

---

---

**Information technology — Biometric  
application programming interface —  
Part 4:  
Biometric sensor function provider  
interface**

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**  
*Technologies de l'information — Interface de programmation  
d'applications biométriques —  
Partie 4: Interface du fournisseur de fonction du capteur biométrique*

ISO/IEC 19784-4:2011

<https://standards.iteh.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-8fcfde6faca4/iso-iec-19784-4-2011>

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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

[ISO/IEC 19784-4:2011](https://standards.iteh.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-8fcfde6faca4/iso-iec-19784-4-2011)

<https://standards.iteh.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-8fcfde6faca4/iso-iec-19784-4-2011>



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

Foreword .....	v
Introduction.....	vi
<b>1</b> <b>Scope</b> .....	<b>1</b>
<b>2</b> <b>Conformance</b> .....	<b>1</b>
<b>3</b> <b>Normative references</b> .....	<b>1</b>
<b>4</b> <b>Terms and definitions</b> .....	<b>1</b>
<b>5</b> <b>Symbols, abbreviated terms, data structure definitions and error codes</b> .....	<b>2</b>
5.1 <b>Symbols and abbreviated terms</b> .....	2
5.2 <b>Data structure definitions and error codes</b> .....	2
<b>6</b> <b>Interface architecture</b> .....	<b>3</b>
<b>7</b> <b>Selecting and loading BFPs and BioAPI_Units</b> .....	<b>4</b>
7.1 <b>Obtaining information about BFPs and BioAPI_Units</b> .....	4
7.2 <b>Loading BFPs</b> .....	5
7.2.1 <b>Loading BFPs when attaching to an unspecified BioAPI_Unit</b> .....	5
7.2.2 <b>Loading BFPs when attaching to a specific BioAPI_Unit</b> .....	5
<b>8</b> <b>BSFPI Definition</b> .....	<b>5</b>
8.1 <b>BSFPI data structures</b> .....	5
8.1.1 <b>Control codes</b> .....	5
8.1.2 <b>BioSFPI_EventHandler</b> .....	6
8.1.3 <b>BioSFPI_GUI_RESPONSE</b> .....	6
8.1.4 <b>BioSFPI_GUI_PROGRESS_EVENT_HANDLER</b> .....	6
8.2 <b>BSFP Functions</b> .....	7
8.2.1 <b>BioSFPI_BSFPLoad</b> .....	7
8.2.2 <b>BioSFPI_UnitAttach</b> .....	8
8.2.3 <b>BioSFPI_UnitDetach</b> .....	9
8.2.4 <b>BioSFPI_QueryUnits</b> .....	9
8.2.5 <b>BioSFPI_Free</b> .....	10
8.2.6 <b>BioSFPI_ControlUnit</b> .....	11
8.2.7 <b>BioSFPI_Cancel</b> .....	11
8.2.8 <b>BioSFPI_SetPowerMode</b> .....	12
8.2.9 <b>BioSFPI_SetGUICallbacks</b> .....	13
8.2.10 <b>BioSFPI_QueryFormatInfo</b> .....	13
8.2.11 <b>BioSFPI_GetPackets</b> .....	14
8.2.12 <b>BioSFPI_DataTransfer</b> .....	15
8.2.13 <b>BioSFPI_SetIndicatorStatus</b> .....	16
8.2.14 <b>BioSFPI_GetIndicatorStatus</b> .....	16
8.2.15 <b>BioSFPI_CalibrateSensor</b> .....	17
8.2.16 <b>BioSFPI_SubscribeToGUIEvents</b> .....	17
8.2.17 <b>BioSFPI_UnsubscribeFromGUIEvents</b> .....	18
<b>Annex A (normative) Imaging biometric sensor units</b> .....	<b>20</b>
<b>A.1</b> <b>General</b> .....	<b>20</b>
A.1.1 <b>Bit and Byte ordering</b> .....	20
A.1.2 <b>Scan sequence</b> .....	20
A.1.3 <b>Pixel aspect ratio</b> .....	20
A.1.4 <b>Greyscale pixel depth</b> .....	20
A.1.5 <b>Colour space</b> .....	20
A.1.6 <b>Video interlacing</b> .....	21

