

**InterNational Committee for Information Technology Standards (INICTS)  
INCITS Secretariat, Information Technology Industry Council (ITI)  
1250 Eye St. NW, Room 200, Washington, DC 20005  
Telephone 202-737-8888; Fax 202-638-4922  
email: [incits@itic.org](mailto:incits@itic.org)**

**Document:** M1/07-0360  
**Date:** July 18, 2007  
**Reply to:** Matt Swayze  
**Phone:** 703-984-4004 / 240-418-6195  
**Email:** [matthew.swayze@daon.com](mailto:matthew.swayze@daon.com)

**BIAS, Revision 6**

## INCITS Project 1823-D

InterNational Committee for Information Technology Standards (INCITS)  
INCITS Secretariat, Information Technology Industry Council (ITI)  
1250 Eye St. NW, Suite 200, Washington, DC 20005  
Telephone 202-737-8888; Fax 202-638-4922  
email: [incits@itic.org](mailto:incits@itic.org)

**Title:** Biometric Identity Assurance Services (BIAS)

**Source:** INCITS M1

**Date:** July 18, 2007

**Revision:** 6

Revision	Date	M1 Document #	Comments
0	February 7, 2006	M1/06-0127	Base document
1	May 18, 2006	M1/06-0432	First draft
2	August 28, 2006	M1/06-0648	Second draft
3	November 8, 2006	M1/06-0888	Third draft
4	December 15, 2006	M1/06-1071	Fourth draft, 1 <sup>st</sup> Letter Ballot text
5	April 13, 2007	M1/07-0198	Fifth draft, 2 <sup>nd</sup> Letter Ballot text
6	July 18, 2007	M1/07-0360	Sixth draft, Public Review text

**Project Editor:**

Matt Swayze

Daon, Inc.

[matthew.swayze@daon.com](mailto:matthew.swayze@daon.com)

703-984-4004 / 240-418-6195

**Contents**

**Foreword** ..... iv

**Introduction** ..... v

**1 Scope** ..... 1

**2 Conformance** ..... 1

**3 Normative References** ..... 1

**4 Terms and Definitions** ..... 2

4.1 Biometric Sample ..... 2

4.2 Claim to Identity ..... 2

4.3 Encounter ..... 2

4.4 Encounter-Centric ..... 2

4.5 Gallery ..... 2

4.6 Identification ..... 2

4.7 Identity Assurance ..... 3

4.8 Person-Centric ..... 3

4.9 Subject ..... 3

4.10 Verification ..... 3

**5 Symbols and Abbreviated Terms** ..... 3

**6 System Context** ..... 3

6.1 Service-Oriented Architectures ..... 3

6.2 BIAS Architecture ..... 5

6.3 BIAS Implementation Considerations ..... 6

**7 Biometric Identity Assurance Services** ..... 8

7.1 BIAS Interface XML Schema ..... 8

7.2 Primitive Services ..... 9

7.2.1 Add Subject To Gallery ..... 9

7.2.2 Check Quality ..... 10

7.2.3 Classify Biometric Data ..... 11

7.2.4 Create Subject ..... 11

7.2.5 Delete Biographic Data ..... 12

7.2.6 Delete Biometric Data ..... 12

7.2.7 Delete Subject ..... 13

7.2.8 Delete Subject From Gallery ..... 13

7.2.9 Get Identify Subject Results ..... 14

7.2.10 Identify Subject ..... 14

7.2.11 List Biographic Data ..... 15

7.2.12 List Biometric Data ..... 16

7.2.13 Perform Fusion ..... 17

7.2.14 Query Capabilities ..... 18

7.2.15	Retrieve Biographic Information .....	24
7.2.16	Retrieve Biometric Information .....	24
7.2.17	Set Biographic Data .....	25
7.2.18	Set Biometric Data .....	26
7.2.19	Transform Biometric Data.....	26
7.2.20	Update Biographic Data .....	27
7.2.21	Update Biometric Data .....	28
7.2.22	Verify Subject .....	28
7.3	Aggregate Services .....	29
7.3.1	Enroll .....	29
7.3.2	Get Enroll Results .....	30
7.3.3	Get Identify Results .....	31
7.3.4	Get Verify Results .....	31
7.3.5	Identify.....	32
7.3.6	Retrieve Information .....	33
7.3.7	Verify .....	34
<b>8</b>	<b>Data Elements and Data Types .....</b>	<b>35</b>
8.1	Biographic Data.....	35
8.1.1	Biographic Data Type.....	35
8.1.2	Biographic Data Item Type.....	36
8.1.3	Biographic Data Set Type.....	36
8.2	Biometric Data.....	38
8.2.1	CBEFF BIR Type.....	38
8.2.2	CBEFF BIR List Type .....	40
8.2.3	Biometric Data Element Type.....	40
8.2.4	Biometric Data List Type .....	41
8.3	Candidate Lists.....	42
8.3.1	Candidate Type .....	42
8.3.2	Candidate List Type .....	42
8.4	Capabilities.....	43
8.4.1	Capability Type.....	43
8.4.2	Capability List Type .....	44
8.5	Fusion Information.....	44
8.5.1	Fusion Information Type.....	44
8.5.2	Fusion Information List Type .....	45
8.6	Other Data Types .....	45
8.6.1	Encounter List Type .....	46
8.6.2	Information Type .....	46
8.6.3	Identity Model Type .....	46
8.6.4	List Filter Type.....	47
8.6.5	Processing Options Type .....	47
8.6.6	Token Type .....	48
<b>9</b>	<b>Error Handling and Notification .....</b>	<b>48</b>
9.1	Successful Service Calls .....	48

9.2 Error Condition Codes..... 48

**10 Security..... 50**

**Annex A: Conformance Requirements ..... 51**

**Annex B: Bibliography ..... 58**

**Annex C: Example Usage Scenarios..... 59**

## Foreword

INCITS (The InterNational Committee for Information Technology Standards) is the ANSI recognized Standards Development Organization for information technology within the United States of America. Members of INCITS are drawn from Government, Corporations, Academia and other organizations with a material interest in the work of INCITS and its Technical Committees. INCITS does not restrict membership and attracts participants in its technical work from 13 different countries, and operates under the rules of the American National Standards Institute.

In the field of Biometrics, INCITS has established the Technical Committee M1. Standards developed by this Technical Committee have reached consensus throughout the development process and have been thoroughly reviewed through several Public Review processes. In addition, the INCITS Executive Board and the ANSI Board of Standards Review have approved this American National Standard for Publication as an INCITS Standard.

## Introduction

Biometric technologies are being used today in a wide variety of applications and environments. At the same time, enterprises – both commercial and government – have been moving towards services-based architectures as the framework for their enterprise infrastructures. As biometrics become a larger part of the greater identity assurance capability, the need to access these services remotely across those services-oriented frameworks will become necessary.

A current gap exists in standards related to the use of biometric technology in a Service-Oriented Architecture (SOA). The Biometric Identity Assurance Services (BIAS) standard is intended to fill that gap by defining a framework for deploying and invoking biometrics-based identity assurance capabilities that can be readily accessed using services (e.g., Web services).

Development of this standard necessarily requires expertise in two distinct technology domains – biometrics and service architectures. The two standards organizations that are the leaders in these areas are INCITS and the Organization for the Advancement of Structured Information Standards (OASIS) respectively. The work has been partitioned between the two organizations such that INCITS develops an INCITS standard for biometric services and OASIS develops an OASIS standard for the Web services integration. These two standards will be separate but interrelated.

The BIAS standard will help ensure biometric-based solutions are robust and maintainable, while providing a mechanism for accessing an organization's biometric services. BIAS should significantly increase the functional opportunities for implementing identity related functions in a services-oriented framework, allowing for platform and application independence. Presently-developed SOA methods for exchanging information, transactions, and security data should provide useful methods, constraints, and patterns for the broader and more robust use of BIAS data. This standard is intended to have the following characteristics:

- Focused on biometrics (though not exclusively)
- Biometric device, type, and vendor independent
- Leverage existing standards where appropriate
- Multi-platform, open
- Primarily focused on remote invocations (services) (i.e., not dealing with local devices).





## 1 Scope

BIAS defines biometric services used for identity assurance that are invoked over a services-based framework. It is intended to provide a generic set of biometric and identity-related functions and associated data definitions to allow remote access to biometric services.

The binding of these services to specific frameworks is not included in this project, but will be the subject of separate standards. The first such standard (for a Web services framework) is planned to be developed by OASIS by the BIAS Integration Technical Committee.

Although focused on biometrics, this standard will necessarily include support for other related identity assurance mechanisms such as biographic and token capabilities. BIAS is intended to be compatible with and used in conjunction with other biometric standards as described in clause 3.

Specification of single-platform biometric functionality (e.g., client-side capture, etc.) is not within the scope of this standard.

Integration of biometric services as part of an authentication service or protocol is not within the scope of this standard; however, it is possible that some of the basic biometric services defined herein may be used by such an implementation in the future.

## 2 Conformance

Annex A specifies the conformance requirements for systems/components claiming conformance to this standard.

## 3 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- OASIS BIAS Messaging Protocol
- ANSI/INCITS 398-2005 Revision 1, American National Standard – Information technology – Common Biometric Exchange Formats Framework (CBEFF)
- ISO/IEC 19784-1:2006, Information Technology – Biometric Application Programming Interface – Part 1: BioAPI Specification

- ISO/IEC 19785-1:2006, Information Technology – Common Biometric Exchange Formats Framework – Part 1: Data Element Specification
- ISO/IEC 19785-2:2006, Information Technology – Common Biometric Exchange Formats Framework – Part 2: Procedures for the Operation of the Biometric Registration Authority

## **4 Terms and Definitions**

For the purposes of this document, the following terms and definitions apply.

### **4.1 Biometric Sample**

Analog or digital representation of biometric characteristics (prior to biometric feature extraction process and obtained from a biometric capture device or biometric capture subsystem)

### **4.2 Claim to Identity**

Assertion that an individual is or is not the source of a specified or unspecified biometric reference in an identity assurance system; also called “biometric claim”.

### **4.3 Encounter**

An interaction with a subject. Each encounter may contain unique information collected during the encounter and/or describing the encounter.

### **4.4 Encounter-Centric**

A system that supports encounter processing, maintaining a one-to-many relationship between subjects and encounters, and which does not necessarily contain a single, unique set of information for each subject.

### **4.5 Gallery**

A group of subjects, related by a common purpose, designation, or status. For example: a watch list, or a set of subjects entitled to a certain benefit.

### **4.6 Identification**

A biometric system function that performs a one-to-many search, in which a biometric sample(s) from one individual is compared against the biometric references of many individuals to return the identifiers of those with a specified degree of similarity.

#### 4.7 Identity Assurance

The process of establishing, determining, and/or confirming a subject identity.

#### 4.8 Person-Centric

A system that maintains a single, unique view of a subject, and which does not support encounter processing.

#### 4.9 Subject

A person who is known to an identity assurance system.

#### 4.10 Verification

A biometric system function that performs a one-to-one comparison, in which a biometric sample(s) from one individual is compared to biometric reference(s) from one individual to produce a comparison score

### 5 Symbols and Abbreviated Terms

AFIS	Automated Fingerprint Identification System
BIAS	Biometric Identity Assurance Services
BIR	Biometric Information Record
CBEFF	Common Biometric Exchange Formats Framework
ESB	Enterprise Service Bus
ID	Identity/Identification/Identifier
OASIS	Organization for the Advancement of Structured Information Standards
SOA	Service-Oriented Architecture

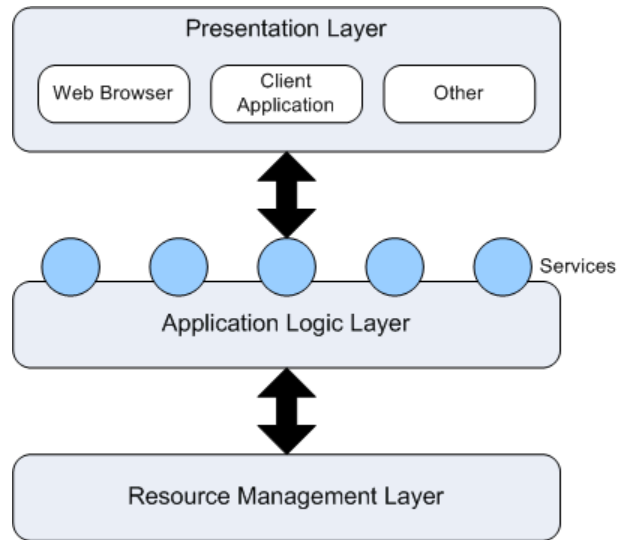
### 6 System Context

This section provides an overview of Service-Oriented Architectures, the BIAS architecture, and BIAS implementation considerations.

