element name="CreateSubject">
2231    <xsd:complexType>
2232    <xsd:annotation>
2233    <xsd:documentation>
2234    Create a new subject record.
2235    </xsd:documentation>
2236    </xsd:annotation>
2237    <xsd:sequence>
2238    <xsd:element name="CreateSubjectRequest" nillable="true">
```

```
2239   <xsd:complexType>
2240   <xsd:complexContent>
2241   <xsd:extension base="tns:RequestTemplate"/>
2242   </xsd:complexContent>
2243   </xsd:complexType>
2244   </xsd:element>
2245   </xsd:sequence>
2246   </xsd:complexType>
2247   </xsd:element>
2248   <xsd:element name="CreateSubjectResponse">
2249   <xsd:complexType>
2250   <xsd:annotation>
2251   <xsd:documentation>
2252   The response to a CreateSubject request, containing the subject
2253   ID of the new subject record.
2254   </xsd:documentation>
2255   </xsd:annotation>
2256   <xsd:sequence>
2257   <xsd:element name="CreateSubjectResponsePackage" nillable="true">
2258   <xsd:complexType>
2259   <xsd:complexContent>
2260   <xsd:extension base="tns:ResponseTemplate">
2261   <xsd:sequence>
2262   <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2263   <xsd:annotation>
2264   <xsd:documentation>Contains the subject ID of the new subject record.</xsd:documentation>
2265   </xsd:annotation>
2266   </xsd:element>
2267   </xsd:sequence>
2268   </xsd:extension>
2269   </xsd:complexContent>
2270   </xsd:complexType>
2271   </xsd:element>
2272   </xsd:sequence>
2273   </xsd:complexType>
2274   </xsd:element>
2275   <xsd:element name="DeleteBiographicData">
2276   <xsd:complexType>
2277   <xsd:annotation>
2278   <xsd:documentation>
2279   Erase all of the biographic data associated with a given
```

2280      subject record or, in the encounter-centric model, with a

2281      given encounter.

2282      `</xsd:documentation>`

2283      `</xsd:annotation>`

2284      `<xsd:sequence>`

2285      `<xsd:element name="DeleteBiographicDataRequest" nillable="true">`

2286      `<xsd:complexType>`

2287      `<xsd:complexContent>`

2288      `<xsd:extension base="tns:RequestTemplate">`

2289      `<xsd:sequence>`

2290      `<xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">`

2291      `<xsd:annotation>`

2292      `<xsd:documentation>`Contains either the subject ID or encounter ID reference.`</xsd:documentation>`

2293      `</xsd:annotation>`

2294      `</xsd:element>`

2295      `</xsd:sequence>`

2296      `</xsd:extension>`

2297      `</xsd:complexContent>`

2298      `</xsd:complexType>`

2299      `</xsd:element>`

2300      `</xsd:sequence>`

2301      `</xsd:complexType>`

2302      `</xsd:element>`

2303      `<xsd:element name="DeleteBiographicDataResponse">`

2304      `<xsd:complexType>`

2305      `<xsd:annotation>`

2306      `<xsd:documentation>`

2307      The response to a DeleteBiographicData request.

2308      `</xsd:documentation>`

2309      `</xsd:annotation>`

2310      `<xsd:sequence>`

2311      `<xsd:element name="DeleteBiographicDataResponsePackage" nillable="true">`

2312      `<xsd:complexType>`

2313      `<xsd:complexContent>`

2314      `<xsd:extension base="tns:ResponseTemplate"/>`

2315      `</xsd:complexContent>`

2316      `</xsd:complexType>`

2317      `</xsd:element>`

2318      `</xsd:sequence>`

2319      `</xsd:complexType>`

2320      `</xsd:element>`

```xml
2321    <xsd:element name="DeleteBiometricData">
2322    <xsd:complexType>
2323    <xsd:annotation>
2324    <xsd:documentation>
2325    Erase all of the biometric data associated with a given
2326    subject record or, in the encounter-centric model, with a
2327    given encounter.
2328    </xsd:documentation>
2329    </xsd:annotation>
2330    <xsd:sequence>
2331    <xsd:element name="DeleteBiometricDataRequest" nillable="true">
2332    <xsd:complexType>
2333    <xsd:complexContent>
2334    <xsd:extension base="tns:RequestTemplate">
2335    <xsd:sequence>
2336    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2337    <xsd:annotation>
2338    <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
2339    </xsd:annotation>
2340    </xsd:element>
2341    </xsd:sequence>
2342    </xsd:extension>
2343    </xsd:complexContent>
2344    </xsd:complexType>
2345    </xsd:element>
2346    </xsd:sequence>
2347    </xsd:complexType>
2348    </xsd:element>
2349    <xsd:element name="DeleteBiometricDataResponse">
2350    <xsd:complexType>
2351    <xsd:annotation>
2352    <xsd:documentation>
2353    The response to a DeleteBiometricData request.
2354    </xsd:documentation>
2355    </xsd:annotation>
2356    <xsd:sequence>
2357    <xsd:element name="DeleteBiometricDataResponsePackage" nillable="true">
2358    <xsd:complexType>
2359    <xsd:complexContent>
2360    <xsd:extension base="tns:ResponseTemplate"/>
2361    </xsd:complexContent>
```

```xsd
2362    </xsd:complexType>
2363    </xsd:element>
2364    </xsd:sequence>
2365    </xsd:complexType>
2366    </xsd:element>
2367    <xsd:element name="DeleteSubject">
2368    <xsd:complexType>
2369    <xsd:annotation>
2370    <xsd:documentation>
2371    Delete an existing subject record and, in an encounter-centric
2372    model, any associated encounter information.
2373    </xsd:documentation>
2374    </xsd:annotation>
2375    <xsd:sequence>
2376    <xsd:element name="DeleteSubjectRequest" nillable="true">
2377    <xsd:complexType>
2378    <xsd:complexContent>
2379    <xsd:extension base="tns:RequestTemplate">
2380    <xsd:sequence>
2381    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2382    <xsd:annotation>
2383    <xsd:documentation>Subject ID of the identity to delete.</xsd:documentation>
2384    </xsd:annotation>
2385    </xsd:element>
2386    </xsd:sequence>
2387    </xsd:extension>
2388    </xsd:complexContent>
2389    </xsd:complexType>
2390    </xsd:element>
2391    </xsd:sequence>
2392    </xsd:complexType>
2393    </xsd:element>
2394    <xsd:element name="DeleteSubjectResponse">
2395    <xsd:complexType>
2396    <xsd:annotation>
2397    <xsd:documentation>
2398    The response to a DeleteSubject request.
2399    </xsd:documentation>
2400    </xsd:annotation>
2401    <xsd:sequence>
2402    <xsd:element name="DeleteSubjectResponsePackage" nillable="true">
```

```xsd
2403    <xsd:complexType>
2404    <xsd:complexContent>
2405    <xsd:extension base="tns:ResponseTemplate"/>
2406    </xsd:complexContent>
2407    </xsd:complexType>
2408    </xsd:element>
2409    </xsd:sequence>
2410    </xsd:complexType>
2411    </xsd:element>
2412    <xsd:element name="DeleteSubjectFromGallery">
2413    <xsd:complexType>
2414    <xsd:annotation>
2415    <xsd:documentation>
2416    Remove the registration of a subject from a gallery or
2417    population group.
2418    </xsd:documentation>
2419    </xsd:annotation>
2420    <xsd:sequence>
2421    <xsd:element name="DeleteSubjectFromGalleryRequest" nillable="true">
2422    <xsd:complexType>
2423    <xsd:complexContent>
2424    <xsd:extension base="tns:RequestTemplate">
2425    <xsd:sequence>
2426    <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2427    <xsd:annotation>
2428    <xsd:documentation>The identifier of the gallery or population group from which the subject will be
2429    deleted.</xsd:documentation>
2430    </xsd:annotation>
2431    </xsd:element>
2432    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2433    <xsd:annotation>
2434    <xsd:documentation>The identity to remove from the gallery.</xsd:documentation>
2435    </xsd:annotation>
2436    </xsd:element>
2437    </xsd:sequence>
2438    </xsd:extension>
2439    </xsd:complexContent>
2440    </xsd:complexType>
2441    </xsd:element>
2442    </xsd:sequence>
2443    </xsd:complexType>
```

```
2444   </xsd:element>
2445   <xsd:element name="DeleteSubjectFromGalleryResponse">
2446   <xsd:complexType>
2447   <xsd:annotation>
2448   <xsd:documentation>
2449   The response to a DeleteSubjectFromGallery request.
2450   </xsd:documentation>
2451   </xsd:annotation>
2452   <xsd:sequence>
2453   <xsd:element name="DeleteSubjectFromGalleryResponsePackage" nillable="true">
2454   <xsd:complexType>
2455   <xsd:complexContent>
2456   <xsd:extension base="tns:ResponseTemplate"/>
2457   </xsd:complexContent>
2458   </xsd:complexType>
2459   </xsd:element>
2460   </xsd:sequence>
2461   </xsd:complexType>
2462   </xsd:element>
2463   <xsd:element name="GetIdentifySubjectResults">
2464   <xsd:complexType>
2465   <xsd:annotation>
2466   <xsd:documentation>
2467   Retrieve the identification results for a specified token,
2468   which was returned by the Identify Subject service.
2469   </xsd:documentation>
2470   </xsd:annotation>
2471   <xsd:sequence>
2472   <xsd:element name="GetIdentifySubjectResultsRequest" nillable="true">
2473   <xsd:complexType>
2474   <xsd:complexContent>
2475   <xsd:extension base="tns:RequestTemplate">
2476   <xsd:sequence>
2477   <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
2478   <xsd:annotation>
2479   <xsd:documentation>A value used to retrieve the results of an IdentifySubject request.</xsd:documentation>
2480   </xsd:annotation>
2481   </xsd:element>
2482   </xsd:sequence>
2483   </xsd:extension>
2484   </xsd:complexContent>
```

```
2485    </xsd:complexType>
2486    </xsd:element>
2487    </xsd:sequence>
2488    </xsd:complexType>
2489    </xsd:element>
2490    <xsd:element name="GetIdentifySubjectResultsResponse">
2491    <xsd:complexType>
2492    <xsd:annotation>
2493    <xsd:documentation>
2494    The response to a GetIdentifySubjectResults request, which includes a candidate list.
2495    </xsd:documentation>
2496    </xsd:annotation>
2497    <xsd:sequence>
2498    <xsd:element name="GetIdentifySubjectResultsResponsePackage" nillable="true">
2499    <xsd:complexType>
2500    <xsd:complexContent>
2501    <xsd:extension base="tns:ResponseTemplate">
2502    <xsd:sequence>
2503    <xsd:element name="CandidateList" type="tns:CandidateListType" minOccurs="1" maxOccurs="1">
2504    <xsd:annotation>
2505    <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input biometric
2506    sample.</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:sequence>
2515    </xsd:complexType>
2516    </xsd:element>
2517    <xsd:element name="IdentifySubject">
2518    <xsd:complexType>
2519    <xsd:annotation>
2520    <xsd:documentation>
2521    Perform an identification search against a given gallery for
2522    a given biometric.
2523    </xsd:documentation>
2524    </xsd:annotation>
2525    <xsd:sequence>
```

```
2526    <xsd:element name="IdentifySubjectRequest" nillable="true">
2527    <xsd:complexType>
2528    <xsd:complexContent>
2529    <xsd:extension base="tns:RequestTemplate">
2530    <xsd:sequence>
2531    <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2532    <xsd:annotation>
2533    <xsd:documentation>The identifier of the gallery or population group which will be
2534    searched.</xsd:documentation>
2535    </xsd:annotation>
2536    </xsd:element>
2537    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2538    <xsd:annotation>
2539    <xsd:documentation>Contains the BIR, a data structure containing the biometric sample for the
2540    search.</xsd:documentation>
2541    </xsd:annotation>
2542    </xsd:element>
2543    <xsd:element name="MaxListSize" type="xsd:positiveInteger" minOccurs="1" maxOccurs="1">
2544    <xsd:annotation>
2545    <xsd:documentation>The maximum size of the candidate list that should be returned.</xsd:documentation>
2546    </xsd:annotation>
2547    </xsd:element>
2548    </xsd:sequence>
2549    </xsd:extension>
2550    </xsd:complexContent>
2551    </xsd:complexType>
2552    </xsd:element>
2553    </xsd:sequence>
2554    </xsd:complexType>
2555    </xsd:element>
2556    <xsd:element name="IdentifySubjectResponse">
2557    <xsd:complexType>
2558    <xsd:annotation>
2559    <xsd:documentation>
2560    The response to an IdentifySubject request, returning a
2561    rank-ordered candidate list.
2562    </xsd:documentation>
2563    </xsd:annotation>
2564    <xsd:sequence>
2565    <xsd:element name="IdentifySubjectResponsePackage" nillable="true">
2566    <xsd:complexType>
```

```xml
2567    <xsd:complexContent>
2568    <xsd:extension base="tns:ResponseTemplate">
2569    <xsd:sequence>
2570    <xsd:element name="IdentifySubjectResult" type="tns:IdentifySubjectResultType" />
2571    </xsd:sequence>
2572    </xsd:extension>
2573    </xsd:complexContent>
2574    </xsd:complexType>
2575    </xsd:element>
2576    </xsd:sequence>
2577    </xsd:complexType>
2578    </xsd:element>
2579    <xsd:complexType name="IdentifySubjectResultType">
2580    <xsd:sequence />
2581    </xsd:complexType>
2582    <xsd:complexType name="CandidateListResultType">
2583    <xsd:complexContent>
2584    <xsd:extension base="tns:IdentifySubjectResultType">
2585    <xsd:sequence>
2586    <xsd:element name="CandidateList" type="tns:CandidateListType">
2587    <xsd:annotation>
2588    <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input biometric
2589    sample; returned with successful synchronous request processing.</xsd:documentation>
2590    </xsd:annotation>
2591    </xsd:element>
2592    </xsd:sequence>
2593    </xsd:extension>
2594    </xsd:complexContent>
2595    </xsd:complexType>
2596    <xsd:complexType name="TokenResultType">
2597    <xsd:complexContent>
2598    <xsd:extension base="tns:IdentifySubjectResultType">
2599    <xsd:sequence>
2600    <xsd:element name="Token" type="tns:TokenType">
2601    <xsd:annotation>
2602    <xsd:documentation>A token used to retrieve the results of the IdentifySubject request; returned with
2603    asynchronous request processing.</xsd:documentation>
2604    </xsd:annotation>
2605    </xsd:element>
2606    </xsd:sequence>
2607    </xsd:extension>
```

```xml
2608    </xsd:complexContent>
2609    </xsd:complexType>
2610    <xsd:element name="ListBiographicData">
2611    <xsd:complexType>
2612    <xsd:annotation>
2613    <xsd:documentation>
2614    Lists the biographic data elements stored for a subject.
2615    </xsd:documentation>
2616    </xsd:annotation>
2617    <xsd:sequence>
2618    <xsd:element name="ListBiographicDataRequest" nillable="true">
2619    <xsd:complexType>
2620    <xsd:complexContent>
2621    <xsd:extension base="tns:RequestTemplate">
2622    <xsd:sequence>
2623    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2624    <xsd:annotation>
2625    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2626    encounter.</xsd:documentation>
2627    </xsd:annotation>
2628    </xsd:element>
2629    </xsd:sequence>
2630    </xsd:extension>
2631    </xsd:complexContent>
2632    </xsd:complexType>
2633    </xsd:element>
2634    </xsd:sequence>
2635    </xsd:complexType>
2636    </xsd:element>
2637    <xsd:element name="ListBiographicDataResponse">
2638    <xsd:complexType>
2639    <xsd:annotation>
2640    <xsd:documentation>
2641    The response to a ListBiographicData request, containing a list
2642    of biographic data elements stored for a subject.  In the
2643    encounter-centric model, the biographic data elements for a
2644    specific encounter are returned.  If an encounter ID is not
2645    specified and encounter data exists for the subject, the list
2646    of encounter IDs which contain biographic data is returned.
2647    </xsd:documentation>
2648    </xsd:annotation>
```

```xml
2649    <xsd:sequence>
2650    <xsd:element name="ListBiographicDataResponsePackage" nillable="true">
2651    <xsd:complexType>
2652    <xsd:complexContent>
2653    <xsd:extension base="tns:ResponseTemplate">
2654    <xsd:sequence>
2655    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2656    <xsd:annotation>
2657    <xsd:documentation>
2658    Contains a list of biographic data elements associated with a
2659    subject or encounter; non-empty if the service was
2660    successful, biographic data exists, and either (a) the
2661    person-centric model is being used or (b) the
2662    encounter-centric model is being used and an encounter
2663    identifier was specified.
2664    </xsd:documentation>
2665    </xsd:annotation>
2666    </xsd:element>
2667    <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
2668    <xsd:annotation>
2669    <xsd:documentation>
2670    A list of encounter ID's associated with a subject and
2671    which contain biographic data; non-empty if the service
2672    was successful, biographic data exists, the
2673    encounter-centric model is being used, and an encounter
2674    identifier was not specified.
2675    </xsd:documentation>
2676    </xsd:annotation>
2677    </xsd:element>
2678    </xsd:sequence>
2679    </xsd:extension>
2680    </xsd:complexContent>
2681    </xsd:complexType>
2682    </xsd:element>
2683    </xsd:sequence>
2684    </xsd:complexType>
2685    </xsd:element>
2686    <xsd:element name="ListBiometricData">
2687    <xsd:complexType>
2688    <xsd:annotation>
2689    <xsd:documentation>
```

2690     Lists the biometric data elements stored for a subject. Note

2691     that no actual biometric data is returned by this service (see

2692     the RetrieveBiometricInformation service to obtain the biometric

2693     data).

2694     </xsd:documentation>

2695     </xsd:annotation>

2696     <xsd:sequence>

2697     <xsd:element name="ListBiometricDataRequest" nillable="true">

2698     <xsd:complexType>

2699     <xsd:complexContent>

2700     <xsd:extension base="tns:RequestTemplate">

2701     <xsd:sequence>

2702     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">

2703     <xsd:annotation>

2704     <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2705     encounter.</xsd:documentation>

2706     </xsd:annotation>

2707     </xsd:element>

2708     <xsd:element name="ListFilter" type="tns:ListFilterType" minOccurs="0" maxOccurs="1">

2709     <xsd:annotation>

2710     <xsd:documentation>Indicates what biometric information should be returned.</xsd:documentation>

2711     </xsd:annotation>

2712     </xsd:element>

2713     </xsd:sequence>

2714     </xsd:extension>

2715     </xsd:complexContent>

2716     </xsd:complexType>

2717     </xsd:element>

2718     </xsd:sequence>

2719     </xsd:complexType>

2720     </xsd:element>

2721     <xsd:element name="ListBiometricDataResponse">

2722     <xsd:complexType>

2723     <xsd:annotation>

2724     <xsd:documentation>

2725     The response to a ListBiometricData request, containing a list

2726     of biometric data elements stored for a subject.  In the

2727     encounter-centric model, the biometric data elements for a

2728     specific encounter are returned.  If an encounter ID is not

2729     specified and encounter data exists for the subject, the list

2730     of encounter IDs which contain biometric data is returned.