A.2	BioSFPI_BSFPImagePropertyID.....	21
A.3	BioSFPI_BSFPImagePropertySchema .....	21
A.4	BioSFPI_UnitImagePropertyID .....	22
A.5	BioSFPI_UnitImagePropertySchema .....	22
A.6	Image sensor functions .....	23
A.6.1	BioSFPI_GetImageParameters .....	23
A.6.2	BioSFPI_SetImageParameters .....	24
A.6.3	BioSFPI_Pause.....	25
A.6.4	BioSFPI_Play.....	25
Annex B	(normative) Signature sensor units .....	27
B.1	General.....	27
B.1.1	Bit and byte ordering.....	27
B.1.2	Coordinate system .....	27
B.1.3	Channel inclusion field .....	28
B.1.4	Sequence of sample points .....	29
B.1.5	Sample Rate .....	29
B.1.6	ChannelDescriptions .....	29
B.2	BioSFPI_BSFPSignaturePropertyID .....	29
B.3	BioSFPI_BSFPSignaturePropertySchema .....	30
B.4	BioSFPI_UnitSignaturePropertyID .....	30
B.5	BioSFPI_UnitSignaturePropertySchema.....	31
B.6	Signature Sensor Functions.....	32
B.6.1	BioSFPI_GetSignatureParameters.....	32
B.6.2	BioSFPI_SetSignatureParameters .....	33
Annex C	(informative) Obtaining biometric sensor data across the SFPI.....	35
C.1	Overall functionality .....	35
C.2	Fundamentals of the data transfer interfaces.....	36
C.3	Memory management.....	36
C.4	Components forming the interface .....	36
C.5	Implementation options for use of the data transfer interface .....	36
Bibliography	.....	38

ITC STANDARD PREVIEW  
(standards.itech.ai)

ISO/IEC 19784-4:2011  
<https://standards.itech.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-8c1dc0faca4/iso-iec-19784-4-2011>

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19784-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

ISO/IEC 19784 consists of the following parts, under the general title *Information technology — Biometric application programming interface*:

- *Part 1: BioAPI specification* [ISO/IEC 19784-4:2011](https://standards.iteh.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-8fcfde6fca4/iso-iec-19784-4-2011)
- *Part 2: Biometric archive function provider interface*
- *Part 4: Biometric sensor function provider interface*

## Introduction

This part of ISO/IEC 19784 specifies a low-level interface that enables a Biometric Service Provider (BSP) module (see ISO/IEC 19784-1) to interact with a biometric sensor from a different vendor [a Biometric Sensor Function Provider (BSFP) module], using only the specification of the standardized interface. The biometric sensor is a physical device with a small amount of associated software. The interface enables a biometric sensor to interwork with any BSP that is designed to handle the technology supported by that sensor. It also enables the BSP module to be independent of details of the actual sensor, although it will remain dependent on the biometric technology it supports (finger, face, vein, etc.).

Biometric sensors can have very different working principles. Also the data formats vary very much depending on the biometric feature and the sensor type. To cover the resulting different requirements for a generic function provider, normative annexes contain the specific functions and data structures for typical sensor device classes or biometric modalities. The philosophy of this part of ISO/IEC 19784 is to add such normative annexes whenever the existing annexes do not cover the requirements for a typical sensor class.

Currently there are two normative annexes. Annex A provides type definitions and function calls designed to support biometric sensors designed to return images (either still images or a sequence of images). Annex B provides type definitions and function calls designed to support biometric sensors designed to return sequences of pen movements.

The major function of a BSP module performing a capture is to produce a complete Biometric Information Record (BIR) for returning to a biometric application (usually via a BioAPI framework).

The major function of a BSFP is to do a capture and to return data in an identified format, either as a single piece of data (use of BioSFPI\_DataTransfer) or as a series of packets (use of BioSFPI\_GetPackets) containing the captured information, or via a stream interface (see Annex C).

The BSP is responsible for turning this data into a Biometric Data Block (BDB) within a BIR for returning across the Service Provider Interface (SPI).

Additional (minor) functions relate to control of the biometric sensor and the parameters of its operation.

# Information technology — Biometric application programming interface —

## Part 4: Biometric sensor function provider interface

### 1 Scope

This part of ISO/IEC 19784 specifies a biometric sensor interface for a Biometric Service Provider (BSP, see ISO/IEC 19784-1). The interface supports a BSP wishing to provide the BioAPI Service Provider Interface (SPI) functions, whilst removing device handling activity from the BSP. This part of ISO/IEC 19784 provides an interface that can be used by all types of biometric sensor, including *inter alia* image streaming sensors (infrared, face, iris, finger, etc.), voice streaming sensors and digital tablets providing dynamic signature data.