#### 6.1 Service-Oriented Architectures

Service-Oriented Architectures are software architectures in which reusable services are deployed onto application servers and then consumed by clients in different applications or business processes. They are intended to decouple the implementation of a software service from the interface that calls that service. This allows clients of a service to rely on a consistent interface regardless of the implementation technology of the service [JDJ].

Biometric services are one of the types of services that can be provided over such a remote interface in a distributed information system across a collection of networks. This can occur in a 2-tier, 3-tier, or N-tier environment. A diagram of a simple N-tier architecture is shown in Figure 6-1, below.



*Figure 6-1. Simple N-tier Architecture*

In this simple diagram, BIAS services are defined between the application logic layer and the resource management layer.

Examples of biometric resources that are of interest may include one or more of the following:

- A fingerprint verification matching server
- A 1:N iris search/match engine
- A facial biometric watch list
- A criminal or civil automated fingerprint identification system (AFIS)
- A name-based biographic identity database
- An archive of biometric identifiers
- A population of subjects.

It is desired that a generic set of services be defined that allows clients to remotely access and manage these capabilities. To the extent possible, domain specific implementations are to be avoided.

NOTE: This standard is intended to support a wide variety of application domains which may include government (e.g., background checking, border management, and criminal justice), enterprise (e.g., logical access control), and commercial biometric identity management implementations (e.g., employee databases).

Services are well defined, self-contained modules that provide standard business functionality and are independent of the state or context of other services and that can be easily assembled to form a collection of autonomous and loosely-coupled business processes.

It is not the intention that specific business logic be instantiated within the service definitions – this logic is more appropriate within the application logic layer – either in the higher level system initiating the series of requests, or within the middleware (e.g., an enterprise service bus [ESB], workflow manager, or biometric middleware) as appropriate. To do so would of necessity make the interface less generic, modular, and flexible and require that the interface be updated each time the logic changed, defeating one of the primary purposes of the services architecture.

The services to be defined are not targeted at a particular SOA implementation or framework. Instead, they are defined in such a manner as to be able to be utilized within any such architecture. This is accomplished by separately defining (in another standard) the bindings to that architecture/implementation. For example, Web services bindings are defined in the OASIS BIAS Messaging Protocol.

## 6.2 BIAS Architecture

The BIAS architecture consists of the following components:

- BIAS services (interface definition)
- BIAS data (schema definition)
- BIAS bindings (defined outside this standard).

The BIAS services expose a common set of operations to external requesters of these operations. These requesters may be an external system, a Web application, or an intermediary. The BIAS services themselves are platform and language independent. The BIAS services may be implemented with differing technologies on multiple platforms. For example, OASIS is defining Web services bindings for the BIAS services.

Figure 6-2 depicts the BIAS services within an application environment. BIAS services provide basic biometric functionality as modular and independent operations which can be assembled in many different ways to perform and/or support a variety of business processes. BIAS services can be publicly exposed directly and/or utilized indirectly in support of a service-provider's own public services.

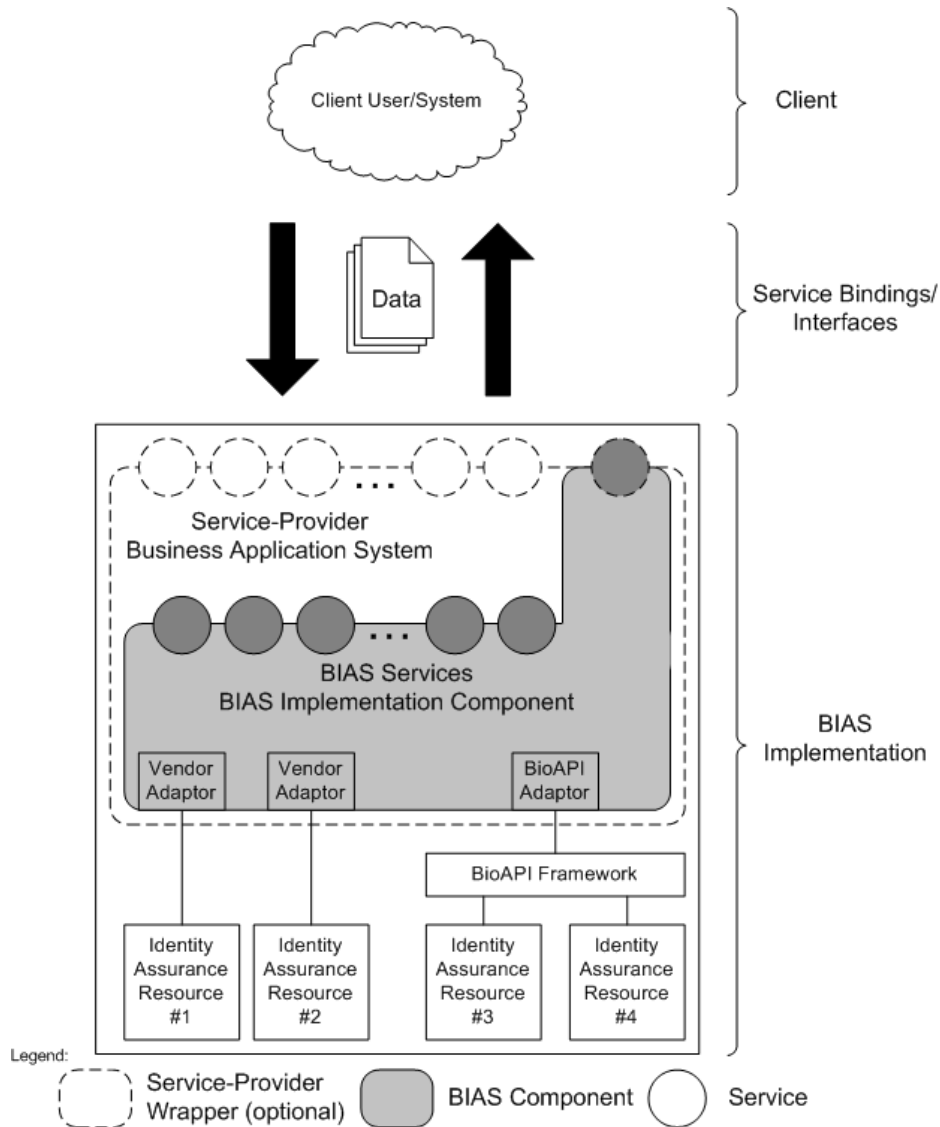


Figure 6-2. BIAS Application Environment

### 6.3 BIAS Implementation Considerations

Biometric services and the applications which use them, particularly in an identity assurance context, have unique characteristics which are summarized below:

- Some services can be performed very quickly while others (such as a 1:N identification within a large population) can take considerable time (on the order of hours) to complete. Therefore, the interface should support both synchronous and asynchronous operations.
- Biometric operations may be singular or multi-biometric.

- Some systems are person-centric and others are encounter-centric. That is, some base transactions on a unique identifier associated with an individual human being while others track “biometric encounters” which may or may not be linked through such an identifier. Figure 6-3 provides context to further explain these concepts.

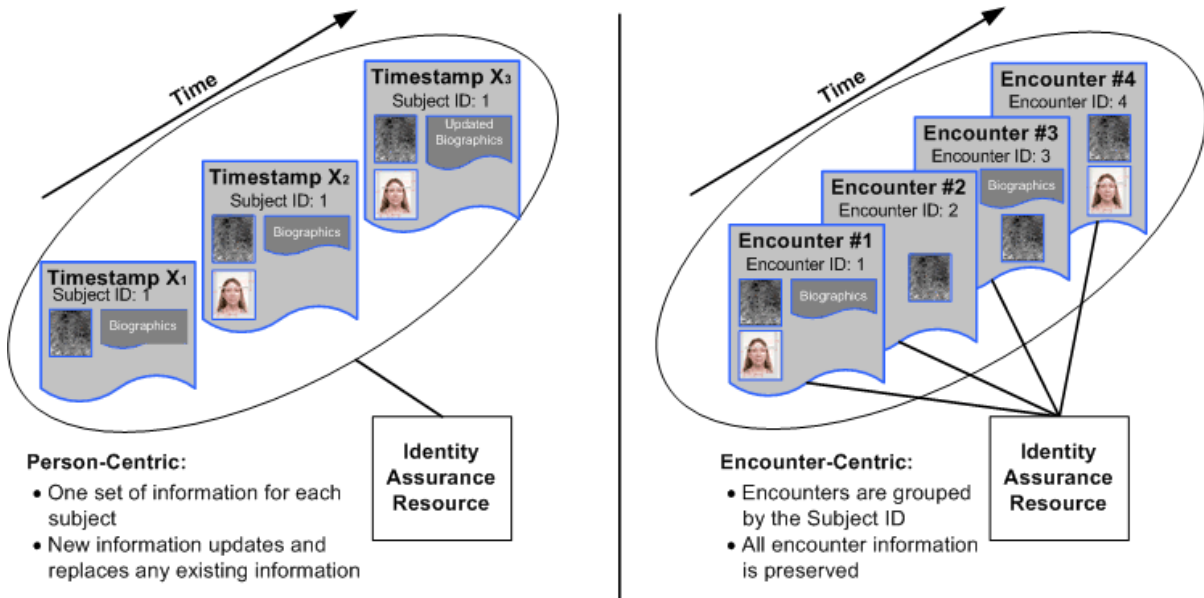


Figure 6-3. Person-Centric and Encounter-Centric Views

- Biometric data is in nearly all cases considered personal information and thus privacy protection is always a consideration.
  - Before a biometric and/or biographic data transaction occurs between two different entities, the terms and conditions of the use of the data should be negotiated and made transparent. The following questions may be addressed:
    - Who will be the recipient of the data to be shared?
    - For what purpose(s) can the recipient use this data?
    - Who/what authorizes this data to be shared with the recipient for this purpose?
    - How long may the recipient retain the data?
    - How must data be destroyed at the end of a retention period?
    - May the recipient share this data with other entities? If so, with whom? For what purpose(s)? How long may the third party retain the data?
  - A recipient would later have to understand what they are accepting and the terms and conditions of the agreement.
- Security is important for any kind of SOA. As in the case of privacy protection, before a biometric and/or biographic data transaction occurs between two

different entities, the security characteristics provided by the BIAS implementation should be known.

- For the purposes of data integrity and quality assurance, a capability for creation of a chain of custody should be created to track events, changes, and transfers of data. For example, the deletion of data should be tracked and logged.
- Methods for service level monitoring and compliance with agreed-upon service level agreements may be critical to requesters and implementers. Terms and conditions for these capabilities may need to be negotiated.

## 7 Biometric Identity Assurance Services

This section defines two categories of BIAS services, primitive and aggregate. Primitive services are lower-level operations that are used to request a specific capability. Aggregate services operate at a higher-level, performing a sequence of primitive operations in a single request. (An example of such a sequence would be a negative search where a 1:N identification which results in a 'no match' is immediately followed by the addition of the biometric sample into that search population.)