```xml
2731    </xsd:documentation>
2732    </xsd:annotation>
2733    <xsd:sequence>
2734    <xsd:element name="ListBiometricDataResponsePackage" nillable="true">
2735    <xsd:complexType>
2736    <xsd:complexContent>
2737    <xsd:extension base="tns:ResponseTemplate">
2738    <xsd:sequence>
2739    <xsd:element name="Identity" type="tns:BIASIdentity">
2740    <xsd:annotation>
2741    <xsd:documentation>
2742    Includes a list of biometric data elements associated
2743    with a subject or encounter or a list of encounter ID's
2744    associated with a subject and which contain biometric
2745    data.
2746    </xsd:documentation>
2747    </xsd:annotation>
2748    </xsd:element>
2749    </xsd:sequence>
2750    </xsd:extension>
2751    </xsd:complexContent>
2752    </xsd:complexType>
2753    </xsd:element>
2754    </xsd:sequence>
2755    </xsd:complexType>
2756    </xsd:element>
2757    <xsd:element name="PerformFusion">
2758    <xsd:complexType>
2759    <xsd:annotation>
2760    <xsd:documentation>
2761    Accepts either match score or match decision information and creates a fused match result.
2762    </xsd:documentation>
2763    </xsd:annotation>
2764    <xsd:sequence>
2765    <xsd:element name="PerformFusionRequest" nillable="true">
2766    <xsd:complexType>
2767    <xsd:complexContent>
2768    <xsd:extension base="tns:RequestTemplate">
2769    <xsd:sequence>
2770    <xsd:element name="FusionInput" type="tns:FusionInformationListType" minOccurs="1" maxOccurs="1">
2771    <xsd:annotation>
```

| | |
|---|---|
| 2772 | <xsd:documentation>Score or decision input information to the fusion method.</xsd:documentation> |
| 2773 | </xsd:annotation> |
| 2774 | </xsd:element> |
| 2775 | </xsd:sequence> |
| 2776 | </xsd:extension> |
| 2777 | </xsd:complexContent> |
| 2778 | </xsd:complexType> |
| 2779 | </xsd:element> |
| 2780 | </xsd:sequence> |
| 2781 | </xsd:complexType> |
| 2782 | </xsd:element> |
| 2783 | <xsd:element name="PerformFusionResponse"> |
| 2784 | <xsd:complexType> |
| 2785 | <xsd:annotation> |
| 2786 | <xsd:documentation> |
| 2787 | The response to the PerformFusion request. |
| 2788 | </xsd:documentation> |
| 2789 | </xsd:annotation> |
| 2790 | <xsd:sequence> |
| 2791 | <xsd:element name="PerformFusionResponsePackage" nillable="true"> |
| 2792 | <xsd:complexType> |
| 2793 | <xsd:complexContent> |
| 2794 | <xsd:extension base="tns:ResponseTemplate"> |
| 2795 | <xsd:sequence> |
| 2796 | <xsd:element name="Match" type="tns:MatchType" minOccurs="1" maxOccurs="1"> |
| 2797 | <xsd:annotation> |
| 2798 | <xsd:documentation>Indicates the result of the fusion method</xsd:documentation> |
| 2799 | </xsd:annotation> |
| 2800 | </xsd:element> |
| 2801 | </xsd:sequence> |
| 2802 | </xsd:extension> |
| 2803 | </xsd:complexContent> |
| 2804 | </xsd:complexType> |
| 2805 | </xsd:element> |
| 2806 | </xsd:sequence> |
| 2807 | </xsd:complexType> |
| 2808 | </xsd:element> |
| 2809 | <xsd:element name="QueryCapabilities"> |
| 2810 | <xsd:complexType> |
| 2811 | <xsd:annotation> |
| 2812 | <xsd:documentation> |

| 2813 | Returns a list of the capabilities, options, galleries, etc. |
| 2814 | that are supported by the BIAS implementation. |
| 2815 | </xsd:documentation> |
| 2816 | </xsd:annotation> |
| 2817 | <xsd:sequence> |
| 2818 | <xsd:element name="QueryCapabilitiesRequest" nillable="true"> |
| 2819 | <xsd:complexType> |
| 2820 | <xsd:complexContent> |
| 2821 | <xsd:extension base="tns:RequestTemplate"/> |
| 2822 | </xsd:complexContent> |
| 2823 | </xsd:complexType> |
| 2824 | </xsd:element> |
| 2825 | </xsd:sequence> |
| 2826 | </xsd:complexType> |
| 2827 | </xsd:element> |
| 2828 | <xsd:element name="QueryCapabilitiesResponse"> |
| 2829 | <xsd:complexType> |
| 2830 | <xsd:annotation> |
| 2831 | <xsd:documentation> |
| 2832 | The response to a QueryCapabilities request. |
| 2833 | </xsd:documentation> |
| 2834 | </xsd:annotation> |
| 2835 | <xsd:sequence> |
| 2836 | <xsd:element name="QueryCapabilitiesResponsePackage" nillable="true"> |
| 2837 | <xsd:complexType> |
| 2838 | <xsd:complexContent> |
| 2839 | <xsd:extension base="tns:ResponseTemplate"> |
| 2840 | <xsd:sequence> |
| 2841 | <xsd:element name="CapabilityList" type="tns:CapabilityListType" minOccurs="1" maxOccurs="1"> |
| 2842 | <xsd:annotation> |
| 2843 | <xsd:documentation>A list of capabilities supported by the BIAS implementation.</xsd:documentation> |
| 2844 | </xsd:annotation> |
| 2845 | </xsd:element> |
| 2846 | </xsd:sequence> |
| 2847 | </xsd:extension> |
| 2848 | </xsd:complexContent> |
| 2849 | </xsd:complexType> |
| 2850 | </xsd:element> |
| 2851 | </xsd:sequence> |
| 2852 | </xsd:complexType> |
| 2853 | </xsd:element> |

```
2854    <xsd:element name="RetrieveBiographicInformation">
2855    <xsd:complexType>
2856    <xsd:annotation>
2857    <xsd:documentation>
2858    Retrieves the biographic data associated with a subject ID.
2859    </xsd:documentation>
2860    </xsd:annotation>
2861    <xsd:sequence>
2862    <xsd:element name="RetrieveBiographicInformationRequest" nillable="true">
2863    <xsd:complexType>
2864    <xsd:complexContent>
2865    <xsd:extension base="tns:RequestTemplate">
2866    <xsd:sequence>
2867    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2868    <xsd:annotation>
2869    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2870    encounter.</xsd:documentation>
2871    </xsd:annotation>
2872    </xsd:element>
2873    </xsd:sequence>
2874    </xsd:extension>
2875    </xsd:complexContent>
2876    </xsd:complexType>
2877    </xsd:element>
2878    </xsd:sequence>
2879    </xsd:complexType>
2880    </xsd:element>
2881    <xsd:element name="RetrieveBiographicInformationResponse">
2882    <xsd:complexType>
2883    <xsd:annotation>
2884    <xsd:documentation>
2885    The response to a RetrieveBiographicInformation request,
2886    containing the biographic data associated with a subject ID. In
2887    the encounter-centric model, the biographic data associated with
2888    a specified encounter is returned. If the encounter ID is not
2889    specified in the encounter-centric model, the biographic
2890    information associated with the most recent encounter is returned.
2891    </xsd:documentation>
2892    </xsd:annotation>
2893    <xsd:sequence>
2894    <xsd:element name="RetrieveBiographicInformationResponsePackage" nillable="true">
```

```xml
2895    <xsd:complexType>
2896    <xsd:complexContent>
2897    <xsd:extension base="tns:ResponseTemplate">
2898    <xsd:sequence>
2899    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2900    <xsd:annotation>
2901    <xsd:documentation>Includes the set of biographic data associated with a subject.</xsd:documentation>
2902    </xsd:annotation>
2903    </xsd:element>
2904    </xsd:sequence>
2905    </xsd:extension>
2906    </xsd:complexContent>
2907    </xsd:complexType>
2908    </xsd:element>
2909    </xsd:sequence>
2910    </xsd:complexType>
2911    </xsd:element>
2912    <xsd:element name="RetrieveBiometricInformation">
2913    <xsd:complexType>
2914    <xsd:annotation>
2915    <xsd:documentation>
2916    Retrieves the biometric data associated with a subject ID.
2917    </xsd:documentation>
2918    </xsd:annotation>
2919    <xsd:sequence>
2920    <xsd:element name="RetrieveBiometricInformationRequest" nillable="true">
2921    <xsd:complexType>
2922    <xsd:complexContent>
2923    <xsd:extension base="tns:RequestTemplate">
2924    <xsd:sequence>
2925    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2926    <xsd:annotation>
2927    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2928    encounter.</xsd:documentation>
2929    </xsd:annotation>
2930    </xsd:element>
2931    <xsd:element name="BiometricType" type="oasis_cbeff:MultipleTypesType" minOccurs="0" maxOccurs="1">
2932    <xsd:annotation>
2933    <xsd:documentation>The type of biological or behavioral data to retrieve.</xsd:documentation>
2934    </xsd:annotation>
2935    </xsd:element>
```

```
2936    </xsd:sequence>
2937    </xsd:extension>
2938    </xsd:complexContent>
2939    </xsd:complexType>
2940    </xsd:element>
2941    </xsd:sequence>
2942    </xsd:complexType>
2943    </xsd:element>
2944    <xsd:element name="RetrieveBiometricInformationResponse">
2945    <xsd:complexType>
2946    <xsd:annotation>
2947    <xsd:documentation>
2948    The response to a RetrieveBiometricInformation request,
2949    containing the biometric data associated with a subject ID. In
2950    the encounter-centric model, the biometric data associated with
2951    a specified encounter is returned. If the encounter ID is not
2952    specified in the encounter-centric model, the biometric
2953    information associated with the most recent encounter is returned.
2954    </xsd:documentation>
2955    </xsd:annotation>
2956    <xsd:sequence>
2957    <xsd:element name="RetrieveBiometricInformationResponsePackage" nillable="true">
2958    <xsd:complexType>
2959    <xsd:complexContent>
2960    <xsd:extension base="tns:ResponseTemplate">
2961    <xsd:sequence>
2962    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2963    <xsd:annotation>
2964    <xsd:documentation>Includes the biometric data associated with a subject.</xsd:documentation>
2965    </xsd:annotation>
2966    </xsd:element>
2967    </xsd:sequence>
2968    </xsd:extension>
2969    </xsd:complexContent>
2970    </xsd:complexType>
2971    </xsd:element>
2972    </xsd:sequence>
2973    </xsd:complexType>
2974    </xsd:element>
2975    <xsd:element name="SetBiographicData">
2976    <xsd:complexType>
```

```
2977    <xsd:annotation>
2978    <xsd:documentation>
2979    Associates biographic data to a given subject record.
2980    </xsd:documentation>
2981    </xsd:annotation>
2982    <xsd:sequence>
2983    <xsd:element name="SetBiographicDataRequest" nillable="true">
2984    <xsd:complexType>
2985    <xsd:complexContent>
2986    <xsd:extension base="tns:RequestTemplate">
2987    <xsd:sequence>
2988    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2989    <xsd:annotation>
2990    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
2991    includes the biographic data to store.</xsd:documentation>
2992    </xsd:annotation>
2993    </xsd:element>
2994    </xsd:sequence>
2995    </xsd:extension>
2996    </xsd:complexContent>
2997    </xsd:complexType>
2998    </xsd:element>
2999    </xsd:sequence>
3000    </xsd:complexType>
3001    </xsd:element>
3002    <xsd:element name="SetBiographicDataResponse">
3003    <xsd:complexType>
3004    <xsd:annotation>
3005    <xsd:documentation>
3006    The response to a SetBiographicData request.
3007    </xsd:documentation>
3008    </xsd:annotation>
3009    <xsd:sequence>
3010    <xsd:element name="SetBiographicDataResponsePackage" nillable="true">
3011    <xsd:complexType>
3012    <xsd:complexContent>
3013    <xsd:extension base="tns:ResponseTemplate">
3014    <xsd:sequence>
3015    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
3016    <xsd:annotation>
```

```
3017    <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
3018    encounter.</xsd:documentation>
3019    </xsd:annotation>
3020    </xsd:element>
3021    </xsd:sequence>
3022    </xsd:extension>
3023    </xsd:complexContent>
3024    </xsd:complexType>
3025    </xsd:element>
3026    </xsd:sequence>
3027    </xsd:complexType>
3028    </xsd:element>
3029    <xsd:element name="SetBiometricData">
3030    <xsd:complexType>
3031    <xsd:annotation>
3032    <xsd:documentation>
3033    Associates biometric data to a given subject record.
3034    </xsd:documentation>
3035    </xsd:annotation>
3036    <xsd:sequence>
3037    <xsd:element name="SetBiometricDataRequest" nillable="true">
3038    <xsd:complexType>
3039    <xsd:complexContent>
3040    <xsd:extension base="tns:RequestTemplate">
3041    <xsd:sequence>
3042    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3043    <xsd:annotation>
3044    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3045    includes the biometric data to store.</xsd:documentation>
3046    </xsd:annotation>
3047    </xsd:element>
3048    </xsd:sequence>
3049    </xsd:extension>
3050    </xsd:complexContent>
3051    </xsd:complexType>
3052    </xsd:element>
3053    </xsd:sequence>
3054    </xsd:complexType>
3055    </xsd:element>
3056    <xsd:element name="SetBiometricDataResponse">
3057    <xsd:complexType>
```

```xml
3058    <xsd:annotation>
3059    <xsd:documentation>
3060    The response to a SetBiometricData request.
3061    </xsd:documentation>
3062    </xsd:annotation>
3063    <xsd:sequence>
3064    <xsd:element name="SetBiometricDataResponsePackage" nillable="true">
3065    <xsd:complexType>
3066    <xsd:complexContent>
3067    <xsd:extension base="tns:ResponseTemplate">
3068    <xsd:sequence>
3069    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
3070    <xsd:annotation>
3071    <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
3072    encounter.</xsd:documentation>
3073    </xsd:annotation>
3074    </xsd:element>
3075    </xsd:sequence>
3076    </xsd:extension>
3077    </xsd:complexContent>
3078    </xsd:complexType>
3079    </xsd:element>
3080    </xsd:sequence>
3081    </xsd:complexType>
3082    </xsd:element>
3083    <xsd:element name="TransformBiometricData">
3084    <xsd:complexType>
3085    <xsd:annotation>
3086    <xsd:documentation>
3087    Transforms or processes a given biometric in one format into a new target format.
3088    </xsd:documentation>
3089    </xsd:annotation>
3090    <xsd:sequence>
3091    <xsd:element name="TransformBiometricDataRequest" nillable="true">
3092    <xsd:complexType>
3093    <xsd:complexContent>
3094    <xsd:extension base="tns:RequestTemplate">
3095    <xsd:sequence>
3096    <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="1" maxOccurs="1">
3097    <xsd:annotation>
```

```
3098    <xsd:documentation>Data structure containing the biometric information to be
3099    transformed.</xsd:documentation>
3100    </xsd:annotation>
3101    </xsd:element>
3102    <xsd:element name="TransformOperation" type="xsd:unsignedLong" minOccurs="1" maxOccurs="1">
3103    <xsd:annotation>
3104    <xsd:documentation>Value indicating the type of transformation to perform.</xsd:documentation>
3105    </xsd:annotation>
3106    </xsd:element>
3107    <xsd:element name="TransformControl" type="xsd:string" minOccurs="0" maxOccurs="1">
3108    <xsd:annotation>
3109    <xsd:documentation> Specifies controls for the requested transform operation.</xsd:documentation>
3110    </xsd:annotation>
3111    </xsd:element>
3112    </xsd:sequence>
3113    </xsd:extension>
3114    </xsd:complexContent>
3115    </xsd:complexType>
3116    </xsd:element>
3117    </xsd:sequence>
3118    </xsd:complexType>
3119    </xsd:element>
3120    <xsd:element name="TransformBiometricDataResponse">
3121    <xsd:complexType>
3122    <xsd:annotation>
3123    <xsd:documentation>
3124    The response to a TransformBiometricData request.
3125    </xsd:documentation>
3126    </xsd:annotation>
3127    <xsd:sequence>
3128    <xsd:element name="TransformBiometricDataResponsePackage" nillable="true">
3129    <xsd:complexType>
3130    <xsd:complexContent>
3131    <xsd:extension base="tns:ResponseTemplate">
3132    <xsd:sequence>
3133    <xsd:element name="OutputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
3134    <xsd:annotation>
3135    <xsd:documentation>Data structure containing the new, transformed biometric
3136    information.</xsd:documentation>
3137    </xsd:annotation>
3138    </xsd:element>
```

```
3139    </xsd:sequence>
3140    </xsd:extension>
3141    </xsd:complexContent>
3142    </xsd:complexType>
3143    </xsd:element>
3144    </xsd:sequence>
3145    </xsd:complexType>
3146    </xsd:element>
3147    <xsd:element name="UpdateBiographicData">
3148    <xsd:complexType>
3149    <xsd:annotation>
3150    <xsd:documentation>
3151    Updates the biographic data for a given subject record.
3152    </xsd:documentation>
3153    </xsd:annotation>
3154    <xsd:sequence>
3155    <xsd:element name="UpdateBiographicDataRequest" nillable="true">
3156    <xsd:complexType>
3157    <xsd:complexContent>
3158    <xsd:extension base="tns:RequestTemplate">
3159    <xsd:sequence>
3160    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3161    <xsd:annotation>
3162    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3163    includes the biographic data to update.</xsd:documentation>
3164    </xsd:annotation>
3165    </xsd:element>
3166    </xsd:sequence>
3167    </xsd:extension>
3168    </xsd:complexContent>
3169    </xsd:complexType>
3170    </xsd:element>
3171    </xsd:sequence>
3172    </xsd:complexType>
3173    </xsd:element>
3174    <xsd:element name="UpdateBiographicDataResponse">
3175    <xsd:complexType>
3176    <xsd:annotation>
3177    <xsd:documentation>
3178    The response to an UpdateBiographicData request.
3179    </xsd:documentation>
```

```
3180    </xsd:annotation>
3181    <xsd:sequence>
3182    <xsd:element name="UpdateBiographicDataResponsePackage" nillable="true">
3183    <xsd:complexType>
3184    <xsd:complexContent>
3185    <xsd:extension base="tns:ResponseTemplate"/>
3186    </xsd:complexContent>
3187    </xsd:complexType>
3188    </xsd:element>
3189    </xsd:sequence>
3190    </xsd:complexType>
3191    </xsd:element>
3192    <xsd:element name="UpdateBiometricData">
3193    <xsd:complexType>
3194    <xsd:annotation>
3195    <xsd:documentation>
3196    Updates a single biometric sample for a given subject record.
3197    </xsd:documentation>
3198    </xsd:annotation>
3199    <xsd:sequence>
3200    <xsd:element name="UpdateBiometricDataRequest" nillable="true">
3201    <xsd:complexType>
3202    <xsd:complexContent>
3203    <xsd:extension base="tns:RequestTemplate">
3204    <xsd:sequence>
3205    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3206    <xsd:annotation>
3207    <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter, and
3208    includes the biometric data to update.</xsd:documentation>
3209    </xsd:annotation>
3210    </xsd:element>
3211    <xsd:element name="Merge" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3212    <xsd:annotation>
3213    <xsd:documentation>Value indicating if the input biometric sample should be merged with any existing biometric
3214    information.</xsd:documentation>
3215    </xsd:annotation>
3216    </xsd:element>
3217    </xsd:sequence>
3218    </xsd:extension>
3219    </xsd:complexContent>
3220    </xsd:complexType>
```

