



FINGERPRINTSCHECK SDK

Developers Guide

[Content description](#)

This is a reference manual and configuration guide for the NeoCheck Fingerprints Check SDK product. It shows how to interact with the .Net library in order to manipulate supported capture devices, detect segmented fingerprints, extract meaningful characteristics (QA) and match different samples in a fast and easy way.

Madrid, December 1st 2017

Disclaimer

The information contained within this document is and shall remain the property of NeoCheck. It must not be produced in whole or in part, or given or communicated to any third party without the prior consent of NeoCheck.

Whilst careful attention have been paid to ensure the appropriate referencing of information and sources, it is possible that a document of this nature might contain some errors. In such cases, upon notification NeoCheck will immediately analyse and make any required corrections or amendments.

©NeoCheck 2017

Release Notes

Version	Date	Description
1.0	18/12/2017	Initial Version
1.1	02/02/2018	Changes in fingerprints results. Suprema scanners integrated.



Index

1	Introduction	3
1.1	Fingerprint Capture.....	3
1.2	Minutiae and relevant data extraction	3
1.3	Fingerprint Matching	4
1.4	Fingerprint QA.....	4
2	Installation	5
2.1	FingerprintsCheck SDK Licensing	5
3	.NET Reference.....	6
3.1	Classes.....	6
3.2	Events.....	6
3.3	Enums.....	6
3.4	EventArgs	8
3.5	Data Types.....	8
3.6	Methods.....	9
3.6.1	FingerprintsSDK.GetInstance method	9
3.6.2	FingerprintsSDK. Connect method.....	9
3.6.3	FingerprintsSDK. Disconnect method	9
3.6.4	FingerprintsSDK. Reset method	10
3.6.5	FingerprintsSDK.StartCaptureSingleFinger method.....	10
3.6.6	FingerprintsSDK.StartCaptureMultipleFingers method	10
3.6.7	FingerprintsSDK.AbortCapture method	10
3.6.8	FingerprintsSDK.ExtractFinger method.....	11
3.6.9	FingerprintsSDK.Match method.....	11
3.6.10	FingerprintsSDK.EnrollFinger method.....	12
3.6.11	FingerprintsSDK.GetEnrolledFingers method	12
3.6.12	FingerprintsSDK.ClearEnrolledFingers method.....	12
3.6.13	FingerprintsSDK.IdentifyFinger method	12
3.6.14	FingerprintsSDK.SegmentFingers method	12
3.6.15	FingerprintsSDK.ToWSQ method.....	13
3.6.16	FingerprintsSDK.ToBitmap method	13



1 Introduction

This user manual describes the different functions and features provided by NeoCheck FingerprintsCheck SDK. The purpose of this SDK is to provide functionality to manage fingerprints scanners and perform fingerprints capture, recognition, matching and QA. The SDK is designed for simple integration into applications, which need biometric identity verification. The SDK has a high-level API that provides functionality for operate with fingerprints.

Features:

1.1 Fingerprint Capture

Fingerprint capture is a process when a fingerprints scanner connected to the computer is started and fingerprints are extracted as images to be processed. This capture process can be done for an isolated fingerprint (1-fingerprint scanner) or for several fingerprints at the same time (4-4-2 fingerprint scanner).

Multi fingerprints (4-4-2) capture considerations

When capturing several fingerprints at the same time, the FingerprintsCheck SDK provides a sequence consisting in 3 steps:

- Capture 4 fingers from the right hand
- Capture 4 fingers from the left hand
- Capture 2 thumbs

To perform a multi fingerprints capture operation, client application must provide the following information to the SDK when starting the capture:

- Sequence step
- Possible missing fingerprints (individual has any issue in some finger, such as fingers amputated, bandaged, burnt)

When capture finishes, FingerprintsCheck SDK communicates the result using an event. The result information includes the provided sequence step, as well as the list of missing fingerprints, so the client application can check that the result is ok. In case of missing fingerprints not provided when starting the capture, or incorrect fingers detected while capturing, result status will reflect the issue (finger not found, invalid finger, duplicate finger)

1.2 Minutiae and relevant data extraction

Fingerprint extraction is a process when a fingerprint can be detected as valid by evaluating the image captured from the subject, performing a segmentation process in case several fingerprints are present in the image. The extraction process identifies the finger position and extracts the valuable fingerprint features, such as minutiae, cores and deltas with a high degree of confidence, even in cases when the fingerprint is rotated.



