TouchChip®

TouchChip TFM/ESS Fingerprint BSP for Linux Specification

Table of Contents

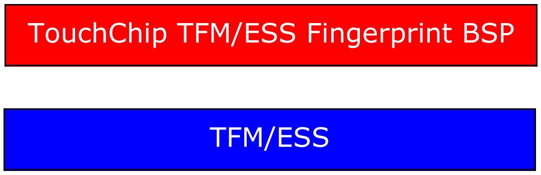
Introduction

The BioAPI specification creates a framework for Biometric identification using various technologies from Biometric Service Providers (BSP's). This document describes the TouchChip TFM/ESS FingerPrint BSP. It is a fingerprint-based identification service that is compliant with the BioAPI specification version 1.1. For more information about BioAPI and the API specification, please visit **http://www.bioapi.org**.

The TouchChip TFM/ESS FingerPrint BSP is BioAPI-compliant Verification BSP developed based on the BioAPI Reference Implementation. It is capable of performing 1:1 matching (or authentication), but not 1:N identification matching.

The TouchChip TFM/ESS FingerPrint BSP is a monolithic BSP; the client/server mode is not supported. It is platform dependent.

The TouchChip TFM/ESS FingerPrint BSP is based on UPEK PerfectTrust® and PerfectMatch® technology. It requires a TFM/ESS fingerprint reader that is licensed for use with this BSP.



TouchChip TFM/ESS Fingerprint BSP

TFM/ESS

Requirements

TouchChip TFM/ESS Fingerprint BSP for Linux needs Linux system installed on PC platform with these requirements:

- Libusb (already installed by most distributions)
- Usbfs (USB Device File system, already installed by most distributions)
- BioAPI 1.10 framework (official download source is at the BioAPI web pages http://www.bioapi.org, however alternative version of BioAPI from http://www.qrivy.net/~michael/blua/bioapi/ is recommended, because it fixes problems with makefile)

Installation and uninstallation

Installation

Extract the BSP package and run the install.sh script. This script installs libtfmessbsp.so file which is in the same directory. If you have installed BioAPI in non-standard directory (which is `/usr/local/lib)`, specify it as a parameter. If you use BioAPI in limited environment where Xserver is not available, install libtfmessbsp.so from console_only/ directory.

Uninstallation

Execute uninstall.sh script which was created while installation.

Running BioAPI applications under standard user account

To install BioAPI for all users we recommend following:

- Install BioAPI framework into the standard location
- Install TouchChip TFM/ESS Fingerprint BSP
- Give access to the BioAPI config directory
  chmod 777 -R /usr/local/var/bioapi/
- Some distributions need to change access rights to the USB device. Directories in the path /proc/bus/usb and subdirectories in this path should have access rights for reading and executing (searching) for everyone (755) and files in this subtree should have access rights for reading and writing for everyone (666).

This configuration allows all users to run BSP applications without root privileges.

Configuration

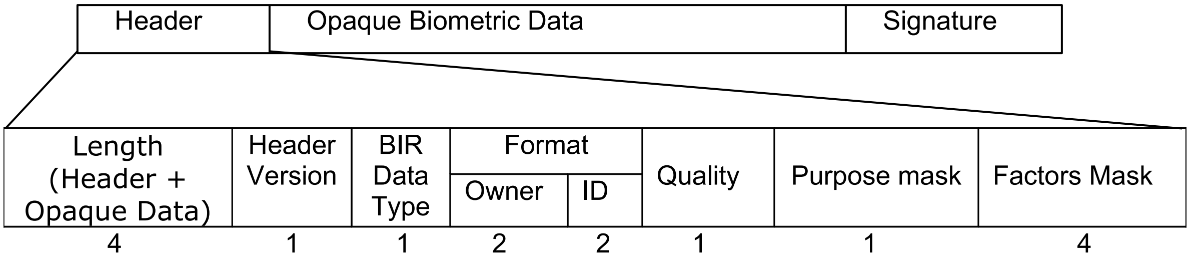
Behavior of this library may be modified by configuration file. This file is stored at /etc/tfmessbsp.cfg, it is created and filled by default values on the first use. For security reasons it is not possible to override this settings by user settings (~/etc/tfmessbsp.cfg is not used at all).
Configuration items:

- `hide-capture-success:` set this item to 1 to disable showing success at the end of BioSPI_Capture(). This may be useful when the application calls BioSPI_Capture() to get user's template and then calls BioSPI_VerifyMatch().