It is not in the scope of this part of ISO/IEC 19784 to define security and privacy requirements for capturing and transferring of biometric data across the Sensor Function Provider Interface (SFPI).

### 2 Conformance

A BSFP shall support all the functions in Clause 8 of this part of ISO/IEC 19784 and all those in the two normative annexes. However, those functions that have a permissible error return of BioAPIERR\_FUNCTION\_NOT\_SUPPORTED can always return this error. Functions that do not list this error as an allowed return are required to perform the function as described, subject to other error returns listed.

### 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.

ISO/IEC 19784-1, *Information technology — Biometric application programming interface — Part 1: BioAPI specification*

ISO/IEC 19794-7, *Information technology — Biometric data interchange formats — Part 7: Signature/sign time series data*

### 4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19784-1 and the following apply.

NOTE Function names and data element names are not included here, but are defined within the body of this part of ISO/IEC 19784.

4.1

**Biometric Sensor Function Provider**

**BSFP**

BioAPI component, attached to a BSP via its interface, for managing and providing information from a biometric sensor

NOTE The biometric sensor can be a camera, a fingerprint device, an iris scanner, or a signature reader using images, pressure, or sound, or any other type of biometric sensor.

4.2

**Biometric Sensor Function Provider Interface**

**BSFPI**

BSP-to-BSFP interface that supports the functions needed to provide biometric data and to manage the BSFP itself and the associated device

4.3

**packet**

fragment of a BDB passing across the SFPI as the result of a single BioSFPI\_GetPackets call

4.4

**streaming data**

data passing across an interface from a source that is operating continuously

5 Symbols, abbreviated terms, data structure definitions and error codes

iTeh STANDARD PREVIEW  
(standards.iteh.ai)

5.1 Symbols and abbreviated terms

**API** – Application Programming Interface

**BDB** – Biometric Data Block <https://standards.iteh.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-8fcfde6faca4/iso-iec-19784-4-2011>

**BFP** – BioAPI Function Provider

**BSP** – Biometric Service Provider

**BSFPI** – Biometric Sensor Function Provider Interface

**FPI** – Function Provider Interface

**ID** – Identity/Identification/Identifier

**SPI** – Service Provider Interface

**sRGB** – Standard Red Green Blue (see ITU-R BT.709-5)

**UUID** – Universally Unique Identifier

5.2 Data structure definitions and error codes

For the purposes of this document, the data structure definitions and error codes defined in ISO/IEC 19784-1 apply.



## 6 Interface architecture

**6.1** ISO/IEC 19784-1:2006 specifies the interface at the top of the BioAPI Framework, which a biometric application uses to interact with BSPs, and through that to biometric sensors either directly or through BFPs (see Figure 1 and Figure 2 of ISO/IEC 19784-1:2006).

**6.2** The BSFPI contains two types of functions:

**6.2.1** General BFP management functions, which are called by the BSP to manage the BFP and its associated BioAPI Units. These functions are not directly related to SPI functions.

- BioSFPI\_BSFPLoad (see Clause 7)
- BioSFPI\_UnitAttach (called in the context of BioAPI\_BSPAttach and BioSPI\_BSPAttach, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_UnitDetach (called in the context of BioAPI\_BSPDetach and BioSPI\_BSPDetach, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_QueryUnits (this function is directly related to the SPI function BioSPI\_QueryUnits which is related to the API function BioAPI\_QueryUnits, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_Free (the relation of this function to the corresponding SPI call depends on the behaviour of the BSP – when the BSP copies data that the BFP has allocated memory for, then it can immediately send BioSFPI\_Free to the BFP, otherwise the BSP will call this function when itself is requested to free the memory for this data)

**6.2.2** Unit management functions, which are called by the BSP to manage BioAPI Units directly. These functions are directly related to SPI functions.