1.3 Fingerprint Matching

Fingerprint matching is a process when a fingerprint can be identified by evaluating its biometric features and comparing them with another fingerprint. This process is called one-to-one matching because both images are from the same fingerprint. Fingerprint matching is a fast way to compare images from the same finger, or a fingerprint with several other fingerprints, which is called one-to-many matching.

1.4 Fingerprint QA

Fingerprint QA is the process to assess if a given fingerprint is compliant with the NFIQ (NIST Fingerprint Image Quality) score. NFIQ score 1,2,3 and 4 are usable for fingerprint matching in practice and 5 is considered as bad fingerprint image quality and fingerprint images with such quality are not recommended to be enrolled for biometric purposes.



2 Installation

There are a set of prerequisites before start using the FingerprintsCheck SDK:

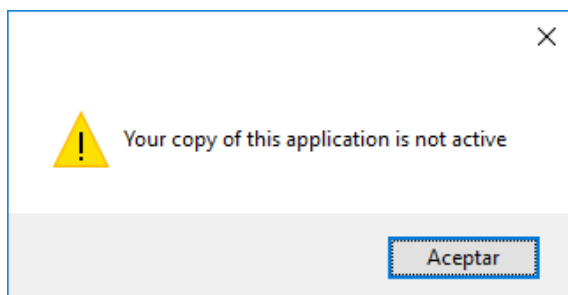
- Windows XP/7/10
- .NET framework 4.6
- Fingerprints scanner SDK

The FingerprintsCheck SDK comes as a set of DLL libraries that are distributed together, along with a Demo application and C# sample code.

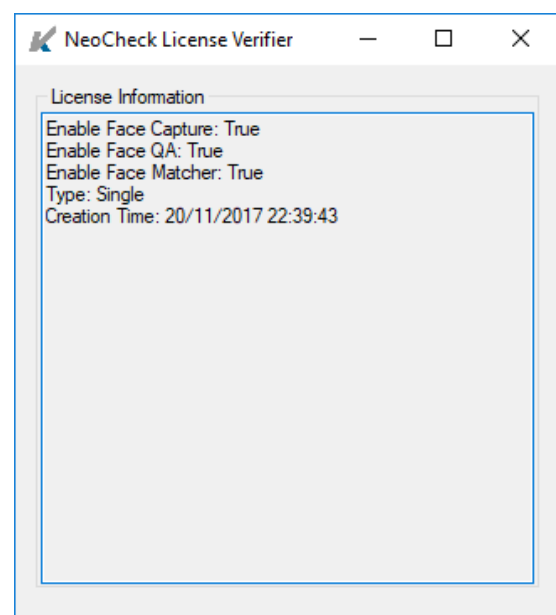
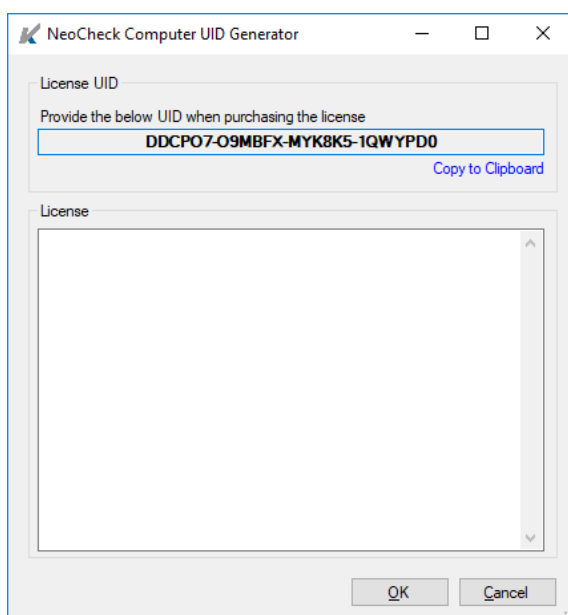
2.1 FingerprintsCheck SDK Licensing