- `security-level:` change security level. Possible values are 1, 2, 3, 4 or 5. See chapter FAR and FRR for details.

TouchChip TFM/ESS FingerPrint BSP BIR

The term **biometric identification record** (BIR) refers to any biometric data that is returned to the application, including raw data, intermediate data, processed sample(s) ready for verification or identification, as well as enrollment data. Typically, the only data stored persistently by the application is the BIR generated for enrollment (i.e. the template). The structure of a BIR is shown below, with the size (in bytes) shown underneath.



Header	Opaque Biometric Data	Signature

Length (Header + Opaque Data)	Header Version	BIR Data Type	Format		Quality	Purpose mask	Factors Mask
			Owner	ID			
4	1	1	2	2	1	1	4

Biometric Identification Record (BIR)

Header data for the TouchChip TFM/ESS FingerPrint BSP BIR are described here:

Format Owner is 0x0012. It has been registered with the BioAPI consortium. Format ID is 0x0.

Factors Mask is BioAPI_FACTOR_FINGERPRINT.

Header Version is 0x1.

Quality is –2. It is not supported by this version.

The TouchChip TFM/ESS FingerPrint BSP supports two purposes:
BioAPI_PURPOSE_VERIFY and BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY.

The TouchChip TFM/ESS FingerPrint BSP supports two BIR Data Types:
BioAPI_BIR_DATA_TYPE_PROCESSED for the biometric templates, used for enrollment and verification. The templates are not signed or encrypted.
BioAPI_BIR_DATA_TYPE_RAW for the raw image data, used for audit purpose. The raw image data is encrypted. This format is not supported by Linux implementation of TouchChip TFM/ESS FingerPrint BSP.

In the TouchChip TFM/ESS FingerPrint BSP, a BIR can be used in one of two ways:
a) By reference to its handle
b) By the BIR itself.

For case a) the memory is allocated inside the BSP, and the application must free the allocated memory by using the BioAPI_FreeBIRHandle function.

Notes:

1. BioAPI_BIR_DATA_TYPE_RAW can be obtained from the functions BioAPI_Capture, BioAPI_Verify and BioAPI_Enroll. If BioAPI_Enroll succeeds, the highest quality image used in template generation is returned.
2. Processing BIR with type BioAPI_BIR_DATA_TYPE_RAW is not supported by this BSP, it cannot be used as an input for any function supported in BSP.

Component Schema

The Component schema describes capabilities of a biometric service provider (BSP) module and attributes of the biometric device attached.

TouchChip TFM/ESS FingerPrint BSP Schema

Field Name	Field Data	Comment
BSPID STRING UUID	"UPEK TFM/ESS BSP"	Uniquely identifying BSP
DeviceID	0	4 byte device ID of attached sensor device
Description	"TouchChip TFM/ESS Fingerprint BSP"	Text descriptive name of the BSP
Path		Path where BSP executable is located
BSPName	"libtfmessbsp.so"	Filename of BSP Module
Spec Version	1.0	BioAPI Specification Version string e.g. 2.0
Product Version	1.0	BSP product version string e.g. 2.0
Vendor	"UPEK, Inc."	Service provider vendor name in text
Supported Formats	{{0x0012,0x0000}}	An array of 2-byte integer pairs. Each pair specifying a supported biometric data format.
Factors Mask	BioAPI_FACTOR_FINGERPRINT	A mask which indicates what forms of authentication are supported
Operations	BioAPI_ENROLL | BioAPI_CAPTURE | BioAPI_CREATETEMPLATE| BioAPI_PROCESS | BioAPI_VERIFYMATCH | BioAPI_VERIFY	Operations supported by service provider (BioAPI_ OPERATIONS_MASK)
Options	BioAPI_APP_GUI | BioAPI_PAYLOAD	Options supported by the BSP BioAPI_OPTIONS_MASK
Payload Policy	0	Minimum FAR value used to determine when to release a payload.
Max Payload Size	150	Maximum size in bytes of a payload
Default Verify Timeout	120000	Default timeout value (in milliseconds) used by a BSP for verify operations when no timeout is set by the application
Default Identify Timeout	120000	Default timeout value (in milliseconds) used by a BSP for identify operations when no timeout is set by the application.
Default Capture Timeout	120000	Default timeout value (in milliseconds) used by a BSP for capture operations when no timeout is set by the application.
Default Enroll Timeout	120000	Default timeout value (in milliseconds) used by a BSP for enroll operations when no timeout is set by the application.
MAX BSP DB size	0	Max size of a BSP owned (internal) database.
MaxIdentify	0	Largest population supported by Identify function.