- BioSFPI\_ControlUnit (this function is directly related to the SPI function BioSPI\_ControlUnit which is related to the API function BioAPI\_ControlUnit, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_Cancel (this function can be called in relation to the corresponding calls of the SPI or the application, but can also be called when there is no corresponding request from the BioAPI framework or application, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_SetPowerMode (this function is directly related to the SPI function BioSPI\_SetPowerMode which is related to the API function BioAPI\_SetPowerMode, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_SetGUICallbacks (this function is directly related to the SPI function BioSPI\_SetGUICallbacks which is related to the API function BioAPI\_SetGUICallbacks, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_SetIndicatorStatus (this function is directly related to the SPI function BioSPI\_SetIndicatorStatus which is related to the API function BioAPI\_SetIndicatorStatus, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_GetIndicatorStatus (this function is directly related to the SPI function BioSPI\_GetIndicatorStatus which is related to the API function BioAPI\_GetIndicatorStatus, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)
- BioSFPI\_CalibrateSensor (this function is directly related to the SPI function BioSPI\_CalibrateSensor which is related to the API function BioAPI\_CalibrateSensor, see Clauses 8 and 9 of ISO/IEC 19784-1:2006)

The following functions are important for the interface between a BSP and an SFP, but do not map directly to any function across the SPI between a BSP and the framework (or an application for frameworkless operation).

- BioSFPI\_QueryFormatInfo (this function is not related to the SPI)
- BioSFPI\_ActivateSensor (this function is not related to the SPI)
- BioSFPI\_GetPackets (this function is not related to the SPI)
- BioSFPI\_DataTransfer (this function is not related to the SPI)
- BioSFPI\_SubscribeToGUIEvents (this function is not related to the SPI)
- BioSFPI\_UnsubscribeFromGUIEvents (this function is not related to the SPI)

## **7 Selecting and loading BFPs and BioAPI\_Units**

### **7.1 Obtaining information about BFPs and BioAPI\_Units**

The component registry contains information about the BioAPI framework, BSPs and BFPs (see 6.3 of ISO/IEC 19784-1:2006).

BFP information is stored in a structure called BioAPI\_BFP\_SCHEMA (see 7.3 of ISO/IEC 19784-1:2006). The BioAPI\_BFP\_SCHEMA contains data elements BFPPropertyID and BFPProperty. The BFPPropertyID is a UUID identifying the type of data structure of the BFPProperty data.

Each BFP that is compliant with ISO/IEC 19784-1 and this part of ISO/IEC 19784 has to provide a standardized BFPPropertyID and a related BFPPropertySchema. Two BFPPropertyIDs and two related BFPPropertySchemas are defined in Annexes A and B. Annex A defines a BFPPropertySchema suitable for image devices, and Annex B defines one suitable for signature devices. These can be used in ISO/IEC 19784-1 calls as BFPPropertyIDs and BFPPropertySchemas.

The information given in the BFPPropertySchema is stored at BFP installation time (see 6.4 of ISO/IEC 19784-1:2006) in the component registry.

An application can obtain this information by calling BioAPI\_EnumBFPs and BioAPI\_QueryBFPs (see 8.1.10 and 8.1.11 of ISO/IEC 19784-1:2006).

A BSP can obtain this information by using the BioSPI\_BFP\_ENUMERATION\_HANDLER (see 9.2.2 of ISO/IEC 19784-1:2006).

An application can retrieve information about all BioAPI\_Units supported by a particular BSP by calling the function BioAPI\_QueryUnits (see 8.1.9 of ISO/IEC 19784-1:2006). This function also works if no attach session is established yet.

A BSP can browse the component registry for all installed BFPs by using the callback mechanism of BioSPI\_BFP\_ENUMERATION\_HANDLER. The BSP analyses the BFPPropertySchemas and decides which BFPs it can support.

Note: The criteria for that decision are out of the scope of this part of ISO/IEC 19784.

During the call BioSPI\_QueryUnits the BSP may load the supported BFPs. A loaded BFP automatically detects the supported BioAPI\_Units and reports the BioAPI\_UNIT\_SCHEMAS in the return of the BioSFPI\_QueryUnits call to the BSP, which reports them to the application.

The BSP may create a list of all BFPUuids and the UnitIDs of the supported devices for each BFP. Additional information can be found in the parameter *UnitManagerUuid* of the BioAPI\_UNIT\_SCHEMA.

## 7.2 Loading BFPs

### 7.2.1 Loading BFPs when attaching to an unspecified BioAPI\_Unit

This sub-clause describes how a BFP is loaded if the application does not choose a specific BioAPI\_Unit in a call of BioAPI\_Attach. (If the application selects a specific BioAPI\_Unit, see 7.2.2).