```
3221   </xsd:element>
3222   </xsd:sequence>
3223   </xsd:complexType>
3224   </xsd:element>
3225   <xsd:element name="UpdateBiometricDataResponse">
3226   <xsd:complexType>
3227   <xsd:annotation>
3228   <xsd:documentation>
3229   The response to an UpdateBiometricData request.
3230   </xsd:documentation>
3231   </xsd:annotation>
3232   <xsd:sequence>
3233   <xsd:element name="UpdateBiometricDataResponsePackage" nillable="true">
3234   <xsd:complexType>
3235   <xsd:complexContent>
3236   <xsd:extension base="tns:ResponseTemplate"/>
3237   </xsd:complexContent>
3238   </xsd:complexType>
3239   </xsd:element>
3240   </xsd:sequence>
3241   </xsd:complexType>
3242   </xsd:element>
3243   <xsd:element name="VerifySubject">
3244   <xsd:complexType>
3245   <xsd:annotation>
3246   <xsd:documentation>
3247   Performs a 1:1 verification match between a given biometric and
3248   either a claim to identity in a given gallery or another given
3249   biometric.
3250   </xsd:documentation>
3251   </xsd:annotation>
3252   <xsd:sequence>
3253   <xsd:element name="VerifySubjectRequest" nillable="true">
3254   <xsd:complexType>
3255   <xsd:complexContent>
3256   <xsd:extension base="tns:RequestTemplate">
3257   <xsd:sequence>
3258   <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
3259   <xsd:annotation>
3260   <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
3261   member.</xsd:documentation>
```

3262    </xsd:annotation>

3263    </xsd:element>

3264    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">

3265    <xsd:annotation>

3266    <xsd:documentation>Includes the identifying information and/or input and reference biometric
3267    samples.</xsd:documentation>

3268    </xsd:annotation>

3269    </xsd:element>

3270    </xsd:sequence>

3271    </xsd:extension>

3272    </xsd:complexContent>

3273    </xsd:complexType>

3274    </xsd:element>

3275    </xsd:sequence>

3276    </xsd:complexType>

3277    </xsd:element>

3278    <xsd:element name="VerifySubjectResponse">

3279    <xsd:complexType>

3280    <xsd:annotation>

3281    <xsd:documentation>

3282    The response to a VerifySubject request.

3283    </xsd:documentation>

3284    </xsd:annotation>

3285    <xsd:sequence>

3286    <xsd:element name="VerifySubjectResponsePackage" nillable="true">

3287    <xsd:complexType>

3288    <xsd:complexContent>

3289    <xsd:extension base="tns:ResponseTemplate">

3290    <xsd:sequence>

3291    <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">

3292    <xsd:annotation>

3293    <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with the
3294    Identity Claim or the Reference BIR.</xsd:documentation>

3295    </xsd:annotation>

3296    </xsd:element>

3297    <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">

3298    <xsd:annotation>

3299    <xsd:documentation>The score if the biometric information matched.</xsd:documentation>

3300    </xsd:annotation>

3301    </xsd:element>

3302    </xsd:sequence>

```
3303    </xsd:extension>
3304    </xsd:complexContent>
3305    </xsd:complexType>
3306    </xsd:element>
3307    </xsd:sequence>
3308    </xsd:complexType>
3309    </xsd:element>
3310    <xsd:element name="Enroll">
3311    <xsd:complexType>
3312    <xsd:annotation>
3313    <xsd:documentation>
3314    The Enroll aggregate service adds a new subject or, in an
3315    encounter-centric model, a new encounter to the system. This may
3316    be accomplished in a number of different ways according to
3317    system requirements and/or resources. If the Enroll aggregate
3318    service is implemented as a synchronous service, the
3319    implementing system immediately processes the request and
3320    returns the results in the ReturnData parameter. If the Enroll
3321    aggregate service is implemented as an asynchronous service, the
3322    implementing system returns a token in the ReturnData
3323    parameter, which is an indication that the request is being
3324    handled asynchronously. In this case, the GetEnrollResults
3325    service is used to poll for the results of the Enroll request.
3326    </xsd:documentation>
3327    </xsd:annotation>
3328    <xsd:sequence>
3329    <xsd:element name="EnrollRequest" nillable="true">
3330    <xsd:complexType>
3331    <xsd:complexContent>
3332    <xsd:extension base="tns:AggregateRequestTemplate"/>
3333    </xsd:complexContent>
3334    </xsd:complexType>
3335    </xsd:element>
3336    </xsd:sequence>
3337    </xsd:complexType>
3338    </xsd:element>
3339    <xsd:element name="EnrollResponse">
3340    <xsd:complexType>
3341    <xsd:annotation>
3342    <xsd:documentation>The response to an Enroll request.</xsd:documentation>
3343    </xsd:annotation>
```

```
3344    <xsd:sequence>
3345    <xsd:element name="EnrollResponsePackage" nillable="true">
3346    <xsd:complexType>
3347    <xsd:complexContent>
3348    <xsd:extension base="tns:AggregateResponseTemplate"/>
3349    </xsd:complexContent>
3350    </xsd:complexType>
3351    </xsd:element>
3352    </xsd:sequence>
3353    </xsd:complexType>
3354    </xsd:element>
3355    <xsd:element name="GetEnrollResults">
3356    <xsd:complexType>
3357    <xsd:annotation>
3358    <xsd:documentation>
3359    The GetEnrollResults aggregate service retrieves the enrollment
3360    results for the specified token. This service is used in
3361    conjunction with the Enroll aggregate service. If the Enroll
3362    aggregate service is implemented as an asynchronous service, the
3363    implementing system returns a token, and the GetEnrollResults
3364    service is used to poll for the results of the original Enroll
3365    request.
3366    </xsd:documentation>
3367    </xsd:annotation>
3368    <xsd:sequence>
3369    <xsd:element name="GetEnrollResultsRequest" nillable="true">
3370    <xsd:complexType>
3371    <xsd:complexContent>
3372    <xsd:extension base="tns:RequestTemplate">
3373    <xsd:sequence>
3374    <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3375    <xsd:annotation>
3376    <xsd:documentation>A value used to retrieve the results of the Enroll request.</xsd:documentation>
3377    </xsd:annotation>
3378    </xsd:element>
3379    </xsd:sequence>
3380    </xsd:extension>
3381    </xsd:complexContent>
3382    </xsd:complexType>
3383    </xsd:element>
3384    </xsd:sequence>
```

| 3385 | </xsd:complexType> |
| 3386 | </xsd:element> |
| 3387 | <xsd:element name="GetEnrollResultsResponse"> |
| 3388 | <xsd:complexType> |
| 3389 | <xsd:annotation> |
| 3390 | <xsd:documentation>The response to a GetEnrollResults request.</xsd:documentation> |
| 3391 | </xsd:annotation> |
| 3392 | <xsd:sequence> |
| 3393 | <xsd:element name="GetEnrollResultsResponsePackage" nillable="true"> |
| 3394 | <xsd:complexType> |
| 3395 | <xsd:complexContent> |
| 3396 | <xsd:extension base="tns:AggregateResponseTemplate"/> |
| 3397 | </xsd:complexContent> |
| 3398 | </xsd:complexType> |
| 3399 | </xsd:element> |
| 3400 | </xsd:sequence> |
| 3401 | </xsd:complexType> |
| 3402 | </xsd:element> |
| 3403 | <xsd:element name="GetIdentifyResults"> |
| 3404 | <xsd:complexType> |
| 3405 | <xsd:annotation> |
| 3406 | <xsd:documentation> |
| 3407 | The GetIdentifyResults aggregate service retrieves the |
| 3408 | identification results for the specified token. This service is |
| 3409 | used in conjunction with the Identify aggregate service. If the |
| 3410 | Identify aggregate service is implemented as an asynchronous |
| 3411 | service, the implementing system returns a token, and the |
| 3412 | GetIdentifyResults service is used to poll for the results of |
| 3413 | the original Identify request. |
| 3414 | </xsd:documentation> |
| 3415 | </xsd:annotation> |
| 3416 | <xsd:sequence> |
| 3417 | <xsd:element name="GetIdentifyResultsRequest" nillable="true"> |
| 3418 | <xsd:complexType> |
| 3419 | <xsd:complexContent> |
| 3420 | <xsd:extension base="tns:RequestTemplate"> |
| 3421 | <xsd:sequence> |
| 3422 | <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1"> |
| 3423 | <xsd:annotation> |
| 3424 | <xsd:documentation>A value used to retrieve the results of the Identify request.</xsd:documentation> |
| 3425 | </xsd:annotation> |

```
3426    </xsd:element>
3427    </xsd:sequence>
3428    </xsd:extension>
3429    </xsd:complexContent>
3430    </xsd:complexType>
3431    </xsd:element>
3432    </xsd:sequence>
3433    </xsd:complexType>
3434    </xsd:element>
3435    <xsd:element name="GetIdentifyResultsResponse">
3436    <xsd:complexType>
3437    <xsd:annotation>
3438    <xsd:documentation>The response to a GetIdentifyResults request.</xsd:documentation>
3439    </xsd:annotation>
3440    <xsd:sequence>
3441    <xsd:element name="GetIdentifyResultsResponsePackage" nillable="true">
3442    <xsd:complexType>
3443    <xsd:complexContent>
3444    <xsd:extension base="tns:AggregateResponseTemplate"/>
3445    </xsd:complexContent>
3446    </xsd:complexType>
3447    </xsd:element>
3448    </xsd:sequence>
3449    </xsd:complexType>
3450    </xsd:element>
3451    <xsd:element name="GetVerifyResults">
3452    <xsd:complexType>
3453    <xsd:annotation>
3454    <xsd:documentation>
3455    The GetVerifyResults aggregate service retrieves the verification
3456    results for the specified token. This service is used in
3457    conjunction with the Verify aggregate service. If the Verify
3458    aggregate service is implemented as an asynchronous service, the
3459    implementing system returns a token, and the GetVerifyResults
3460    service is used to poll for the results of the original Verify
3461    request.
3462    </xsd:documentation>
3463    </xsd:annotation>
3464    <xsd:sequence>
3465    <xsd:element name="GetVerifyResultsRequest" nillable="true">
3466    <xsd:complexType>
```

```
3467    <xsd:complexContent>
3468    <xsd:extension base="tns:RequestTemplate">
3469    <xsd:sequence>
3470    <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
3471    <xsd:annotation>
3472    <xsd:documentation>A value used to retrieve the results of the Verify request.</xsd:documentation>
3473    </xsd:annotation>
3474    </xsd:element>
3475    </xsd:sequence>
3476    </xsd:extension>
3477    </xsd:complexContent>
3478    </xsd:complexType>
3479    </xsd:element>
3480    </xsd:sequence>
3481    </xsd:complexType>
3482    </xsd:element>
3483    <xsd:element name="GetVerifyResultsResponse">
3484    <xsd:complexType>
3485    <xsd:annotation>
3486    <xsd:documentation>The response to a GetVerifyResults request.</xsd:documentation>
3487    </xsd:annotation>
3488    <xsd:sequence>
3489    <xsd:element name="GetVerifyResultsResponsePackage" nillable="true">
3490    <xsd:complexType>
3491    <xsd:complexContent>
3492    <xsd:extension base="tns:AggregateResponseTemplate">
3493    <xsd:sequence>
3494    <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3495    <xsd:annotation>
3496    <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with the
3497    Identity Claim or the Reference BIR.</xsd:documentation>
3498    </xsd:annotation>
3499    </xsd:element>
3500    <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
3501    <xsd:annotation>
3502    <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3503    </xsd:annotation>
3504    </xsd:element>
3505    </xsd:sequence>
3506    </xsd:extension>
3507    </xsd:complexContent>
```

```
3508    </xsd:complexType>
3509    </xsd:element>
3510    </xsd:sequence>
3511    </xsd:complexType>
3512    </xsd:element>
3513    <xsd:element name="Identify">
3514    <xsd:complexType>
3515    <xsd:annotation>
3516    <xsd:documentation>
3517    The Identify aggregate service performs an identification
3518    function according to system requirements and/or resources. If
3519    the Identify aggregate service is implemented as a synchronous
3520    service, the implementing system immediately processes the
3521    request and returns the results in the ReturnData parameter. If
3522    the Identify aggregate service is implemented as an asynchronous
3523    service, the implementing system returns a token in the
3524    ReturnData parameter, which is an indication that the request is
3525    being handled asynchronously. In this case, the
3526    GetIdentifyResults service is used to poll for the results of
3527    the Identify request.
3528    </xsd:documentation>
3529    </xsd:annotation>
3530    <xsd:sequence>
3531    <xsd:element name="IdentifyRequest" nillable="true">
3532    <xsd:complexType>
3533    <xsd:complexContent>
3534    <xsd:extension base="tns:AggregateRequestTemplate"/>
3535    </xsd:complexContent>
3536    </xsd:complexType>
3537    </xsd:element>
3538    </xsd:sequence>
3539    </xsd:complexType>
3540    </xsd:element>
3541    <xsd:element name="IdentifyResponse">
3542    <xsd:complexType>
3543    <xsd:annotation>
3544    <xsd:documentation>The response to an Identify request.</xsd:documentation>
3545    </xsd:annotation>
3546    <xsd:sequence>
3547    <xsd:element name="IdentifyResponsePackage" nillable="true">
3548    <xsd:complexType>
```

```
3549    <xsd:complexContent>

3550    <xsd:extension base="tns:AggregateResponseTemplate"/>

3551    </xsd:complexContent>

3552    </xsd:complexType>

3553    </xsd:element>

3554    </xsd:sequence>

3555    </xsd:complexType>

3556    </xsd:element>

3557    <xsd:element name="RetrieveInformation">

3558    <xsd:complexType>

3559    <xsd:annotation>

3560    <xsd:documentation>

3561    The RetrieveInformation aggregate service retrieves requested

3562    information about a subject, or in an encounter-centric model

3563    about an encounter. In a person-centric model, this aggregate

3564    service may be used to retrieve both biographic and biometric

3565    information for a subject record. In an encounter-centric model,

3566    this aggregate service may be used to retrieve biographic and/or

3567    biometric information for either a single encounter or all

3568    encounters. Either a SubjectID or EncounterID must be specified

3569    in the Identify parameter.

3570    </xsd:documentation>

3571    </xsd:annotation>

3572    <xsd:sequence>

3573    <xsd:element name="RetrieveInformationRequest" nillable="true">

3574    <xsd:complexType>

3575    <xsd:complexContent>

3576    <xsd:extension base="tns:RequestTemplate">

3577    <xsd:sequence>

3578    <xsd:element name="ProcessingOptions" type="tns:ProcessingOptionsType" minOccurs="1" maxOccurs="1">

3579    <xsd:annotation>

3580    <xsd:documentation>Options that guide how the service request is processed, and may identify what type(s) of
3581    information should be returned.</xsd:documentation>

3582    </xsd:annotation>

3583    </xsd:element>

3584    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">

3585    <xsd:annotation>

3586    <xsd:documentation>Includes the identifier of the subject or encounter.</xsd:documentation>

3587    </xsd:annotation>

3588    </xsd:element>

3589    </xsd:sequence>
```

```
3590    </xsd:extension>
3591    </xsd:complexContent>
3592    </xsd:complexType>
3593    </xsd:element>
3594    </xsd:sequence>
3595    </xsd:complexType>
3596    </xsd:element>
3597    <xsd:element name="RetrieveInformationResponse">
3598    <xsd:complexType>
3599    <xsd:annotation>
3600    <xsd:documentation>The response to a RetrieveInformation request.</xsd:documentation>
3601    </xsd:annotation>
3602    <xsd:sequence>
3603    <xsd:element name="RetrieveInformationResponsePackage" nillable="true">
3604    <xsd:complexType>
3605    <xsd:complexContent>
3606    <xsd:extension base="tns:AggregateResponseTemplate"/>
3607    </xsd:complexContent>
3608    </xsd:complexType>
3609    </xsd:element>
3610    </xsd:sequence>
3611    </xsd:complexType>
3612    </xsd:element>
3613    <xsd:element name="Verify">
3614    <xsd:complexType>
3615    <xsd:annotation>
3616    <xsd:documentation>
3617    The Verify aggregate service performs a 1:1 verification
3618    function according to system requirements and/or resources.
3619    Either the IdentityClaim or ReferenceBIR input data elements in
3620    the Identity parameter are required. If the Verify aggregate
3621    service is implemented as a synchronous service, the
3622    implementing system immediately processes the request and returns
3623    the results in the ReturnData parameter. If the Verify aggregate
3624    service is implemented as an asynchronous service, the
3625    implementing system returns a token in the ReturnData parameter,
3626    which is an indication that the request is being handled
3627    asynchronously. In this case, the GetVerifyResults service is
3628    used to poll for the results of the Verify request.
3629    </xsd:documentation>
3630    </xsd:annotation>
```

```
3631    <xsd:sequence>
3632    <xsd:element name="VerifyRequest" nillable="true">
3633    <xsd:complexType>
3634    <xsd:complexContent>
3635    <xsd:extension base="tns:AggregateRequestTemplate">
3636    <xsd:sequence>
3637    <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3638    <xsd:annotation>
3639    <xsd:documentation>Includes either the IdentityClaim or ReferenceBIR.</xsd:documentation>
3640    </xsd:annotation>
3641    </xsd:element>
3642    <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
3643    <xsd:annotation>
3644    <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
3645    member.</xsd:documentation>
3646    </xsd:annotation>
3647    </xsd:element>
3648    </xsd:sequence>
3649    </xsd:extension>
3650    </xsd:complexContent>
3651    </xsd:complexType>
3652    </xsd:element>
3653    </xsd:sequence>
3654    </xsd:complexType>
3655    </xsd:element>
3656    <xsd:element name="VerifyResponse">
3657    <xsd:complexType>
3658    <xsd:annotation>
3659    <xsd:documentation>The response to a Verify request.</xsd:documentation>
3660    </xsd:annotation>
3661    <xsd:sequence>
3662    <xsd:element name="VerifyResponsePackage" nillable="true">
3663    <xsd:complexType>
3664    <xsd:complexContent>
3665    <xsd:extension base="tns:AggregateResponseTemplate">
3666    <xsd:sequence>
3667    <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3668    <xsd:annotation>
3669    <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with the
3670    Identity Claim or the Reference BIR.</xsd:documentation>
3671    </xsd:annotation>
```

3672        </xsd:element>

3673        <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">

3674        <xsd:annotation>

3675        <xsd:documentation>The score if the biometric information matched.</xsd:documentation>

3676        </xsd:annotation>

3677        </xsd:element>

3678        </xsd:sequence>

3679        </xsd:extension>

3680        </xsd:complexContent>

3681        </xsd:complexType>

3682        </xsd:element>

3683        </xsd:sequence>

3684        </xsd:complexType>

3685        </xsd:element>

3686        <xsd:element name="BIASFault" type="tns:BIASFaultDetail"/>

3687        </xsd:schema>

3688        </wsdl:types>

3689        <wsdl:message name="BIASFaultMessage">

3690        <wsdl:part name="parameters" element="tns:BIASFault"/>

3691        </wsdl:message>

3692        <wsdl:message name="AddSubjectToGalleryRequestMessage">

3693        <wsdl:part name="parameters" element="tns:AddSubjectToGallery"/>

3694        </wsdl:message>

3695        <wsdl:message name="AddSubjectToGalleryResponseMessage">

3696        <wsdl:part name="parameters" element="tns:AddSubjectToGalleryResponse"/>

3697        </wsdl:message>

3698        <wsdl:message name="CheckQualityRequestMessage">

3699        <wsdl:part name="parameters" element="tns:CheckQuality"/>

3700        </wsdl:message>

3701        <wsdl:message name="CheckQualityResponseMessage">

3702        <wsdl:part name="parameters" element="tns:CheckQualityResponse"/>

3703        </wsdl:message>

3704        <wsdl:message name="ClassifyBiometricDataRequestMessage">

3705        <wsdl:part name="parameters" element="tns:ClassifyBiometricData"/>

3706        </wsdl:message>

3707        <wsdl:message name="ClassifyBiometricDataResponseMessage">

3708        <wsdl:part name="parameters" element="tns:ClassifyBiometricDataResponse"/>

3709        </wsdl:message>

3710        <wsdl:message name="CreateSubjectRequestMessage">

3711        <wsdl:part name="parameters" element="tns:CreateSubject"/>

3712        </wsdl:message>