TouchChip TFM/ESS FingerPrint BSP Biometric Device Schema

The information in the biometric device registry entry is updated each time a biometric device is attached to or removed from the service provider.

Field Name	Field Data Type	Comment
ModuleID	"UPEK TFM/ESS BSP"	UUID (in string format) uniquely identifying service provider module
DeviceID	0	4 byte device ID
Supported Formats	{{0x0012,0x0000}}	BIR Formats supported by BSP + device See BioAPI_BIR_BIOMETRIC_DATA_FORMAT
Supported Events	0	A BioAPI_MODULE_EVENT_MASK indicating which types of events are supported
Device Vendor	"UPEK, Inc."	Text name of device Vendor
Device Description	"TFM/ESS Fingerprint Device"	Text description of the biometric device
Device Serial Number	""	Serial Number of biometric device
Device Hardware Version	1.0	Device hardware version string (in dotted high/low format – e.g. 2.0).
Device Firmware Version	1.0	Device Firmware version string (in dotted high/low format – e.g. 2.0).
Authenticated Device	FALSE	An indication of whether the device has been authenticated.

Functions supported

The following table is a summary of functions supported the TouchChip TFM/ESS FingerPrint BSP.

Function	Verification BSP Mandatory	TouchChip TFM/ESS FingerPrint BSP Implementation
Module Management Functions		
BioSPI_ModuleLoad	X	X
BioSPI_ModuleUnload	X	X
BioSPI_ModuleAttach	X	X
BioSPI_ModuleDetach	X	X
Handle Functions		
BioSPI_FreeBIRHandle	X	X
BioSPI_GetBIRFromHandle	X	X
BioSPI_GetHeaderFromHandle	X	X
Callback and Event Functions		
BioSPI_EnableEvents		
BioSPI_SetGUICallbacks		X
BioSPI_SetStreamCallback		
BioSPI_StreamInputOutput		
Biometric Functions		
BioSPI_Capture		X
BioSPI_CreateTemplate		X
BioSPI_Process		X
BioSPI_VerifyMatch		X
BioSPI_IdentifyMatch		
BioSPI_Enroll	X	X
BioSPI_Verify	X	X
BioSPI_Identify		
BioSPI_Import		
BioSPI_SetPowerMode		
Database Functions		
BioSPI_DbOpen		
BioSPI_DbClose		
BioSPI_DbCreate		
BioSPI_DbDelete		
BioSPI_DbSetCursor		
BioSPI_DbFreeCursor		
BioSPI_DbStoreBIR		
BioSPI_DbGetBIR		
BioSPI_DbGetNextBIR		
BioSPI_DbQueryBIR		
BioSPI_DbDeleteBIR		

Note: Module Management Functions are exported directly from the TouchChip TFM/ESS FingerPrint BSP. All other functions are exported by using function pointer with BioSPI_ModuleAttach.