In this case the application calls the function BioAPI\_BSPAttach with the value BioAPI\_DONT\_CARE in the particular element of the parameter *UnitList*.

The BSP then performs the operation described in 7.1 using the callback mechanism of BioSPI\_BFP\_ENUMERATION\_HANDLER and followed by a call of BioSFPI\_QueryUnits. As a result, the BSP has a list of all units supported and their supporting BFPs. The BSP chooses one of the BioAPI\_Units and then loads the appropriate BSFP and completes the operation with a BioSFPI\_BSFPLoad function call followed by a BioSFPI\_BSFPAttach function call.

Note: The criteria used to choose one of the supported BioAPI\_Units are not defined in this part of ISO/IEC 19784.

### 7.2.2 Loading BFPs when attaching to a specific BioAPI\_Unit

After issuing a BioAPI\_BSPLoad, the framework will load the BSP (if supported) and issue a BioSPI\_BSPLoad.

On return, the application calls the function BioAPI\_QueryUnits to determine the units supported by that BSP, and then calls BioAPI\_Attach giving the specific Unit it wishes to use. The application retrieves the UnitSchemaArray from the BioAPI\_QueryUnits and can browse this for the BioAPI\_Unit information. The application selects the *UnitId* it wants to attach to and calls the function BioAPI\_BSPAttach. The BSP may decide to load an appropriate BFP at this time depending on either the list being built as described at the end of 7.1 or by analysing the parameter *UnitManagerUuid* of the BioAPI\_UNIT\_SCHEMA.

If the BSP can support at least one BSFP supporting the requested Unit, it proceeds as specified in 7.1. If it cannot, it will give an error return. If there are multiple choices of BSFP for the selected Unit, the BSP makes a choice (the criteria for doing this are not standardised).

## 8 BSFPI Definition

### 8.1 BSFPI data structures

#### 8.1.1 Control codes

The function BioSFPI\_ControlUnit uses 32bit values as codes to indicate a specific operation to be performed. To avoid conflicts between standardized and vendor-specific codes these control codes are structured.

The upper 16 bits are used to structure the purpose of control codes. The lower 16 bits are used to identify control codes of each category.

The following three categories are standardized:

```
#define BioSFPI_BFPControlCode (0x8000)
#define BioSFPI_BioAPIUnitControlCode (0x4000)
#define BioSFPI_VendorSpecificControlCode (0x1000)
```

All other category values are reserved for future use.

### 8.1.2 BioSFPI\_EventHandler

This sub-clause defines the event handler interface to receive asynchronous notification of events of type BioAPI\_EVENT from a BioAPI Unit. Example events include insertion or removal of a BioAPI Unit (e.g. insertion or withdrawal of a biometric sensor device).

This event handler is passed to the BSFP during BioSFPI\_BFPLoad. This is the single event handler that all BioAPI Units managed by this BSFP should use to notify the BSP of event types that occur in a loaded BSFP.

```
typedef BioAPI_RETURN (BioAPI *BioSFPI_EventHandler)  
    (const BioAPI_UUID *BSFPUuid,  
     BioAPI_UNIT_ID UnitID,  
     BioAPI_UNIT_SCHEMA *UnitSchema,  
     BioAPI_EVENT EventType);
```

*BSFPUuid (input)* - The UUID of the BSFP raising the event.

*UnitID (input)* – The unit ID of the BioAPI Unit (biometric sensor device) associated with the event.

*UnitSchema (input)* – The schema of the BioAPI Unit raising the event.

*EventType (input)* - The BioAPI\_EVENT that has occurred.

### 8.1.3 BioSFPI\_GUI\_RESPONSE

An enumeration of the possible actions to be performed by a BFP after a GUI event notification callback made by the BFP has returned control to the BFP. Before returning from a notification callback, a BSP assigns a value of this type to an output parameter of the callback (Response).

```
typedef uint8_t BioSFPI_GUI_RESPONSE; ISO/IEC 19784-4:2011  
https://standards.iteh.ai/catalog/standards/sist/6b777a06-6dc6-41f4-8e41-  
#define BioSFPI_GUI_RESPONSE_PROGRESS_CONTINUE (1) std/iso/iec-19784-4-2011  
#define BioSFPI_GUI_RESPONSE_PROGRESS_ABORT (2)
```

The value BioSFPI\_GUI\_RESPONSE\_PROGRESS\_CONTINUE can only be returned in response to a GUI progress event notification callback. It indicates that the BFP should continue performing the biometric capture.

