SDK for IdentiFI Product Series (30, 45, 45i, 50, 50i and 60) iOS API Reference

Version 1.6.0.0

Released on 2025-03-19

Table of Contents

Features

Release	Date	Description
1.6.0.0	2025-03-12	Fix the connect method which fails to connect on iOS 18.x, iOS Simulator library for Apple Mx computer, legacy firmware update integration.
1.5.0.1	2024-12-19	Documentation corrections.
1.5.0.1	2024-01-24	Updated IdentiFI-45i / IdentiFI-50i battery reading for unit with FW version 2.9.4 and above.
1.5.0.0	2023-12-20	Added latest Iris capture methods and callbacks.
1.4.0.0	2022-01-31	Change the way the IdentiFI save fingerprint template into the circular buffer using a new savedAtIndex parameter to provide more flexibility. The following existing methods/callback now have a savedAtIndex parameter: startCaptureOneFinger, startCaptureTwoFinger, startCaptureFourFinger, startCaptureRollFinger, clearSavedFpImages and onSavedFpImagesCleared. **Note: IdentiFI firmware version 2.5.0 and up needed.**
1.3.2.0	2021-06-08	Fixed getFirmwareVersion,getModelNumber and getDeviceSerialNumber callback processing.
1.3.1.0	2020-12-20	Changed connect and onConnectionTimeOut timing.
1.3.0.0	2020-10-05	We added the SDK the capability to update firmware using startFirmwareUpdate methode. Up on request, S.I.C. Biometrics will build a double encrypted file update with serial number lock. This firmware update methode offer more flexibility in the field and provide as well the capability to update other files in the IdentiFI reader, like Integrated Biometrics library.
1.3.0.0	2020-10-05	We added the capability to crop WSQ image using an optional croppedImage parameter for getWSQEncodedFpImageFromImageSavedAt method.
1.2.1.0	2020-06-02	We added support for IdentiFI-60.
1.2.1.0	2020-06-02	We added the capability to save battery power when no command activity is detected for a specified duration. Using the new setReaderSuspendMode: secondsToSuspend method, a time between 1 minutes to 24 hours can be configured and save to file. When configured, this mode will auto-disconnect the IdentiFI reader from the application and shutdown the fingerprint sensor power.
1.2.1.0	2020-06-02	We added the capability for an application to take over the IdentiFI's LED control using the new setLEDControlForPowerLED: FpLED: ComLED: mSecOn: mSecOff: method.
1.2.1.0	2020-06-02	We added the capability* for an application to set the IdentiFI LED brightness using the new setLEDBrightness: ledBrightness methode, which provide a range of 10 LED brigthness level. *May not work with customized IdentiFI LED brightness control circuit.
1.2.1.0	2020-06-02	Updated onCaptureError: -600 Duplicate extraction failed. Added securityLevel parameter to isFingerDuplicated method.
1.2.0.0	2020-04-14	We added support for IdentiFI-30.
1.2.0.0	2020-04-14	We changed the WSQ compression ratio based on IdentiFI model as follow: Compression ratio: 1:5 (bitRate = 2.25) for IdentiFI-30, IdentiFI-45 1:15 (bitRate = 0.75) for IdentiFI-50,IdentiFI-50.
1.2.0.0	2020-04-14	We added a new capability to the IdentiFI firmware to capture only fingerprint image that meat a minimum NFIQ score using the new setMinimumNFIQScore method. If the captured fingerprint image does not meat the minimum expected NFIQ score, the new onCaptureError callback is triggered. We also added a rolling smear detection witch calls the onCapture callback when rolling smear is detected in the final capture.
1.2.0.0	2020-04-14	We aded a new isFingerDuplicated: savedAtIndex: securityLevel: method which can verify if a specified saved fingerprint image is a duplicate from the other saved fingerprint images.
1.1.7.3	2020-02-17	We renamed the library from "IdentiFI-45" to "IdentiFI" to be less device specific. We added to the SDK a fat library that now provides capability to work with the Xcode Simulator.

Release History

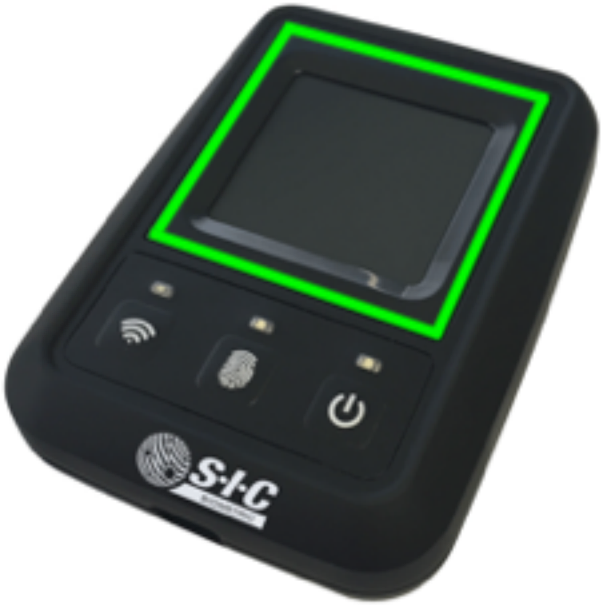
Release	Date	Description
1.6.0.0	2024-02-03	Fix the connect method which fails to connect on iOS 18.x Changed the startFirmwareUpdate(NSData *) newFirmware to add a legacy option for IdentiFI running with firmware version < v2.3.0. (void)startFirmwareUpdate:(NSData *) newFirmware toLegacyFirmware: (BOOL) legacyFirmwareUpdateMethod IMPORTANT! Some legacy firmware may not fire the onFirmwareTransferCompleted callback. In that specific situation, the user should power off the reader (pressing both Power button + WiFi button at the same time) to complet the firmware update on the next power on. Rebuild for Apple Mx IC iOS XcodeSimulator.
1.5.0.1	2024-01-24	Updated IdentiFI battery reading for IdentiFI-45i/-50i with FW version 2.9.4 and above.
1.5.0.0	2023-12-20	Add iris capture related methods for IdentiFI-45I/IdentiFI-50I support. onCaptureError renamed onFpCaptureStatus Changed setLEDControlForPowerLED: to add iris LED control Add setPowerOffMode:(int); and onSetPowerOffMode:(int); Add getPowerOffMode:(int); and onGetPowerOffMode:(int); Deprecated getReaderSuspendMode / onGetReaderSuspendMode. Deprecated setReaderSuspendMode / onSetReaderSuspendMode.
1.4.0.0	2022-01-31	Add a savedAtIndex parameter to the following existing methods/callback: startCaptureOneFinger, startCaptureTwoFinger, startCaptureFourFinger, startCaptureRollFinger, clearSavedFpImages and onSavedFpImagesCleared. Note: IdentiFI firmware version 2.5.0 and up needed.
1.3.2.0	2021-06-08	Fixed getFirmwareVersion, getModelNumber, getDeviceSerialNumber callback processing.
1.3.1.0	2020-12-20	Changed connect and onConnectionTimeOut timing.
1.3.0.0	2020-10-05	**New features available from IdentiFI Firmware v2.3.1 and up.** Added -(void) startFirmwareUpdate:(NSData *) newFirmware and -(void) onFirmwareTransferCompleted:(long) transferResult callback Added croppedImage parameter to getWSQEncodedFpImageFromImageSavedAt
1.2.2.0	2020-07-10	Fix onGetWSQEncodedFpImage callback issue.
1.2.1.0	2020-06-02	**New features available from IdentiFI Firmware v2.2.1 and up.** Added support for IdentiFI-60. Added - (void) setReaderSuspendMode: secondsToSuspend onSetReaderSuspendMode: secondsToSuspend Added - (void) getReaderSuspendMode onGetReaderSuspendMode: secondsToSuspend Added - (void) setLEDControlForPowerLED: FpLED: ComLED: mSecOn: mSecOff: onSetLEDControlForPowerLED: FpLED: ComLED: mSecOn: mSecOff: Added - (void) setLEDBrightness: ledBrightness onSetLEDBrightness: ledBrightness Added - (void) getLEDBrightness onGetLEDBrightness: ledBrightness Updated onCaptureError: -600 Duplicate extraction failed. Added securityLevel parameter to isFingerDuplicated method.