Optional Sub-functions

Capability	Supported	Not Supported
Return of raw/audit data	X	
Return of quality		X
Application-controlled GUI	X	
GUI streaming callbacks		X
Detection of source presence		X
Payload carry	X	
BIR signing		X
BIR encryption		X
Return of FRR		X
Model adaptation		X
Binning		X
Client/server communications		X
Supports self-contained device		X

FAR and FRR

There are two possible criteria for the results of a match: False Accept Rate (FAR) and False Reject Rate (FRR). FAR is the probability that samples falsely match the presented template, whereas FRR is the probability that the samples are falsely rejected (i.e. what should match, doesn't).

The BioAPI specification provides the application with the possibility to request FAR or FRR thresholds, and the possibility to have the FAR and FRR achieved by the matcher returned to the application.

The thresholds are specified by a 32-bit integer value (N) that indicates a probable False Accept Rate of N/(2**31-1).

Similarly, A 32-bit integer value (N) indicates a probable False Accept Rate of N/(2**31-1).

The TouchChip TFM/ESS FingerPrint BSP functions allow the application to request a match threshold in terms of maximum FAR value (i.e., a limit on the probability of a false match,) and an optional maximum FRR value. If both are provided, the application must tell the BSP which one should take precedence.

The TouchChip TFM/ESS FingerPrint BSP internally selects one of 5 security levels, based on the requested FAR/FRR levels. This selection is done on the basis of past matcher performance on certain large databases. The following table shows the security level selected based on the requested FAR or FRR values. If both values are requested, the one taking precedence will prevail, and the other will be ignored.

FAR		FRR		Security level
2148	(0.000001)	654983	(0.001902)	`V_MATCH_MAX_SEC`
310419	(0.000145)	500364	(0.000233)	`V_MATCH_HIGHER_SEC`
500364	(0.000233)	464931	(0.000217)	`V_MATCH_MEDIUM_SEC`
2515777	(0.001172)	358630	(0.000167)	`V_MATCH_LOWER_SEC`
4083440	(0.001902)	346819	(0.000162)	`V_MATCH_MIN_SEC`

Security level	Value
`V_MATCH_MIN_SEC`	1
`V_MATCH_LOWER_SEC`	2
`V_MATCH_MEDIUM_SEC`	3
`V_MATCH_HIGHER_SEC`	4
`V_MATCH_MAX_SEC`	5

Because the achieved values of FAR and FRR cannot be determined for any single invocation of the matcher, the achieved values always return 0.

V_MATCH_MEDIUM_SEC is used by default. To override change `security-level in configuration file tfmessbsp.cfg.`

User Interface

The TouchChip TFM/ESS FingerPrint BSP comes with a built-in user interface. It also allows the application to control the “look and feel” of the user interface by allowing the application to provide callback for the TouchChip TFM/ESS FingerPrint BSP.

GUI callback function

A callback function allows the TouchChip TFM/ESS FingerPrint BSP to indicate GUI state information to the application, and to receive responses back. The applications must supply the callback function defined as below:

```
typedef BioAPI_RETURN (BioAPI *BioAPI_GUI_STATE_CALLBACK)
(void *GuiStateCallbackCtx,
BioAPI_GUI_STATE GuiState,
BioAPI_GUI_RESPONSE Response,
BioAPI_GUI_MESSAGE Message,
BioAPI_GUI_PROGRESS Progress,
BioAPI_GUI_BITMAP_PTR SampleBuffer);
```

Parameters:

> *GuiStateCallbackCtx (input)* - A generic pointer to context information that was provided by the original requester and is being returned to its originator.
>
> *GuiState (input)* – an indication of the current state of the service provider with respect to the GUI, plus an indication of what others parameters are available.
>
> *Response (output)* – The response from the application back to the service provider on return from the Callback (this parameter cannot be used; it is a bug in the BioAPI specification version 1.1; see notes).
>
> *Message (input/optional)* – The number of a message to display to the user. Message numbers are service-provider dependent. *GuiState* indicates if a *Message* is provided; if not the parameter is NULL.
>
> *Progress (input/optional)* – A Value that indicates (as a percentage) the amount of progress in the development of a Sample/BIR. The value may be used to display a progress bar. Not all service providers support a progress indication. *GuiState* indicates if a sample *Progress value* is provided in the call; if not the parameter is NULL.
>
> *SampleBuffer (input/optional)* – The current sample buffer for the application to display. *GuiState* indicates if a sample *Buffer* is provided; if not the parameter is NULL.