### 7.1 BIAS Interface XML Schema

A goal for this BIAS Standard is to be as language and protocol independent as possible. To that end, the services are specified using a simple XML schema as defined below.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="BIAS_Interface"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:complexType name="InterfaceType">
    <xs:sequence>
      <xs:element name="parameter" type="ParameterType"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required" />
  </xs:complexType>
  <xs:complexType name="ParameterType">
    <xs:attribute name="name" type="xs:string" use="required" />
    <xs:attribute name="type" type="xs:string" use="required" />
    <xs:attribute name="direction" type="DirectionType"
      use="required" />
    <xs:attribute name="use" type="UseType" use="optional"
      default="required" />
  </xs:complexType>
  <xs:simpleType name="DirectionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="in" />
      <xs:enumeration value="out" />
      <xs:enumeration value="inout" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UseType">
    <xs:restriction base="xs:string">
```



```

    <xs:enumeration value="required" />
    <xs:enumeration value="optional" />
    <xs:enumeration value="conditional" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="interface" type="InterfaceType"></xs:element>
</xs:schema>

```

Each service is identified by an *<interface>* tag and must include a *name* attribute. Service parameters are identified by a *<parameter>* tag and must include a *name*, *type*, and *direction* attribute. The *direction* attribute specifies whether the parameter is an input parameter (*in*), an output parameter (*out*), or an input/output parameter (*inout*). Parameters may also include a *use* attribute to indicate if the parameter is required, optional, or conditional. If the parameter is conditional, the service description must identify the conditions.

## 7.2 Primitive Services

BIAS specifies the following set of primitive services.

### 7.2.1 Add Subject To Gallery

```

<interface name="AddSubjectToGallery">
  <parameter name="GalleryID" type="xs:string" direction="in" />
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="IdentityClaim"
    type="xs:string" direction="in" use="optional" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>

```

#### 7.2.1.1 Description

The Add Subject To Gallery service shall register a subject to a given gallery or population group. As an optional parameter, the value of the claim to identity by which the subject is known to the gallery may be specified. This claim to identity shall be unique across the gallery. If no claim to identity is specified, the subject ID (assigned with the Create Subject service) shall be used as the claim to identity. Additionally, in the encounter-centric model, the encounter ID associated with the subject's biometrics that will be added to the gallery shall be specified.

#### 7.2.1.2 Parameters

*Gallery ID (input)* – the identifier of the gallery or population group to which the subject will be added

*Subject ID (input)* – the identifier of the subject

*Identity Claim (input, optional)* – the identifier by which the subject is known to the gallery

*Encounter ID (input, conditional)* – the identifier of the encounter, required for encounter-centric models

*Return (output)* – return value indicating success or specifying a particular error condition

## 7.2.2 Check Quality

```
<interface name="CheckQuality">
  <parameter name="BIR" type="CBEFF BIR Type" direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="QualityScore" type="xs:int" direction="out" />
  <parameter name="AlgorithmVendor"
    type="xs:string" direction="inout" use="optional" />
  <parameter name="AlgorithmVendorProductID"
    type="xs:string" direction="inout" use="conditional" />
  <parameter name="AlgorithmVersion"
    type="xs:string" direction="out" />
</interface>
```

### 7.2.2.1 Description

The Check Quality service shall return a quality score for a given biometric. The biometric input is provided in a CBEFF basic structure or CBEFF record, which in this standard 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 shall provide the algorithm vendor and algorithm vendor product ID that were used to calculate the biometric quality as output parameters.

### 7.2.2.2 Parameters

*BIR (input)* – data structure containing a single biometric sample for which a quality score is to be determined

*Return (output)* – return value indicating success or specifying a particular error condition

*Quality Score (output)* – the quality of the biometric

*Algorithm Vendor (input, optional / output)* – the vendor of the quality algorithm used to determine the quality

*Algorithm Vendor Product ID (input, conditional / output)* – the vendor's ID for the algorithm used to determine the quality; required as input if algorithm vendor is provided

*Algorithm Version (output)* – the version of the algorithm used to determine the quality

### 7.2.3 Classify Biometric Data

```
<interface name="ClassifyBiometricData">
  <parameter name="BIR" type="CBEFF BIR Type" direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="ClassificationAlgorithmType" type="xs:string"
    direction="out" />
  <parameter name="Classification" type="xs:string" direction="out" />
</interface>
```

#### 7.2.3.1 Description

The Classify Biometric Data service shall attempt 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). The types of classification algorithms and classes are not specified here, rather they are left for the implementing system to define.

#### 7.2.3.2 Parameters

*BIR (input)* – data structure containing a single biometric sample for which the classification is to be determined

*Return (output)* – return value indicating success or specifying a particular error condition

*Classification Algorithm Type (output)* – identifies the type of classification algorithm that was used to perform the classification (e.g., for fingerprints, Henry classification)

*Classification (output)* – the result of the classification

### 7.2.4 Create Subject

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

#### 7.2.4.1 Description

The Create Subject service shall create a new subject record and associate a subject ID to that record. As an optional parameter, the subject ID may be specified by the caller. If no subject ID is specified, the Create Subject service shall generate one.

#### 7.2.4.2 Parameters

*Subject ID (input/output, optional)* – the identifier of the subject

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.5 Delete Biographic Data

```
<interface name="DeleteBiographicData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.5.1 Description

The Delete Biographic Data service shall erase all of the biographic data associated with a given subject record. In the encounter-centric model the service shall erase all of the biographic data associated with a given encounter, and therefore the encounter ID shall 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.

#### 7.2.5.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, conditional)* – the identifier of the encounter, required for encounter-centric models

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.6 Delete Biometric Data

```
<interface name="DeleteBiometricData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.6.1 Description

The Delete Biometric Data service shall remove biometric data from a given subject record. In the encounter-centric model, the encounter ID shall 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.

### 7.2.6.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, conditional)* – the identifier of the encounter, required for encounter-centric models

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.7 Delete Subject

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

#### 7.2.7.1 Description

The Delete Subject service shall delete an existing subject record and, in an encounter-centric model, any associated encounter information from the system. This service shall also remove the subject from any registered galleries. When deleting a subject, BIAS implementations may completely erase the subject information in order to prevent the ability to reconstruct a record or records in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

#### 7.2.7.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.8 Delete Subject From Gallery

```
<interface name="DeleteSubjectFromGallery">
  <parameter name="GalleryID" type="xs:string" direction="in" />
  <parameter name="SubjectID" type="xs:string" direction="in"
    use="conditional" />
  <parameter name="IdentityClaim"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.8.1 Description

The Delete Subject From Gallery service shall remove the registration of a subject from a gallery or population group. The subject shall be identified by either the

subject ID or the claim to identity that was specified in the Add Subject To Gallery service.

### 7.2.8.2 Parameters

*Gallery ID (input)* – the identifier of the gallery or population group from which the subject will be deleted

*Subject ID (input, conditional)* – the identifier of the subject; required if an identity claim is not provided

*Identity Claim (input, conditional)* – the identifier by which the subject is known to the gallery; required if a subject ID is not provided

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.9 Get Identify Subject Results

```
<interface name="GetIdentifySubjectResults">
  <parameter name="Token" type="TokenType" direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="CandidateList" type="CandidateListType"
    direction="out" />
</interface>
```

#### 7.2.9.1 Description

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

#### 7.2.9.2 Parameters

*Token (input)* – a value used to retrieve the results of the Identify Subject request

*Return (output)* – return value indicating success or specifying a particular error condition

*Candidate List (output)* – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample

### 7.2.10 Identify Subject

```
<interface name="IdentifySubject">
  <parameter name="GalleryID" type="xs:string" direction="in" />
  <parameter name="BIR" type="CBEFF BIR Type" direction="in" />
  <parameter name="MaxListSize" type="xs:int" direction="in" />
</interface>
```

```

<parameter name="Return" type="xs:unsignedLong" direction="out" />
<parameter name="CandidateList" type="CandidateListType"
  direction="out" use="conditional" />
<parameter name="Token" type="TokenType"
  direction="out" use="conditional" />
</interface>

```

### 7.2.10.1 Description

The Identify Subject service shall perform an identification search against a given gallery for a given biometric, returning a rank-ordered candidate list of a given maximum size.

If the Identify Subject service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the candidate list. If the Identify Subject service is implemented as an asynchronous service, the implementing system shall return a token, which is an indication that the request is being handled asynchronously. In this case, the Get Identify Subject Results service shall be used to poll for the results of the Identify Subject request.

### 7.2.10.2 Parameters

*Gallery ID (input)* – the identifier of the gallery or population group which will be searched; this parameter may also be used to identify an external system where the identification request should be forwarded, if this capability is supported by the implementing system

*BIR (input)* – data structure containing the biometric sample for the search

*Max List Size (input)* – the maximum size of the candidate list that should be returned

*Return (output)* – return value indicating success or specifying a particular error condition

*Candidate List (output, conditional)* – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample; returned with successful, synchronous request processing

*Token (output, conditional)* – a token used to retrieve the results of the Identify Subject request; returned with asynchronous request processing

### 7.2.11 List Biographic Data

```

<interface name="ListBiographicData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="BiographicDataElements" type="BiographicDataType"

```

```

        direction="out" use="conditional" />
        <parameter name="EncounterList" type="EncounterListType"
            direction="out" use="conditional" />
    </interface>

```

### 7.2.11.1 Description

The List Biographic Data service shall list the biographic data elements stored for a subject using the Biographic Data Elements output parameter. In the encounter-centric model, an encounter ID may be specified to indicate that only the biographic data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain biographic data using the Encounter List output parameter, and the Biographic Data Elements output parameter shall be empty.

### 7.2.11.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, optional)* – the identifier of the encounter

*Return (output)* – return value indicating success or specifying a particular error condition

*Biographic Data Elements (output, conditional)* – 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

*Encounter List (output, conditional)* – 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

### 7.2.12 List Biometric Data

```

<interface name="ListBiometricData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="ListFilter"
    type="ListFilterType" direction="in" use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="BiometricDataList" type="BiometricDataListType"
    direction="out" use="conditional" />
  <parameter name="EncounterList" type="EncounterListType"
    direction="out" use="conditional" />
</interface>

```



### 7.2.12.1 Description

The List Biometric Data service shall list the biometric data elements stored for a subject using the Biometric Data List output parameter. Note that no actual biometric data is returned by this service (see the Retrieve Biometric Information service to obtain the biometric data). In the encounter-centric model, an encounter ID may be specified to indicate that only the biometric data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain biometric data using the Encounter List output parameter, and the Biometric Data List output parameter shall be empty.

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

### 7.2.12.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, optional)* – the identifier of the encounter

*List Filter (input, optional)* – indicates what biometric information should be returned

*Return (output)* – return value indicating success or specifying a particular error condition

*Biometric Data List (output, conditional)* - a list of biometric data associated with a subject or encounter; non-empty if the service was successful, biometric 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

*Encounter List (output, conditional)* – 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

### 7.2.13 Perform Fusion

```
<interface name="PerformFusion">
  <parameter name="FusionInput" type="FusionInformationListType"
    direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="Match" type="xs:boolean" direction="out" />
</interface>
```

### 7.2.13.1 Description

The Perform Fusion service shall accept either match score or match decision information and create a fused match result. The FusionInformationListType, through the FusionInformationType, provides specific elements for match score input and match decision input (see clause 8.5). The fusion method and processes are left to the implementing system.

### 7.2.13.2 Parameters

*Fusion Input (input)* – score or decision input information to the fusion method

*Return (output)* – return value indicating success or specifying a particular error condition

*Match (output)* – indicates the result of the fusion method

## 7.2.14 Query Capabilities

```
<interface name="QueryCapabilities">
  <parameter name="CapabilityList" type="CapabilityListType"
    direction="out" />
</interface>
```

### 7.2.14.1 Description

The Query Capabilities service shall return a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation. Table 7.1 provides a list of capabilities. Refer to Annex A for conformance requirements regarding which capability names an implementation must use in the Query Capabilities service.