Release	Date	Description
1.2.0.0	2020-04-14	**New features available from IdentiFI Firmware v2.1.0 and up.** Added support for IdentiFI-30. Implemented WSQ compression ratio change based on IdentiFI model as follow: Compression ratio: 1:5 (bitRate = 2.25) for IdentiFI-30 1:15 (bitRate = 0.75) for IdentiFI-45,-50. Added setMinimumNFIQScore / onSetMinimumNFIQScore. Added onCaptureError for rolling smear detection and minimum NFIQ score. Added isFingerDuplicated / onIsFingerDuplicated.
1.1.7.3	2020-02-17	Renamed the library from "IdentiFI-45" to "IdentiFI" to be less device specific and merged with IdentiFI-50 related methods and callback. Added to the SDK a fat library that now provides capability to work with the Xcode Simulator.
1.1.7	2019-01-18	**New features available from IdentiFI Firmware v2.0.4 and up.** Implemented - (void) getSegmentedFpImageSavedAt:(int) savedAtIndex; - (void) onGetSegmentedFpImage_RAW:(NSData *) rawFpImageData fromImageSavedAt:(int)savedAtIndex;
1.1.6	2018-12-11	Fixed Connect / onConnection that would be raised on a second connect call while the connection is not yet completed. Change back the onConnectionTimeOut from 90 to 10 sec.
1.1.5	2018-11-15	All preview image size are now 1/4th (150 000 bytes/frame) of the original full size(600 000 bytes/frame).
1.1.4	2018-08-29	Special customer release. Increased onConnectionTimeOut from 10 to 90 sec.
1.1.3	2018-08-29	Changed the way onConnectionTimeOut is handle and increased the timeout delay from 6 to 10 sec. Add an new delegate - (void) onConnectionError:(NSString *) errorDescription;
1.1.2	2018-08-05	Fixed Cancel Capture issue.
1.1.1	2018-07-10	Replaced getBatteryLevel/ onGetBatteryLevel: by getBatteryPercentage/onGetBatteryPercentage:. The return charge level is a percentage value between 1% to 100% in steps of 5% resolution.
1.1.0	2018-07-01	All fingerprint capture process have been redesigned to reduce time per capture. WSQ and NFIQ score are no longer generated at the end of each capture, and no more part of the delegate LastFrame callback. Each fingerprint capture is now saved into the IdentiFI RAM (volatile) memory. Powering off the IdentiFI will lose all temporarily saved fp image capture. Up to 10 fingerprint images can be kept into the IdentiFI memory. At the end of a fingerprint capture, the delegate callback will receive a savedAtIndex value, between 0 to 9, indication at which memory slot the fp image capture is saved. Each WSQ encoded image and NFIQ score can the be retrieved at a later time using a savedAtIndex value range between 0 to 9. If all the 10 memory slots have been used (0 through 9), any additional fp capture will be saved, and overwrite previously save fp image at index 9.
1.0.2	2017-02-28	Added delegate method to return RAW fingerprint image and NFIQ score. - (void) onLastFrame:(NSData *) rawFpImageData withNFIQ: (int) : nfiqScore Note: Supported by IdentiFI firmware version 0.8.0 and up.
1.0.1	2017-01-14	Added fingerprint power management methods to extend battery power: - getFpPowerStatus / onGetFpPowerStatus - setFpPowerOn / onGetFpPowerOn - setFpPowerOff / onSetFpPowerOff Add WSQ fingerprint image encoding method: - getWSQEncodedFingerprintImage / onGetWSQEncodedFingerprintImage Note: Supported by IdentiFI firmware version 0.8.0 and up.
1.00	2016-12-04	Initial release.

Important operational note

The IdentiFI's fingerprint sensor is surrounded by metal frame (green on the image below). When a person places his fingertips on the sensor, it is important that the fingers also touch the metal frame. Doing so will generate a signal to the sensor that will be used to balance the contrast and brightness of the captured image for optimal quality.

If the fingers do not touch the frame, the sensor will not capture, nor will the image be displayed on your mobile device.



Getting Started

The IdentiFI iOS SDK was developed and fully tested for Xcode development platform. Using third-party development platform is really risky and will not be supported or will be support with support fees by SIC team, to its sole discretion.