```
3713    <wsdl:message name="CreateSubjectResponseMessage">
3714    <wsdl:part name="parameters" element="tns:CreateSubjectResponse"/>
3715    </wsdl:message>
3716    <wsdl:message name="DeleteBiographicDataRequestMessage">
3717    <wsdl:part name="parameters" element="tns:DeleteBiographicData"/>
3718    </wsdl:message>
3719    <wsdl:message name="DeleteBiographicDataResponseMessage">
3720    <wsdl:part name="parameters" element="tns:DeleteBiographicDataResponse"/>
3721    </wsdl:message>
3722    <wsdl:message name="DeleteBiometricDataRequestMessage">
3723    <wsdl:part name="parameters" element="tns:DeleteBiometricData"/>
3724    </wsdl:message>
3725    <wsdl:message name="DeleteBiometricDataResponseMessage">
3726    <wsdl:part name="parameters" element="tns:DeleteBiometricDataResponse"/>
3727    </wsdl:message>
3728    <wsdl:message name="DeleteSubjectRequestMessage">
3729    <wsdl:part name="parameters" element="tns:DeleteSubject"/>
3730    </wsdl:message>
3731    <wsdl:message name="DeleteSubjectResponseMessage">
3732    <wsdl:part name="parameters" element="tns:DeleteSubjectResponse"/>
3733    </wsdl:message>
3734    <wsdl:message name="DeleteSubjectFromGalleryRequestMessage">
3735    <wsdl:part name="parameters" element="tns:DeleteSubjectFromGallery"/>
3736    </wsdl:message>
3737    <wsdl:message name="DeleteSubjectFromGalleryResponseMessage">
3738    <wsdl:part name="parameters" element="tns:DeleteSubjectFromGalleryResponse"/>
3739    </wsdl:message>
3740    <wsdl:message name="GetIdentifySubjectResultsRequestMessage">
3741    <wsdl:part name="parameters" element="tns:GetIdentifySubjectResults"/>
3742    </wsdl:message>
3743    <wsdl:message name="GetIdentifySubjectResultsResponseMessage">
3744    <wsdl:part name="parameters" element="tns:GetIdentifySubjectResultsResponse"/>
3745    </wsdl:message>
3746    <wsdl:message name="IdentifySubjectRequestMessage">
3747    <wsdl:part name="parameters" element="tns:IdentifySubject"/>
3748    </wsdl:message>
3749    <wsdl:message name="IdentifySubjectResponseMessage">
3750    <wsdl:part name="parameters" element="tns:IdentifySubjectResponse"/>
3751    </wsdl:message>
3752    <wsdl:message name="ListBiographicDataRequestMessage">
3753    <wsdl:part name="parameters" element="tns:ListBiographicData"/>
```

```
3754    </wsdl:message>
3755    <wsdl:message name="ListBiographicDataResponseMessage">
3756    <wsdl:part name="parameters" element="tns:ListBiographicDataResponse"/>
3757    </wsdl:message>
3758    <wsdl:message name="ListBiometricDataRequestMessage">
3759    <wsdl:part name="parameters" element="tns:ListBiometricData"/>
3760    </wsdl:message>
3761    <wsdl:message name="ListBiometricDataResponseMessage">
3762    <wsdl:part name="parameters" element="tns:ListBiometricDataResponse"/>
3763    </wsdl:message>
3764    <wsdl:message name="PerformFusionRequestMessage">
3765    <wsdl:part name="parameters" element="tns:PerformFusion"/>
3766    </wsdl:message>
3767    <wsdl:message name="PerformFusionResponseMessage">
3768    <wsdl:part name="parameters" element="tns:PerformFusionResponse"/>
3769    </wsdl:message>
3770    <wsdl:message name="QueryCapabilitiesRequestMessage">
3771    <wsdl:part name="parameters" element="tns:QueryCapabilities"/>
3772    </wsdl:message>
3773    <wsdl:message name="QueryCapabilitiesResponseMessage">
3774    <wsdl:part name="parameters" element="tns:QueryCapabilitiesResponse"/>
3775    </wsdl:message>
3776    <wsdl:message name="RetrieveBiographicInformationRequestMessage">
3777    <wsdl:part name="parameters" element="tns:RetrieveBiographicInformation"/>
3778    </wsdl:message>
3779    <wsdl:message name="RetrieveBiographicInformationResponseMessage">
3780    <wsdl:part name="parameters" element="tns:RetrieveBiographicInformationResponse"/>
3781    </wsdl:message>
3782    <wsdl:message name="RetrieveBiometricInformationRequestMessage">
3783    <wsdl:part name="parameters" element="tns:RetrieveBiometricInformation"/>
3784    </wsdl:message>
3785    <wsdl:message name="RetrieveBiometricInformationResponseMessage">
3786    <wsdl:part name="parameters" element="tns:RetrieveBiometricInformationResponse"/>
3787    </wsdl:message>
3788    <wsdl:message name="SetBiographicDataRequestMessage">
3789    <wsdl:part name="parameters" element="tns:SetBiographicData"/>
3790    </wsdl:message>
3791    <wsdl:message name="SetBiographicDataResponseMessage">
3792    <wsdl:part name="parameters" element="tns:SetBiographicDataResponse"/>
3793    </wsdl:message>
3794    <wsdl:message name="SetBiometricDataRequestMessage">
```

```
3795    <wsdl:part name="parameters" element="tns:SetBiometricData"/>
3796    </wsdl:message>
3797    <wsdl:message name="SetBiometricDataResponseMessage">
3798    <wsdl:part name="parameters" element="tns:SetBiometricDataResponse"/>
3799    </wsdl:message>
3800    <wsdl:message name="TransformBiometricDataRequestMessage">
3801    <wsdl:part name="parameters" element="tns:TransformBiometricData"/>
3802    </wsdl:message>
3803    <wsdl:message name="TransformBiometricDataResponseMessage">
3804    <wsdl:part name="parameters" element="tns:TransformBiometricDataResponse"/>
3805    </wsdl:message>
3806    <wsdl:message name="UpdateBiographicDataRequestMessage">
3807    <wsdl:part name="parameters" element="tns:UpdateBiographicData"/>
3808    </wsdl:message>
3809    <wsdl:message name="UpdateBiographicDataResponseMessage">
3810    <wsdl:part name="parameters" element="tns:UpdateBiographicDataResponse"/>
3811    </wsdl:message>
3812    <wsdl:message name="UpdateBiometricDataRequestMessage">
3813    <wsdl:part name="parameters" element="tns:UpdateBiometricData"/>
3814    </wsdl:message>
3815    <wsdl:message name="UpdateBiometricDataResponseMessage">
3816    <wsdl:part name="parameters" element="tns:UpdateBiometricDataResponse"/>
3817    </wsdl:message>
3818    <wsdl:message name="VerifySubjectRequestMessage">
3819    <wsdl:part name="parameters" element="tns:VerifySubject"/>
3820    </wsdl:message>
3821    <wsdl:message name="VerifySubjectResponseMessage">
3822    <wsdl:part name="parameters" element="tns:VerifySubjectResponse"/>
3823    </wsdl:message>
3824    <wsdl:message name="EnrollRequestMessage">
3825    <wsdl:part name="parameters" element="tns:Enroll"/>
3826    </wsdl:message>
3827    <wsdl:message name="EnrollResponseMessage">
3828    <wsdl:part name="parameters" element="tns:EnrollResponse"/>
3829    </wsdl:message>
3830    <wsdl:message name="GetEnrollResultsRequestMessage">
3831    <wsdl:part name="parameters" element="tns:GetEnrollResults"/>
3832    </wsdl:message>
3833    <wsdl:message name="GetEnrollResultsResponseMessage">
3834    <wsdl:part name="parameters" element="tns:GetEnrollResultsResponse"/>
3835    </wsdl:message>
```

```
3836    <wsdl:message name="GetIdentifyResultsRequestMessage">
3837    <wsdl:part name="parameters" element="tns:GetIdentifyResults"/>
3838    </wsdl:message>
3839    <wsdl:message name="GetIdentifyResultsResponseMessage">
3840    <wsdl:part name="parameters" element="tns:GetIdentifyResultsResponse"/>
3841    </wsdl:message>
3842    <wsdl:message name="GetVerifyResultsRequestMessage">
3843    <wsdl:part name="parameters" element="tns:GetVerifyResults"/>
3844    </wsdl:message>
3845    <wsdl:message name="GetVerifyResultsResponseMessage">
3846    <wsdl:part name="parameters" element="tns:GetVerifyResultsResponse"/>
3847    </wsdl:message>
3848    <wsdl:message name="IdentifyRequestMessage">
3849    <wsdl:part name="parameters" element="tns:Identify"/>
3850    </wsdl:message>
3851    <wsdl:message name="IdentifyResponseMessage">
3852    <wsdl:part name="parameters" element="tns:IdentifyResponse"/>
3853    </wsdl:message>
3854    <wsdl:message name="RetrieveInformationRequestMessage">
3855    <wsdl:part name="parameters" element="tns:RetrieveInformation"/>
3856    </wsdl:message>
3857    <wsdl:message name="RetrieveInformationResponseMessage">
3858    <wsdl:part name="parameters" element="tns:RetrieveInformationResponse"/>
3859    </wsdl:message>
3860    <wsdl:message name="VerifyRequestMessage">
3861    <wsdl:part name="parameters" element="tns:Verify"/>
3862    </wsdl:message>
3863    <wsdl:message name="VerifyResponseMessage">
3864    <wsdl:part name="parameters" element="tns:VerifyResponse"/>
3865    </wsdl:message>
3866    <wsdl:portType name="BIAS_v1">
3867    <wsdl:operation name="AddSubjectToGallery">
3868    <wsdl:input name="AddSubjectToGallery" message="tns:AddSubjectToGalleryRequestMessage"/>
3869    <wsdl:output name="AddSubjectToGalleryResponse" message="tns:AddSubjectToGalleryResponseMessage"/>
3870    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3871    </wsdl:operation>
3872    <wsdl:operation name="CheckQuality">
3873    <wsdl:input name="CheckQuality" message="tns:CheckQualityRequestMessage"/>
3874    <wsdl:output name="CheckQualityResponse" message="tns:CheckQualityResponseMessage"/>
3875    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3876    </wsdl:operation>
```

```
3877    <wsdl:operation name="ClassifyBiometricData">
3878    <wsdl:input name="ClassifyBiometricData" message="tns:ClassifyBiometricDataRequestMessage"/>
3879    <wsdl:output name="ClassifyBiometricDataResponse" message="tns:ClassifyBiometricDataResponseMessage"/>
3880    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3881    </wsdl:operation>
3882    <wsdl:operation name="CreateSubject">
3883    <wsdl:input name="CreateSubject" message="tns:CreateSubjectRequestMessage"/>
3884    <wsdl:output name="CreateSubjectResponse" message="tns:CreateSubjectResponseMessage"/>
3885    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3886    </wsdl:operation>
3887    <wsdl:operation name="DeleteBiographicData">
3888    <wsdl:input name="DeleteBiographicData" message="tns:DeleteBiographicDataRequestMessage"/>
3889    <wsdl:output name="DeleteBiographicDataResponse" message="tns:DeleteBiographicDataResponseMessage"/>
3890    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3891    </wsdl:operation>
3892    <wsdl:operation name="DeleteBiometricData">
3893    <wsdl:input name="DeleteBiometricData" message="tns:DeleteBiometricDataRequestMessage"/>
3894    <wsdl:output name="DeleteBiometricDataResponse" message="tns:DeleteBiometricDataResponseMessage"/>
3895    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3896    </wsdl:operation>
3897    <wsdl:operation name="DeleteSubject">
3898    <wsdl:input name="DeleteSubject" message="tns:DeleteSubjectRequestMessage"/>
3899    <wsdl:output name="DeleteSubjectResponse" message="tns:DeleteSubjectResponseMessage"/>
3900    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3901    </wsdl:operation>
3902    <wsdl:operation name="DeleteSubjectFromGallery">
3903    <wsdl:input name="DeleteSubjectFromGallery" message="tns:DeleteSubjectFromGalleryRequestMessage"/>
3904    <wsdl:output name="DeleteSubjectFromGalleryResponse"
3905    message="tns:DeleteSubjectFromGalleryResponseMessage"/>
3906    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3907    </wsdl:operation>
3908    <wsdl:operation name="GetIdentifySubjectResults">
3909    <wsdl:input name="GetIdentifySubjectResults" message="tns:GetIdentifySubjectResultsRequestMessage"/>
3910    <wsdl:output name="GetIdentifySubjectResultsResponse"
3911    message="tns:GetIdentifySubjectResultsResponseMessage"/>
3912    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3913    </wsdl:operation>
3914    <wsdl:operation name="IdentifySubject">
3915    <wsdl:input name="IdentifySubject" message="tns:IdentifySubjectRequestMessage"/>
3916    <wsdl:output name="IdentifySubjectResponse" message="tns:IdentifySubjectResponseMessage"/>
3917    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
```

```
3918    </wsdl:operation>
3919    <wsdl:operation name="ListBiographicData">
3920    <wsdl:input name="ListBiographicData" message="tns:ListBiographicDataRequestMessage"/>
3921    <wsdl:output name="ListBiographicDataResponse" message="tns:ListBiographicDataResponseMessage"/>
3922    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3923    </wsdl:operation>
3924    <wsdl:operation name="ListBiometricData">
3925    <wsdl:input name="ListBiometricData" message="tns:ListBiometricDataRequestMessage"/>
3926    <wsdl:output name="ListBiometricDataResponse" message="tns:ListBiometricDataResponseMessage"/>
3927    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3928    </wsdl:operation>
3929    <wsdl:operation name="PerformFusion">
3930    <wsdl:input name="PerformFusion" message="tns:PerformFusionRequestMessage"/>
3931    <wsdl:output name="PerformFusionResponse" message="tns:PerformFusionResponseMessage"/>
3932    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3933    </wsdl:operation>
3934    <wsdl:operation name="QueryCapabilities">
3935    <wsdl:input name="QueryCapabilities" message="tns:QueryCapabilitiesRequestMessage"/>
3936    <wsdl:output name="QueryCapabilitiesResponse" message="tns:QueryCapabilitiesResponseMessage"/>
3937    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3938    </wsdl:operation>
3939    <wsdl:operation name="RetrieveBiographicInformation">
3940    <wsdl:input name="RetrieveBiographicInformation"
3941    message="tns:RetrieveBiographicInformationRequestMessage"/>
3942    <wsdl:output name="RetrieveBiographicInformationResponse"
3943    message="tns:RetrieveBiographicInformationResponseMessage"/>
3944    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3945    </wsdl:operation>
3946    <wsdl:operation name="RetrieveBiometricInformation">
3947    <wsdl:input name="RetrieveBiometricInformation"
3948    message="tns:RetrieveBiometricInformationRequestMessage"/>
3949    <wsdl:output name="RetrieveBiometricInformationResponse"
3950    message="tns:RetrieveBiometricInformationResponseMessage"/>
3951    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3952    </wsdl:operation>
3953    <wsdl:operation name="SetBiographicData">
3954    <wsdl:input name="SetBiographicData" message="tns:SetBiographicDataRequestMessage"/>
3955    <wsdl:output name="SetBiographicDataResponse" message="tns:SetBiographicDataResponseMessage"/>
3956    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
3957    </wsdl:operation>
3958    <wsdl:operation name="SetBiometricData">
3959    <wsdl:input name="SetBiometricData" message="tns:SetBiometricDataRequestMessage"/>
```

```
3960    <wsdl:output name="SetBiometricDataResponse" message="tns:SetBiometricDataResponseMessage"/>

3961    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3962    </wsdl:operation>

3963    <wsdl:operation name="TransformBiometricData">

3964    <wsdl:input name="TransformBiometricData" message="tns:TransformBiometricDataRequestMessage"/>

3965    <wsdl:output name="TransformBiometricDataResponse"
3966    message="tns:TransformBiometricDataResponseMessage"/>

3967    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3968    </wsdl:operation>

3969    <wsdl:operation name="UpdateBiographicData">

3970    <wsdl:input name="UpdateBiographicData" message="tns:UpdateBiographicDataRequestMessage"/>

3971    <wsdl:output name="UpdateBiographicDataResponse" message="tns:UpdateBiographicDataResponseMessage"/>

3972    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3973    </wsdl:operation>

3974    <wsdl:operation name="UpdateBiometricData">

3975    <wsdl:input name="UpdateBiometricData" message="tns:UpdateBiometricDataRequestMessage"/>

3976    <wsdl:output name="UpdateBiometricDataResponse" message="tns:UpdateBiometricDataResponseMessage"/>

3977    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3978    </wsdl:operation>

3979    <wsdl:operation name="VerifySubject">

3980    <wsdl:input name="VerifySubject" message="tns:VerifySubjectRequestMessage"/>

3981    <wsdl:output name="VerifySubjectResponse" message="tns:VerifySubjectResponseMessage"/>

3982    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3983    </wsdl:operation>

3984    <wsdl:operation name="Enroll">

3985    <wsdl:input name="Enroll" message="tns:EnrollRequestMessage"/>

3986    <wsdl:output name="EnrollResponse" message="tns:EnrollResponseMessage"/>

3987    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3988    </wsdl:operation>

3989    <wsdl:operation name="GetEnrollResults">

3990    <wsdl:input name="GetEnrollResults" message="tns:GetEnrollResultsRequestMessage"/>

3991    <wsdl:output name="GetEnrollResultsResponse" message="tns:GetEnrollResultsResponseMessage"/>

3992    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3993    </wsdl:operation>

3994    <wsdl:operation name="GetIdentifyResults">

3995    <wsdl:input name="GetIdentifyResults" message="tns:GetIdentifyResultsRequestMessage"/>

3996    <wsdl:output name="GetIdentifyResultsResponse" message="tns:GetIdentifyResultsResponseMessage"/>

3997    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>

3998    </wsdl:operation>

3999    <wsdl:operation name="GetVerifyResults">

4000    <wsdl:input name="GetVerifyResults" message="tns:GetVerifyResultsRequestMessage"/>
```