Table 7.1 – List of Capability Items

Capability Name	Capability Description
AggregateInputDataOptional	<p>A capability information item shall be provided for each data element accepted as optional input by the implementing system for the aggregate services.</p> <p>The Capability Name shall be set to “AggregateInputDataOptional”.</p> <p>The Capability Value shall be equal to the name of the data element accepted by the aggregate services.</p> <p>The Capability Supporting Value shall indicate which aggregate services support the data element, using one or more of the following values, each separated by a comma:</p> <ul style="list-style-type: none"> <li>• “Enroll”</li> <li>• “Identify”</li> <li>• “Verify”</li> <li>• “Retrieve”</li> <li>• “All”</li> </ul>
AggregateInputDataRequired	<p>A capability information item shall be provided for each data element required as input by the implementing system for the aggregate services.</p> <p>The Capability Name shall be set to “AggregateInputDataRequired”.</p> <p>The Capability Value shall be equal to the name of the data element required by the aggregate services.</p> <p>The Capability Supporting Value shall indicate which aggregate services support the data element, using one or more of the following values, each separated by a comma:</p> <ul style="list-style-type: none"> <li>• “Enroll”</li> <li>• “Identify”</li> <li>• “Verify”</li> <li>• “Retrieve”</li> <li>• “All”</li> </ul>

Capability Name	Capability Description
AggregateProcessingOption	<p>A capability information item shall be provided for each processing option supported by the implementing system for the aggregate services.</p> <p>The Capability Name shall be set to "AggregateProcessingOption".</p> <p>The Capability Value shall be equal to the value for the Processing Option parameter in the aggregate services.</p> <p>The Capability Supporting Value shall indicate which aggregate services support the processing option, using one or more of the following values, each separated by a comma:</p> <ul style="list-style-type: none"> <li>• "Enroll"</li> <li>• "Identify"</li> <li>• "Verify"</li> <li>• "Retrieve"</li> <li>• "All"</li> </ul>
AggregateReturnData	<p>A capability information item shall be provided for each data element returned by the implementing system for the aggregate services.</p> <p>The Capability Name shall be set to "AggregateReturnData".</p> <p>The Capability Value shall be equal to the name of the data element returned by the aggregate services.</p> <p>The Capability Supporting Value shall indicate which aggregate services support the data element, using one or more of the following values, each separated by a comma:</p> <ul style="list-style-type: none"> <li>• "Enroll"</li> <li>• "Identify"</li> <li>• "Verify"</li> <li>• "Retrieve"</li> <li>• "All"</li> </ul>

Capability Name	Capability Description
AggregateServiceDescription	<p>A capability information item shall be provided for each aggregate service supported by the implementing system, describing the processing logic of the aggregate system.</p> <p>The Capability Name shall be set to "AggregateServiceDescription".</p> <p>The Capability Value shall contain a description of the processing logic of the aggregate services.</p> <p>The Capability Supporting Value shall indicate which aggregate service is described by this capability information item, using one of the following values:</p> <ul style="list-style-type: none"> <li>• "Enroll"</li> <li>• "Identify"</li> <li>• "Verify"</li> <li>• "Retrieve"</li> </ul>
BiographicDataSet	<p>A capability information item shall be provided to identify the biographic data sets supported by the implementing system.</p> <p>The Capability Name shall be set to "BiographicDataSet".</p> <p>The Capability Value shall contain the name of the supported biographic data format (e.g., "EFTS" or "NIEM").</p> <p>The Capability Supporting Value shall contain the version of the supported biographic data format.</p>
CBEFFPatronFormat	<p>A capability information item shall be provided for each patron format supported by the implementing system.</p> <p>The Capability Name shall be set to "CBEFFPatronFormat".</p> <p>The Capability Value shall contain the format owner.</p> <p>The Capability Supporting Value shall contain the format type.</p>

Capability Name	Capability Description
ClassificationAlgorithmType	<p>A capability information item shall be provided for each classification algorithm type supported by the implementing system.</p> <p>The Capability Name shall be set to "ClassificationAlgorithmType".</p> <p>The Capability Value shall be set to the name of the supported classification algorithm type.</p>
ConformanceClass	<p>A capability information item shall be provided to identify the conformance class of the BIAS implementation (see Annex A for more information).</p> <p>The Capability Name shall be set to "ConformanceClass".</p> <p>The Capability Value shall be set to one of the following:</p> <ul style="list-style-type: none"> <li>• "1" (for Class 1 conformance)</li> <li>• "2" (for Class 2 conformance)</li> <li>• "3" (for Class 3 conformance)</li> <li>• "4" (for Class 4 conformance)</li> <li>• "5" (for Class 5 conformance)</li> </ul>
Gallery	<p>A capability information item shall be provided for each gallery or population ground supported by the implementing system.</p> <p>The Capability Name shall be set to "Gallery".</p> <p>The Capability Value shall be equal to the value for the Gallery ID parameter in the Add Subject to Gallery, Delete Subject From Gallery, Identify Subject, and Verify Subject services.</p>
IdentityModel	<p>A capability information item shall be provided to identify whether the implementing system is person-centric or encounter-centric based.</p> <p>The Capability Name shall be set to "IdentityModel".</p> <p>The Capability Value shall be set to one of the following:</p> <ul style="list-style-type: none"> <li>• "person"</li> <li>• "encounter"</li> </ul>

Capability Name	Capability Description
QualityAlgorithm	<p>A capability information item shall be provided for each quality algorithm vendor and algorithm vendor product ID supported by the implementing system.</p> <p>The Capability Name shall be set to "QualityAlgorithm".</p> <p>The Capability Value shall contain the algorithm vendor.</p> <p>The Capability Supporting Value shall contain the algorithm vendor product ID.</p>
SupportedBiometric	<p>A capability information item shall be provided for each biometric type supported by the implementing system.</p> <p>The Capability Name shall be set to "SupportedBiometric".</p> <p>The Capability Value shall be set to the biometric type, as defined by CBEFF. The CBEFF hexadecimal value must be literally transformed to a string (for example, the biometric type for finger 0x00000008 should be represented as the string "00000008").</p> <p>The Capability Supporting Value shall indicate if the implementing system supports matching for the biometric type, using one of the following values:</p> <ul style="list-style-type: none"> <li>• "1" (identification)</li> <li>• "2" (verification)</li> <li>• "3" (identification and verification)</li> <li>• "4" (no matching supported)</li> </ul>
TransformOperation	<p>A capability information item shall be provided for each transform operation type supported by the implementing system.</p> <p>The Capability Name shall be set to "TransformOperation".</p> <p>The Capability Value shall be equal to the value for the Transform Operation parameter in the Transform Biometric Data service.</p>

**7.2.14.2 Parameters**

*Capability List* – a list of capabilities supported by the BIAS implementation

## 7.2.15 Retrieve Biographic Information

```
<interface name="RetrieveBiographicInformation">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="BiographicData" type="BiographicDataType"
    direction="out" />
</interface>
```

### 7.2.15.1 Description

The Retrieve Biographic Information service shall retrieve the biographic data associated with a subject ID. In the encounter-centric model, the encounter ID may be specified and the service shall return the biographic data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the service shall return the biographic information associated with the most recent encounter.

### 7.2.15.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, optional)* – the identifier of the encounter

*Return (output)* – return value indicating success or specifying a particular error condition

*Biographic Data (output)* – a list of biographic data elements associated with the subject or encounter

## 7.2.16 Retrieve Biometric Information

```
<interface name="RetrieveBiometricInformation">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="BiometricType"
    type="xs:string" direction="in" use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="BIRList"
    type=" CBEFF BIR ListType " direction="out" />
</interface>
```

### 7.2.16.1 Description

The Retrieve Biometric Information service shall retrieve the biometric data associated with a subject ID. In the encounter-centric model, the encounter ID may be specified and the service shall return the biometric data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the service shall return the biometric information associated with the most recent



encounter. The service provides an optional input parameter to specify that only biometric data of a certain type should be retrieved.

### 7.2.16.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, optional)* – the identifier of the encounter

*Biometric Type (input, optional)* – the type of biological or behavioral data to retrieve

*Return (output)* – return value indicating success or specifying a particular error condition

*BIR List (output)* – data structure containing the retrieved biometric samples

### 7.2.17 Set Biographic Data

```
<interface name="SetBiographicData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="IdentityModel" type="IdentityModelType"
    direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="inout" use="optional" />
  <parameter name="BiographicData" type="BiographicDataType"
    direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.17.1 Description

The Set Biographic Data service shall associate biographic data to a given subject record. An input flag shall indicate whether the biographic information should replace any existing biographic information (person-centric model) or if a new encounter should be created and associated with the subject (encounter-centric model). For encounter-centric models, the encounter ID may be specified by the caller in order to link biographic and biometric information (assuming biometric information was previously associated using the Set Biometric Data service). If the encounter ID is omitted for the encounter-centric model, the service shall return a system-assigned encounter ID.

#### 7.2.17.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Identity Model (input)* – indicates a person-centric or encounter-centric model

*Encounter ID (input/output, optional)* – the identifier of the encounter

*Biographic Data (input)* – a list of biographic data to associate with the subject or encounter

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.18 Set Biometric Data

```
<interface name="SetBiometricData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="IdentityModel" type="IdentityModelType"
    direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="inout" use="optional" />
  <parameter name="BIRList"
    type="CBEFF BIR ListType" direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.18.1 Description

The Set Biometric Data service shall associate biometric data to a given subject record. The identity model parameter shall indicate 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 Set Biographic Data service). If the encounter ID is omitted for the encounter-centric model, the service shall return a system-assigned encounter ID.

#### 7.2.18.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Identity Model (input)* – indicates a person-centric or encounter-centric model

*Encounter ID (input/output, optional)* – the identifier of the encounter

*BIR List (input)* – data structure containing the new biometric sample(s)

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.19 Transform Biometric Data

```
<interface name="TransformBiometricData">
  <parameter name="InputBIR" type="CBEFF BIR Type" direction="in" />
  <parameter name="TransformOperation"
    type="xs:unsignedLong" direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

```
<parameter name="OutputBIR" type="CBEFF BIR Type" direction="out" />
</interface>
```

### 7.2.19.1 Description

The Transform Biometric Data service shall transform or process a given biometric in one format into a new target format. Examples of transformations include:

- Feature Extraction
- Centering or cropping biometric images
- Standard biometric data format conversion

### 7.2.19.2 Parameters

*Input BIR (input)* – data structure containing the biometric information to be transformed

*Transform Operation (input)* – value indicating the type of transformation to perform

*Return (output)* – return value indicating success or specifying a particular error condition

*Output BIR (output)* – data structure containing the new, transformed biometric information

## 7.2.20 Update Biographic Data

```
<interface name="UpdateBiographicData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="BiographicData" type="BiographicDataType"
    direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

### 7.2.20.1 Description

The Update Biographic Data service shall update the biographic data for an existing subject record. The service shall replace any existing biographic data with the new biographic data. In the encounter-centric model, the encounter ID shall be specified.

### 7.2.20.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, conditional)* – the identifier of the encounter, required for encounter-centric models

*Biographic Data (input)* – list of updated biographic data elements

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.21 Update Biometric Data

```
<interface name="UpdateBiometricData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="Merge"
    type="xs:boolean" direction="in" use="optional" />
  <parameter name="BIR" type="CBEFF BIR Type" direction="in" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.21.1 Description

The Update Biometric Data service shall update a single biometric sample for an existing subject record. The service includes an optional parameter indicating if the new biometric sample should be merged with the existing biometric sample. If this parameter is set to “False” or is not used in the service request, the service shall replace the existing biometric sample with the new biometric sample. In the encounter-centric model, the encounter ID shall be specified. In the person-centric model, an input flag shall indicate if the input biometric data should either replace or be merged with the existing biometric data.

#### 7.2.21.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Encounter ID (input, conditional)* – the identifier of the encounter, required for encounter-centric models

*Merge (input, optional)* – value indicating if the input biometric sample should be merged with any existing biometric information

*BIR (input)* – data structure containing the new biometric sample

*Return (output)* – return value indicating success or specifying a particular error condition

### 7.2.22 Verify Subject

```
<interface name="VerifySubject">
  <parameter name="InputBIR" type="CBEFF_BIR_Type" direction="in" />
  <parameter name="GalleryID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="IdentityClaim"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="ReferenceBIR"
    type="CBEFF_BIR_Type" direction="in" use="conditional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

```
<parameter name="Match" type="xs:boolean" direction="out" />
<parameter name="Score" type="xs:int" direction="out" />
</interface>
```

### 7.2.22.1 Description

The Verify Subject service shall perform a 1:1 verification match between a given biometric and either a claim to identity in a given gallery or another given biometric. As such either the Identity Claim or Reference BIR input parameters are required.