The FingerprintsCheck SDK uses a third-party software provided to perform finger 1:1 matching and extended QA. The software requires to obtain a license on this component.

In order to activate the SDK, it is necessary to execute the NeoCheckLicenseVerifier tool. The following message will be displayed until a valid NeoCheckLicense.lic file is found:



A new window will be loaded showing your machine UID which should be sent to licenses@neocheck.com for generating the activation code. Once you paste the provided code in the lower textbox, a .lic file is going to be created in the execution route.



3 .NET Reference

NeoCheck FingerprintsCheck SDK uses a set of components for working with fingerprints' scanners connected to the computer. These perform fingerprint recognition, matching and QA.

3.1 Classes

NAME	DESCRIPTION
FingerprintsSDK	The main point of entry of the Fingerprints SDK. Provides methods and events for working with fingerprints scanners and performing fingerprint recognition, matching and QA.
GenericPringerprintScanner	Represents the base class to operate with fingerprints scanners connected to the computer, providing methods to capture fingerprints.
CogentFingerprintScanner	Cogent 4-4-2 Fingerprints scanner implementation.
SupremaFingerprintScanner	Suprema 4-4-2 Fingerprints scanner implementation.
SupremaSingleFingerprintScanner	Suprema Single-Fingerprint scanner implementation.
IFingerprintsMatcher	Represents the base class to perform fingerprints recognition, matching and QA.
NeuroTechnologyFingerprintsMatcher	Fingerprints matcher implementation, 3rd party module.

3.2 Events

NAME	DESCRIPTION
FingerprintsPreviewImage	Event raised by the Fingerprints SDK when capture operation is in progress and a preview image from the fingerprints scanner is available.
FingerprintsAvailable	Event raised by a fingerprints SDK when a capture process has finished. It includes information in the event argument that contains all the fingerprints captured, and the global result of the capture process. See FingerprintsAvailableEventArgs in chapt. 3.4.

3.3 Enums

NAME	DESCRIPTION
FingerprintsScannerImplementation	The implementations available to interact with fingerprints scanners connected to the computer. Values: SupremaRealScanG10 SupremaRealScanG1 CogentCS500 CrossmatchPatrol CrossmatchVerifier320
FingerprintsScannerCaptureType	The type of fingerprints capture of the fingerprints scanner implementation. Values: SingleFinger MultipleFinger
FingerprintsMatcherImplementation	The fingerprints matcher implementations available to perform fingerprint recognition, matching and QA. Values: NeuroTechnology



FingerResultStatus	The fingerprint status result when performing a matching operation. Values: NotStarted Ok MatchNotFound FingerNotFound FingerDuplicate FingerInvalid Canceled
MultipleFingersType	The type of multiple fingers extraction, when working with 4-4-2 fingerprints scanners. Values: LeftFourFingers RightFourFinges Thumbs
FingerPosition	The position of the finger extracted. Values: Unknown RightThumb RightIndex RightMiddle RightRing RightLittle LeftThumb LeftIndex LeftMiddle LeftRing LeftLittle
FingerPatternType	The finger pattern type. Values: Unknown PlainArch TentedArch RadialLoop UlnarLoop PlainWhorl CentralPocketLoop DoubleLoop AccidentalWhorl Whorl RightSlantLoop LeftSlantLoop Scar Amputation
FingerNfiqQuality	The finger NFIQ score. NFIQ score 1, 2, 3 and 4 are usable for fingerprint matching and 5 is considered as bad fingerprint image and not recommended for biometric purposes. Values: Unknown = 0 Poor = 5 Fair = 4 Good = 3 VeryGood = 2 Excellent = 1
FingerMinutiaeType	The type of every minutiae extracted from the fingerprint. Values: Unknown End Bifurcation