```
4001    <wsdl:output name="GetVerifyResultsResponse" message="tns:GetVerifyResultsResponseMessage"/>
4002    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4003    </wsdl:operation>
4004    <wsdl:operation name="Identify">
4005    <wsdl:input name="Identify" message="tns:IdentifyRequestMessage"/>
4006    <wsdl:output name="IdentifyResponse" message="tns:IdentifyResponseMessage"/>
4007    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4008    </wsdl:operation>
4009    <wsdl:operation name="RetrieveInformation">
4010    <wsdl:input name="RetrieveInformation" message="tns:RetrieveInformationRequestMessage"/>
4011    <wsdl:output name="RetrieveInformationResponse" message="tns:RetrieveInformationResponseMessage"/>
4012    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4013    </wsdl:operation>
4014    <wsdl:operation name="Verify">
4015    <wsdl:input name="Verify" message="tns:VerifyRequestMessage"/>
4016    <wsdl:output name="VerifyResponse" message="tns:VerifyResponseMessage"/>
4017    <wsdl:fault name="BIASFault" message="tns:BIASFaultMessage"/>
4018    </wsdl:operation>
4019    </wsdl:portType>
4020    <wsdl:binding name="BIAS_v1HttpBinding" type="tns:BIAS_v1">
4021    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
4022    <wsdl:operation name="AddSubjectToGallery">
4023    <soap:operation soapAction="AddSubjectToGallery"/>
4024    <wsdl:input name="AddSubjectToGallery">
4025    <soap:body use="literal"/>
4026    </wsdl:input>
4027    <wsdl:output name="AddSubjectToGalleryResponse">
4028    <soap:body use="literal"/>
4029    </wsdl:output>
4030    <wsdl:fault name="BIASFault">
4031    <soap:fault name="BIASFault" use="literal"/>
4032    </wsdl:fault>
4033    </wsdl:operation>
4034    <wsdl:operation name="CheckQuality">
4035    <soap:operation soapAction="CheckQuality"/>
4036    <wsdl:input name="CheckQuality">
4037    <soap:body use="literal"/>
4038    </wsdl:input>
4039    <wsdl:output name="CheckQualityResponse">
4040    <soap:body use="literal"/>
4041    </wsdl:output>
```

```
4042    <wsdl:fault name="BIASFault">
4043    <soap:fault name="BIASFault" use="literal"/>
4044    </wsdl:fault>
4045    </wsdl:operation>
4046    <wsdl:operation name="ClassifyBiometricData">
4047    <soap:operation soapAction="ClassifyBiometricData"/>
4048    <wsdl:input name="ClassifyBiometricData">
4049    <soap:body use="literal"/>
4050    </wsdl:input>
4051    <wsdl:output name="ClassifyBiometricDataResponse">
4052    <soap:body use="literal"/>
4053    </wsdl:output>
4054    <wsdl:fault name="BIASFault">
4055    <soap:fault name="BIASFault" use="literal"/>
4056    </wsdl:fault>
4057    </wsdl:operation>
4058    <wsdl:operation name="CreateSubject">
4059    <soap:operation soapAction="CreateSubject"/>
4060    <wsdl:input name="CreateSubject">
4061    <soap:body use="literal"/>
4062    </wsdl:input>
4063    <wsdl:output name="CreateSubjectResponse">
4064    <soap:body use="literal"/>
4065    </wsdl:output>
4066    <wsdl:fault name="BIASFault">
4067    <soap:fault name="BIASFault" use="literal"/>
4068    </wsdl:fault>
4069    </wsdl:operation>
4070    <wsdl:operation name="DeleteBiographicData">
4071    <soap:operation soapAction="DeleteBiographicData"/>
4072    <wsdl:input name="DeleteBiographicData">
4073    <soap:body use="literal"/>
4074    </wsdl:input>
4075    <wsdl:output name="DeleteBiographicDataResponse">
4076    <soap:body use="literal"/>
4077    </wsdl:output>
4078    <wsdl:fault name="BIASFault">
4079    <soap:fault name="BIASFault" use="literal"/>
4080    </wsdl:fault>
4081    </wsdl:operation>
4082    <wsdl:operation name="DeleteBiometricData">
```

```
4083    <soap:operation soapAction="DeleteBiometricData"/>
4084    <wsdl:input name="DeleteBiometricData">
4085    <soap:body use="literal"/>
4086    </wsdl:input>
4087    <wsdl:output name="DeleteBiometricDataResponse">
4088    <soap:body use="literal"/>
4089    </wsdl:output>
4090    <wsdl:fault name="BIASFault">
4091    <soap:fault name="BIASFault" use="literal"/>
4092    </wsdl:fault>
4093    </wsdl:operation>
4094    <wsdl:operation name="DeleteSubject">
4095    <soap:operation soapAction="DeleteSubject"/>
4096    <wsdl:input name="DeleteSubject">
4097    <soap:body use="literal"/>
4098    </wsdl:input>
4099    <wsdl:output name="DeleteSubjectResponse">
4100    <soap:body use="literal"/>
4101    </wsdl:output>
4102    <wsdl:fault name="BIASFault">
4103    <soap:fault name="BIASFault" use="literal"/>
4104    </wsdl:fault>
4105    </wsdl:operation>
4106    <wsdl:operation name="DeleteSubjectFromGallery">
4107    <soap:operation soapAction="DeleteSubjectFromGallery"/>
4108    <wsdl:input name="DeleteSubjectFromGallery">
4109    <soap:body use="literal"/>
4110    </wsdl:input>
4111    <wsdl:output name="DeleteSubjectFromGalleryResponse">
4112    <soap:body use="literal"/>
4113    </wsdl:output>
4114    <wsdl:fault name="BIASFault">
4115    <soap:fault name="BIASFault" use="literal"/>
4116    </wsdl:fault>
4117    </wsdl:operation>
4118    <wsdl:operation name="GetIdentifySubjectResults">
4119    <soap:operation soapAction="GetIdentifySubjectResults"/>
4120    <wsdl:input name="GetIdentifySubjectResults">
4121    <soap:body use="literal"/>
4122    </wsdl:input>
4123    <wsdl:output name="GetIdentifySubjectResultsResponse">
```

```
4124        <soap:body use="literal"/>
4125      </wsdl:output>
4126      <wsdl:fault name="BIASFault">
4127        <soap:fault name="BIASFault" use="literal"/>
4128      </wsdl:fault>
4129    </wsdl:operation>
4130    <wsdl:operation name="IdentifySubject">
4131      <soap:operation soapAction="IdentifySubject"/>
4132      <wsdl:input name="IdentifySubject">
4133        <soap:body use="literal"/>
4134      </wsdl:input>
4135      <wsdl:output name="IdentifySubjectResponse">
4136        <soap:body use="literal"/>
4137      </wsdl:output>
4138      <wsdl:fault name="BIASFault">
4139        <soap:fault name="BIASFault" use="literal"/>
4140      </wsdl:fault>
4141    </wsdl:operation>
4142    <wsdl:operation name="ListBiographicData">
4143      <soap:operation soapAction="ListBiographicData"/>
4144      <wsdl:input name="ListBiographicData">
4145        <soap:body use="literal"/>
4146      </wsdl:input>
4147      <wsdl:output name="ListBiographicDataResponse">
4148        <soap:body use="literal"/>
4149      </wsdl:output>
4150      <wsdl:fault name="BIASFault">
4151        <soap:fault name="BIASFault" use="literal"/>
4152      </wsdl:fault>
4153    </wsdl:operation>
4154    <wsdl:operation name="ListBiometricData">
4155      <soap:operation soapAction="ListBiometricData"/>
4156      <wsdl:input name="ListBiometricData">
4157        <soap:body use="literal"/>
4158      </wsdl:input>
4159      <wsdl:output name="ListBiometricDataResponse">
4160        <soap:body use="literal"/>
4161      </wsdl:output>
4162      <wsdl:fault name="BIASFault">
4163        <soap:fault name="BIASFault" use="literal"/>
4164      </wsdl:fault>
```

```
4165    </wsdl:operation>
4166    <wsdl:operation name="PerformFusion">
4167    <soap:operation soapAction="PerformFusion"/>
4168    <wsdl:input name="PerformFusion">
4169    <soap:body use="literal"/>
4170    </wsdl:input>
4171    <wsdl:output name="PerformFusionResponse">
4172    <soap:body use="literal"/>
4173    </wsdl:output>
4174    <wsdl:fault name="BIASFault">
4175    <soap:fault name="BIASFault" use="literal"/>
4176    </wsdl:fault>
4177    </wsdl:operation>
4178    <wsdl:operation name="QueryCapabilities">
4179    <soap:operation soapAction="QueryCapabilities"/>
4180    <wsdl:input name="QueryCapabilities">
4181    <soap:body use="literal"/>
4182    </wsdl:input>
4183    <wsdl:output name="QueryCapabilitiesResponse">
4184    <soap:body use="literal"/>
4185    </wsdl:output>
4186    <wsdl:fault name="BIASFault">
4187    <soap:fault name="BIASFault" use="literal"/>
4188    </wsdl:fault>
4189    </wsdl:operation>
4190    <wsdl:operation name="RetrieveBiographicInformation">
4191    <soap:operation soapAction="RetrieveBiographicInformation"/>
4192    <wsdl:input name="RetrieveBiographicInformation">
4193    <soap:body use="literal"/>
4194    </wsdl:input>
4195    <wsdl:output name="RetrieveBiographicInformationResponse">
4196    <soap:body use="literal"/>
4197    </wsdl:output>
4198    <wsdl:fault name="BIASFault">
4199    <soap:fault name="BIASFault" use="literal"/>
4200    </wsdl:fault>
4201    </wsdl:operation>
4202    <wsdl:operation name="RetrieveBiometricInformation">
4203    <soap:operation soapAction="RetrieveBiometricInformation"/>
4204    <wsdl:input name="RetrieveBiometricInformation">
4205    <soap:body use="literal"/>
```

```
4206    </wsdl:input>
4207    <wsdl:output name="RetrieveBiometricInformationResponse">
4208    <soap:body use="literal"/>
4209    </wsdl:output>
4210    <wsdl:fault name="BIASFault">
4211    <soap:fault name="BIASFault" use="literal"/>
4212    </wsdl:fault>
4213    </wsdl:operation>
4214    <wsdl:operation name="SetBiographicData">
4215    <soap:operation soapAction="SetBiographicData"/>
4216    <wsdl:input name="SetBiographicData">
4217    <soap:body use="literal"/>
4218    </wsdl:input>
4219    <wsdl:output name="SetBiographicDataResponse">
4220    <soap:body use="literal"/>
4221    </wsdl:output>
4222    <wsdl:fault name="BIASFault">
4223    <soap:fault name="BIASFault" use="literal"/>
4224    </wsdl:fault>
4225    </wsdl:operation>
4226    <wsdl:operation name="SetBiometricData">
4227    <soap:operation soapAction="SetBiometricData"/>
4228    <wsdl:input name="SetBiometricData">
4229    <soap:body use="literal"/>
4230    </wsdl:input>
4231    <wsdl:output name="SetBiometricDataResponse">
4232    <soap:body use="literal"/>
4233    </wsdl:output>
4234    <wsdl:fault name="BIASFault">
4235    <soap:fault name="BIASFault" use="literal"/>
4236    </wsdl:fault>
4237    </wsdl:operation>
4238    <wsdl:operation name="TransformBiometricData">
4239    <soap:operation soapAction="TransformBiometricData"/>
4240    <wsdl:input name="TransformBiometricData">
4241    <soap:body use="literal"/>
4242    </wsdl:input>
4243    <wsdl:output name="TransformBiometricDataResponse">
4244    <soap:body use="literal"/>
4245    </wsdl:output>
4246    <wsdl:fault name="BIASFault">
```

```
4247    <soap:fault name="BIASFault" use="literal"/>
4248    </wsdl:fault>
4249    </wsdl:operation>
4250    <wsdl:operation name="UpdateBiographicData">
4251    <soap:operation soapAction="UpdateBiographicData"/>
4252    <wsdl:input name="UpdateBiographicData">
4253    <soap:body use="literal"/>
4254    </wsdl:input>
4255    <wsdl:output name="UpdateBiographicDataResponse">
4256    <soap:body use="literal"/>
4257    </wsdl:output>
4258    <wsdl:fault name="BIASFault">
4259    <soap:fault name="BIASFault" use="literal"/>
4260    </wsdl:fault>
4261    </wsdl:operation>
4262    <wsdl:operation name="UpdateBiometricData">
4263    <soap:operation soapAction="UpdateBiometricData"/>
4264    <wsdl:input name="UpdateBiometricData">
4265    <soap:body use="literal"/>
4266    </wsdl:input>
4267    <wsdl:output name="UpdateBiometricDataResponse">
4268    <soap:body use="literal"/>
4269    </wsdl:output>
4270    <wsdl:fault name="BIASFault">
4271    <soap:fault name="BIASFault" use="literal"/>
4272    </wsdl:fault>
4273    </wsdl:operation>
4274    <wsdl:operation name="VerifySubject">
4275    <soap:operation soapAction="VerifySubject"/>
4276    <wsdl:input name="VerifySubject">
4277    <soap:body use="literal"/>
4278    </wsdl:input>
4279    <wsdl:output name="VerifySubjectResponse">
4280    <soap:body use="literal"/>
4281    </wsdl:output>
4282    <wsdl:fault name="BIASFault">
4283    <soap:fault name="BIASFault" use="literal"/>
4284    </wsdl:fault>
4285    </wsdl:operation>
4286    <wsdl:operation name="Enroll">
4287    <soap:operation soapAction="Enroll"/>
```

```
4288    <wsdl:input name="Enroll">
4289    <soap:body use="literal"/>
4290    </wsdl:input>
4291    <wsdl:output name="EnrollResponse">
4292    <soap:body use="literal"/>
4293    </wsdl:output>
4294    <wsdl:fault name="BIASFault">
4295    <soap:fault name="BIASFault" use="literal"/>
4296    </wsdl:fault>
4297    </wsdl:operation>
4298    <wsdl:operation name="GetEnrollResults">
4299    <soap:operation soapAction="GetEnrollResults"/>
4300    <wsdl:input name="GetEnrollResults">
4301    <soap:body use="literal"/>
4302    </wsdl:input>
4303    <wsdl:output name="GetEnrollResultsResponse">
4304    <soap:body use="literal"/>
4305    </wsdl:output>
4306    <wsdl:fault name="BIASFault">
4307    <soap:fault name="BIASFault" use="literal"/>
4308    </wsdl:fault>
4309    </wsdl:operation>
4310    <wsdl:operation name="GetIdentifyResults">
4311    <soap:operation soapAction="GetIdentifyResults"/>
4312    <wsdl:input name="GetIdentifyResults">
4313    <soap:body use="literal"/>
4314    </wsdl:input>
4315    <wsdl:output name="GetIdentifyResultsResponse">
4316    <soap:body use="literal"/>
4317    </wsdl:output>
4318    <wsdl:fault name="BIASFault">
4319    <soap:fault name="BIASFault" use="literal"/>
4320    </wsdl:fault>
4321    </wsdl:operation>
4322    <wsdl:operation name="GetVerifyResults">
4323    <soap:operation soapAction="GetVerifyResults"/>
4324    <wsdl:input name="GetVerifyResults">
4325    <soap:body use="literal"/>
4326    </wsdl:input>
4327    <wsdl:output name="GetVerifyResultsResponse">
4328    <soap:body use="literal"/>
```

```
4329   </wsdl:output>
4330   <wsdl:fault name="BIASFault">
4331   <soap:fault name="BIASFault" use="literal"/>
4332   </wsdl:fault>
4333   </wsdl:operation>
4334   <wsdl:operation name="Identify">
4335   <soap:operation soapAction="Identify"/>
4336   <wsdl:input name="Identify">
4337   <soap:body use="literal"/>
4338   </wsdl:input>
4339   <wsdl:output name="IdentifyResponse">
4340   <soap:body use="literal"/>
4341   </wsdl:output>
4342   <wsdl:fault name="BIASFault">
4343   <soap:fault name="BIASFault" use="literal"/>
4344   </wsdl:fault>
4345   </wsdl:operation>
4346   <wsdl:operation name="RetrieveInformation">
4347   <soap:operation soapAction="RetrieveInformation"/>
4348   <wsdl:input name="RetrieveInformation">
4349   <soap:body use="literal"/>
4350   </wsdl:input>
4351   <wsdl:output name="RetrieveInformationResponse">
4352   <soap:body use="literal"/>
4353   </wsdl:output>
4354   <wsdl:fault name="BIASFault">
4355   <soap:fault name="BIASFault" use="literal"/>
4356   </wsdl:fault>
4357   </wsdl:operation>
4358   <wsdl:operation name="Verify">
4359   <soap:operation soapAction="Verify"/>
4360   <wsdl:input name="Verify">
4361   <soap:body use="literal"/>
4362   </wsdl:input>
4363   <wsdl:output name="VerifyResponse">
4364   <soap:body use="literal"/>
4365   </wsdl:output>
4366   <wsdl:fault name="BIASFault">
4367   <soap:fault name="BIASFault" use="literal"/>
4368   </wsdl:fault>
4369   </wsdl:operation>
```