Note 1: Naming convention

Throughout this document, the words "IdentiFI device" refer to IdentiFI-30, IdentiFI-45/45i, IdentiFI-50/50i and IdentiFI-60.

Note 2: The Send command / Wait for response rule.

The IdentiFI can only handle one command/process at a time. It's important that the calling application waits for a command delegate callback (response) to complete before it can call a new method (sends any new command to the IdentiFI).

The cancelFpCapture/cancelIrisCapture methods is the only exception to this rule, since it can be called to stop an ongoing capture process on the IdentiFI.

Note 3: IdentiFI circular buffer.

The IdentiFI device temporarily saves segmented fingerprint images into a circular buffer.

This circular buffer can hold up to 10 segmented fingerprint images.

When a fingerprint capture is completed, the *onLastFrame* callback returns a *fpImageSavedAt* value which is the last index position where the last segmented fingerprint image was saved.

When more than one finger is captured at a time (2 or 4), the segmentation process begins by saving the first finger on the left.

This *fpImageSavedAt* value can go from 0 to 9.

When 10 fingerprints memory spaces have been used (*fpImageSavedAt = 9)*, if a new capture is completed, the new fingerprint image will overwrite the oldest pre-existing fingerprint image and the *fpImageSavedAt* value will go back to 0.

The circular buffer content can be cleared at all time by calling *clearSavedFpImages*.

The circular buffer is using volatile memory, and the data is not persistent after powering off the IdentiFI.

If needed, the calling application can specify a savedAtIndex parameter when calling any of the startCapture… methods in order to control in which part of the circular buffer the resulting image(s) should be saved at.

Integrating the SDK in your Xcode project

- Open your existing or new Xcode project.
- From the menu File->Add Files to “YourProjectName…” add the .\lib\libIdentiFI.a file and the .\include\IdentiFI.h file to your project.
- In your YourControllerNameViewController.h header file add the following:

```
#import “IdentiFI.h”

@interface ViewController : UIViewController
{

   IdentiFI *myIdentiFI_Device;

…

}
```

- The library needs to be initialized before it’s used in YourControllerNameViewController.m. In the -(void) viewDidLoad method, after the [super viewDidLoad] call, add the following:

```
myIdentiFI_Device = [[IdentiFI alloc] init];  //init is the initializer method for IdentiFI class

[myIdentiFI_Device setDelegate:self];  //Set the IdentiFI’s class delegate for callback
```

- Add the methods you plan to use in YourControllerNameViewController.m, for the delegates to be called back:

```
//Communication Link
- (void) onConnection { }
- (void) onConnectionError:(NSString *) errorDescription { }
- (void) onConnectionTimeOut { }
- (void) onDisconnection { }
//Finger capture preview
- (void) onCancelFpCapture { }
- (void) onFpCaptureStatus: (int) fpCaptureStatus {}
- (void) onStreaming: (UIImage *) fpImage { }
- (void) onStreamingRolledFp: (UIImage *) fpImage rollingState:(int) currentState   verticalLineX:(int) currentXPosition { }
//Finger finale capture
- (void) onLastFrame:(UIImage *) fpImage  fpImageSavedAt: (int) savedAtIndex { }
- (void) onLastFrame_RAW: (NSData *)  rawFpImageData   fpImageSavedAt: (int) savedAtIndex { }
- (void) onLastFrameRolledFp:(UIImage *) fpImage fpImageSavedAt: (int) savedAtIndex { }
- (void) onLastFrameRolledFp_RAW: (NSData *)  rawFpImageData   fpImageSavedAt: (int) savedAtIndex { }
//IdentiFI's settings
- (void) onGetLEDBrightness:(int) ledBrightness { }
- (void) onSetLEDBrightness:(int) ledBrightness { }
- (void) onGetPowerOffMode:(int) seconds { }
- (void) onSetPowerOffMode:(int) seconds { }
- (void) onSetMinimumNFIQScore:(int) minimumNFIQScore { }
- (void) onSetReaderSuspendMode:(int) secondsToSuspend { }
//Saved fingerprint images.
- (void) onGetNfiqScore:(int) nfiqScore fromImageSavedAt: (int) savedAtIndex { }
- (void) onGetSegmentedFpImage_RAW:(NSData *) rawFpImageData fromImageSavedAt:(int)savedAtIndex { }
- (void) onGetWSQEncodedFpImage:(NSData *) wsqEncodedFpImageData fromImageSavedAt: (int) savedAtIndex { }
- (void) onIsFingerDuplicated:(int) isFingerDuplicated { }
- (void) onSavedFpImagesCleared { }
//Iris capture
- (void) onCancelIrisCapture { }
- (void) onIrisCaptureStatus:(int) irisCaptureStatus { }
- (void) onStreamingLeftIris:(UIImage *) leftIrisImage RightIris:(UIImage *) rightIrisImage { }
-(void) onLastFrameLeftIris:(UIImage *) leftIrisImage RightIris:(UIImage *) rightIrisImage
  LeftIrisTotalScore:(int) LTS LeftIrisUsableArea:(int) LUA RightIrisTotalScore:(int) RTS RightIrisUsableArea:(int) RUA { }
//Power management
- (void) onGetFpPowerStatus:(Boolean) fpPowerStatus { }
- (void) onSetFpPowerOn:(Boolean) fpPowerStatus { }
- (void) onSetFpPowerOff { }
//Device information
- (void) onGetBatteryPercentage:(int) percentLevel { }
- (void) onGetDeviceSerialNumber:(NSString*) serialNumber { }
- (void) onGetFirmwareVersion:(NSString *) version { }
- (void) onGetModelNumber:(NSString *) model { }
- (void) onGetReaderDescription:(NSString *) deviceDescription { }
//Keypad's LEDs control
- (void) onSetLEDControlForPowerLED:(int) pwr FpLED:(int) fp ComLED:(int) com IrisLED:(int) iris mSecOn:(int) mOn mSecOff:(int) mOff { }
```

- From that point on, once you have established a Wi-Fi connection link between the host and the IdentiFI device, your code needs to call the connect method like so:

```
[myIdentiFI_Device connect];
```

- Once the onConnection has been called and until the onDisconnection is called, your code can call the library methods described in this document. To close the link to the device connection, your code then needs to call

```
[myIdentiFI_Device disconnect];
```

IdentiFI's Methods and Delegates Callback Reference

Communication link

connect

- Description

Connects the instance to the remote IdentiFI device.
Prerequisite : A secured Wi-Fi link has been established between the host and the IdentiFI device

- Callback

```
- (void) onConnection;
- (void) onConnectionError:(NSString *) errorDescription;
- (void) onConnectionTimeOut;
```

onConnection()

- Description

Delegate method called one time at the beginning of a new session, received after the connect method has been called.
Prerequisite that a secured Wi-Fi link has been established between your host and the IdentiFI device.

onConnectionError:(NSString *) errorDescription

- Description

Delegate method called if a connection error occurs.