	Other
FingerCoreType	The type of the finger core extracted from the fingerprint. Values: Single Double
FingerprintsCaptureStatus	The result of a fingerprints scanner capture process. Values: NotStarted Finished FingerNotFound FingerInvalid FingerDuplicated Canceled

3.4 EventArgs

NAME	DESCRIPTION
FingerprintsPreviewImageEventArgs	Event raised by the fingerprints SDK when performing a fingerprints capture operation. It includes the fingerprint preview image that is being captured by the fingerprints scanner, in case client application wants to show it.
FingerprintsAvailableEventArgs	Base event raised by the fingerprints SDK when performing a fingerprints capture operation. It has two implementations, detailed below, for the cases when the capture operation is for a single fingerprints or for several fingerprints at the same time.
SingleFingerprintAvailableEventArgs	Concrete implementation of the FingerprintsAvailableEventArgs, corresponding to a single fingerprints capture operation. It contains information about the fingerprint captured and the global result of the capture operation. See FingerprintsCaptureStatus in Chapter 2.3.
MultipleFingerprintsAvailableEventArgs	Concrete implementation of the FingerprintsAvailableEventArgs, corresponding to a multiple fingerprints capture operation. It contains information about all the fingerprints captured, the sequence step, the missing fingerprints provided when starting the capture and the global result of the capture operation. See FingerprintsCaptureStatus in Chapter 2.3.

3.5 Data Types

NAME	DESCRIPTION
Fingerprint	Information about the Fingerprint captured by a fingerprints scanner. It includes the fingerprint Bitmap image and the FingerPosition
FingerFeatures	Set of fingerprint features extracted, including Cores, Deltas and Minutiae
FingerCore	Information about the Fingerprint Core extracted, including: <ul style="list-style-type: none"> - CoreType: if it is a Single or Double core. - X and Y Core coordinates. - Core rotation angle (in degrees).
FingerDelta	Information about every Fingerprint Delta extracted, including: <ul style="list-style-type: none"> - X and Y Delta coordinates. - Delta triangle angles (in degrees).
FingerMinutiae	Information about every Fingerprint Delta extracted, including: <ul style="list-style-type: none"> - MinutiaeType: type of the minutiae. See chapter 2.3 for values.

	<ul style="list-style-type: none"> - X and Y Delta coordinates. - Minutiae rotation angle (in degrees). - Minutiae curvature. - Minutiae G value. - Minutiae quality.
FingerResult	<p>Information about all features extracted from a fingerprint, including:</p> <ul style="list-style-type: none"> - Image: Fingerprint image processed. - ImageQA: Fingerprint image with overlay showing all features extracted. - Features: FingerFeatures object with information about Cores, Deltas and Minutiae. - Position: finger position value. See chapter 2.3 for values. - PatternType: fingerprint pattern type. See chapter 2.3 for values. - NfiqQuality: fingerprint NFIQ score. See chapter 2.3 for values. - Quality: quality measure, with value ranging from 0 to 100. - Status: FingerResultStatus value. See chapter 2.3 for values.

3.6 Methods

Following a list of all public methods from FingerprintsCheck SDK, scanners and fingerprints matcher implementations.

3.6.1 FingerprintsSDK.GetInstance method

Returns a unique instance (Singleton) of the Fingerprint SDK.

PARAMETER	DESCRIPTION
FingerprintsScannerImplementation	The fingerprints scanner implementation used to interact devices connected to the computer.
FingerprintsMatcherImplementation	The fingerprints matcher implementation used perform fingerprint recognition, matching and QA.

