



# Information technology — Biometric application programming interface —

## Part 2: Biometric archive function provider interface

### TECHNICAL CORRIGENDUM 1

*Technologies de l'information — Interface de programmation d'applications biométriques —*

*Partie 2: Interface du fournisseur de fonction d'archives biométriques*

*RECTIFICATIF TECHNIQUE 1*

Technical Corrigendum 1 to ISO/IEC 19784-2:2007 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

Page 6 5.2.1

Replace the Description section with the following:

#### **Description**

This function initializes a BFP. Initialization includes registering the BSP event handler for the specified BFP and enabling all events. The BSP can choose to provide an event handler function to receive notification of events. Many BSPs can independently and concurrently load the same BFP, and each BSP can establish its own event handler. They will all receive notification of an event. The same or different event handlers can be used if a BSP loads multiple BFPs.

A BSP may establish as many event handlers as it wishes, for a given BFP, by calling BioAFPI\_BAFPLoad one or more times for that BFP. An event handler is identified by the address of the notification.

When an event occurs in a BFP, the BFP may send an event notification to the BSP by calling the BSP's event handler.

If a BSP has set up multiple event handlers, they shall be called one at a time (in any order chosen by the BFP) rather than concurrently.

Event notifications may occur at any time, either during a BioSPI call (related or unrelated to the event) or while there is no BioSPI call in execution. BSP writers should ensure that all callbacks are properly and safely handled by the BSP, no matter when the BSP receives them.

NOTE This usually requires the use of thread synchronization techniques and discipline in the actions performed by the BSP code placed in event handlers.

There is a "use count" on the establishment of event handlers; they have to be de-established (by BioAFPI\_BAFPLoad) as many times as they were established. When a BFP is loaded (BioAFPI\_BAFPLoad), it shall raise an "insert" event immediately for each present BioAPI Unit. This will indicate to the BSP that it can go ahead and do a BioAFPI\_UnitAttach. If the hardware component for a specific functionality is not present, the "insert" event cannot be raised until the hardware component has been plugged in.

The function BioAFPI\_UnitAttach can be invoked multiple times for each call to BioAFPI\_BAFPLoad.

A callback is used to notify the BSP of events of type BioAPI\_EVENT. The BAFP shall retain this information for later use.

NOTE See Annex A for additional information on loading BFPs.

Page 24

Insert the following annex after 5.2.19:

## Annex A (informative)

### Selecting and loading BFPs and BioAPI\_Units

#### A.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 BFPPPropertyID and BFPPProperty. The BFPPPropertyID is a UUID identifying the type of data structure of the BFPPProperty data.

Each BFP that is compliant with ISO/IEC 19784-1 and this part of ISO/IEC 19784 must provide a standardized BFPPPropertyID and a related BFPPPropertySchema.

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 **BioAFPI\_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*.

## A.2 Loading BFPs

### A.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\_BSPAttach**. (If the application selects a specific *BioAPI\_Unit*, see A.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 A.1 using the callback mechanism of

*BioSPI\_BFP\_ENUMERATION\_HANDLER* and followed by a call of **BioAFPI\_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 BAFP and completes the operation with a **BioAFPI\_BAFPLoad** function call followed by a **BioAFPI\_BAFPAttach** function call.

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

### A.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\_BSPAttach** giving the specific *Unit* it wishes to use. More precisely, 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 A.1 or by analysing the parameter *UnitManagerUuid* of the *BioAPI\_UNIT\_SCHEMA*.

On return, the application calls the function **BioAPI\_QueryUnits** to determine the units supported by that BSP, and then calls **BioAPI\_BSPAttach** giving the specific *Unit* it wishes to use. More precisely, the