- Parameter

Parameter	Description
errorDescription	Error description.

onConnectionTimeOut()

- Description

Delegate method called one time after connect() call failed to connect to an IdentiFI within 15 seconds.

disconnect

- Description

Disconnects the instance from the remote IdentiFI device.

- Callback

The API provides one delegate callback related to this method:

```
- (void) onDisconnection;
```

onDisconnection()

- Description

Delegate callback method called when the Wi-Fi host is disconnected from the IdentiFI.

Finger capture

startCaptureOneFinger:(int) savedAtIndex
startCaptureTwoFinger:(int) savedAtIndex
startCaptureFourFinger:(int) savedAtIndex

- Description

These methods are used to start a new fingerprint capture for one, two or four fingers.

startCaptureTwoFinger can only be called on IdentiFI-45 and up. When using this method it is mandatory to place 2 fingers on the sensor to complete a finger capture process.

startCaptureFourFinger can only be called on IdentiFI-50 and up. When using this method it is mandatory to place 4 fingers on the sensor to complete a finger capture process.

- Parameter

Parameter	Description
savedAtIndex	The fingerprint circular buffer memory index value where the fingerprint image will be saved at. If the savedAtIndex value is -1, the fingerprint image(s) will be saved into the IdentiFI circular buffer next space from previous capture. If the target space already have a fingerprint, it will be delete and replace by the new capture. If the savedAtIndex value is between 0 and 9, the fingerprint image will be save starting at that index and existing fingerprint image will be deleted and replace by the new capture. Valid Range Value: -1, 0-9.

- Callback

The API provides a set of three delegate callbacks related to this method:

- (void) onStreaming: (UImage *) fpImage
- (void) onLastFrame: (UIImage *) fpImage fpImageSavedAt: (int) savedAtIndex
- (void) onLastFrame_RAW:(NSData *) rawFpImageData fpImageSavedAt: (int) savedAtIndex
- (void) onFpCaptureStatus: (int) fpCaptureStatus;

startCaptureRollFinger:(int) savedAtIndex

- Description

This method is used to start a new fingerprint capture with a single rolling finger.
Note: Only available on IdentiFI-45, IdentiFI-50 and IdentiFI-60.

- Parameter

Parameter	Description
savedAtIndex	The fingerprint circular buffer memory index value where the fingerprint image will be saved at. If the savedAtIndex value is -1, the fingerprint image will be saved into the IdentiFI circular buffer next space from previous capture. If the target space already have a fingerprint, it will be delete and replace by the new capture. If the savedAtIndex value is between 0 and 9, the fingerprint image will be save starting at that index and existing fingerprint image will be deleted and replace by the new capture. Valid Range Value: -1, 0-9.

- Callback

The API provides a set of three delegate callbacks related to this method:

- (void) onStreamingRolledFp:(UIImage *) fpImage rollingState:(int) state verticalLineX:(int) XPosition;
- (void) onLastFrameRolledFp:(UIImage *) fpImage fpImageSavedAt: (int) savedAtIndex;
- (void) onLastFrameRolledFp_RAW: (NSData *) rawFpImageData fpImageSavedAt: (int) savedAtIndex;
- (void) onFpCaptureStatus: (int) fpCaptureStatus;

onStreaming(UIImage *) fpImage

- Description

Delegate callback for the startCaptureOneFinger, startCaptureTwoFinger and startCaptureFourFinger methods.
This method is called many times per second while acquisition previews are ready to be displayed to the user.
The image is zoomed at 1/4th of the onLastFrame image.

- Parameter

Parameter	Description
fpImage	A preview of the most recent capture Image data size: 50 000 bytes for the IdentiFI-30. 150 000 bytes for the IdentiFI-45. 400 000 bytes for the IdentiFI-50. 600 000 bytes for the IdentiFI-60. fpImage.size.width: 200 for the IdentiFI-30, 400 for the IdentiFI-45, 800 for IdentiFI-50. 800 for IdentiFI-60. fpImage.size.heigh: 250 for the IdentiFI-30. 375 for the IdentiFI-45, 500 for the IdentiFI-50. 750 for the IdentiFI-60.

onStreamingRolledFp:(UIImage *) fpImage rollingState:(int) state verticalLineX:(int) xPosition;

- Description

Delegate callback for the startCaptureRollFinger method.
This method is called many times per second while acquisition previews are ready to be displayed to the user.
The image is zoomed out at 1/4th of the onLastFrame image.

- Parameter

Parameter	Description
fpImage	A preview of the most recent capture Image data size: 600 000 bytes fpImage.size.width: 800 for all IdentiFI fpImage.size.heigh: 750 for all IdentiFI
state	0: Flat finger not yet detected.(Typically no vertical guide is displayed) 1: Begin rolling. (Typically red vertical guide line) 2: Keep rolling. (Typically green vertical guide line)
xPosition	The X position of the vertical guide line to display. Value between 0 to 400.

onLastFrame:(UIImage *) fpImage fpImageSavedAt:(int) savedAtIndex;

- Description

Delegate callback for the startCaptureOneFinger, startCaptureTwoFinger and startCaptureFourFinger methods.

This delegate is called once at the end of a final capture.