### 7.2.22.2 Parameters

*Input BIR (input)* – data structure containing the biometric sample for the search

*Gallery ID (input, optional)* – the identifier of the gallery or population group of which the subject must be a member

*Identity Claim (input, conditional)* – the identifier by which the subject is known to the gallery, required if no Reference BIR is provided

*Reference BIR (input, conditional)* – data structure containing the biometric sample that will be compared to the Input BIR, required if no Identity Claim is provided

*Return (output)* – return value indicating success or specifying a particular error condition

*Match (output)* – indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR

*Score (output)* – the score if the biometric information matched

## 7.3 Aggregate Services

BIAS offers the following set of aggregate services. The intent of BIAS is to standardize the service request; system requirements and organizational business rules will determine how the service is implemented. While the description for an aggregate service may provide examples of how each one may be implemented using the primitive services defined in clause 7.2, service providers are not required to utilize any of the primitive services when implementing the aggregate services.

### 7.3.1 Enroll

```
<interface name="Enroll">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="InputData" type="InformationType" direction="in">
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="ReturnData" type="InformationType" direction="out">
</interface>
```

### 7.3.1.1 Description

The Enroll aggregate service shall add a new subject or, in an encounter-centric model, a new encounter to the system. This may be accomplished in a number of different ways according to system requirements and/or resources. For example, this aggregate service may initiate one or more Identify Subject primitive service requests to determine if the given subject is already known to the system. If the subject is not previously known to the system, any or all of the Create Subject, Set Biographic Data, Set Biometric Data, and Add Subject to Gallery primitive services may be utilized to add subject information to the system. If the subject is previously known to the system, the service may (1) do nothing; (2) initiate an Update Biographic Data and/or Update Biometric Data primitive service request in a person-centric model; or (3) initiate a Set Biographic Data and/or Set Biometric Data primitive service request in an encounter-centric model.

If the Enroll aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the Enroll aggregate service is implemented as an asynchronous service, the implementing system shall return a token in the Return Data parameter, which is an indication that the request is being handled asynchronously. In this case, the Get Enroll Results service shall be used to poll for the results of the Enroll request.

### 7.3.1.2 Parameters

*Processing Options (input)* – options that guide how the service request is processed

*Input Data (input)* – contains a subject enrollment record

*Return (output)* – return value indicating success or specifying a particular error condition

*Return Data (output)* – contains a return data record

## 7.3.2 Get Enroll Results

```
<interface name="GetEnrollResults">
  <parameter name="Token" type="TokenType" direction="in">
    <parameter name="Return" type="xs:unsignedLong" direction="out" />
    <parameter name="ReturnData" type="InformationType" direction="out">
  </interface>
```

### 7.3.2.1 Description

The Get Enroll Results aggregate service shall retrieve the enrollment results for the specified token. This service is used in conjunction with the Enroll aggregate service. If the Enroll aggregate service is implemented as an asynchronous service, the

implementing system returns a token, and the Get Enroll Results service is used to poll for the results of the original Enroll request.

### 7.3.2.2 Parameters

*Token (input)* – a value used to retrieve the results of the Enroll request

*Return (output)* – return value indicating success or specifying a particular error condition

*Return Data (output)* – contains a return data record

### 7.3.3 Get Identify Results

```
<interface name="GetIdentifyResults">
  <parameter name="Token" type="TokenType" direction="in">
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="ReturnData" type="InformationType" direction="out">
</interface>
```

#### 7.3.3.1 Description

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

#### 7.3.3.2 Parameters

*Token (input)* – a value used to retrieve the results of the Identify request

*Return (output)* – return value indicating success or specifying a particular error condition

*Return Data (output)* – contains a return data record

### 7.3.4 Get Verify Results

```
<interface name="GetVerifyResults">
  <parameter name="Token" type="TokenType" direction="in">
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="Match" type="xs:boolean" direction="out" />
  <parameter name="Score" type="xs:int" direction="out" />
  <parameter name="ReturnData" type="InformationType" direction="out">
</interface>
```

#### 7.3.4.1 Description

The Get Verify Results aggregate service shall retrieve the verification results for the specified token. This service is used in conjunction with the Verify aggregate service.

If the Verify aggregate service is implemented as an asynchronous service, the implementing system returns a token, and the Get Verify Results service is used to poll for the results of the original Verify request.

#### 7.3.4.2 Parameters

*Token (input)* – a value used to retrieve the results of the Verify request

*Return (output)* – return value indicating success or specifying a particular error condition

*Match (output)* – indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR

*Score (output)* – the score if the biometric information matched

*Return Data (output)* – contains a return data record

#### 7.3.5 Identify

```
<interface name="Identify">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="InputData" type="InformationType" direction="in">
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="ReturnData" type="InformationType" direction="out">
</interface>
```

##### 7.3.5.1 Description

The Identify aggregate service shall perform an identification function according to system requirements and/or resources. For example, a system may have multiple galleries of subjects, and may utilize any or all of these galleries, via calls to the Identify Subject primitive service, to perform a system-level identification function. The system may perform additional actions based on input flags and/or results of the Identify Subject primitive service requests. For example, in an encounter-centric model, this aggregate service may search three separate galleries of subjects, and if a match is found it may then utilize the Set Biographic Data and/or Set Biometric Data primitive services to create a new encounter for the subject.

If the Identify aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the Identify aggregate service is implemented as an asynchronous service, the implementing system shall return a token in the Return Data parameter, which is an indication that the request is being handled asynchronously. In this case, the Get Identify Results service shall be used to poll for the results of the Identify request.



### 7.3.5.2 Parameters

*Processing Options (input)* – options that guide how the service request is processed

*Input Data (input)* – contains an input data record, which at a minimum must include biometric data

*Return (output)* – return value indicating success or specifying a particular error condition

*Return Data (output)* – contains a return data record

### 7.3.6 Retrieve Information

```
<interface name="RetrieveInformation">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="SubjectID"
    type="xs:string" direction="in" use="conditional">
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="conditional">
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="ReturnData" type="InformationType" direction="out">
</interface>
```

#### 7.3.6.1 Description

The Retrieve Information aggregate service shall retrieve requested information about a subject, or in an encounter-centric model about an encounter. In a person-centric model, this aggregate service may be used to retrieve both biographic and biometric information for a subject record. In an encounter-centric model, this aggregate service may be used to retrieve biographic and/or biometric information for either a single encounter or all encounters. Either a subject ID or encounter ID must be specified.

#### 7.3.6.2 Parameters

*Processing Options (input)* – options that guide how the service request is processed, and may identify what type(s) of information should be returned

*Subject ID (input, conditional)* – the identifier of the subject; required if no encounter ID is provided

*Encounter ID (input, conditional)* – the identifier of the encounter; required if no subject ID is provided

*Return (output)* – return value indicating success or specifying a particular error condition

*Return Data (output)* – contains a return data record

### 7.3.7 Verify

```
<interface name="Verify">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="InputData" type="InformationType" direction="in">
  <parameter name="IdentityClaim"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="ReferenceBIR"
    type="CBEFF BIR Type" direction="in" use="conditional" />
  <parameter name="GalleryID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
  <parameter name="Match" type="xs:boolean" direction="out" />
  <parameter name="Score" type="xs:int" direction="out" />
  <parameter name="ReturnData" type="InformationType" direction="out">
</interface>
```

#### 7.3.7.1 Description

The Verify aggregate service shall perform a 1:1 verification function according to system requirements and/or resources. Either the Identity Claim or Reference BIR input parameters are required. The system may perform additional actions based on input flags and/or results of verification. For example, in an encounter-centric model, this aggregate service may initiate a request to the Verify Subject primitive service, and if a match is found it may then utilize the Set Biographic Data and/or Set Biometric Data primitive services to create a new encounter for the subject.

If the Verify aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the Verify aggregate service is implemented as an asynchronous service, the implementing system shall return a token in the Return Data parameter, which is an indication that the request is being handled asynchronously. In this case, the Get Verify Results service shall be used to poll for the results of the Verify request.

#### 7.3.7.2 Parameters

*Processing Options (input)* – options that guide how the service request is processed, and may identify what type(s) of information should be returned

*Input Data (input)* – contains an input data record, which at a minimum must include biometric data

*Identity Claim (input, conditional)* – the identifier by which the subject is known to the gallery, required if no Reference BIR is provided

*Reference BIR (input, conditional)* – data structure containing the biometric sample that will be compared to the Input BIR, required if no Identity Claim is provided

*Gallery ID (input, optional)* – the identifier of the gallery or population group of which the subject must be a member

*Return (output)* – return value indicating success or specifying a particular error condition

*Match (output)* – indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR

*Score (output)* – the score if the biometric information matched

*Return Data (output)* – contains a return data record

## 8 Data Elements and Data Types

A goal of BIAS is to be flexible to the amount and types of biographic and biometric information available to and used by a system. The parameters “Biographic Data” and “Biometric Data” are meant to be general in this sense in order to allow this flexibility. This section includes information on how this flexibility can be specified and supported by implementing systems.

### 8.1 Biographic Data

BIAS defines three data types to provide flexibility for the amount and types of biographic data supported by implementing systems. The Biographic Data Item Type shall represent a single biographic data item, and the Biographic Data Set Type shall represent a set of biographic information in a specified format. The Biographic Data Type is a common type that shall represent either a set or list of biographic data.

#### 8.1.1 Biographic Data Type

```
<xs:complexType name="BiographicDataType">
  <xs:choice maxOccurs="unbounded">
    <xs:element name="LastName" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="FirstName" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="BiographicDataItem"
      type="BiographicDataItemType" maxOccurs="unbounded" />
    <xs:element name="BiographicDataSet"
      type="BiographicDataSetType" maxOccurs="1" />
  </xs:choice>
</xs:complexType>
```

##### 8.1.1.1 Description

The Biographic Data Type defines a set of biographic data elements, utilizing either the Biographic Data Item Type to represent a list of elements or the Biographic Data Set Type to represent a complete, formatted set of biographic information. This type

also includes optional fixed fields for representing first and last names, two common biographic data elements.