Note: The current implementation of the TouchChip TFM/ESS FingerPrint BSP supports only the *Message.* If (*GuiState* & BioAPI_GUI_MESSAGE) *Message* is provided from BSP.

An application can cancel an active TFM/ESS BSP operation by returning the BioAPI_CANCEL code (as a return value) from a GUI callback function. To continue an active operation in a normal way an application must return the BioAPI_OK code.

Note: This behavior is not according to the actual BioAPI specification version 1.1 but this solution bypasses a bug in the BioAPI specification version 1.1 which makes impossible to cancel an active operation via a GUI callback from an application.

GUI callback messages definition

The following table is a summary of messages defined in the TouchChip TFM/ESS FingerPrint BSP. They are used to notify the application the status of the TouchChip TFM/ESS FingerPrint BSP and used by application to build the "look and feel" of the user interface.

Message	Code	Meaning	Used
TFMESS_GUIMSG_GOOD_IMAGE	0	Scanned good image	X
TFMESS_GUIMSG_NO_FINGER	1	No finger detected	X
TFMESS_GUIMSG_TOO_LIGHT	2	Finger image is too light	
TFMESS_GUIMSG_TOO_DRY	3	Finger is too dry	
TFMESS_GUIMSG_TOO_DARK	4	Finger image is too dark	
TFMESS_GUIMSG_TOO_HIGH	5	Finger is too high	X
TFMESS_GUIMSG_TOO_LOW	6	Finger is too low	X
TFMESS_GUIMSG_TOO_LEFT	7	Finger is too left	X
TFMESS_GUIMSG_TOO_RIGHT	8	Finger is too right	X
TFMESS_GUIMSG_TOO_SMALL	9	Finger image is too small	
TFMESS_GUIMSG_TOO_STRANGE	10	Finger image is too strange	X
TFMESS_GUIMSG_BAD_QUALITY	11	Finger has bad quality	X
TFMESS_GUIMSG_PUT_FINGER	12	Put finger 1st time	X
TFMESS_GUIMSG_PUT_FINGER2	13	Put finger 2nd time	X
TFMESS_GUIMSG_PUT_FINGER3	14	Put finger 3rd time	X
TFMESS_GUIMSG_REMOVE_FINGER	15	Remove finger	X
TFMESS_GUIMSG_CONSOLIDATION_FAIL	16	Consolidation failed	X
TFMESS_GUIMSG_CONSOLIDATION_SUCCEED	17	Consolidation succeed	X
TFMESS_GUIMSG_CLEAN_SENSOR	18	Clean the sensor	
TFMESS_GUIMSG_KEEP_FINGER	19	Keep finger on the sensor	X
TFMESS_GUIMSG_START	20	Non specific GUI start	X
TFMESS_GUIMSG_VERIFY_START	21	Verification GUI start	X
TFMESS_GUIMSG_ENROLL_START	22	Enrollment GUI start	X
TFMESS_GUIMSG_FINGER_DETECT_START	23	Finger detect GUI start	
TFMESS_GUIMSG_GUI_FINISH	24	Non specific GUI end	X
TFMESS_GUIMSG_GUI_FINISH_SUCCEED	25	Operation Succeeded	X
TFMESS_GUIMSG_GUI_FINISH_FAIL	26	Operation Failed	X
TFMESS_GUIMSG_CALIB_START	27	Sensor calibration start	
TFMESS_GUIMSG_TOO_FAST	28	Swipe too fast	X
TFMESS_GUIMSG_TOO_SKEWED	29	Swipe too skewed	X
TFMESS_GUIMSG_TOO_SHORT	30	Swipe too short	X
TFMESS_GUIMSG_TOUCH_SENSOR	31	Touch sensor with finger	
TFMESS_GUIMSG_PROCESSING_IMAGE	32	Processing image	X
TFMESS_GUIMSG_SWIPE_IN_PROGRESS	33	Swipe in progress	X
TFMESS_GUIMSG_BACKWARD_MOVEMENT	34	Backward movement detected	
TFMESS_GUIMSG_JOINT_DETECTED	35	Finger joint detected	
TFMESS_GUIMSG_CENTER_AND_PRESS_HARDER	36	Center finger and press harder	