3.6.2 FingerprintsSDK. Connect method

This method initializes the fingerprint scanner.

3.6.3 FingerprintsSDK. Disconnect method

This method disconnects the fingerprint scanner, releasing all resources used.

3.6.4 FingerprintsSDK.Reset method

This method resets the fingerprint scanner, by disconnecting and connecting again.

3.6.5 FingerprintsSDK.StartCaptureSingleFinger method

This method allows the client application to perform a single finger capture process using the fingerprints scanner. It is an asynchronous process, limited in time by a timeout value that can be passed as parameter. Either if capture is successful or any error or timeout occur, the client application can be informed about the result of the capture process by subscribing to the event `FingerprintsCaptureCompleted`, described in Chapter 1.2.

PARAMETER	DESCRIPTION
FingerPosition <code>fingerPosition</code>	The finger position to capture
bool <code>roll</code>	Optional parameter to capture a fingerprint by rolling it. (this feature is not available for all fingerprints scanners). The default value is false.
Int <code>timeout</code>	The maximum time (in milliseconds) for performing a fingerprint capture. The default value is 10000 (10 seconds), and a 0 value means infinite time.

3.6.6 FingerprintsSDK.StartCaptureMultipleFingers method

This method allows the client application to perform multiple fingers capture process using the fingerprints scanner. It must receive as parameter a `MultipleFingersType` value (See chapter 2.3 for values), and optionally a list of missing fingers, if any. It is an asynchronous process, limited in time by a timeout value that can be passed as parameter. Either if capture is successful or any error or timeout occur, the client application can be informed about the result of the capture process by subscribing to the event `FingerprintsCaptureCompleted`, described in Chapter 1.2.

PARAMETER	DESCRIPTION
MultipleFingersType <code>multipleFingersType</code>	The type of multiple fingers capture to be performed. See Chapter 2.3 for values.
List<FingerPosition> <code>missingFingers</code>	The list of missing fingers, by their <code>FingerPosition</code> values. The default value is null.
Int <code>timeout</code>	The maximum time (in milliseconds) for performing a fingerprint capture. The default value is 10000 (10 seconds), and a 0 value means infinite time.

3.6.7 FingerprintsSDK.AbortCapture method

This method aborts a running capture process in the fingerprints scanner. It raises a `FingerprintsCaptureCompleted` event when fingerprints scanner has aborted the capture process and a new one can be started.



3.6.8 FingerprintsSDK.ExtractFinger method

Returns a FingerResult object with all the information extracted, or null of fingerprint could not be processed. See Chapter 2.5 for FingerResult information.

PARAMETER	DESCRIPTION
Bitmap finger Or Byte[] finger	The image to perform the fingerprint extraction.

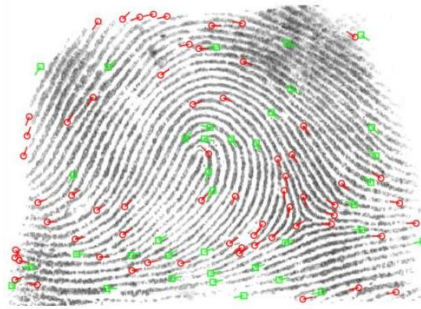


Figure 1 – Minutiae extraction

3.6.9 FingerprintsSDK.Match method

Returns a value indicating the similarity of the fingerprints passed as parameters. The return value is a float between 0 and 1.

PARAMETER	DESCRIPTION
Bitmap finger1 Or Byte[] finger1	The first image to perform the fingerprints matching.
Bitmap finger2 Or Byte[] finger2	The second image to perform the fingerprints matching.