```
4370    </wsdl:binding>
4371    <wsdl:service name="BIAS_v1">
4372    <wsdl:port name="BIAS_v1HttpBindingPort" binding="tns:BIAS_v1HttpBinding">
4373    <soap:address location="http://your_URI/IRI_to_your_BIAS_implementation"/>
4374    </wsdl:port>
4375    </wsdl:service>
4376    </wsdl:definitions>
4377
```

# Annex B.    BIAS Patron format specification

4379 The BIAS SOAP Profile defines an XML CBEFF Patron Format based on, but tailored from, Clause 13/15
4380 of ISO/IEC 19785-3 [CBEFF3] as specified below.

4381

## B.1  Patron

4383 Organization for the Advancement of Structured Information Standards (OASIS)

4384

## B.2  Patron identifier

4386 **82 (0052 Hex).**

4387

4388 This has been allocated by the Registration Authority for ISO/IEC 19785-2.

4389

## B.3  Patron format name

4391 **OASIS BIAS CBEFF XML Patron Format**

4392

## B.4  Patron format identifier

4394 **01 (0001 Hex).**

4395

4396 This has been registered in accordance with ISO/IEC 19785-2.

4397

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

4399 **No ASN.1 object identifiers are assigned to this patron format**

4400

## B.6  Domain of use

4402 This clause specifies a patron format based on XML that is designed to be friendly with code generation
4403 tools. It defines a CBEFF structure that allows for the creation of simple, complex, and multi-modal BIRs
4404 for use within BIAS transactions.

4405

## B.7  Version identifier

4407 This patron format specification has a version identifier of (major 1, minor 0).

4408

## B.8  CBEFF version

4410 This specification conforms to CBEFF version (major 2, minor 0).

4411

## B.9 General

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

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

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

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

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

## B.10 Specification

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

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

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

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

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

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

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

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

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

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

## B.11 Element <BIR>

### B.11.1 Syntax

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

  a)  an optional **<Version>** element (see B.12);

  b)  an optional **<CBEFFVersion>** element (see B.13);

  c)  zero or more application-specific elements;

  d)  a mandatory **<BIRInfo>** element (see B.14);

  e)  an optional **<BDBInfo>** element (see B.15);

  f)  an optional **<SBInfo>** element (see B.16);

  g)  zero or more **<BIR>** elements (see B.11);

  h)  either an optional **<BDB>** element that shall contain a valid representation of an octect string (see B.18), or an optional **<bdbX>** element that shall contain a valid XML string;

  i)  an optional **<SB>** element – the content of this element shall be a valid representation of an octet string.

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

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

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

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

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


### B.11.2 Semantics

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

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

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

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

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

4493  **B.11.2.6**         The **&lt;BDB&gt;** or **&lt;bdbX&gt;** element (if present) carries the biometric data block (BDB) of the
4494  BIR.

4495         NOTE – A **&lt;BDB&gt;** or **&lt;bdbX&gt;** element and a **&lt;BIR&gt;** element cannot coexist as children of
4496         the same **&lt;BIR&gt;** element (see B.11.1.2).

4497  **B.11.2.7**         The **&lt;SB&gt;** element (if present) carries the security block (SB) of the BIR.

4498         NOTE – A **&lt;SB&gt;** element can coexist with either a **&lt;BIR&gt;** element or a **&lt;BDB&gt; or**
4499         **&lt;bdbX&gt;** element that is a child of the same **&lt;BIR&gt;** element.

4500  **B.11.2.8**         The **&lt;BIRInfo&gt;** element carries information about both the BIR and (possibly) about its
4501  descendant BIRs (if the **&lt;BIR&gt;** element has one or more child **&lt;BIR&gt;** elements), as specified in B.14.2.1.

4502  **B.11.2.9**         The **&lt;BDBInfo&gt;** element (if present) carries information about either the BDB of the BIR
4503  (if the **&lt;BIR&gt;** element has a child **&lt;BDB&gt;** or **&lt;bdbX&gt;** element) or about the BDBs of the descendant BIRs
4504  that have a child **&lt;BDB&gt;** or **&lt;bdbX&gt;** element (if the **&lt;BIR&gt;** element has one or more child **&lt;BIR&gt;**
4505  elements), as specified in B.15.2.1.

4506  **B.11.2.10**         The **&lt;SBInfo&gt;** element (if present) carries information about either the SB of the BIR (if
4507  the **&lt;BIR&gt;** element has a child **&lt;SB&gt;** element) or about the SBs of the descendant BIRs that have a child
4508  **&lt;SB&gt;** element (if the **&lt;BIR&gt;** element has one or more child **&lt;BIR&gt;** elements but no child **&lt;SB&gt;** element),
4509  as specified in B.16.2.1.

4510

# 4511  B.12 Element &lt;Version&gt;

## 4512  B.12.1 Syntax

4513  This element shall have contents consisting of the following (in order):

4514  a) a required  **&lt;Major&gt;** element – the value of this element shall be a valid representation of a non-
4515      negative integer.

4516  b) a required  **&lt;Minor&gt;** element – the value of this element shall be a valid representation of a non-
4517      negative integer.

## 4518  B.12.2 Semantics

4519  **B.12.2.1**         This element represents the data element CBEFF_patron_header_version, and carries
4520  the (major and minor) version number of the patron format.  The number assigned to this version of the
4521  patron format is major 1, minor 0.

4522  **B.12.2.2**         The **&lt;Major&gt;** element represents the major version number (1 in this version).

4523  **B.12.2.3**         The **&lt;Minor&gt;** element represents the minor version number (0 in this version).

4524  **B.12.2.4**         If this element is not present, the values **Major="1" Minor="0"** are implied.

4525  **B.12.2.5**         A child **&lt;BIR&gt;** element shall have the same (major and minor) version number as its
4526  parent **&lt;BIR&gt;** element.

4527         NOTE – This implies that the **&lt;Version&gt;** element, if present in a child **&lt;BIR&gt;** element,
4528         has to carry the same values as the **&lt;Version&gt;** element in the parent **&lt;BIR&gt;** element.
4529         This is equivalent to omitting the **&lt;Version&gt;** element.  Therefore, this element is
4530         normally omitted in child **&lt;BIR&gt;** elements.

4531

## B.13 Element <CBEFFVersion>

### B.13.1 Syntax

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

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

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

### B.13.2 Semantics

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

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

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

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

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

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


## B.14 Element <BIRInfo>

### B.14.1 Syntax

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

a) an optional **<Creator>** element – the content of this element shall be a string of ISO/IEC 10646 characters;

b) an optional **<Index>** element – the content of this element shall be a valid representation of a universally unique identifier (see B.20), and shall not inherit its value from any other level BIR;

c) an optional **<Payload>** element – the content of this element shall be a valid representation of an octet string, and shall not inherit its value from any other level BIR.

d) a required **<Integrity>** element – the value of this element shall be one of the character strings in the third cell of the corresponding row of Table B.1;

e) an optional **<CreationDate>** element – the value of this element shall be a valid representation of a date and time of the day (see B.19);

f) an optional **<NotValidBefore>** element – the value of this element shall be a valid representation of a date and time of the day;

g) an optional **<NotValidAfter>** element – the value of this element shall be a valid representation of a date and time of the day.

## B.14.2 Semantics

4569  **B.14.2.1**   The `<BIRInfo>` element carries information about the BIR.  In addition, if the BIR has
4570  one or more child BIRs (the `<BIR>` element has one or more child `<BIR>` elements), the information
4571  carried by the attributes and child elements of the `<BIRInfo>` element is inherited by those child BIRs
4572  except where overridden by a corresponding attribute or child element of the `<BIRInfo>` element of a
4573  child BIR.  The information inherited by a BIR applies to that BIR, and (if the BIR has itself child BIRs) is
4574  further inherited by its child BIRs in the same way (and so on recursively).

4575      NOTE – Since the `Integrity` element is required and the `<BIRInfo>` element is
4576      mandatory in all `<BIR>` elements, inheritance of the `Integrity` element can never
4577      occur.

4578  **B.14.2.2**   The `Integrity` element indicates whether integrity information about this BIR is
4579  provided within the security block (SB) of the BIR (the child `<SB>` element of the parent `<BIR>` element of
4580  this `<BIRInfo>` element).

4581      NOTE – This information may consist of a digital signature or MAC, a reference to a key
4582      or certificate, an encrypted key (with or without a reference to the key used to encrypt
4583      that key), or other parameters of the digital signing (or MAC) process.

4584  **B.14.2.3**   If the value of the `<Integrity>` element is "`true`", then the parent `<BIR>` element of
4585  this `<BIRInfo>` element shall have a child `<SB>` element.

4586  **B.14.2.4**   Table B.1 specifies the correspondence between the attributes and child elements of this
4587  element and CBEFF data elements, and specifies the supported abstract values and their encodings (see
4588  also B.10.7).

4589      NOTE - This element represents all CBEFF data elements whose name begins with
4590      "CBEFF_BIR_".

4591                                   **Table B.1 – BIR information**

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

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

4592

## B.15  Element <BDBInfo>

### B.15.1  Syntax

4595    **B.15.1.1**        This element shall have a content consisting of the following (in order):

4596    a)  an optional **<ChallengeResponse>** element – the content of this element shall be a valid
4597        representation of an octet string (see B.18);

4598    b)  an optional **<Index>** element – the content of this element shall be a valid representation of a
4599        universally unique identifier (see B.20).

4600    c)  an optional **<FormatOwner>** element – the value of this element shall be a valid representation
4601        of an integer in the range 1 to 65535 (see B.17);

4602    d)  an optional **<FormatType>** element – the value of this element shall be a valid representation of
4603        an integer in the range 1 to 65535;

4604    e)  an optional **<Encryption>** element – the value of this element shall be one of the character
4605        strings in the third cell of the corresponding row of Table B.2;

4606    f)  an optional **<CreationDate>** element – the value of this element shall be a valid
4607        representation of a date and time of the day (see B.19);

4608    g)  an optional **<NotValidBefore>** element – the value of this element shall be a valid
4609        representation of a date and time of the day;

4610    h)  an optional **<NotValidAfter>** element – the value of this element shall be a valid
4611        representation of a date and time of the day;

4612     i)   an optional **`<Type>`** element – the value of this element shall be one of the character strings in
4613          the third cell of the corresponding row of Table B.2;

4614     j)   an optional **`<Subtype>`** element – the value of this element shall be one of the character strings
4615          in the third cell of the corresponding row of Table B.2;

4616     k)   an optional **`<Level>`** element – the value of this element shall be one of the character strings in
4617          the third cell of the corresponding row of Table B.2;

4618     l)   an optional **`<ProductOwner>`** element – the value of this element shall be a valid
4619          representation of an integer in the range 1..65535 (see B.17);

4620     m) an optional **`<ProductType>`** element – the value of this element shall be a valid representation
4621          of an integer in the range 1..65535;

4622     n)   an optional **`<CaptureDeviceOwner>`** element – the value of this element shall be a valid
4623          representation of an integer in the range 1..65535 (see B.17);

4624     o)   an optional **`<CaptureDeviceType>`** element – the value of this element shall be a valid
4625          representation of an integer in the range 1..65535;

4626     p)   an optional **`<FeatureExtractionAlgorithmOwner>`** element – the value of this element
4627          shall be a valid representation of an integer in the range 1..65535 (see B.17);

4628     q)   an optional **`<FeatureExtractionAlgorithmType>`** element – the value of this element shall
4629          be a valid representation of an integer in the range 1..65535;

4630     r)   an optional **`<ComparisonAlgorithmOwner>`** element – the value of this element shall be a
4631          valid representation of an integer in the range 1..65535 (see B.17);

4632     s)   an optional **`<ComparisonAlgorithmType>`** element – the value of this element shall be a valid
4633          representation of an integer in the range 1..65535;

4634     t)   an optional **`<QualityAlgorithmOwner>`** element – the value of this element shall be a valid
4635          representation of an integer in the range 1..65535 (see B.17);

4636     u)   an optional **`<QualityAlgorithmType>`** element – the value of this element shall be a valid
4637          representation of an integer in the range 1..65535;

4638     v)   an optional **`<CompressionAlgorithmOwner>`** element – the value of this element shall be a
4639          valid representation of an integer in the range 1..65535 (see B.17);

4640     w) an optional **`<CompressionAlgorithmType>`** element – the value of this element shall be a
4641          valid representation of an integer in the range 1..65535;

4642     x)   an optional **`<Purpose>`** element – the value of this element shall be one of the character strings
4643          in the third cell of the corresponding row of Table B.2;

4644     y)   an optional **`<Quality>`** element – the value of this element shall be a valid representation of an
4645          integer in the range –2..100 (see B.17), as specified in the third cell of the corresponding row of
4646          Table B.2.

4647 **B.15.1.3**      If the parent **`<BIR>`** element has a child **`<BDB>`** element, then the **`<Encryption>`**
4648 element shall be present in this **`<BDBInfo>`** element unless it is present in the child **`<BDBInfo>`** element
4649 of an ancestor **`<BIR>`** element (see also B.11.1.4).

4650 **B.15.1.4** If the parent **<BIR>** element has a child **<BDB>** element, then the **<FormatOwner>**
4651 element shall be present in this **<BDBInfo>** element unless it is present in the child **<BDBInfo>** element
4652 of an ancestor **<BIR>** element (see also B.11.1.4).

4653 **B.15.1.5** If the parent **<BIR>** element has a child **<BDB>** element, then the **<FormatType>**
4654 element shall be present in this **<BDBInfo>** element unless it is present in the child **<BDBInfo>** element
4655 of an ancestor **<BIR>** element (see also B.11.1.4).

4656 NOTE – The ancestor **<BIR>** elements mentioned in the last three subclauses above
4657 need not be the same.

4658

## 4659 B.15.2 Semantics

4660 **B.15.2.1** If the BIR has a BDB (the **<BIR>** element has a child **<BDB>** element), then the
4661 **<BDBInfo>** element carries information about that BDB. Otherwise, the information carried by the
4662 attributes and child elements of the **<BDBInfo>** element is inherited by all the BIRs that are children of
4663 the BIR except where overridden by a corresponding attribute or child element of the **<BDBInfo>**
4664 element of a child BIR. The information inherited by a BIR with a BDB applies to that BDB, whereas the
4665 information inherited by a BIR that has itself child BIRs is further inherited by all the BIRs that are children
4666 of the BIR in the same way (and so on recursively).

4667 **B.15.2.2** If the BIR has a BDB and encryption is applied to that BDB (either by including the
4668 **encryption** attribute with the value "**true**" in the **<BDBInfo>** element or by having the BIR inherit that
4669 attribute value from its parent BIR), then the BDB in the **<BDB>** element shall be encrypted.

4670 **B.15.2.3** If the BDB of a BIR is encrypted, information about the encryption process may be
4671 provided within the security block (SB) of that BIR (the child **<SB>** element of the parent **<BIR>** element
4672 of this **<BIRInfo>** element).

4673 NOTE – This information may consist of a reference to an encryption key, an encrypted
4674 key (with or without a reference to the key used to encrypt that key), or other parameters
4675 of the encryption process.

4676 **B.15.2.4** Table B.2 specifies the correspondence between the attributes and child elements of this
4677 element and CBEFF data elements, and specifies the supported abstract values and their encodings (see
4678 also F.10.7).

4679 NOTE – This element represents all CBEFF data elements whose name begins with
4680 "CBEFF_BDB_".

4681 **Table B.2 – BDB information**

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

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

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

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

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

4682

## B.16  Element <SBInfo>

### B.16.1  Syntax

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

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

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

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

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

4696     NOTE 1 – The ancestor **`<BIR>`** elements mentioned in the last two subclauses above
4697     need not be the same.

4698     NOTE 2 – When the parent **`<BIR>`** element has a child **`<SB>`** element and one omits both
4699     children of the **`<SBInfo>`** element, the **`<SBInfo>`** element will have no attributes and an
4700     empty content.  Omission of the **`<SBInfo>`** element is not allowed in this case (see
4701     B.11.1.5).

4702

## B.16.2 Semantics

4704 **B.16.2.1**     If the BIR has an SB (the **`<BIR>`** element has a child **`<SB>`** element), then the **`<SBInfo>`**
4705 element carries information about that SB.  In addition, if the BIR has one or more child BIRs (the **`<BIR>`**
4706 element has one or more child **`<BIR>`** elements), the information carried by the child element of the
4707 **`<SBInfo>`** element is inherited by those child BIRs except where overridden by a corresponding child
4708 element of the **`<SBInfo>`** element of a child BIR.  The information inherited by a BIR with an SB applies
4709 to that SB, and (if the BIR has itself child BIRs) is further inherited by its child BIRs in the same way (and
4710 so on recursively).

4711 **B.16.2.2**     Table B.3 specifies the correspondence between the attributes and child elements of this
4712 element and CBEFF data elements, and specifies the supported abstract values and their encodings (see
4713 also B.10.7).

4714     NOTE – This element represents all CBEFF data elements whose name begins with
4715     "CBEFF_SB_".

4716

4717 **Table B.3 – SB information**

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

## B.17 Representation of Integers

4719 **B.17.1** A non-negative integer shall be represented as a string of one or more ISO/IEC 10646 characters
4720 in the range DIGIT ZERO to DIGIT NINE ("**0**" to "**9**") in decimal notation.

4721 **B.17.2** A negative integer shall be represented as the corresponding positive integer, preceded by a
4722 HYPHEN-MINUS character ("**-**").

4723 **B.17.3** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the
4724 encoding.

4725

## B.18 Representation of Octet Strings

4727 **B.18.1** An octet string shall be represented as a string of the following ISO/IEC 10646 characters:

4728 a) LATIN CAPITAL LETTER A to LATIN CAPITAL LETTER Z;

4729     b) LATIN SMALL LETTER A to LATIN SMALL LETTER Z;

4730     c) DIGIT ZERO to DIGIT NINE;

4731     d) PLUS SIGN;

4732     e) SOLIDUS;

4733     f) EQUALS SIGN.

4734 forming the Base64 encoding of the octet string (see IETF RFC 2045), with all whitespace removed.

4735 **B.18.2** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the
4736 encoding.

4737

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

4739 **B.19.1** A date and time of the day shall be represented as a string of ISO/IEC 10646 characters in the
4740 following format, which conforms to ISO 8601.

4741 **B.19.2** The encoding shall be the concatenation of all the following components (in order):

4742     a) the "year" component, consisting of the year encoded in four digits ("`2000`" to "`2999`") ;

4743     b) the hyphen character "`-`"

4744     c) the "month" component, consisting of the month encoded in two digits ("`01`" to "`12`");

4745     d) the hyphen character "`-`"

4746     e) the "day" component, consisting of the day encoded in two digits ("`01`" to "`31`");

4747     f) the letter "`T`";

4748     g) the "hour" component, consisting of the hour encoded in two digits ("`00`" to "`23`");

4749     h) the colon character ":"

4750     i) the "minute" component, consisting of the minute encoded in two digits ("`00`" to "`59`");

4751     j) the colon character ":"

4752     k) the "second" component, consisting of the second encoded in two digits ("`00`" to "`59`");

4753     l) the letter "`Z`".

4754 **B.19.3** The "year", "month", "day", "hour", "minute", and "second" components shall be present.

4755 **B.19.4** The letter "`T`" shall be present.

4756 **B.19.5** The letter "`Z`" shall be present whether or not the "hour" component is present.

4757     NOTE    This letter indicates that the date and time of the day are UTC.

4758 **B.19.6** Arbitrary whitespace is allowed before and after the encoding, but is forbidden inside the
4759 encoding.

4760

## B.20  Representation of Universally Unique Identifiers

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

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

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

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

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

d)  HYPHEN-MINUS ("`-`").

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

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


## B.21  Patron format conformance statement

### B.21.1  Identifying information

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


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

| CBEFF data element name | Mandatory/ optional | Patron format field name | Abstract values specified? | Encodings specified? |
|---|---|---|---|---|
| CBEFF_BDB_format_owner | Mandatory (specified or inherited) if a BDB is present | `<FormatOwner>` child of `<BDBInfo>` | Yes | Yes |
| CBEFF_BDB_format_type | Mandatory (specified | `<FormatType>` child of | Yes | Yes |

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

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

4782

## 4783 B.22 XML schema of the BIAS patron format

```
4784    <?xml version='1.0' encoding="utf-8"?>
4785    <xs:schema
4786      xmlns:xs="http://www.w3.org/2001/XMLSchema"
4787      xmlns=" http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/"
4788      targetNamespace=" http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/"
4789      elementFormDefault="qualified">
4790
4791      <xs:element name="BIR" type="BIRType"/>
4792
4793      <xs:complexType name="BIRType">
4794        <xs:sequence>
4795          <xs:element name="Version" type="VersionType" minOccurs="0"/>
4796          <xs:element name="CBEFFVersion" type="VersionType" minOccurs="0"/>
4797          <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
4798          <xs:element name="BIRInfo" type="BIRInfoType"/>
4799          <xs:element name="BDBInfo" type="BDBInfoType" minOccurs="0"/>
4800          <xs:element name="SBInfo" type="SBInfoType" minOccurs="0"/>
4801          <xs:element name="BIR" type="BIRType" minOccurs="0" maxOccurs="unbounded"/>
4802          <xs:element name="BDB" type="xs:base64Binary" minOccurs="0"/>
4803          <xs:element name="SB" type="xs:base64Binary" minOccurs="0"/>
4804        </xs:sequence>
```