### 8.1.1.2 Definitions

*Last Name (optional)* – the last name of a subject

*First Name (optional)* – the first name of a subject

*Biographic Data Item* – a single biographic data element

*Biographic Data Set* – a set of biographic data information

## 8.1.2 Biographic Data Item Type

```
<xs:complexType name="BiographicDataType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="Name" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="Type" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="Value" type="xs:string"
      minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

### 8.1.2.1 Description

The Biographic Data Item Type defines a single biographic data element. The biographic data item *name* and *type* are required elements, while the *value* is optional.

### 8.1.2.2 Definitions

*Name* – the name of the biographic data item (i.e., "PersonName")

*Type* – the data type for the biographic data item (i.e., "xs:string")

*Value (optional)* – the value assigned to the biographic data item (i.e., "John Doe")

## 8.1.3 Biographic Data Set Type

```
<xs:complexType name="BiographicDataSetType">
  <xs:sequence>
    <xs:any namespace="##any" />
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required" />
  <xs:attribute name="version" type="xs:string" use="optional" />
  <xs:attribute name="source" type="xs:string" use="required" />
  <xs:attribute name="type" type="xs:string" use="required" />
</xs:complexType>
```

**8.1.3.1 Description**

The Biographic Data Set Type defines a set of biographic data that is formatted according to the specified format. This data type allows biographic information in an Electronic Fingerprint Transmission Specification (EFTS) or pre-defined XML format, for example, to be used with BIAS messages.

**8.1.3.2 Definitions**

*name* – the name of the biographic data format (e.g. "EFTS")

*version (optional)* – the version of the biographic data format (e.g. "7.1" or "1.0")

*source* – reference to a URI describing the biographic data format

*type* – the biographic data format type (e.g. "XML")

**8.1.3.3 Common Biographic Data Format References**

In order to provide a consistent representation of some common biographic data formats, this specification will establish common values for the attributes used in the Biographic Data Set Type. The common biographic data formats are: the EFTS Type-2 record, the Electronic Biometric Transmission Specification (EBTS) Type-2 record, the National Information Exchange Model (NIEM), the OASIS Customer Information Quality (CIQ) extensible Name and Address Language (xNAL), and the Human Resources extensible Markup Language (HR-XML). If any of these common biographic data formats are used, then values found in Table 8-1 for the *name*, *version*, *source*, and *type* attributes shall be specified:

*Table 8.1 – Common Biographic Data Format References*

<b>Biographic Data Format</b>	<b><i>name</i></b>	<b><i>version</i></b>	<b><i>source</i></b>	<b><i>type</i></b>
EFTS Type-2	EFTS	7.1	<a href="http://www.fbi.gov/">http://www.fbi.gov/</a>	ASCII
EBTS Type-2	EBTS	1.2	<a href="http://www.biometrics.dod.mil/">http://www.biometrics.dod.mil/</a>	ASCII
NIEM	NIEM	1.0 2.0	<a href="http://www.niem.gov/">http://www.niem.gov/</a>	XML
CIQ xNAL	xNAL	2.0 3.0	<a href="http://www.oasis-open.org/">http://www.oasis-open.org/</a>	XML
HR-XML	HR-XML	2.5	<a href="http://www.hr-xml.org/">http://www.hr-xml.org/</a>	XML

## 8.2 Biometric Data

This section defines how to represent biometric data in the BIAS services.

### 8.2.1 CBEFF BIR Type

Biometric information shall be packaged as a biometric information record (BIR) in a CBEFF structure, called a CBEFF-BIR in this standard, with the biometric data embedded in the biometric data block, as defined by INCITS 398-2005 Revision 1 or ISO/IEC 19785-1:2006. This standard does not require any specific CBEFF patron format, and recognizes that BIAS implementations may support only one patron format. Applications and implementations may also choose to support multiple patron formats.

The CBEFF BIR Type schema shown below shall be used to represent biometric information. This schema provides for either a non-XML and an XML representation. For some implementations, such as the OASIS-defined Web services bindings, it will be helpful to replicate the CBEFF data elements as separate XML elements. This schema represents CBEFF data elements from both the US (INCITS 398-2005 Revision 1) and International (ISO/IEC 19785-1:2006) versions of the CBEFF standard. The data elements in the two standards are fairly similar. However, a data element may have a different meaning and/or a different set of valid values based on which standard is used. The *<CBEFFVersion>* data element specifies whether the US or International version is being used. An additional data element, *<Other>*, has been added to this schema to capture non-standard information and to allow for flexibility to any future changes in the existing CBEFF standards.

For a description or definition of each data element, see the referenced CBEFF standards.

```
<xs:complexType name="CBEFF_BIR_Type">
  <xs:choice minOccurs="1" maxOccurs="1">
    <xs:any namespace="##any" />
    <xs:element name="XML_BIR" type="CBEFF_XML_BIR_Type" />
  </xs:choice>
</xs:complexType>

<xs:complexType name="CBEFF_XML_BIR_Type">
  <xs:sequence>
    <xs:element name="patron-format-identifier" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="version" type="Version"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="cbeff-version" type="Version"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="bir-info" type="BIR-info"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="bdb-info" type="BDB-info"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="sb-info" type="SB-info"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="other" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
```

```

        <xs:sequence>
          <xs:any namespace="##any" processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="BIR" type="xs:base64Binary"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Version">
  <xs:sequence>
    <xs:element name="major" type="xs:integer"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="minor" type="xs:integer"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BIR-info">
  <xs:sequence>
    <xs:element name="creator" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="index" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="payload" type="xs:anyType"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="integrity-options" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="creation-date" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="validity-period" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="patron-format-owner" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="patron-format-type" type="xs:string"
      minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BDB-info">
  <xs:sequence>
    <xs:element name="challenge-response" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="index" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="format-owner" type="xs:integer"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="format-type" type="xs:integer"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="security-Encryption-options" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="creation-date" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="validity-period" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="type" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="subtype" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="data-type" type="xs:string"
      minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="product-owner" type="xs:integer"
  minOccurs="1" maxOccurs="1" />
<xs:element name="product-type" type="xs:integer"
  minOccurs="1" maxOccurs="1" />
<xs:element name="purpose" type="xs:string"
  minOccurs="0" maxOccurs="1" />
<xs:element name="quality" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="value" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="algorithm-owner" type="xs:string"
        minOccurs="0" maxOccurs="1" />
      <xs:element name="algorithm-type" type="xs:string"
        minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="device-owner" type="xs:integer"
  minOccurs="0" maxOccurs="1" />
<xs:element name="device-type" type="xs:integer"
  minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>

<xs:complexType name="SB-info">
  <xs:sequence>
    <xs:element name="format-owner" type="xs:integer"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="format-type" type="xs:integer"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</complexType>

```

## 8.2.2 CBEFF BIR List Type

```

<xs:complexType name="CBEFF BIR ListType">
  <xs:sequence>
    <xs:element name="BIR"
      type="CBEFF BIR Type"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

```

### 8.2.2.1 Description

The CBEFF BIR List Type shall provide a list of CBEFF-BIR elements.

### 8.2.2.2 Definitions

*BIR* – CBEFF structure containing information about a biometric sample

## 8.2.3 Biometric Data Element Type

```

<xs:complexType name="BiometricDataElementType">
  <xs:sequence>
    <xs:element name="BiometricType" type="xs:hexBinary"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="BiometricTypeCount" type="xs:positiveInteger"

```



```

        minOccurs="0" maxOccurs="1" />
    <xs:element name="BiometricSubType" type="xs:hexBinary"
        minOccurs="0" maxOccurs="1" />
    <xs:element name="BDBFormatOwner" type="xs:string"
        minOccurs="1" maxOccurs="1" />
    <xs:element name="BDBFormatType" type="xs:string"
        minOccurs="1" maxOccurs="1" />
</xs:sequence>
</xs:complexType>

```

### 8.2.3.1 Description

The Biometric Data Element Type shall provide descriptive information about biometric data, such as the biometric type, subtype, and format, contained in the BDB of the CBEFF-BIR.

### 8.2.3.2 Definitions

*Biometric Type* – the type of biological or behavioral data stored in the biometric record, as defined by CBEFF

*Biometric Type Count (optional)* – the number of biometric records having the biometric type recorded in the biometric type field

*Biometric Subtype (optional)* – more specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF

*BDB Format Owner* – identifies the standards body, working group, industry consortium, or other CBEFF biometric organization that has defined the format for the biometric data

*BDB Format Type* – identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field

## 8.2.4 Biometric Data List Type

```

<xs:complexType name="BiometricDataListType">
  <xs:sequence>
    <xs:element name="BiometricDataElement"
      type="BiometricDataElementType"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

```

### 8.2.4.1 Description

The Biometric Data List Type shall provide a list of biometric data elements.

### 8.2.4.2 Definitions

*Biometric Data Element* – data structure containing information about a biometric record

## 8.3 Candidate Lists

Candidate lists are returned in the response to a biometric identification request. BIAS defines two data types to represent candidate lists. The Candidate Type shall represent a single candidate, and the Candidate List Type shall represent a set or list of candidates.

### 8.3.1 Candidate Type

```
<xs:complexType name="CandidateType">
  <xs:sequence>
    <xs:element name="Score" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="BiographicData" type="BiographicDataType"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="BIRList" type="CBEFF_BIR_ListType"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="rank" type="xs:int" use="required" />
</xs:complexType>
```

#### 8.3.1.1 Description

The Candidate Type defines a single candidate as a possible match in response to a biometric identification request. The candidate *BIR* is a required element, while the *score* and *biographic data* are optional.

#### 8.3.1.2 Definitions

*rank* – the rank of the candidate in relation to other candidates for the same biometric identification operation

*Score (optional)* – the match score

*Biographic Data (optional)* – biographic data associated with the candidate match

*BIRList* – biometric data associated with the candidate match

### 8.3.2 Candidate List Type

```
<xs:complexType name="CandidateListType">
  <xs:sequence>
    <xs:element name="Candidate" type="CandidateType"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 8.3.2.1 Description

The Candidate List Type defines a set of candidates, utilizing the Candidate Type to represent each element in the set.

### 8.3.2.2 Definitions

*Candidate* – a single candidate

## 8.4 Capabilities

Implementing systems will have various capabilities to support BIAS services. These capabilities include information on supported biometric matching capabilities, supported galleries, supported processing options for the aggregate services, supported biographic formats, etc. BIAS defines two data types to represent these capabilities. The Capability Type shall represent a single capability, and the Capability List Type shall represent a set of capabilities.

### 8.4.1 Capability Type

```
<xs:complexType name="CapabilityType">
  <xs:sequence>
    <xs:element name="CapabilityName" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="CapabilityID" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="CapabilityDescription" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="CapabilityValue" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="CapabilitySupportingValue" type="xs:string"
      minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

#### 8.4.1.1 Description

The Capability Type defines a single capability supported by an implementing system. Each supported capability shall be identified by a *Capability Name*. Some supported capabilities will have an associated *Capability Value* (e.g., supported galleries will have a Gallery ID value), while others will not (e.g., biometric matching support for a specific biometric type).

#### 8.4.1.2 Definitions

*Capability Name* – the name of the capability, as defined by the implementing system

*Capability ID (optional)* – an identifier assigned to the capability by the implementing system

*Capability Description (optional)* – a description of the capability

*Capability Value (optional)* – a value assigned to the capability

*Capability Supporting Value (optional)* – a secondary value supporting the capability

## 8.4.2 Capability List Type

```
<xs:complexType name="CapabilityListType">
  <xs:sequence>
    <xs:element name="Capability" type="CapabilityType"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 8.4.2.1 Description

The Capability List Type defines a set of capabilities, utilizing the Capability Type to represent each element in the set.

### 8.4.2.2 Definitions

*Capability* – a single capability

## 8.5 Fusion Information

Fusion information is sent as input to fusion services. BIAS defines two data types to represent fusion information. The Fusion Information Type shall represent a single set of fusion information, and the Fusion Information List Type shall represent a set or list of fusion information elements.

### 8.5.1 Fusion Information Type

```
<xs:complexType name="FusionInformationType">
  <xs:sequence>
    <xs:element name="BiometricType" type="xs:hexBinary"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="BiometricSubType" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="AlgorithmOwner" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="AlgorithmType" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:choice minOccurs="1" maxOccurs="1">
      <xs:element name="Score" type="xs:unsignedLong" />
      <xs:element name="Decision" type="xs:string" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

### 8.5.1.1 Description

The Fusion Information Type represents the information necessary to perform a fusion operation. It shall include the biometric type and subtype, if applicable, for which the biometric match was performed, the matching algorithm identifying information, and either a score (for score-level fusion) or a decision (for decision-level fusion).

### 8.5.1.2 Definitions

*Biometric Type* – the type of biological or behavioral data stored in the biometric record, as defined by CBEFF

*Biometric Subtype (optional)* – more specifically defines the type of biometric data stored in the biometric record

*Algorithm Owner* – the owner or vendor of the algorithm used to determine the score or decision

*Algorithm Type* – the Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision

*Score* – the similarity score assigned by the matching algorithm

*Decision* – the match decision assigned by the matching algorithm

## 8.5.2 Fusion Information List Type

```
<xs:complexType name="FusionInformationListType">
  <xs:sequence>
    <xs:element name="FusionElement" type="FusionInformationType"
      minOccurs="2" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 8.5.2.1 Description

The Fusion Information List Type shall contain at a minimum two sets of fusion input elements, utilizing the Fusion Information Type to represent a single set of fusion information.