TouchChip TFM/ESS FingerPrint BSP Specified Error Codes

Error code	**Value**	**Meaning**
BioAPIERR_TFMESS_ACTION_CANCELLED_BY_USER	0x1801	Action cancelled by user
BioAPIERR_TFMESS_INVALID_BIR_HANDLE	0x1802	Invalid BIR handle
BioAPIERR_TFMESS_AUDIT_DATA_ENCRYPTION_ERROR	0x1803	Audit data encryption failure
BioAPIERR_TFMESS_PM_INIT_FPFLTR_FAIL	0x1866	InitFpfltr failure
BioAPIERR_TFMESS_PM_INIT_MATCHER_FAIL	0x1867	InitMather failure
BioAPIERR_TFMESS_PM_SET_SECURITY_LEVEL_FAIL	0x1868	SetSecurityLevel failure
BioAPIERR_TFMESS_PM_MATCH_FAIL	0x1869	matchprints failure
BioAPIERR_TFMESS_PT_GENERAL_ERROR	0x18CA	General or unknown error status
BioAPIERR_TFMESS_PT_API_NOT_INIT	0x192E	PerfectTrust API wasn't initialized
BioAPIERR_TFMESS_PT_API_ALREADY_INITIALIZED	0x192F	PerfectTrust API has been already initialized
BioAPIERR_TFMESS_PT_INVALID_PARAMETER	0x18CD	Invalid parameter error
BioAPIERR_TFMESS_PT_INVALID_HANDLE	0x1931	Invalid handle error
BioAPIERR_TFMESS_PT_NOT_ENOUGH_MEMORY	0x1932	Not enough memory to process given operation
BioAPIERR_TFMESS_PT_MALLOC_FAILED	0x1933	Failure of extern memory allocation function
BioAPIERR_TFMESS_PT_DATA_TOO_LARGE	0x1934	Passed data are too large
BioAPIERR_TFMESS_PT_NOT_ENOUGH_PERMANENT_MEMORY	0x1935	Not enough permanent memory to store data
BioAPIERR_TFMESS_PT_MORE_DATA	0x1936	There is more data to return than the supplied buffer can contain
BioAPIERR_TFMESS_PT_FUNCTION_FAILED	0x194D	Function failed
BioAPIERR_TFMESS_PT_INVALID_INPUT_BIR_FORM	0x1950	Invalid form of PT_INPUT_BIR

		structure
BioAPIERR_TFMESS_PT_WRONG_RESPONSE	0x1951	TFM has returned wrong or unexpected response
BioAPIERR_TFMESS_PT_NOT_ENOUGH_TFM_MEMORY	0x1952	Not enough memory on TFM to process given operation
BioAPIERR_TFMESS_PT_ALREADY_OPENED	0x18CC	Connection is already opened
BioAPIERR_TFMESS_PT_CANNOT_CONNECT	0x1954	Cannot connect to TFM
BioAPIERR_TFMESS_PT_TIMEOUT	0x1955	Timeout elapsed
BioAPIERR_TFMESS_PT_BAD_BIO_TEMPLATE	0x1956	Bad biometric template
BioAPIERR_TFMESS_PT_SLOT_NOT_FOUND	0x1957	Requested slot was not found
BioAPIERR_TFMESS_PT_ANTISPOOFING_EXPORT	0x1958	Attempt to export antispoofing info from TFM
BioAPIERR_TFMESS_PT_ANTISPOOFING_IMPORT	0x1959	Attempt to import antispoofing info to TFM
BioAPIERR_TFMESS_PT_ACCESS_DENIED	0x18C9	Access to operation is denied
BioAPIERR_TFMESS_PT_NO_TEMPLATE	0x195D	No template was captured in current session
BioAPIERR_TFMESS_PT_BIOMETRIC_TIMEOUT	0x195E	Timeout for biometric operation has expired
BioAPIERR_TFMESS_PT_CONSOLIDATION_FAILED	0x185F	Failure of template consolidation
BioAPIERR_TFMESS_PT_BIO_OPERATION_CANCELED	0x1801	Biometric operation canceled
BioAPIERR_TFMESS_PT_AUTHENTIFICATION_FAILED	0x18EE	Authentification failed
BioAPIERR_TFMESS_PT_UNKNOWN_COMMAND	0x1962	Unknown command
BioAPIERR_TFMESS_PT_GOING_TO_SLEEP	0x1963	Power off attempt failed
BioAPIERR_TFMESS_PT_NOT_IMPLEMENTED	0x1964	Function or