```
4805                    </xs:complexType>
4806
4807                    <xs:complexType name="VersionType">
4808                     <xs:sequence>
4809                       <xs:element name="Major" type="xs:nonNegativeInteger" />
4810                       <xs:element name="Minor" type="xs:nonNegativeInteger" />
4811                      </xs:sequence>
4812                    </xs:complexType>
4813
4814                    <xs:complexType name="BIRInfoType">
4815                     <xs:sequence>
4816                      <xs:element name="Creator" type="xs:string" minOccurs="0"/>
4817                      <xs:element name="Index" type="UUIDType" minOccurs="0"/>
4818                      <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
4819                      <xs:element name="Integrity" type="xs:boolean" />
4820                      <xs:element name="CreationDate" type="xs:dateTime" minOccurs="0" />
4821                      <xs:element name="NotValidBefore" type="xs:dateTime" minOccurs="0" />
4822                      <xs:element name="NotValidAfter" type="xs:dateTime" minOccurs="0" />
4823                      </xs:sequence>
4824                    </xs:complexType>
4825
4826                    <xs:complexType name="BDBInfoType">
4827                     <xs:sequence>
4828                      <xs:element name="ChallengeResponse" type="xs:base64Binary" minOccurs="0"/>
4829                      <xs:element name="Index" type="UUIDType" minOccurs="0"/>
4830                      <xs:element name="FormatOwner" type="xs:positiveInteger" minOccurs="0" />
4831                      <xs:element name="FormatType" type="xs:positiveInteger" minOccurs="0" />
4832                      <xs:element name="Encryption" type="xs:boolean" minOccurs="0" />
4833                      <xs:element name="CreationDate" type="xs:dateTime" minOccurs="0" />
4834                      <xs:element name="NotValidBefore" type="xs:dateTime" minOccurs="0" />
4835                      <xs:element name="NotValidAfter" type="xs:dateTime" minOccurs="0" />
4836                      <xs:element name="Type" type="MultipleTypesType" minOccurs="0" />
4837                      <xs:element name="Subtype" type="SubtypeType" minOccurs="0" />
4838                      <xs:element name="Level" type="ProcessedLevelType" minOccurs="0" />
4839                      <xs:element name="ProductOwner" type="xs:positiveInteger" minOccurs="0" />
4840                      <xs:element name="ProductType" type="xs:positiveInteger" minOccurs="0" />
4841                      <xs:element name="CaptureDeviceOwner" type="xs:positiveInteger" minOccurs="0" />
4842                      <xs:element name="CaptureDeviceType" type="xs:positiveInteger" minOccurs="0" />
4843                      <xs:element name="FeatureAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4844                      <xs:element name="FeatureAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4845                      <xs:element name="ComparisonAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4846                      <xs:element name="ComparisonAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4847                      <xs:element name="QualityAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4848                      <xs:element name="QualityAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4849                      <xs:element name="CompressionAlgorithmOwner" type="xs:positiveInteger" minOccurs="0" />
4850                      <xs:element name="CompressionAlgorithmType" type="xs:positiveInteger" minOccurs="0" />
4851                      <xs:element name="Purpose" type="PurposeType" minOccurs="0" />
4852                      <xs:element name="Quality" type="QualityType" minOccurs="0" />
4853                      </xs:sequence>
4854                    </xs:complexType>
4855
4856                    <xs:complexType name="SBInfoType">
4857                     <xs:sequence>
4858                      <xs:element name="FormatOwner" type="xs:positiveInteger" minOccurs="0" />
4859                      <xs:element name="FormatType" type="xs:positiveInteger" minOccurs="0" />
4860                      </xs:sequence>
4861                    </xs:complexType>
4862
4863                    <xs:simpleType name="QualityType">
4864                     <xs:restriction base="xs:integer">
4865                      <xs:minInclusive value="-2"/>
4866                      <xs:maxInclusive value="100"/>
4867                      </xs:restriction>
4868                    </xs:simpleType>
4869
4870                    <xs:simpleType name="SingleTypeType">
4871                     <xs:restriction base="xs:string">
4872                      <xs:enumeration value="Scent"/>
4873                      <xs:enumeration value="DNA"/>
4874                      <xs:enumeration value="Ear "/>
4875                      <xs:enumeration value="Face"/>
```

```
4876                    <xs:enumeration value="Finger"/>
4877                    <xs:enumeration value="Foot"/>
4878                    <xs:enumeration value="HandGeometry"/>
4879                    <xs:enumeration value="Vein"/>
4880                    <xs:enumeration value="Iris"/>
4881                    <xs:enumeration value="Retina"/>
4882                    <xs:enumeration value="Voice"/>
4883                    <xs:enumeration value="Gait"/>
4884                    <xs:enumeration value="Keystroke"/>
4885                    <xs:enumeration value="LipMovement"/>
4886                    <xs:enumeration value="SignatureSign"/>
4887                   </xs:restriction>
4888                  </xs:simpleType>
4889
4890                  <xs:simpleType name="MultipleTypesType">
4891                   <xs:list itemType="SingleTypeType"/>
4892                  </xs:simpleType>
4893
4894                  <xs:simpleType name="SingleAnySubtypeType">
4895                   <xs:restriction base="xs:string">
4896                    <xs:enumeration value="Left"/>
4897                    <xs:enumeration value="Right"/>
4898                    <xs:enumeration value="Thumb"/>
4899                    <xs:enumeration value="IndexFinger"/>
4900                    <xs:enumeration value="MiddleFinger"/>
4901                    <xs:enumeration value="RingFinger"/>
4902                    <xs:enumeration value="LittleFinger"/>
4903                   </xs:restriction>
4904                  </xs:simpleType>
4905
4906                  <xs:simpleType name="SingleVeinOnlySubtypeType">
4907                   <xs:restriction base="xs:string">
4908                    <xs:enumeration value="LeftVein"/>
4909                    <xs:enumeration value="RightVein"/>
4910                    <xs:enumeration value="Palm"/>
4911                    <xs:enumeration value="BackOfHand"/>
4912                    <xs:enumeration value="Wrist"/>
4913                    <xs:enumeration value="Reserved1"/>
4914                    <xs:enumeration value="Reserved2"/>
4915                   </xs:restriction>
4916                  </xs:simpleType>
4917
4918                  <xs:simpleType name="MultipleAnySubtypesType">
4919                   <xs:list itemType="SingleAnySubtypeType"/>
4920                  </xs:simpleType>
4921
4922                  <xs:simpleType name="MultipleVeinOnlySubtypesType">
4923                   <xs:list itemType="SingleVeinOnlySubtypeType"/>
4924                  </xs:simpleType>
4925
4926                  <xs:simpleType name="SubtypeType">
4927                   <xs:union memberTypes="MultipleAnySubtypesType MultipleVeinOnlySubtypesType"/>
4928                  </xs:simpleType>
4929
4930                  <xs:simpleType name="ProcessedLevelType">
4931                   <xs:restriction base="xs:string">
4932                    <xs:enumeration value="Raw"/>
4933                    <xs:enumeration value="Intermediate"/>
4934                    <xs:enumeration value="Processed"/>
4935                   </xs:restriction>
4936                  </xs:simpleType>
4937
4938                  <xs:simpleType name="PurposeType">
4939                   <xs:restriction base="xs:string">
4940                    <xs:enumeration value="Verify"/>
4941                    <xs:enumeration value="Identify"/>
4942                    <xs:enumeration value="Enroll"/>
4943                    <xs:enumeration value="EnrollVerify"/>
4944                    <xs:enumeration value="EnrollIdentify"/>
4945                    <xs:enumeration value="Audit"/>
4946                   </xs:restriction>
```

```
4947                    </xs:simpleType>
4948
4949                    <xs:simpleType name="UUIDType">
4950                     <xs:restriction base="xs:string">
4951                      <xs:pattern value="[a-fA-F0-9]{8}\-([a-fA-F0-9]{4}\-){3}[a-fA-F0-9]{12}"/>
4952                     </xs:restriction>
4953                    </xs:simpleType>
4954                  </xs:schema>
4955
```

4956    NOTE   NO VALUE AVAILABLE is encoded by the absence of optional fields in the XML
4957    encoding. There is little value in, for example, having the following string appear in a
4958    record: <level> no value available <level>.

4959

## 4960 B.23  Sample BIR encoding

4961 An example of a simple BIR in XML encoding (complying with the XSD schema and the normative textual
4962 description) follows.

4963

```
4964            <?xml version="1.0" encoding="utf-8"?>
4965            <BIR xmlns=" http://docs.oasis-open.org/bias/ns/biaspatronformat-1.0/">
4966                    <Version>
4967                            <Major>1</Major>
4968                            <Minor>0</Minor>
4969                    </Version>
4970                    <CBEFFVersion>
4971                            <Major>2</Major>
4972                            <Minor>0</Minor>
4973                    </CBEFFVersion>
4974                    <BIRInfo>
4975                            <Creator>ABCDE</Creator>
4976                            <Index>86CA3100-43F3-0D23-A941-7871E519A00E</Index>
4977                            <Payload>a2V2aW4ubWFuZ29sZEBuaXN0Lmdvdg==</Payload>
4978                            <Integrity>true</Integrity>
4979                            <CreationDate>2004-03-02T15:03:15Z</CreationDate>
4980                            <NotValidBefore>2004-03-02T15:00:00Z</NotValidBefore>
4981                            <NotValidAfter>2004-03-03T15:00:00Z</NotValidAfter>
4982                    </BIRInfo>
4983                    <BDBInfo>
4984
4985                    <ChallengeResponse>VmlzaXQgaHR0cDovL2J3cy5uaXN0Lmdvdi8iBmb3Igc29tZSBhd2Vzb21lIGJpb21ldH
4986            JpY3Mvd2ViHNIcnZpY2UgcHJvamVjdHMh</ChallengeResponse>
4987                            <Index>86CA3100-43F3-0D23-A941-7871E519A00E</Index>
4988                            <FormatOwner>51</FormatOwner>
4989                            <FormatType>88</FormatType>
4990                            <Encryption>true</Encryption>
4991                            <CreationDate>2004-03-02T15:00:00Z</CreationDate>
4992                            <NotValidBefore>2004-03-02T15:00:00Z</NotValidBefore>
4993                            <NotValidAfter>2004-03-02T15:00:00Z</NotValidAfter>
4994                            <Type>Iris</Type>
4995                            <Subtype>Left</Subtype>
4996                            <Level>Processed</Level>
4997                            <ProductOwner>16</ProductOwner>
```

```
4998                          <ProductType>2</ProductType>
4999                          <Purpose>Verify</Purpose>
5000                          <Quality>100</Quality>
5001                  </BDBInfo>
5002                  <SBInfo>
5003                          <FormatOwner>51</FormatOwner>
5004                          <FormatType>99</FormatType>
5005                  </SBInfo>
5006                  <BDB>a2V2aW4ubWFuZ29sZEBuaXN0Lmdvdg==</BDB>
5007                  <SB>TmF0aW9uYWwgSW5zdGl0dXRlIG9mIFN0YW5kYXJkcyBhbmQgVGVjaG5vbG9neQ==</SB>
5008          </BIR>
```

# Annex C.    Use Cases (non-normative)

5009

5010 The intent of this annex is to provide operational sequence diagrams / flow charts that show how the
5011 higher level usage scenarios within [INCITS-BIAS] could be implemented using the BIAS SOAP profile.
5012 The following use cases are given:

5013  • Verification (synchronous/aggregate)

5014  • Verification (asynchronous/aggregate)

5015  • Verification (primitive)

5016  • Identification (primitive)

5017  • Enrollment (aggregate)

5018  • Enrollment (primitive)

## C.1  Verification Use Case

5019

5020 This use case uses the aggregate Verify operation in which a single request results in some set of
5021 operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service
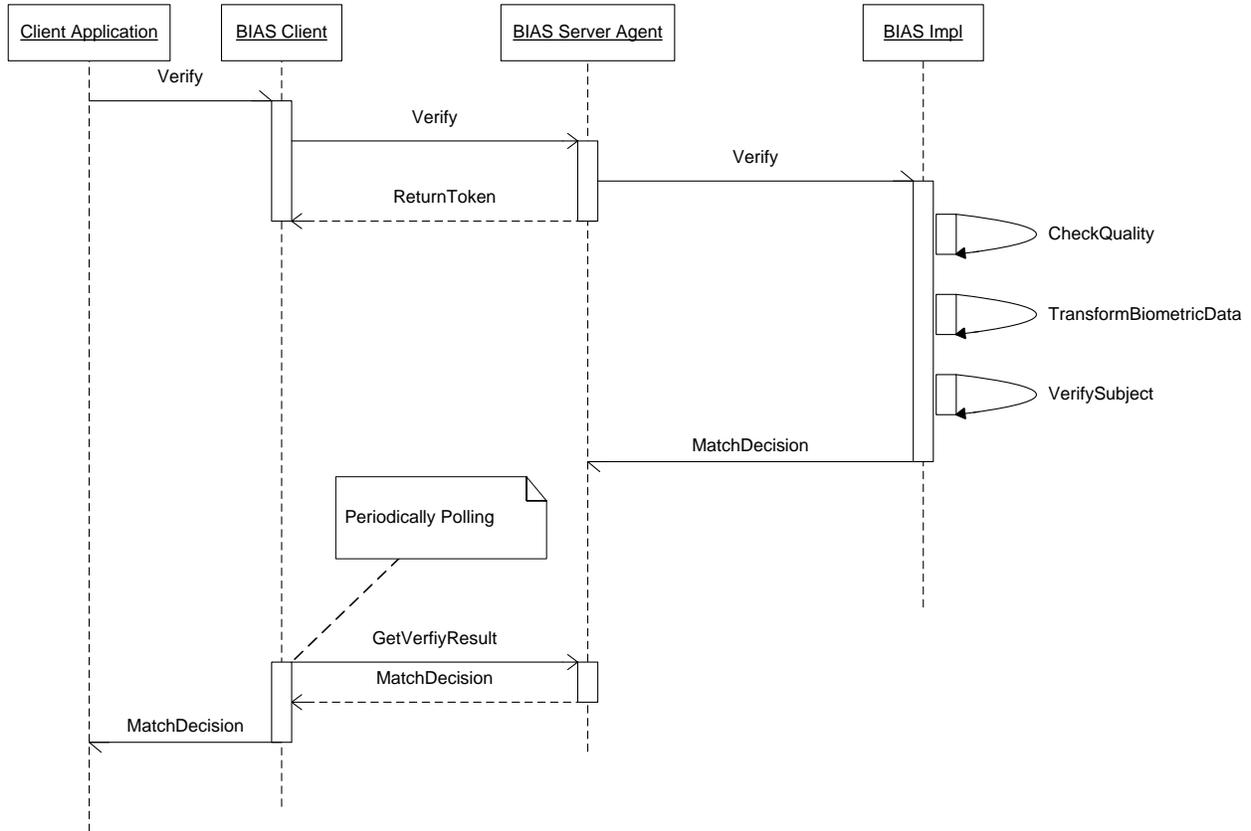5022 provider.

5023
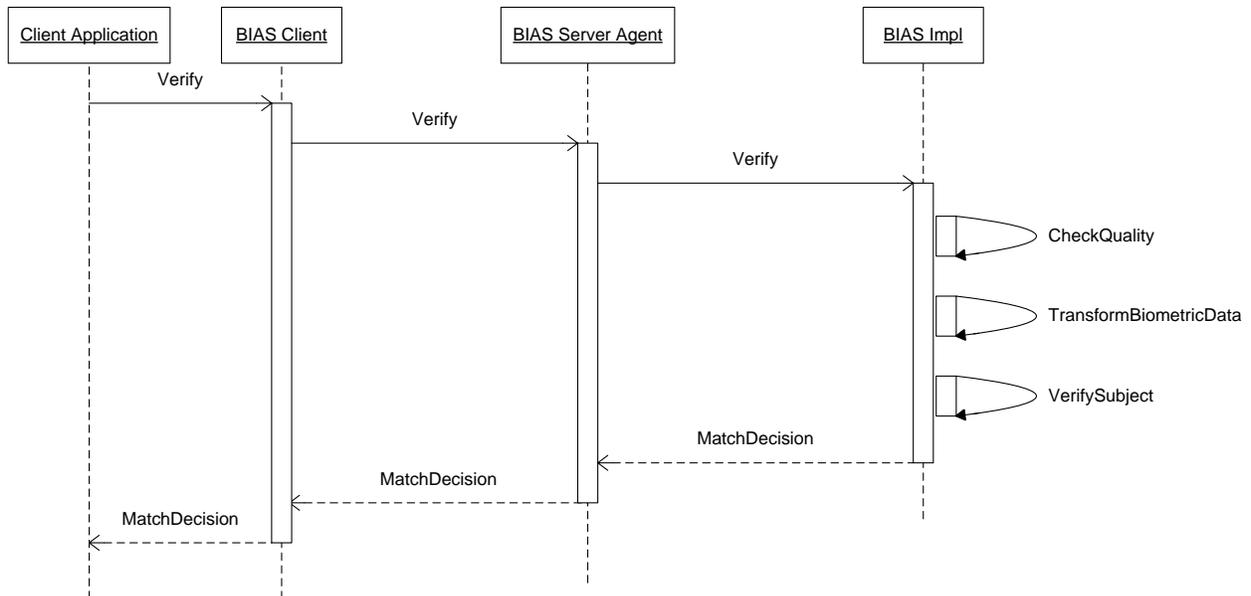


Note that
1. CheckQuality, TransformBiometricData, VerifySubject can be exposed as interfaces of BIAS server agent.

5024

5025

## C.2 Asynchronous Verification Use Case

5027 In this use case, the requester issues two requests – the BIAS Verify request to initiate the operation
5028 followed by a BIAS GetVerifyResult request to retrieve the results of that operation.
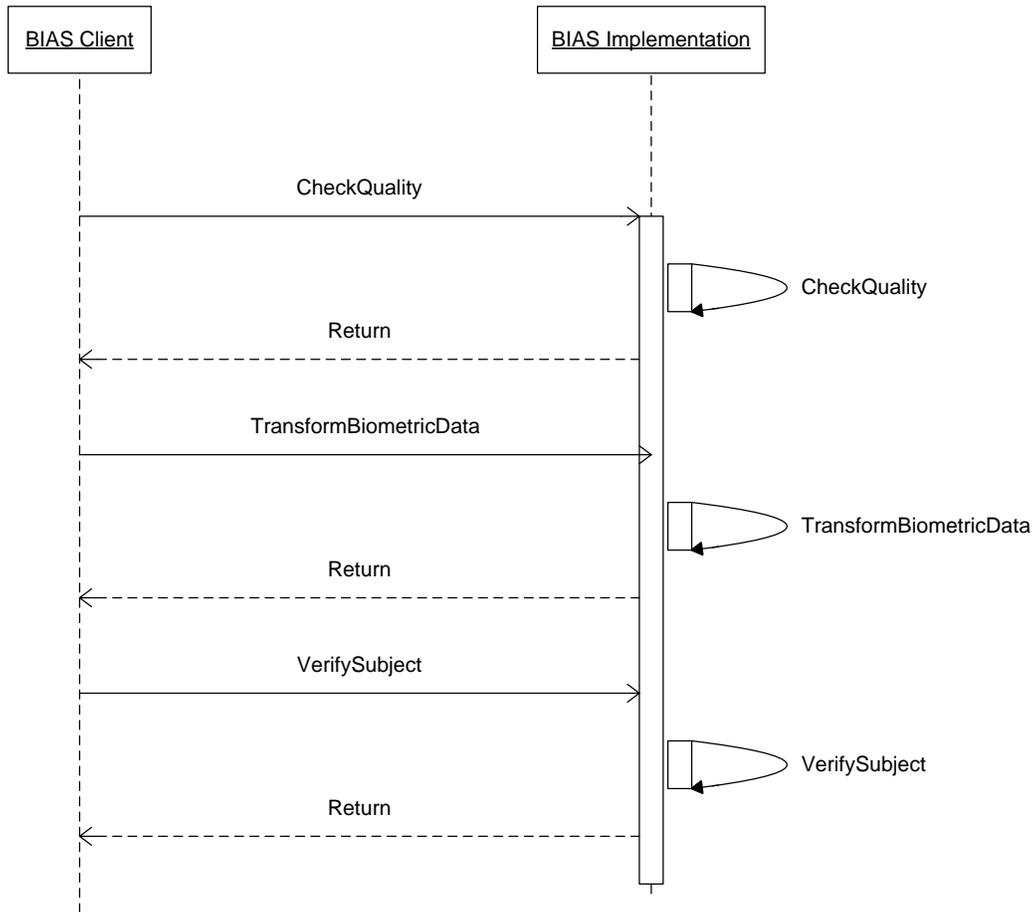
5029



5030
5031
5032

## C.3 Primitive Verification Use Case

5034 In this use case, the verification operation is performed as a series of requests using the BIAS primitive
5035 operations.  In this case, the client rather than the service provider controls the workflow of the higher
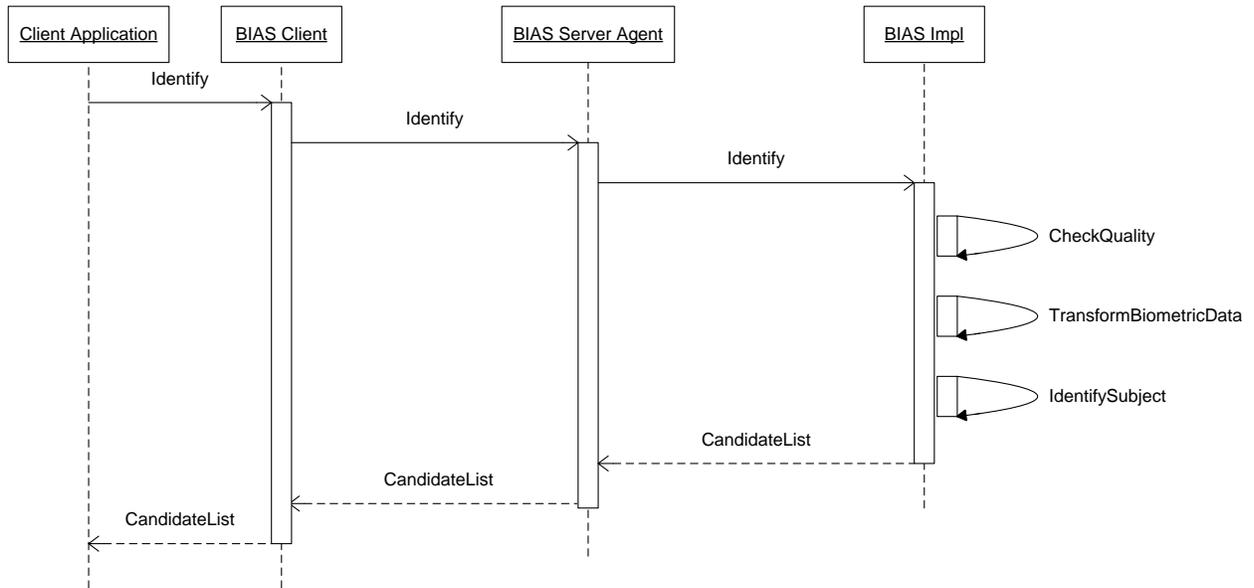5036 level operation.

5037



5038

5039

## C.4  Identification Use Case

5041 This use case uses the aggregate Identify operation in which a single request results in some set of
5042 operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service
5043 provider.

5044

5045

5046