- Parameter

Parameter	Description
fpImage	UIImage containing the final fingerprint capture. Image data size (.size.width x .height): 200 000 bytes (400 x 500) for the IdentiFI-30. 600 000 bytes (800 x 750) for the IdentiFI-45. 1 600 000 bytes (1600 x 1000) for the IdentiFI-50. 2 400 000 bytes (1600 x 1500) for the IdentiFI-60.
savedAtIndex	The memory index of where the fpImage was saved for later processing of the NFIQ score and WSQ encoding. Possible valide returned value range between 0 to 9.

onLastFrame_RAW: (NSData *) rawFpImageData fpImageSavedAt: (int) savedAtIndex;

- Description

Delegate callback for the startCaptureOneFinger, startCaptureTwoFinger and startCaptureFourFinger methods.

This delegate is called once at the end of a valid final capture.

- Parameter

Parameter	Description
rawFpImageData	NSData containing a RAW copy of the final fingerprint capture. RAW Image size (width x height): 200 000 bytes (400 x 500) for the IdentiFI-30. 600 000 bytes (800 x 750) for the IdentiFI-45. 1 600 000 bytes (1600 x 1000) for the IdentiFI-50. 2 400 000 bytes (1600 x 1500) for the IdentiFI-60.
savedAtIndex	The memory index of where the fpImage was saved for later processing of the NFIQ score and WSQ encoding. Possible valide returned value range between 0 to 9.

onLastFrameRolledFp:(UIImage *) fpImage fpImageSavedAt:(int) savedAtIndex;

- Description

Delegate callback for the startCaptureRollFinger method.

This delegate is called once at the end of a final capture.

- Parameter

Parameter	Description
fpImage	UIImage containing the final fingerprint capture.
savedAtIndex	The memory index of where the fpImage was saved for later processing of the NFIQ score and WSQ encoding. Possible valide returned value range between 0 to 9.

•

onLastFrameRolledFp_RAW: (NSData *) rawFpImageData fpImageSavedAt: (int) savedAtIndex;

- Description

Delegate callback for the startCaptureRollFinger method.

This delegate is called once at the end of a valid final capture.

- Parameter

Parameter	Description
rawFpImageData	NSData containing a RAW copy of the final fingerprint capture. RAW Image size (width x height): 600 000 bytes (800 x 750) for all IdentiFI models.
savedAtIndex	The memory index of where the fpImage was saved for later processing of the NFIQ score and WSQ encoding. Possible valide returned value range between 0 to 9.

onFpCaptureStatus: (int) fpCaptureStatus

- Description

Delegate callback for startCaptureOneFinger, startCaptureTwoFinger, startCaptureFourFinger and startCaptureRollFinger methods called when a capture error occurs.

If this callback is called, the last frame captured ended with an error code (described below).

* If the error is caused by the minimum NFIQ Score not being met (captureErrorCode = 1), no final fingerprint image will be returned and the capture process is restarted until a final capture is completed and meets the expected minimum NFIQ Score or until the cancelFpCapture( ) is called.

- Parameter

Parameter	Description
captureErrorCode	An int value with the following possible error codes. 1: Finger capture didn't meet minimum NFIQ score. 9: Failed to start FP capture, FP sensor not powered. 304: Rolling smear detected. 305: Rolled finger was shifted horizontally. 306: Rolled finger was shifted vertically. 307: Rolled finger was shifted both horizontally and vertically. 399: Segmentation error occurred. Capture will be restarted. -600: Failed to extract details for duplicate. nnn: Other errors or status code may exist; please refer to Integrated Biometrics documentation.

cancelFpCapture

- Description

This method is used to cancel an ongoing fingerprint capture on the current device

- Delegate callback method

The API provides one delegate callback related to this method:

- (void) onCancelFpCapture

onCancelFpCapture()

- Description

Delegate callback for the cancelFpCapture() method. This method is called each time a fingerprint capture is interrupted by the user.

IdentiFI settings

setMinimumNFIQScore:(int) minimumNFIQScore

- Description

This method is used to set the minimum NFIQ score that should be met by the finale frame capture. If this minimum NFIQ score is not met, an onFpCaptureStatus callback will be called and the capture process will be restarted.

Note: At power up, the IdentiFI device always sets the minimum NFIQ score to 0 by default.

- Parameter

Parameter	Description
minimumNFIQScore	Valid value range 0-5. 0: No minimum NFIQ score is tested and no error will be generated. 1-5: Minimum NFIQ score to be met on the finale capture.

- Delegate callback method

The API provides one delegate callback related to this method:

- (void) onSetMinimumNFIQScore:(int) minimumNFIQScore;

onSetMinimumNFIQScore:(int) minimumNFIQScore

- Description

Delegate callback for setMinimumNFIQScore method.

- Parameter

Parameter	Description
minimumNFIQScore	New minimum NFIQ score that will be used for future fingerprint captures. Not saved to file.

setLEDBrightness:(int) ledBrightness

- Description

This method is used to set the IdentiFI's keypad LED brightness.

This value is saved into the IdentiFI and persists after power-off.

- Parameter

Parameter	Description
LEDBrightness	LED brightness level. Valid value range [1 - 10]. 1: Lowest keypad's LED brightness. 10: Highest keypad's LED brightness.

- Callback

The API provides one delegate callback related to this method:

- (void) onSetLEDBrightness:(int) ledBrightness;

onSetLEDBrightness:(int) ledBrightness

- Description

Delegate callback for setLEDBrightness method.

- Parameter

Parameter	Description
ledBrightness	Saved value for the keypad LEDs brightness.

getLEDBrightness

- Description

This method is used to get the IdentiFI's keypad LED brightness.

This value is saved into the IdentiFI memory and persists after power-off.

- Callback

The API provides one delegate callback related to this method:

- (void) onGetLEDBrightness:(int) ledBrightness;

onGetLEDBrightness:(int) ledBrightness

- Description

Delegate callback for getLEDBrightness method.

- Parameter

Parameter	Description
ledBrightness	Saved value for the keypad LEDs brightness.

Saved fingerprint images

clearSavedFpImages:(int) savedAtIndex

- Description

This method is used to clear(delete) fingerprint image saved in the IdentiFI reader.

- Parameter

Parameter	Description
savedAtIndex	The fingerprint circular buffer memory index value where the fingerprint image was previously saved at. If the savedAtIndex value is -1, all fingerprint images saved into the IdentiFI reader are deleted. If the savedAtIndex value is between 0 and 9, only the specified fingerprint image will be deleted. Valid Range Value: -1, 0-9.