### 8.5.2.2 Definitions

*Fusion Element* – a set of fusion information

## 8.6 Other Data Types

This section describes the remaining data types defined by this standard.

## 8.6.1 Encounter List Type

```
<xs:complexType name="EncounterListType">
  <xs:sequence>
    <xs:element name="EncounterID" type="xs:string"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 8.6.1.1 Description

The Encounter List Type defines a set of encounters.

### 8.6.1.2 Definition

*Encounter ID* – the identifier of an encounter

## 8.6.2 Information Type

```
<xs:complexType name="InformationType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 8.6.2.1 Description

The Information Type provides a method to represent any type of data element. It could represent a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system.

### 8.6.2.2 Definitions

The Information Type allows for an unlimited number of data element types, and it does not specify nor require any particular data element.

## 8.6.3 Identity Model Type

```
<xs:simpleType name="IdentityModelType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="encounter" />
    <xs:enumeration value="person" />
  </xs:restriction>
</xs:simpleType>
```

### 8.6.3.1 Description

The Identity Model Type defines the requested or supported identity model.

### 8.6.3.2 Definitions

*encounter* – specifies an encounter-centric identity model

*person* – specifies a person-centric identity model

### 8.6.4 List Filter Type

```
<xs:complexType name="ListFilterType">
  <xs:sequence>
    <xs:element name="BiometricTypeFilter" type="xs:hexBinary"
      minOccurs="1" maxOccurs="unbounded" />
    <xs:element name="IncludeBiometricSubtype" type="xs:boolean"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

#### 8.6.4.1 Description

The List Filter Type provides a method to filter the amount of information returned in a search of biometric data.

#### 8.6.4.2 Definitions

*Biometric Type Filter* – limits the returned information to a specific type of biometric, as defined by CBEFF

*Include Biometric Subtype* – a Boolean flag indicating if biometric subtype information should be returned

### 8.6.5 Processing Options Type

```
<xs:complexType name="ProcessingOptionsType">
  <xs:sequence>
    <xs:element name="Option" type="xs:string"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

#### 8.6.5.1 Description

BIAS aggregate services support the ability to include various processing options which direct and possibly control the business logic for that service. The Process Options Type provides a method to represent those options. Processing options should be defined by the implementing system.

#### 8.6.5.2 Definitions

*Option* – an option supported by the implementing system

## 8.6.6 Token Type

```
<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element name="TokenValue" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="Expiration" type="xs:date"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

### 8.6.6.1 Description

Some of the BIAS services may be handled asynchronously, such as the Identify Subject service. Upon receiving a request for the service, the implementing system may either process the request synchronously and return the results directly or it may process the request asynchronously and return a token that can be used to retrieve the results at some later time. If a token is returned, the client/requester will be responsible for polling for the results, using the token as the input parameter. The Token Type defines the token that is returned for asynchronous processing.

### 8.6.6.2 Definitions

*Token Value* – a value returned by the implementing system that is used to retrieve the results to a service request at a later time

*Expiration* – a date and time at which point the token expires and the service results are no longer guaranteed to be available

## 9 Error Handling and Notification

As with any messaging interface, there is a need to define effective measures for handling and propagating error conditions. All BIAS services have a Return parameter for conveying return values and error condition codes.

### 9.1 Successful Service Calls

BIAS defines a return value for successful service calls.