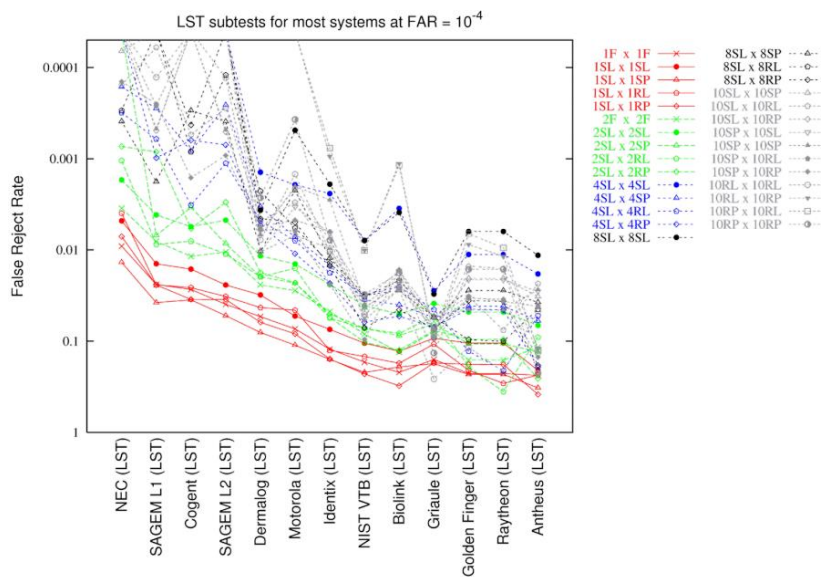


Figure 2 – NIST FpVTE Vendors comparison test 2012



3.6.10 FingerprintsSDK.EnrollFinger method

This method enrolls a fingerprint in a given position, as a previous step to perform a 1-N matching (through the Identify method). The return value is the result of the enrolment.

PARAMETERS	DESCRIPTION
Bitmap finger Or Byte[] finger	The finger image.
FingerPosition position	The position of the finger. See chapter 2.3 for values.

3.6.11 FingerprintsSDK.GetEnrolledFingers method

This method returns all enrolled fingerprints. The enrolled fingerprints will contain the information extracted, if they were enrolled using the Capture operation.

3.6.12 FingerprintsSDK.ClearEnrolledFingers method

This method clears all enrolled fingerprints. It must be called to restart the enrolment process with other fingerprints set, or to remove any already enrolled fingerprint, by providing the FingerPositions of the fingerprints to remove.

PARAMETER	DESCRIPTION
List<FingerPosition> fingerPositions	The list of finger positions to remove from the list of enrolled fingerprints. If null, the entire list of enrolled fingerprints will be cleared.

3.6.13 FingerprintsSDK.IdentifyFinger method

Returns the FingerPosition value for the fingerprint, or null if finger was not found in the set of enrolled fingerprints. This method is used to perform 1-N matching, because it will compare the fingerprint given as input with the fingerprints already enrolled.

PARAMETER	DESCRIPTION
Bitmap finger Or Byte[] finger	The finger to identify.

3.6.14 FingerprintsSDK.SegmentFingers method

Returns a set of FingerResult objects after segmenting an image composed of several fingerprints. The method must receive as parameter the type of image captured from a 4-4-2 fingerprints scanner, along with any missing fingers from the individual whose hand was captured.

PARAMETER	DESCRIPTION
Bitmap fingers Or Byte[] fingers	The fingers to segment.
MultipleFingersType multipleFingersType	The scan type performed (in a 4-4-2 scanner). See chapter 2.3 for values.
IList<FingerPosition> missingFingers	The missing fingers, if any





Figure 3 – Fingerprints Slap Segmentation

3.6.15 FingerprintsSDK.ToWSQ method

Method to convert a fingerprint image from Bitmap to WSQ byte array.

PARAMETER	DESCRIPTION
Bitmap finger	The finger image to convert.

3.6.16 FingerprintsSDK.ToBitmap method

Method to convert a fingerprint WSQ byte array to Bitmap.

PARAMETER	DESCRIPTION
Byte[] finger	The WSQ byte array fingerprint to convert.