- Callback

The API provides one callback related to this method:

- (void) onSavedFpImagesCleared();

onSavedFpImagesCleared:(int) savedAtIndex

- Description

Delegate callback for the clearSavedFpImages:(int) savedAtIndex method received once the specified fingerprint image(s) has(ve) been erased from memory.

- Parameter

Parameter	Description
savedAtIndex	-1: All fingerprint images saved into the IdentiFI reader were deleted. 0-9: The index of the fingerprint image that was deleted.

getNfiqScoreFromImageSavedAt:(int) SavedAtIndex

- Description

This method is used to retrieve the NFIQ score from a saved fingerprint image from a previous capture saved into the IdentiFI memory.

- Parameter

Parameter	Description
savedAtIndex	The memory index value where the fingerprint image was previously saved at. Valid Range Value: 0-9.

- Callback

The API provides one delegate callback related to this method:

- (void) onGetNfiqScore:(int) nfiqScore fromImageSavedAt: (int) savedAtIndex;

onGetNfiqScore:(int) nfiqScore fromImageSavedAt: (int) savedAtIndex

- Description

Delegate callback for the getNfiqScoreFromImageSavedAt method.

- Parameter

Parameter	Description
nfiqScore	NFIQ score of the saved image that was requested. Note: 0 mean that no image was found at the requested index.
savedAtIndex	The index of the previously saved fingerprint image for which the NFIQ score was requested.

getSegmentedFpImageSavedAt: (int) savedAtIndex;

- Description

This method is used to retrieve a saved segmented fingerprint image from a previous capture saved into the IdentiFI's memory.

- Parameter

Parameter	Description
savedAtIndex	The memory index value where the fingerprint image was previously saved at. Valid Range Value: 0-9

- Callback

The API provides one delegate callback related to this method:

- (void) onGetSegmentedFpImage_RAW:(NSData *) rawFpImageData fromImageSavedAt: (int) savedAtIndex;

onGetSegmentedFpImage_RAW:(NSData *) rawFpImageData fromImageSavedAt:(int) savedAtIndex

- Description

Delegate callback for the getSegmentedFpImageSavedAt method.

- Parameter

Parameter	Description
rawFpImageData	NSData containing a RAW copy of the final fingerprint capture. The RAW format is 600000 bytes = Width:800 X Heigth:750.
savedAtIndex	The memory index of saved fingerprint images for which the WSQ encoding was requested. Possible valid returned value range between 0 to 9.

getWSQEncodedFpImageFromImageSavedAt: (int) savedAtIndex croppedImage:(BOOL) croppedImage;

- Description

This method is used to retrieve the WSQ encoded copy of the fingerprint capture previously saved into the IdentiFI memory.

- Parameter

Parameter	Description
savedAtIndex	The memory index value where the fingerprint image was previously saved at. Valid Range Value: 0-9.
croppedImage	A boolean value indicating if the returned WSQ image should be cropped or not.

- Callback

The API provides one delegate callback related to this method:

- (void) onGetWSQEncodedFpImage:(NSData *) wsqFpImageData fromImageSavedAt: (int) savedAtIndex;

onGetWSQEncodedFpImage:(NSData *) wsqFpImageData fromImageSavedAt:(int) savedAtIndex

- Description

Delegate callback for the getWSQEncodedFpImageFromImageSavedAt method.

- Parameter

Parameter	Description
wsqFpImageData	WSQ encoded requested saved image. Compression ratio: 1:5 (bitRate = 2.25) for IdentiFI-30 1:15 (bitRate = 0.75) for IdentiFI-45,-50,-60. A wsqEncodedFpImageData of length = 0 means that no image was found at the requested index.
savedAtIndex	The memory index of requested WSQ image. Possible valide returned value range between 0 to 9.

isFingerDuplicated:(int) savedAtIndex securityLevel:(int) securityLevel;

- Description

This method is used to verify if the finger saved at savedAtIndex is a duplicate from the other fingers saved into the IdentiFI's memory.

- Parameter

Parameter	Description
savedAtIndex	The memory index value where the fingerprint image was previously saved at. Valid Range Value: 0-9.
securityLevel	The SecurityLevel designates how "strict" the similarity criteria between prints should be for them to be considered duplicates. Value of 1 matches fewer minutiae than a value of 7 which requires the highest number. Therefore, a value of 1 provides the least consideration for whether a duplicate finger is present whereas a value of 7 is the strictest. Valid Range Value: 1-7.

- Callback

The API provides one delegate callback related to this method:

- (void) onIsFingerDuplicated:(int) isFingerDuplicated;

onIsFingerDuplicated:(int) isFingerDuplicated

- Description

Delegate callback for the isFingerDuplicated method.

Note: This callback may take up to 6.5 seconds to be called with duplicate check results when no duplicate exist and testing with 10 registered flat fingers. For rolled finger in the same conditions, it may take up to 12.75 seconds to be called.

- Parameter

Parameter	Description
isFingerDuplicated	Valid value range: -1: No image saved at the savedAtIndex requested for duplicate check. 0: No duplicate found for finger saved as savedAtIndex. 1: Finger saved at savedAtIndex is a duplicate.

Iris capture

startCaptureIris()

- Description

This method is used to start a new iris capture.
Note: The iris sensor must be powered and initialized by calling setIrisPowerOn and then waiting for the callback onSetIrisPowerOn before startCaptureIris method can be called for the first time.

- Callback

The API provides a set of two callbacks related to this method:

```
- void onStreamingIris(Bitmap LeftIrisBm, Bitmap RightIrisBm)

- void onLastFrameIris(Bitmap LeftIrisBm, Bitmap RightIrisBm,
                      int LeftIrisTotalScore, int LeftIrisUsableArea,
                      int RightIrisTotalScore, int RightIrisUsableArea)
```

onIrisCaptureStatus:(unsigned int) irisCaptureStatus

- Description

Callback for the StartCaptureIris method.
This method is called to provide error or status code on the current iris capture process.