		service is not implemented
BioAPIERR_TFMESS_PT_COMM_ERROR	0x1965	General communication error
BioAPIERR_TFMESS_PT_SESSION_TERMINATED	0x1966	Session was terminated
BioAPIERR_TFMESS_PT_TOUCH_CHIP_ERROR	0x1967	Touch chip error occured
BioAPIERR_TFMESS_PT_I2C_EEPROM_ERROR	0x1968	I2C EEPROM error occured
BioAPIERR_TFMESS_PT_INVALID_PURPOSE	0x1969	Purpose parameter (or BIR's purpose) is invalid for given operation
BioAPIERR_TFMESS_PT_SWIPE_TOO_BAD	0x196A	Finger swipe is too bad for image reconstruction
BioAPIERR_TFMESS_PT_NOT_SUPPORTED	0x18CE	Requested functionality or value of parameter is not supported
BioAPIERR_TFMESS_PT_CALIBRATION_FAILED	0x196C	Calibration failed
BioAPIERR_TFMESS_PT_ANTISPOOFING_NOT_CAPTURED	0x196D	Antispoofing data were not captured
BioAPIERR_TFMESS_PT_LATCHUP_DETECTED	0x196E	Sensor latch-up event detected
BioAPIERR_TFMESS_PT_DIAGNOSTICS_FAILED	0x196F	Diagnostics failed
BioAPIERR_TFMESS_PT_SAME_VERSION	0x1970	Attempt to upgrade to same firmware version
BioAPIERR_TFMESS_PT_NO_SENSOR	0x1971	No sensor
BioAPIERR_TFMESS_PT_SENSOR_OUT_OF_LIMITS	0x1972	The measured values are out of allowable limits
BioAPIERR_TFMESS_PT_TOO_MANY_BAD_LINES	0x1973	Too many bad lines
BioAPIERR_TFMESS_PT_SENSOR_NOT_REPAIRABLE	0x1974	Sensor is not repairable
BioAPIERR_TFMESS_PT_GAIN_OFFSET	0x1975	Gain offset calibration error
BioAPIERR_TFMESS_PT_POWER_SHUTOFF	0x1976	Asynchronous power shut down