```
Client Application    BIAS Client              BIAS Server Agent              BIAS Impl

        Identify
                    Identify
                                        Identify
                                                        CheckQuality

                                                        TransformBiometricData

                                                        IdentifySubject

                                        CandidateList
                    CandidateList
        CandidateList
```

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

5047

5048

## 5049 C.5 Biometric Enrollment Use Case

5050 This use case uses the aggregate Enroll operation in which a single request results in some set of
5051 operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service
5052 provider.

5053 Here, if the result of the IdentifySubject is no matches found, then the subject is added to the gallery. If a
5054 match had been found then other logic may have been applied (e.g., return candidate list, add encounter
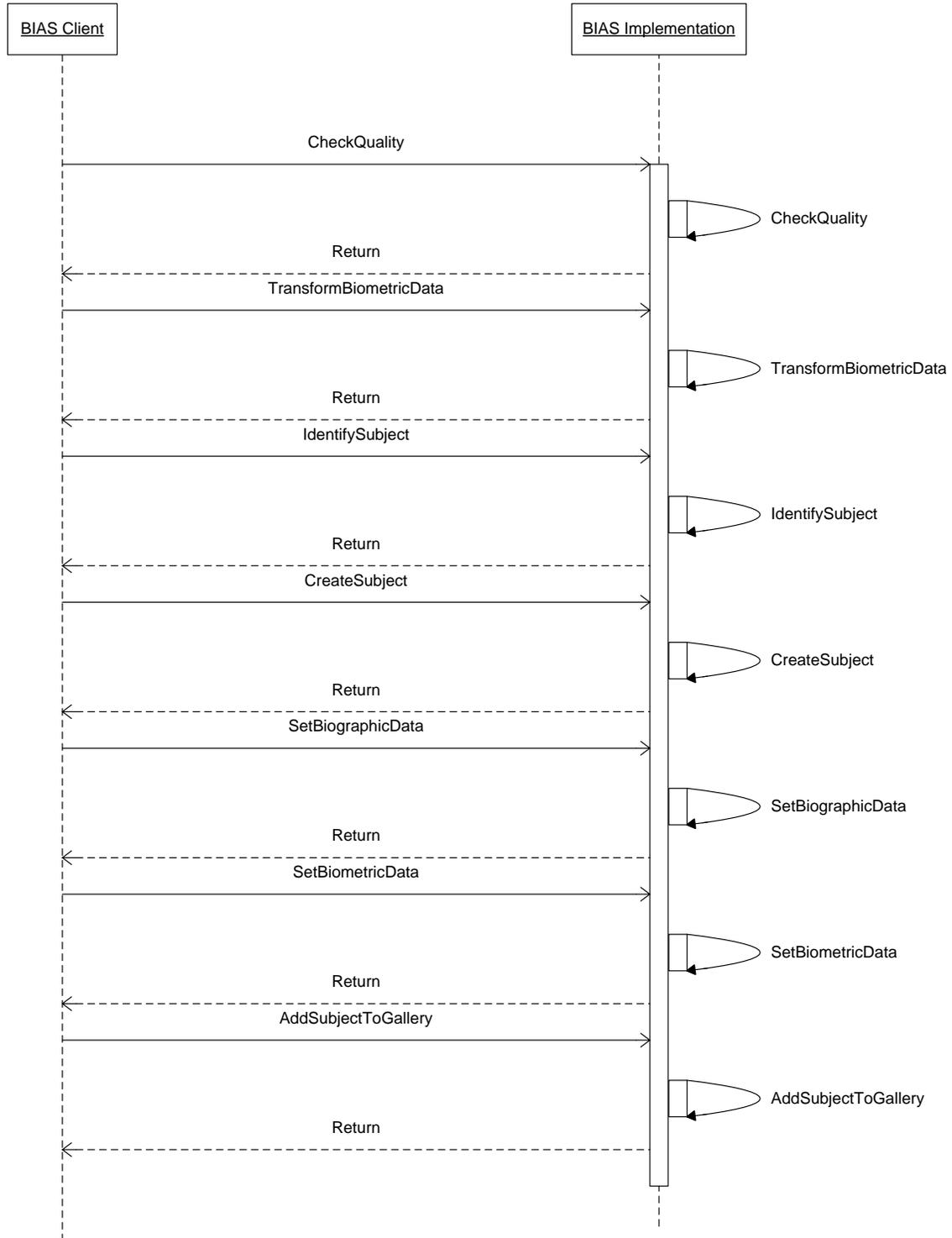5055 for existing subject, etc.).

5056
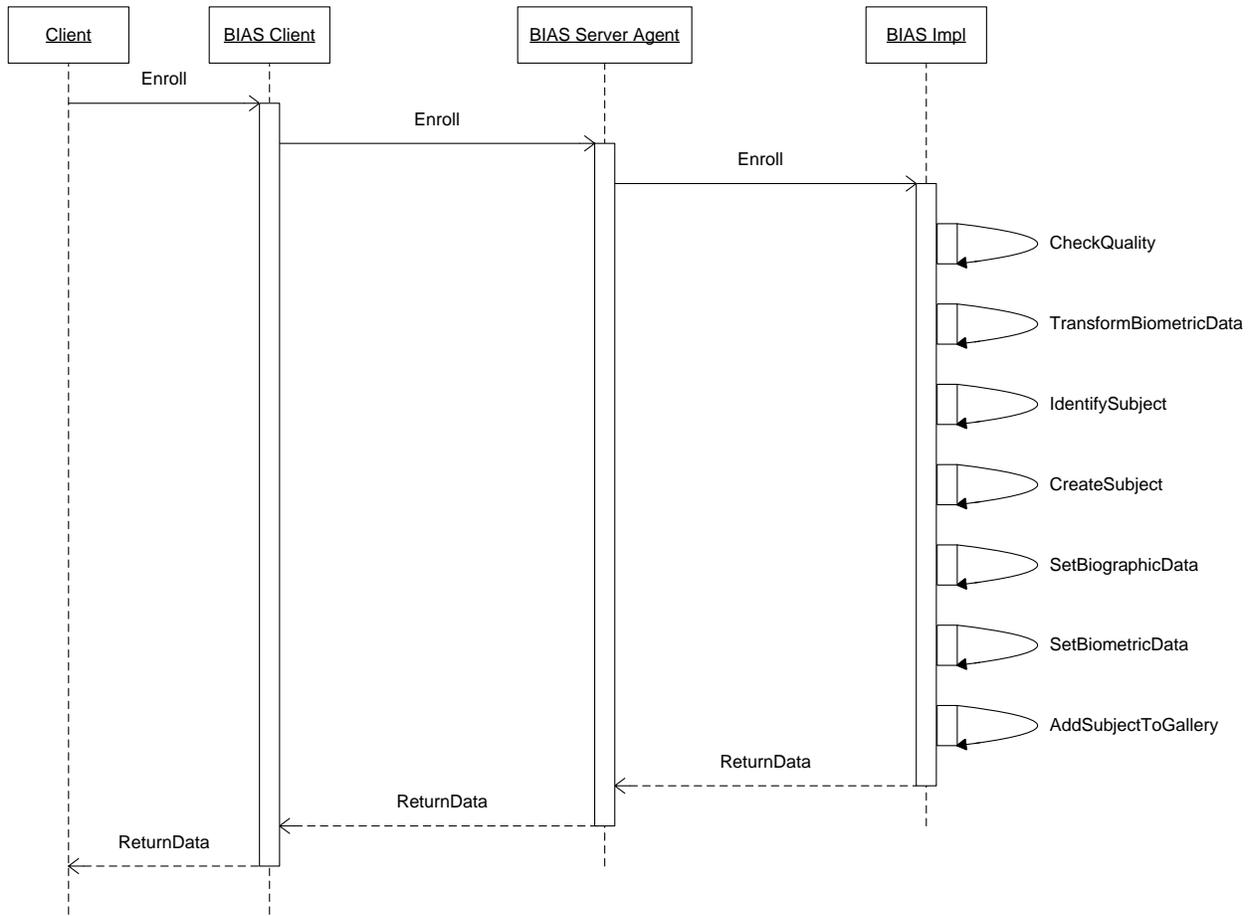


5057
5058
5059

## C.6 Primitive Enrollment Use Case

5061 In this use case, the enrollment operation is performed as a series of requests using the BIAS primitive
5062 operations.  In this case, the client rather than the service provider controls the workflow of the higher
5063 level operation.

5064



5065

# Annex D.    Samples (non-normative)

## D.1  Create Subject Request/Response Example

*INCITS BIAS Specification*

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

*OASIS BIAS Examples*

**Simple Create Subject Request:**

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

     xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/" />
     </soap:Body>
</soap:Envelope>
```

**Create Subject Request with SubjectID Parameter:**

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

```
5105    </soap:Envelope>
```

**5106** **Create Subject Request with Optional OASIS BIAS Content:**

```
5107    POST /bias HTTP/1.1
5108    Host: www.acme.com
5109    Content-Type: application/soap+xml; charset="utf-8"
5110    Content-Length: nnnn
5111    SOAPAction: "CreateSubject"
5112    <?xml version="1.0"?>
5113    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5114        <soap:Body>
5115            <tns:CreateSubject
5116    xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5117                <tns:GenericRequestParameters>
5118                    <tns:Application>BIAS Application</tns:Application>
5119                    <tns:ApplicationUser>BIAS User</tns:ApplicationUser>
5120                </tns:GenericRequestParameters>
5121                <tns:Identity>
5122                    <tns:SubjectID>123456789</tns:SubjectID>
5123                </tns:Identity>
5124            </tns:CreateSubject>
5125        </soap:Body>
5126    </soap:Envelope>
```

**5127** **Simple Create Subject Response:**

```
5128    HTTP/1.1 200 OK
5129    Content-Type: application/soap+xml; charset="utf-8"
5130    Content-Length: nnnn
5131    <?xml version="1.0"?>
5132    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5133        <soap:Body>
5134            <tns:CreateSubjectResponse
5135    xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5136                <tns:ResponseStatus>
5137                    <tns:Return>0</tns:Return>
5138                </tns:ResponseStatus>
5139                <tns:Identity>
5140                    <tns:SubjectID>123456789</tns:SubjectID>
5141                </tns:Identity>
5142            </tns:CreateSubjectResponse>
5143        </soap:Body>
5144    </soap:Envelope>
```

**5145** **Create Subject Response with Optional OASIS BIAS Content:**

```
5146    HTTP/1.1 200 OK
5147    Content-Type: application/soap+xml; charset="utf-8"
5148    Content-Length: nnnn
5149    <?xml version="1.0"?>
5150    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5151         <soap:Body>
5152              <tns:CreateSubjectResponse
5153    xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5154                   <tns:ResponseStatus>
5155                        <tns:Return>0</tns:Return>
5156                        <tns:Message>Subject ID 123456789 successfully
5157    created.</tns:Message>
5158                   </tns:ResponseStatus>
5159                   <tns:Identity>
5160                        <tns:SubjectID>123456789</tns:SubjectID>
5161                   </tns:Identity>
5162              </tns:CreateSubjectResponse>
5163         </soap:Body>
5164    </soap:Envelope>
5165
```

## 5166   D.2  Set Biographic Data Request/Response Example

5167   *INCITS BIAS Specification*

```
5168    <interface name="SetBiographicData">
5169         <parameter name="SubjectID" type="xs:string" direction="in" />
5170         <parameter name="IdentityModel" type="IdentityModelType"direction="in"
5171    />
5172         <parameter name="EncounterID"type="xs:string" direction="inout"
5173    use="optional" />
5174         <parameter name="BiographicData"
5175    type="BiographicDataType"direction="in" />
5176         <parameter name="Return" type="xs:unsignedLong" direction="out" />
5177    </interface>
```

5178   *OASIS BIAS Examples*

5179   **Set Biographic Data Request:**

```
5180    POST /bias HTTP/1.1
5181    Host: www.acme.com
5182    Content-Type: application/soap+xml; charset="utf-8"
5183    Content-Length: nnnn
5184    SOAPAction: "SetBiographicData"
5185    <?xml version="1.0"?>
5186    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5187         <soap:Body>
```

```
5188            <tns:SetBiographicData

5189

5190    xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5191                <tns:Identity>
5192                    <tns:SubjectID>123456789></tns:SubjectID>
5193                    <tns:BiographicData>
5194                        <tns:BiographicDataItem>
5195                            <tns:Name>Last</tns:Name>
5196                            <tns:Type>string</tns:Type>
5197                            <tns:Value>Doe</tns:Value>
5198                        </tns:BiographicDataItem>
5199                    </tns:BiographicData>
5200                </tns:Identity>
5201                <tns:IdentityModel>person</tns:IdentityModel>
5202            </tns:SetBiographicData>
5203        </soap:Body>
5204    </soap:Envelope>
```

5205 **Set Biographic Data Response:**

```
5206    HTTP/1.1 200 OK
5207    Content-Type: application/soap+xml; charset="utf-8"
5208    Content-Length: nnnn
5209    <?xml version="1.0"?>
5210    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5211        <soap:Body>
5212            <tns:SetBiographicDataResponse

5213

5214    xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5215                <tns:ResponseStatus>
5216                    <tns:Return>0</tns:Return>
5217                </tns:ResponseStatus>
5218            </tns:SetBiographicDataResponse>
5219        </soap:Body>
5220    </soap:Envelope>
```

5221

## 5222 D.3 Set Biometric Data Request/Response Example

5223 *INCITS BIAS Specification*

```
5224    <interface name="SetBiometricData">
5225        <parameter name="SubjectID" type="xs:string" direction="in" />
5226        <parameter name="IdentityModel" type="IdentityModelType"direction="in"
5227    />
5228        <parameter name="EncounterID"type="xs:string" direction="inout"
5229    use="optional" />
```

```
5230            <parameter name="BIRList"type="CBEFF_BIR_ListType" direction="in" />
5231            <parameter name="Return" type="xs:unsignedLong" direction="out" />
5232    </interface>
```

5233    *OASIS BIAS Examples*

5234    **Set Biometric Data Request:**

```
5235    POST /bias HTTP/1.1
5236    Host: www.acme.com
5237    Content-Type: application/soap+xml; charset="utf-8"
5238    Content-Length: nnnn
5239    SOAPAction: "SetBiometricData"
5240    <?xml version="1.0"?>
5241    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5242         <soap:Body>
5243                <tns:SetBiometricData
5244
5245         xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5246                    <tns:Identity>
5247                         <tns:SubjectID>123456789></tns:SubjectID>
5248                         <tns:BiometricData>
5249                              <tns:BIRList>
5250                                    <tns:BIR>biometric data</tns:BIR>
5251                              </tns:BIRList>
5252                         </tns:BiometricData>
5253                    </tns:Identity>
5254                    <tns:IdentityModel>person</tns:IdentityModel>
5255              </tns:SetBiometricData>
5256         </soap:Body>
5257    </soap:Envelope>
```

5258    **Set Biometric Data Response:**

```
5259    HTTP/1.1 200 OK
5260    Content-Type: application/soap+xml; charset="utf-8"
5261    Content-Length: nnnn
5262    <?xml version="1.0"?>
5263    <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
5264         <soap:Body>
5265                <tns:SetBiometricDataResponse
5266
5267         xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
5268                    <tns:ResponseStatus>
5269                         <tns:Return>0</tns:Return>
5270                    </tns:ResponseStatus>
5271              </tns:SetBiometricDataResponse>
```

```
5272          </soap:Body>
5273     </soap:Envelope>
```

# Annex E.    Acknowledgements

5275 The following individuals have participated in the creation of this specification and are gratefully
5276 acknowledged:

5277

5278 **Participants:**

5279

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

5280

5281 * Though no longer members of the BIAS TC at time of publication, these individuals contributed in the
5282 early stages of the development of this standard.

5283 In addition, the inputs from the INCITS technical committee M1 are also gratefully appreciated.

# Annex F. Revision History

5284

5285

| Revision | Date | Editor | Changes Made |
|---|---|---|---|
| 0.01 | 2008-05-23 | TBD | Initial draft |
| 0.02 | 2008-07-23 | TBD | Inserted data dictionary<br>Added normative references<br>Updated sec 3 & 5 + Annex B |
| 0.03 | 2008-08-19 | TBD | WSDL updated |
| 0.04 | 2008-09-11 | TBD | Updated references<br>Added security requirements<br>Corrected Fig. 3 |
| 0.05 | 2008-09-29 | TBD | SSL/TLS requirement clarified<br>Reordered material in 5.3 & App C/D<br>Updated references<br>2 new use cases added (App C)<br>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:<br>- Breaking Clause 3 into 2 clauses for data elements and operations<br>- Specification of URI & IRI<br>- 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<br>[Also published as CD01] |
| 0.12 | 2010-11-04 | Mangold/Tilton/Swayze | Incorporation of public review comments<br>Update WSDL |
| 0.13 | 2011-01-03 | Tilton/Mangold | Clarification regarding xsd:any<br>Updated WSDL |
| 0.14 | 2011-06-15 | Mangold/Tilton | Inserted new Annex B – CBEFF Patron Format miscellaneous editorial changes |

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

5286