```
SUCCESS=0
```

### 9.2 Error Condition Codes

BIAS defines two levels of errors that are represented by the error condition codes: System and Service errors. In addition, BIAS also allows for implementations to define their own error condition codes to represent error conditions that are not already defined by this standard.



System-level errors occur when the implementing system cannot service a request. They could result due to an internal logic error or because the implementing system does not support a particular request. Service-level errors occur when there is a problem transmitting or representing the service request. They could result due to an invalid service request or because of a communications error. This standard will define the error condition codes for system-level errors. Companion standards, such as the OASIS BIAS Messaging Protocol, should define the error-condition codes for service-level errors.

BIAS defines the following set of system-level error codes:

UNKNOWN\_ERROR=1

The service failed for an unknown reason.

UNSUPPORTED\_CAPABILITY=2

A requested capability is not supported by the service implementation.

INVALID\_INPUT=10

The data in a service input parameter is invalid.

BIR\_QUALITY\_ERROR=11

Biometric sample quality is too poor for the service to succeed.

INVALID\_BIR=12

The input BIR is empty or in an invalid or unrecognized format.

BIR\_SIGNATURE\_FAILURE=13

The service could not validate the signature, if used, on the input BIR.

BIR\_DECRYPTION\_FAILURE=14

The service could not decrypt an encrypted input BIR.

INVALID\_ENCOUNTER\_ID=15

The input encounter ID is empty or in an invalid format.

INVALID\_SUBJECT\_ID=16

The input subject ID is empty or in an invalid format.

UNKNOWN\_SUBJECT=17

The subject referenced by the input subject ID does not exist.

UNKNOWN\_GALLERY=18

The gallery referenced by the input gallery ID does not exist.

UNKNOWN\_ENCOUNTER=19

The encounter referenced by the input encounter ID does not exist.

## 10 Security

Security is important for any kind of distributed computing environment, and even more so when distributed computing occurs over open networks. Web services, for example, generally operate as a public internet Web service, an intranet Web service, or a combination of both. BIAS implementers will ultimately choose the level of security and the security capabilities provided by their system(s). BIAS bindings, which are defined outside of this standard, need to include security capabilities and provide BIAS implementers with these options.

BIAS bindings shall provide options for message confidentiality and integrity. BIAS bindings shall provide options for message transport confidentiality and integrity. BIAS bindings shall provide options to support access controls in order to authenticate users of a service.

**Annex A**  
(normative)

**Conformance Requirements**

**A.1 General**

Conformance to this standard falls into the following classes:

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

Conformance requirements for these classes are defined in clause A.2.

**A.2 Class Conformance Requirements**

To claim conformance to this standard, implementations shall provide the mandatory services and capability item information for their conformance class, as defined below, in accordance with the service definitions in clause 7.

Implementations shall accept all valid input parameters and return valid outputs, as defined in clauses 7 and 8. Implementations shall perform error handling as defined in clause 9. Service bindings shall implement the security requirements as defined in clause 10.

All classes shall implement the Query Capabilities service.

The following table is a summary of conformance requirements by class. Details are provided in the following sub-clauses.

*Table A.1 – BIAS Conformance Classes*

<b>Service/Capability</b>	<b>Class 1</b>	<b>Class 2</b>	<b>Class 3</b>	<b>Class 4</b>	<b>Class 5</b>
<b>Primitive Services</b>					
Add Subject To Gallery	X		X		
Check Quality	X				
Classify Biometric Data	X				
Create Subject	X		X	X	
Delete Biographic Data	X		X	X	

Service/Capability	Class 1	Class 2	Class 3	Class 4	Class 5
Delete Biometric Data	X		X	X	
Delete Subject	X		X	X	
Delete Subject From Gallery	X		X		
Get Identify Subject Results	X		X		
Identify Subject	X		X		
List Biographic Data	X		X	X	
List Biometric Data	X		X	X	
Perform Fusion	X				
Query Capabilities	X	X	X	X	X
Retrieve Biographic Information	X		X	X	
Retrieve Biometric Information	X		X	X	
Set Biographic Data	X		X	X	
Set Biometric Data	X		X	X	
Transform Biometric Data	X				
Update Biographic Data	X		X	X	
Update Biometric Data	X		X	X	
Verify Subject	X		X	X	
<b>Aggregate Services</b>					
Enroll		X			X
Get Enroll Results		X			X
Get Identify Results		X			
Get Verify Results		X			X
Identify		X			
Retrieve Information		X			X
Verify		X			X
<b>Capability Information Items</b>					
AggregateInputDataOptional		X			X
AggregateInputDataRequired		X			X

Service/Capability	Class 1	Class 2	Class 3	Class 4	Class 5
AggregateProcessingOption		X			X
AggregateReturnData		X			X
AggregateServiceDescription		X			X
BiographicDataSet	X	X	X	X	X
CBEFFPatronFormat	X	X	X	X	X
ClassificationAlgorithmType	X				
ConformanceClass	X	X	X	X	X
Gallery	X	X	X		
IdentityModel	X	X	X	X	X
QualityAlgorithm	X				
SupportedBiometric	X	X	X	X	X
TransformOperation	X				

**A.2.1 Class 1: Full Primitive Services Implementation**

A Full Primitive Services Implementation shall provide the following BIAS services:

- Add Subject To Gallery
- Check Quality
- Classify Biometric Data
- Create Subject
- Delete Biographic Data
- Delete Biometric Data
- Delete Subject
- Delete Subject From Gallery
- Get Identify Subject Results
- Identify Subject
- List Biographic Data
- List Biometric Data
- Perform Fusion
- Query Capabilities
- Retrieve Biographic Information
- Retrieve Biometric Information
- Set Biographic Data
- Set Biometric Data

- Transform Biometric Data
- Update Biographic Data
- Update Biometric Data
- Verify Subject

A Full Primitive Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

- BiographicDataSet
- CBEFFPatronFormat
- ClassificationAlgorithmType
- ConformanceClass
- Gallery
- IdentityModel
- QualityAlgorithm
- SupportedBiometric
- TransformOperation

#### **A.2.2 Class 2: Full Aggregate Services Implementation**

A Full Aggregate Services Implementation shall provide the following BIAS services:

- Enroll
- Get Enroll Results
- Get Identify Results
- Get Verify Results
- Identify
- Retrieve Information
- Verify
- Query Capabilities

A Full Aggregate Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

- AggregateInputDataOptional
- AggregateInputDataRequired
- AggregateProcessingOption
- AggregateReturnData
- AggregateServiceDescription
- BiographicDataSet
- CBEFFPatronFormat
- ConformanceClass
- Gallery
- IdentityModel

- SupportedBiometric

### **A.2.3 Class 3: Limited Primitive Services Implementation**

A Limited Primitive Services Implementation provides many of the primitive services, except for Check Quality, Classify Biometric Data, Perform Fusion, and Transform Biometric Data. A Limited Primitive Services Implementation shall provide the following BIAS services:

- Add Subject To Gallery
- Create Subject
- Delete Biographic Data
- Delete Biometric Data
- Delete Subject
- Delete Subject From Gallery
- Get Identify Subject Results
- Identify Subject
- List Biographic Data
- List Biometric Data
- Query Capabilities
- Retrieve Biographic Information
- Retrieve Biometric Information
- Set Biographic Data
- Set Biometric Data
- Update Biographic Data
- Update Biometric Data
- Verify Subject

A Limited Primitive Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

- BiographicDataSet
- CBEFFPatronFormat
- ConformanceClass
- Gallery
- IdentityModel
- SupportedBiometric

### **A.2.4 Class 4: Minimum Primitive Services Implementation**

A Minimum Primitive Services Implementation does not provide biometric identification services. A Minimum Primitive Services Implementation shall provide the following BIAS services:

- Create Subject

- Delete Biographic Data
- Delete Biometric Data
- Delete Subject
- List Biographic Data
- List Biometric Data
- Query Capabilities
- Retrieve Biographic Information
- Retrieve Biometric Information
- Set Biographic Data
- Set Biometric Data
- Update Biographic Data
- Update Biometric Data
- Verify Subject

The optional Gallery ID parameter shall not be used in the Verify Subject service for Class 4 implementations, as the service for adding a subject to a gallery are not required.

A Minimum Primitive Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

- BiographicDataSet
- CBEFFPatronFormat
- ConformanceClass
- IdentityModel
- SupportedBiometric

#### **A.2.5 Class 5: Minimum Aggregate Services Implementation**

A Minimum Aggregate Services Implementation does not provide biometric identification services. A Minimum Aggregate Services Implementation shall provide the following BIAS services:

- Enroll
- Get Enroll Results
- Get Verify Results
- Retrieve Information
- Verify
- Query Capabilities

A Minimum Aggregate Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

- AggregateInputDataOptional
- AggregateInputDataRequired



- AggregateProcessingOption
- AggregateReturnData
- AggregateServiceDescription
- BiographicDataSet
- CBEFFPatronFormat
- ConformanceClass
- IdentityModel
- SupportedBiometric

## **Annex B**

(informative)

### **Bibliography**

- ISO/IEC 19784-2 Information Technology – Biometric Application Programming Interface – Part 2: Biometric Archive Function Provider Interface
- ISO/IEC FCD 19785-3 Information Technology – Common Biometric Exchange Formats Framework – Part 3: Patron Format Specifications
- ISO/IEC CD 24708, BioAPI Interworking Protocol (BIP)
- <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnpag2/html/wssp.asp>
- [JDJ] Service-Oriented Architecture: Beyond Web Services, JDJ, [http://java.sys-con.com/read/44368\\_p.htm](http://java.sys-con.com/read/44368_p.htm)

## **Annex C**

(informative)

### **Example Usage Scenarios**

#### **C.1 General**

The following usage scenarios illustrate the use of BIAS services in various application domains. The following usage scenarios are provided:

- Border Management – Benefits
- Border Management – Entry
- Border Management – Entry with Watch List Hit
- Online Banking
- Transportation Worker Identification

#### **C.2 Usage Scenario: Border Management – Benefits**

An individual within the United States is applying for immigration benefits for which an in-person interview is required. There is neither a watch list hit nor a derogatory Federal Bureau of Investigation (FBI) Integrated Automated Fingerprint Identification System (IAFIS) record for the Individual.

##### **C.2.1 Basic Flow of Events**

1. Individual applies for immigration status modification.
2. Staff member reviews the individual's status change application and enters the application data.
3. System does not locate a record for the individual based on the information entered (Verify Subject).
4. Individual is sent for fingerprinting.
5. Staff member scans the machine-readable documents.
6. Staff member records and enters the individual's 10 fingerprint images (Create Subject, Set Biographic Data, Set Biometric Data).
7. System retrieves and displays data to staff member (Retrieve Biographic Information).
8. Staff member notifies the individual of interview request.
9. Individual appears for interview.
10. Adjudicator scans the machine-readable documentation.
11. Adjudicator captures the individual's fewer-than-10 verification fingerprint images.
12. System confirms the match of the captured verification prints against the 10-print on file (Verify Subject).
13. System retrieves and displays to the Adjudicator the Individual's application data (Retrieve Biographic Information).

14. Adjudicator makes and records the status decision.
15. System links the decision and current encounter data to the individual's record (Update Biographic Data).

### **C.2.2 Pre-Conditions**

- Machine-readable documentation available
- No biometric watch list hit
- No derogatory IAFIS records located
- Individual appearing for interview at a later date

### **C.2.3 Post-Conditions**

- Individual enrolled into the identity assurance system
- Individual's data displayed to the Adjudicator
- Status decision made and recorded

## **C.3 Usage Scenario: Border Management – Entry**

An individual is attempting to enter the United States. There is neither a watch list hit nor a derogatory FBI IAFIS record for the individual.

### **C.3.1 Basic Flow of Events**

1. Individual arrives at the border.
2. Officer scans the individual's machine-readable documentation.
3. System locates 2-print record in biometric database based on the information entered (Verify Subject).
4. The system retrieves and displays to the Officer the individual's primary view data (Retrieve Biographic Information, Retrieve Biometric Information).
5. Officer captures the individual's 10 fingerprint images.
6. Officer makes and records the entry decision.
7. System links the decision and current encounter data to the individual's record (Set Biographic Data, Set Biometric Data).

### **C.3.2 Pre-Conditions**

- Machine-readable documentation available
- 2-print record on file
- No biometric watch list hit
- No derogatory IAFIS records located
- No secondary inspection necessary

### C.3.3 Post-Conditions

- Individual's data displayed to the Officer
- Existing 2-print updated by the newly-captured 10-print
- Entry decision made and recorded
- Current encounter record linked to the individual

## C.4 Usage Scenario: Border Management – Entry with Watch List Hit

An individual is attempting to enter the United States. There is a watch list hit, but there are no derogatory FBI IAFIS records for the individual.

### C.4.1 Basic Flow of Events

1. Individual arrives at the border.
2. Officer scans the individual's machine-readable documentation.
3. System locates a 2-print watch list record for the individual based on the information entered (Verify).
4. Officer sends the individual to secondary inspection.
5. The system retrieves and displays to the Secondary Officer the individual's secondary view data, including watch list hit (Retrieve Information).
6. Secondary Officer captures the individual's 10 fingerprint images.
7. Secondary Officer makes and records the entry decision.
8. System links the decision and current encounter data to the individual's record (Set Biographic Data, Set Biometric Data).

### C.4.2 Pre-Conditions

- Machine-readable documentation available
- Biometric watch list hit
- No derogatory IAFIS records located

### C.4.3 Post-Conditions

- Individual's data displayed to the Officer
- Biometric watch list hit confirmed
- Entry decision made and recorded
- Current encounter record linked to the individual

## C.5 Online Banking

An individual has a bank account at XYZ Bank. He would like to access his account information and perform transactions related to his account. The account holder uses his home PC with a biometric device (e.g., an iris camera) installed. In lieu of a

password, the bank has configured their online banking web application to use biometric verification.

### **C.5.1 Basic Flow of Events**

The following sections describe enrolling an individual's biometric and accessing a bank account.

#### **C.5.1.1 Enrollment**

The bank has issued the individual a one-time password to allow him to enroll his biometric into the system. The individual accesses the online banking site and selects 'biometric enrollment'. He enters his account number and one-time password to access this function. Once verified, the enrollment application is initiated. The individual follows the steps to capture his biometric data and to perform a local 1:1 match against that data to ensure it will be matchable. Once suitable data is acquired, it is submitted to the bank as an enrollment [Set Biometric Data]. At this point, the individual's biometric data has been associated with his identity (account).

Note – enrollment could also be performed in person at the bank, but a similar scenario would apply, less the one-time password.

#### **C.5.1.2 Account Access**

Once an individual is biometrically enrolled, he would like to perform an online transaction. He accesses the online banking site and enters his account number. At this point, the individual is challenged to present his biometric (e.g., capture his iris). The individual interacts with the device to capture the biometric data. This data is then transmitted to the bank for verification [Verify Subject]. If the verification is successful, the bank will provide access to the transaction screens for the individual's account.

### **C.5.2 Pre-Conditions**

- Individual has already been enrolled and setup with an account.
- If the individual has not used his biometric device online before (i.e., with his browser), an active-X control (or Java applet) may have to be downloaded before he can use it with the online banking application. This may be transparent to him (or partially or not at all, depending on his security settings). It is possible that it may not be required at all, depending on what software came with the device and was loaded when it was installed.

### **C.5.3 Post-Conditions**

- Individual's biometrics are associated with his account.

## C.6 Transportation Worker Identification

This use case describes the use of biometrics for the purpose of transportation worker identification. In particular, the use of biometric services in this process are highlighted.

### C.6.1 Detailed Description

Per the Transportation Security Administration (TSA), the following steps are involved in the issuance of a biometric identity credential:

1. A registered employer (or local facility) initiates a request for a transportation worker identification card (TWIC) to be generated.
2. TWIC applicant completes pre-enrollment and in-person enrollment.
3. Enrollment record request sent to IDMS.
4. 1:N check of Reference Biometric performed.
5. Name-based threat assessment initiated and go, no-go results are returned to IDMS.
6. Authorization to produce TWIC and requisite data sent to Card Production Facility.
7. The user's card is personalized and encoded.
8. Card is securely shipped to the designated Enrollment Center.
9. TWIC Applicant returns to Enrollment Center, validates his identity using the reference biometric, and the card is electronically unlocked and issued.
10. IDMS is notified of TWIC issuance and activation.
11. TWIC holder requests access privileges at transportation facility.
12. Local facility notifies IDMS that access privileges have been granted.
13. Threat/intelligence information is received, and the generated watch list is compared to IDMS.
14. TWIC Hotlist is broadcast to all facilities, as well as a specific notification to any site where privileges have been granted.
15. Revocation and Disposition.

The above, however, does not address the subsequent use of the identity credential for access control purposes. This includes the use for both physical and logical access (i.e., access to secure areas/buildings and to computer/network resources respectively).

### C.6.2 Basic Flow of Events

The following sections concentrate on those portions of the workflow involving biographic or biometric identity data and operations. Major sub-operations (simplified) are defined as:

- Pre-enrollment (optional)

- Enrollment
- Enrollment processing
- Credential issuance
- Privilege granting
- Access control

#### **C.6.2.1 Pre-Enrollment**

Each transportation worker to be issued a TWIC card may optionally pre-enroll. This involves accessing a web-site and entering biographic data. This data is stored for the applicant. [Create Subject, Set Biographic Data] At this time, the applicant may also schedule an enrollment.

#### **C.6.2.2 Enrollment**

At the enrollment session, the operator (trusted agent) works with the applicant to:

- Enter (if not pre-enrolled) or verify and edit (if pre-enrolled) biographic data [Create Subject, Set Biographic Data] or [Retrieve Biographic Information, Update Biographic Data]
- Scan/Validate source documents
- Capture ten-prints (fingerprint images) [Set Biometric Data]
- Capture facial photograph [Set Biometric Data]

#### **C.6.2.3 Enrollment Processing**

Once all of the biographic and biometric data has been collected and sent to the IDMS, enrollment processing is initiated. This consists of:

- 1:N duplicate check against TWIC fingerprint database [Identify Subject, Add Subject to Gallery]
- Watch list (and/or other threat screening) check [Identify Subject (if performed locally)]
- Criminal History Records Check (external interface to FBI IAFIS)
- Name-based checks (external)

#### **C.6.2.4 Credential Issuance**

If all enrollment processing completes with no adverse information, resulting in an “approval” decision, then the TWIC credential may be issued as follows:

- Card production package is generated [Retrieve Biographic Information, Retrieve Biometric Information]
- Card data is received by the production facility, which produces the card and send it to the enrollment center or origin



- The worker is notified that his/her card is ready for pickup and returns to the enrollment center
- The worker's fingerprint(s) is captured and verified against the IDMS [Verify Subject]
- The fingerprint may also be verified against the fingerprint stored on the card.
- The card is activated.

#### **C.6.2.5 Privilege Granting**

At each transportation facility to which the worker requires access, privileges must be granted. This involves the following:

- The worker presents their TWIC card to the facility agent who validates the card (this may involve checking of expiration date, hotlist/CRL, and/or other security features of the card)
- The worker's fingerprint is matched against that stored on the card (local operation)
- Optionally, the worker may be enrolled in a local (operational biometric) [Create Subject, Set Biometric Data]
- The worker is added to the local logical or physical access control system (account created)

#### **C.6.2.6 Access Control**

- The worker present's their card to the reader
  - Biometric is read off the card (if match-to-card is used)
- The worker's biometric is captured and either
  - Matched against that read off the card, or
  - Matched against the corresponding record in the (central) access control database [Verify Subject]

#### **C.6.3 Special Requirements**

- Card management functions, though necessary, are not addressed in the use case.
- Revocation is based on a cryptographic CRL and a card hotlist and does not involve biometric operations.

#### **C.6.4 Pre-Conditions**

- Sponsorship is a requirement prior to enrollment, but is not addressed in the use case.

### C.6.5 Post-Conditions

- Periodically, the biometrics of a worker may be queried against one or more threat watch lists; however, this is not depicted in the use case.