- Parameter

Parameter	Description
irisCaptureStatus	1: Capture will begin. 2: Eyes are detected! Keep moving. 3: Capture completed. 5: Capture Aborted error, Iris sensor capture stopped and powered off. 9: Failed to start Iris capture, Iris sensor not powered. nnn: Other IriTech errors or status code may exist; please refer to IriTech documentation.

onStreamingLeftIris:(UIImage *) leftIrisImage RightIris:(UIImage *) rightIrisImage

- Description

Callback for the StartCaptureIris functions.
This callback function is call when an acquisition is ready to be previewed by the user.

Parameter	Description
leftIrisImage	An acquisition of the most recent capture (320 x 240)
rightIrisImage	An acquisition of the most recent capture (320 x 240)

- Parameter

onLastFrameLeftIris:(UIImage *) leftIrisImage RightIris:(UIImage *) rightIrisImage LeftIrisTotalScore:(int) LTS LeftIrisUsableArea:(int) LUA RightIrisTotalScore:(int) RTS RightIrisUsableArea:(int) RUA;

- Description

Callback for the StartCaptureIris.
This callback function is call one time at the end of a completed iris capture.

- Parameter

Parameter	Description
leftIrisImage	Finale iris capture full size 640 x 480.
rightIrisImage	Finale iris capture full size 640 x 480.
LTS	Left iris total score
LUA	Left iris usable area
RTS	Right iris total score
RUA	Right iris usable area

cancelIrisCapture

- Description

This method is used to cancel an iris capture on the current device

- Callback function

The API provides one callback related to this function:

- (void) onCancelFpCapture

onCancelIrisCapture

- Description

Callback function for the cancelIrisCapture() method. This method is called each time an iris capture is interrupted by the user.

Power management

getFpPowerStatus

- Description

This method is used to query the power status of the IdentiFI's fingerprint sensor.

Note: Upon Wi-Fi link connection to the IdentiFI, the fingerprint sensor is powered up.
Upon disconnection of the Wi-Fi link, the fingerprint sensor is powered down.

- Callback

The API provides one delegate callback related to this method:

- (void) onGetFpPowerStatus:(Boolean *) fpPowerStatus

onGetFpPowerStatus:(Boolean) fpPowerStatus

- Description

Delegate callback for the getFpPowerStatus() method.

- Parameter

Parameter	Description
fpPowerStatus	A boolean value of the fingerprint sensor power status. False: Fingerprint sensor power is off. True: Fingerprint sensor power is on and ready to capture.

setFpPowerOn

- Description

This method is used to power up the IdentiFI's fingerprint sensor.

Note: Upon Wi-Fi link connection to the IdentiFI, the fingerprint sensor is powered up.
Upon disconnection of the Wi-Fi link, the fingerprint sensor is powered down.

- Callback

The API provides one delegate callback related to this method:

- (void) onSetFpPowerOn:(Boolean *) fpPowerStatus

onSetFpPowerOn:(Boolean) fpPowerStatus

- Description

Delegate callback for the setFpPowerOn() method.

- Parameter

Parameter	Description
fpPowerStatus	A boolean value of the fingerprint sensor power status. False: Fingerprint sensor power is off. True: Fingerprint sensor power is on and ready to capture.

setFpPowerOff

- Description

This method is used to power down the IdentiFI's fingerprint sensor.

Note: Upon Wi-Fi link connection to the IdentiFI, the fingerprint sensor is powered up.
Upon disconnection of the Wi-Fi link, the fingerprint sensor is powered down.

- Callback

The API provides one delegate callback related to this method:

- (void) onSetFpPowerOff

onSetFpPowerOff

- Description

Delegate callback for the setFpPowerOff() method.

getIrisPowerStatus

- Description

  This method is used to query the power status of the IdentiFI's iris sensor.

  Note: Calling the disconnect() method does power down the iris sensor.

- Callback

  The API provides one callback related to this method:

  onGetIrisPowerStatus:(Boolean) irisPowerStatus

onGetIrisPowerStatus:(Boolean) irisPowerStatus

- Description

  Callback function for the getIrisPowerStatus() method.

- Parameter

Parameter	Description
irisPowerStatus	A boolean value of the fingerprint sensor power status. False: Iris sensor power is off. True: Iris sensor power is on and ready to capture.

setIrisPowerOn

- Description

  This method is used to power up the IdentiFI's iris sensor.

  Note: Calling the disconnect() method does power down the iris sensor.

- Callback

  The API provides one callback related to this method:

  void onSetIrisPowerOn:(Boolean) irisPowerStatus

onSetIrisPowerOn:(Boolean) irisPowerStatus

- Description

  Callback for the setIrisPowerOn() method.

- Parameter

Parameter	Description
irisPowerStatus	A boolean value of the iris sensor power status. False: Failed to set the iris sensor power to on. True: Iris sensor power is on and ready to capture.

SetIrisPowerOff

- Description

  This method is used to power down the IdentiFI's iris sensor.

  Note: Calling the disconnect() method does power down the iris sensor.

- Callback

  The API provides one callback related to this method:

  - (void) onSetFpPowerOff

onSetIrisPowerOff

- Description

  Callback for the setIrisPowerOff() method which mean that the iris sensor power is now off.

setPowerOffMode:(int) SecondsToPowerOff

- Description

  This method is used to configure the IdentiFI self power-off mode when the reader is left unused.

- Parameter

Parameter	Description
SecondsToPower-Off	0: Disabled the self power-off mode of the IdentiFI. 1 - < 60: Set to default minimum value of 60 seconds. >= 60 : As is or set to the maximum accepted of 86400. Valid Range Value: 0-86400.

- Callback

  The API provides one callback related to this method:

  - (void) onSetPowerOffMode(int SecondsToPowerOff)

onSetPowerOffMode:(int) SecondsToPowerOff;

- Description

  Callback for the SetPowerOffMode() method, return the value for the self power-off saved into the unit.

- Parameter

Parameter	Description
SecondsToPower-Off	0: Power-off mode disabled. 60 - 86400: Idle unused seconds before the unit self power-off.

getPowerOffMode

- Description

  This method is used to get the IdentiFI saved self power-off mode value when left unused.

- Callback

  The API provides one callback related to this method:

  - (void) onGetPowerOffMode: (int) powerOffModeSeconds

onGetPowerOffMode: (int) powerOffModeSeconds

- Description

  Delegate callback for the getPowerOffMode( ) method, return the value for the self power-off saved into the unit.