BioAPIERR_TFMESS_PT_OLD_VERSION	0x1977	Attempt to upgrade to older firmware version
BioAPIERR_TFMESS_PT_SUSPEND	0x1978	Connection interrupted because of suspend request
BioAPIERR_TFMESS_PT_DEVICE_NOT_FOUND	0x18CB	Device not found
BioAPIERR_TFMESS_PT_DEVICE_SICK	0x18DF	Device doesn't work as expected
BioAPIERR_TFMESS_PT_UNSUPPORTED_SPEED	0x197B	Host hardware doesn't support requested communication speed
BioAPIERR_TFMESS_PT_SENSOR_NOT_CALIBRATED	0x197C	Sensor is not calibrated
BioAPIERR_TFMESS_PT_SAFE_MODE	0x197D	Firmware is missing or corrupted, device is running in safe mode
BioAPIERR_TFMESS_PT_SENSOR_HW_ERROR	0x197E	Sensor hardware error occured
BioAPIERR_TFMESS_PT_SESSION_NOT_AUTHENTICATED	0x197F	Session was not authenticated yet
BioAPIERR_TFMESS_PT_SECURE_CHANNEL_ALREADY_ESTABLISHED	0x1980	Secure channel has been already established
BioAPIERR_TFMESS_PT_OTP_SEQUENCE_NUMBER_OVERFLOW	0x1981	Overflow of One Time Password sequence number
BioAPIERR_TFMESS_PT_NVM_ERROR	0x1982	General NVM error
BioAPIERR_TFMESS_PT_NVM_CANNOT_WRITE	0x1983	NVM write operation failed
BioAPIERR_TFMESS_PT_NVM_CANNOT_READ	0x1984	NVM read operation failed
BioAPIERR_TFMESS_PT_NVM_INVALID_FILE_ID	0x1985	Attempt to access non-existing internal NVM file

BioAPIERR_TFMESS_PT_CRYPTO_ERROR	0x1986	General crypto error
BioAPIERR_TFMESS_PT_CRYPTO_MECHANISM_NOT_SUPPORTED	0x1987	Requested cryptographic mechanism is not supported
BioAPIERR_TFMESS_PT_CRYPTO_PADDING_ERROR	0x1988	Padding error detected during crypto operation
BioAPIERR_TFMESS_PT_CRYPTO_KEY_TOO_LONG	0x1989	Key too long (probably due to the export regulations)
BioAPIERR_TFMESS_PT_CRYPTO_SYM_BAD_KEY	0x198A	Bad symmetric key used
BioAPIERR_TFMESS_PT_HW_RNG_INIT_ERROR	0x198B	HW random number generator initialization failed
BioAPIERR_TFMESS_PT_SC_BASE	0x19F4	PT SC BSP error code base
BioAPIERR_TFMESS_PT_SC_ERROR	0x194F	General smart-card error
BioAPIERR_TFMESS_PT_SC_NOT_SUPPORTED	0x19F5	Communication with this card is not supported
BioAPIERR_TFMESS_PT_SC_COMM_FAIL	0x19F6	Failure during communication with the card
BioAPIERR_TFMESS_PT_SC_BAD_PARAM	0x19F7	Incorrect parameter detected
BioAPIERR_TFMESS_PT_SC_NO_CARD	0x19F9	The card is not present in the reader

Obsolete TouchChip TFM/ESS FingerPrint BSP Error Codes (for backward compatibility with existing TouchChip TFM/ESS FingerPrint BSP applications):

Error code	**Value**	**Meaning**
BioAPIERR_BSP_ACTION_CANCELLED_BY_USER	0x1801	Action cancelled by user
BioAPIERR_BSP_INVALID_BIR_HANDLE	0x1802	Invalid BIR handle
BioAPIERR_BSP_AUDIT_DATA_ENCRYPTION_ERROR	0x1803	Audit data encryption failure
BioAPIERR_BSP_PM_INIT_FPFLTR_FAIL	0x1866	InitFpfltr failure
BioAPIERR_BSP_PM_INIT_MATCHER_FAIL	0x1867	InitMather failure
BioAPIERR_BSP_PM_SET_SECURITY_LEVEL_FAIL	0x1868	SetSecurityLevel failure
BioAPIERR_BSP_PM_MATCH_FAIL	0x1869	matchprints failure
BioAPIERR_BSP_PT_NODEVICE	0x18CB	No device
BioAPIERR_BSP_PT_SENSOR_COMMUNICATION	0x18E5	Sensor communication error
BioAPIERR_BSP_PT_INVALID_LICENSE	0x18EE	Invalid sensor license

Information furnished is believed to be accurate and reliable. However, Upek, Inc assumes no responsibility for the consequences of use of such information not for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Upek, Inc. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Upek, Inc's products are not authorized for use as critical components in life support devices or systems without express written approval of Upek, Inc.

Upek, Inc.
http://www.upek.com