The value BioSFPI\_GUI\_RESPONSE\_PROGRESS\_ABORT can only be returned in response to a GUI progress event notification callback. It indicates that the BFP should abort the biometric capture and produce an error code. After this response from the BSP, an error return from the original call is expected from the BFP.

If a BSP returns a response value other than those permitted in each situation, the BFP shall abort the biometric capture and cause the original function call to return an error code. No further GUI event notification callbacks are expected from the BFP for the same function.

### 8.1.4 BioSFPI\_GUI\_PROGRESS\_EVENT\_HANDLER

#### Callback function

```
typedef BioAPI_RETURN (BioAPI *BioSFPI_GUI_PROGRESS_EVENT_HANDLER)  
    (BioAPI_UNIT_ID UnitID,  
     const void *GUIProgressEventHandlerCtx,  
     const BioAPI_GUI_BITMAP_ARRAY *Bitmaps,  
     BioSFPI_GUI_RESPONSE *Response);
```

#### Description

This is a function pointer type for a BSP's GUI event handler function that is to handle GUI progress event notification callbacks coming from a BFP. In order to receive GUI progress event notifications, a BSP shall register a callback function of type BioSFPI\_GUI\_PROGRESS\_EVENT\_HANDLER by providing the callback address of the function, along with a context address, in a call to BioSFPI\_SubscribeToGUIEvents (see 8.2.16).

The framework makes a callback to an application function of this type each time it receives an incoming callback to a function of type `BioSPI_GUI_PROGRESS_EVENT_HANDLER` which it exposes to a BFP. The parameters of the callback (except `GUIProgressEventHandlerCtx`) shall be set from the parameters of the incoming callback with the same names; the parameter `GUIProgressEventHandlerCtx` shall be set from the GUI progress context address originally provided by the BSP in its call to `BioSFPI_SubscribeToGUIEvents`; and the callback address shall be set from the GUI progress callback address originally provided by the BSP in the call to `BioSFPI_SubscribeToGUIEvents`.

A BFP can generate GUI progress events only during the execution of a call to `BioSFPI_DataTransfer`, `BioSFPI_Play` or any other function that involves a biometric capture. They can be generated at any time, even multiple times, during any biometric capture.

The main purpose of the GUI progress event notification is to send to the BSP streaming data (as a series of bitmaps) collected by the biometric sensor. One possible use of this information is for the BSPs to show the bitmaps to the subject or an operator. The BSP shall respond to each GUI progress notification by indicating whether the biometric capture should continue or abort (see 8.1.3).

The GUI progress event handler function and any functions invoked (directly or indirectly) by that function shall not call any `BioSFPI` function.

If the BSP returns a value different from `BioAPI_OK`, the BFP shall cancel the current biometric capture and produce an error code.

#### Parameters

*UnitID (input)* – The unit ID of the biometric sensor unit of the BSP to which the GUI event is related.

*GUIProgressEventHandlerCtx (input)* – The context address originally provided by the subscriber application as part of the GUI event subscription.

*Bitmaps (input)* – An array of bitmaps containing image(s) of the samples captured in the biometric capture, which are intended for display to the subject or to an operator. This array usually contains zero or one bitmaps, but may contain multiple bitmaps if the BFP has the ability to capture multiple instances of a biometric modality in a single capture operation.

*Response (output)* – A value indicating the response from the BSP to the GUI select event notification (see 8.1.3).

#### Return Value

A `BioAPI_RETURN` value indicating success or specifying a particular error condition. The value `BioAPI_OK` indicates success. All other values represent an error condition.

#### Errors

`BioAPIERR_USER_CANCELLED`  
`BioAPIERR_FUNCTION_FAILED`  
`BioAPIERR_FUNCTION_NOT_SUPPORTED`

## 8.2 BSFP Functions

### 8.2.1 BioSFPI\_BSFP\_Load

#### **BioAPI\_RETURN BioAPI BioSFPI\_BSFP\_Load**

```
(const BioAPI_UUID *BSFPUuid,  
BioSFPI_EventHandler BioSFPINotifyCallback);
```