- Parameter

Parameter	Description
powerOffModeSeconds	0: The self power-off mode of the IdentiFI is disabled. 60-86400: Number of seconds before the IdentiFI self power-off if no new command are sent.

Device information

getBatteryPercentage

- Description

This method is used to get the IdentiFI's battery charge level.

- Delegate callback method

The API provides one delegate callback related to this method:

- (void) onGetBatteryPercentage(int) level

onGetBatteryPercentage:(int) percentLevel

- Description

Delegate callback for the getBatteryPercentage() method.

- Parameter

Parameter	Description
percentLevel	An integer value that contains the percentage of charge left in the IdentiFI's battery. Range of the returned value is from 1 to 100 (%), by steps of 5%. Ex: 1,5,10….,95,100

getDeviceSerialNumber

- Description

This method is used to get the serial number of the connected IdentiFI.

- Delegate callback method

The API provides one delegate callback related to this method:

- (void) onGetDeviceSerialNumber:(NSString *) serial

onGetDeviceSerialNumber:(NSString *) serialNumber

- Description

Delegate callback for the getDeviceSerialNumber() method.

- Parameter

Parameter	Description
serialNumber	A NString that contains the unique ID of the connected device.

getFirmwareVersion

- Description

This method is used to get the firmware version of the connected IdentiFI.

- Delegate callback method

The API provides one delegate callback related to this method:

- (void) onGetFirmwareVersion:(NSString *) firmwareVersion

onGetFirmwareVersion:(NSString *) firmwareVersion

- Description

Delegate callback for the getFirmwareVersion() method.

- Parameter

Parameter	Description
firmwareVersion	A NString that contains the firmware version of the connected device

(NSString *) getLibraryVersion

- Description

Returns the version of the libIdentiFI.a in a NSString *.

getModelNumber

- Description

This method is used to get the model number of the connected IdentiFI.

- Delegate callback method

The API provides one delegate callback related to this method:

-(void) onGetModelNumber:(NSString *) model

onGetModelNumber:(NSString *) model

- Description

Delegate callback for the getModelNumber() method.

- Parameter

Parameter	Description
model	A NString containing the model number of the connected device

getReaderDescription

- Description

This method is used to get information about the connected IdentiFI and the fingerprint module.

- Delegate callback method

The API provides one delegate callback related to this method:

- (void) onGetReaderDescription(NSString *) deviceDescription

onGetReaderDescription(NSString *) deviceDescription

- Description

Delegate callback for the getReaderDescription() method.

- Parameter

Parameter	Description
deviceDescription	A NString that contains the hardware description of the connected device

Keypad LED control

setLEDControlForPowerLED: power FpLED: fp ComLED: com IrisLED: com mSecOn: mOn mSecOff: mOff

- Description

This method is used to control the IdentiFI's keypad LEDs.
The LED blinking cycle has a maximum duration of 2 seconds.

mOn + mOff total duration must be <= 2000 msec.

- Parameter

Parameter	Description
(int) power	Power LED blinking value. Valid value range [0-3]. 0: LED is off. 1: LED will be blinking green. 2: LED will be blinking amber. 3: LED will be blinking red.
(int) fp	Fingerprint LED blinking value. Valid value range [0-3]. 0: LED is off. 1: LED will be blinking green. 2: LED will be blinking amber. 3: LED will be blinking red.
(int) com	Com LED blinking value. Valid value range [0-1]. 0: LED is off. 1: LED will be blinking blue.
(int) iris	Iris LED blinking value. Valid value range [0-1]. 0: LED is off. 1: LED will be blinking red.
(int) mOn	msec duration for the On part of a blinking cycle. Valid value range [0-2000].
(int) mOff	msec duration for the Off part of a blinking cycle. Valid value range [0-2000].

- Callback

The API provides one delegate callback related to this method:

- (void) onSetLEDControlForPowerLED: FpLED: ComLED: IrisLED: mSecOn: mSecOff:

onSetLEDControlForPowerLED:(int) power FpLED:(int) fp ComLED:(int) com IrisLED:(int) iris … mSecOn:(int) mOn mSecOff:(int) mOff;

- Description

Delegate callback for the setLEDControlForPowerLED: power FpLED: fp ComLED: com mSecOn: mOn mSecOff: mOff method.

- Parameter

Parameter	Description
(int) power	Power LED blinking value. Valid value range [0-3]. 0: LED is off. 1: LED will be blinking green. 2: LED will be blinking amber. 3: LED will be blinking red.
(int) fp	Fingerprint LED blinking value. Valid value range [0-3]. 0: LED is off. 1: LED will be blinking green. 2: LED will be blinking amber. 3: LED will be blinking red.
(int) com	Com LED blinking value. Valid value range [0-1]. 0: LED is off. 1: LED will be blinking blue.
(int) iris	Iris LED blinking value. Valid value range [0-1]. 0: LED is off. 1: LED will be blinking red.
(int) mOn	msec duration for the On part of a blinking cycle. Valid value range [0-2000].
(int) mOff	msec duration for the Off part of a blinking cycle. Valid value range [0-2000].

Firmware Update

startFirmwareUpdate:(NSData *) newFirmware toLegacyFirmware: (BOOL) legacyFirmwareUpdateMethod

- Description

This method is used to update the IdentiFI firmware.

- Parameter

Parameter	Description
(NSData *) newFirmware	A dedicated and encrypted firmware file.
legacyFirmwareUpdateMethod	Should be true if the target IdentiFI reader is running with firmware version < v2.3.0

- Callback

The API provides one delegate callback related to this method:

- (void) onFirmwareTransferCompleted:(long) transferResult
-

onFirmwareTransferCompleted:(long) transferResult

- Description

Delegate callback for the startFirmwareUpdate: method.

- Parameter

Parameter	Description
(long) transferResult	Transfer status code. 0: Firmware update transfert completed; unit is powering down; update will complete at next power up. 1: Error with the firmware update file decryption. 2: Firmware update unauthorized for the attached IdentiFI. 3: Invalid IdentiFI serial number decoding. 4: Error in IdentiFI serial number verification. 5: Error with update extraction.