Green Bit

MultiScan SDK

Overview

V2.9

September 2013

GREEN BIT

BIOMETRIC SYSTEMS

Table of Contents

1 Introduction....5
1.1 Documentation conventions....6
1.1.1 General Conventions....6
1.2 Documentation revisions....7
2 Software components description....8
2.1 Supported devices....9
2.2 Software Structure....10
2.3 DactyMatch SDK....12
2.4 Supported software license....13
2.5 Advantages of using MultiScan SDK....14
3 Supported Devices Technical Features....15
3.1 Specific scanner libraries....16
3.2 MultiScan1000....17
3.2.1 Scan Areas....17
3.2.2 Image characteristics....17
3.2.3 Certificates....18
3.3 MultiScan500....19
3.3.1 Scan Areas....19
3.3.2 Image characteristics....19
3.3.3 Certificates....20
3.4 MC500....21
3.5 MSC500....22
3.6 DactyScan84(n)....23
3.6.1 Scan Areas....23
3.6.2 Image characteristics....24
3.6.3 Certificates....24
3.7 DactyScan84(c)....25
3.7.1 Scan Areas....25
3.7.2 Image characteristics....26
3.7.3 Certificates....26
3.8 PoliScan2....27
3.8.1 Scan Areas....27
3.8.2 Image characteristics....28
3.8.3 Certificates....28
3.9 VisaScan3....29
3.9.1 Image characteristics (Photo Scanner only) 29
3.10 DactyScan26 and DactyScan26i....30
3.10.1 Scan Areas....30
3.10.2 Image characteristics....31
3.10.3 Certificates....31

3.11 DactyScan40....32
3.11.1 Scan Areas....32
3.11.2 Image characteristics....33
3.11.3 Certificates....33
3.12 DactyScan40i....34
3.12.1 Scan Areas....34
3.12.2 Image characteristics....35
3.12.3 Certificates....35
3.13 DactyScan32....36
3.13.1 Scan Areas....36
3.13.2 Image characteristics....37
3.13.3 Certificates....37
3.14 MC517....38
3.14.1 Scan Areas....38
3.14.2 Image characteristics....39
3.14.3 Certificates....39
3.15 MSC517....40
4 BASIC SDK....41
4.1 BASIC SDK: C Interface Libraries....42
4.1.1 Level 0: Serial BUS Drivers....42
4.1.2 Level 1: Individual Scanner SDKs (Low level SDK)....42
4.1.3 Level 2: GBMSAPI SDK....43
4.2 BASIC SDK: .NET-MONO WRAPPER....46
4.3 BASIC SDK: Java Wrapper....47
4.4 BASIC SDK: Programming Examples....48
4.4.1 GettingStarted C++ Example....48
4.4.2 GettingStarted C# Example....49
4.4.3 BASIC C++ Example (for Windows only)....49
4.4.4 BASIC C# Example....49
4.5 BASIC SDK: Application deployment....50
5 FULL ENHANCED SDK (Level 3)....51
5.1 FULL ENHANCED SDK: C Interface Libraries....52
5.1.1 Graphical Interface libraries (GBMSGUI, GBMSGUI_QT)....52
5.1.2 Image Processing Tools Libraries....54
5.2 FULL ENHANCED SDK: .NET-MONO WRAPPER....57
5.3 FULL ENHANCED SDK: Java Wrapper....58
5.4 FULL ENHANCED SDK: Programming examples....59
5.4.1 Image Processing Tool C++ Example....59
5.4.2 Image Processing Tool C# Example....59
5.4.3 Gbfinimg C# Example....59
5.4.4 GBFIR C++ Example....60
5.4.5 GBMSDemo C# Example....60
5.4.6 GBMSGUI_QT Test C++ Example....60
5.4.7 Libraries source code (GBMSGUI_NET and GBMSGUI_QT)....60
5.5 FULL ENHANCED SDK: Application Deployment....61

6 Troubleshooting.....62

1 Introduction

This document provides an overview of Green Bit MultiScan SDK for fingerprint images acquisition and processing.

1.1 Documentation conventions

1.1.1 General Conventions

Green Bit saves the right to make changes, integrations or enhancements to this manual without notice, and this cannot be a reason to consider this present publication inadequate.

To highlight some important text parts the following graphical structure is used:

	Note: The signal is used to highlight important technical information or suggestions.

In this manual the following acronyms are used:

- IAFIS Integrated Automated Fingerprint Identification System
- IQS Image Quality Specifications

Microsoft Windows 2000®, Microsoft Windows XP® and Microsoft Windows Server2003® are trademarks of Microsoft Corporation.

1.2 Documentation revisions

Revision	Date	Description
V1.0	01/31/2009	Original
V1.1	03/16/2009	Added support for DactyScan 40 device. Bugs fixing
V2.0	05/18/2009	- New release of MS500u.dll - Added new object for DactyScan 40 device - Bugs fixing - All .NET components recompiled with strong name signature - New version of VSRoll.dll library
V2.1	06/29/2009	- Bugs fixing - All .NET components recompiled with strong name signature
V2.2	07/28/2009	- Bugs fixing - New MS500 Low Level library
V2.3	November 2009	See Modification List file
V2.4	February 2010	See Modification List file
V2.5	September 2010	See Modification List file
V2.6	March 2011	See Modification List file
V2.7	March 2012	See Modification List file
V2.8	September 2012	See Modification List file
V2.9	September 2013	See Modification List file

2 Software components description

MULTISCAN SDK library allows developers to interface several Green Bit devices (see paragraph 2.1 for a detailed list of supported devices).
For components list see the Modification List file.

2.1 Supported devices

The Green Bit devices that are currently supported by this software are:

- MultiScan500 (on Windows only at moment)
- MultiScan1000 (on Windows only at moment)
- DactyScan84
- DactyScan84n
- DactyScan84c
- Visascan3 (on Windows only)
- Poliscan2 (on Windows only)
- DactyScan26 and DactyScan26i
- DactyScan40 (on Windows only)
- DactyScan40i
- DactyScan32
- MC517 and MSC517

2.2 Software Structure

MULTISCAN SDK is provided with two packets:

- **BASIC SDK**: for device management and image acquisition
- **FULL ENHANCED SDK**[1]: it provides a graphical user interface (GBMSGUI) and a library set for image processing (segmentation, image enhancement, matching score calculation for the sequence check, quality evaluation, image compression) and formatting (compliant with ANSI/NIST ITL 1-2007 standard).

Note that by using the FULL ENHANCED SDK, developers have on hand a complete solution (fully customizable since GBMSGUI is provided with the complete source code). For getting information, please contact Green Bit commercial department (info@greenbit.com).

Regarding the interface with Green Bit devices, four levels of abstraction are provided:

- LEVEL 0: peripheral drivers (USB)
- LEVEL 1: low level device interface
- LEVEL 2: Green Bit Generic Scanner interface
- LEVEL 3: Enhanced GUI interface and image processing

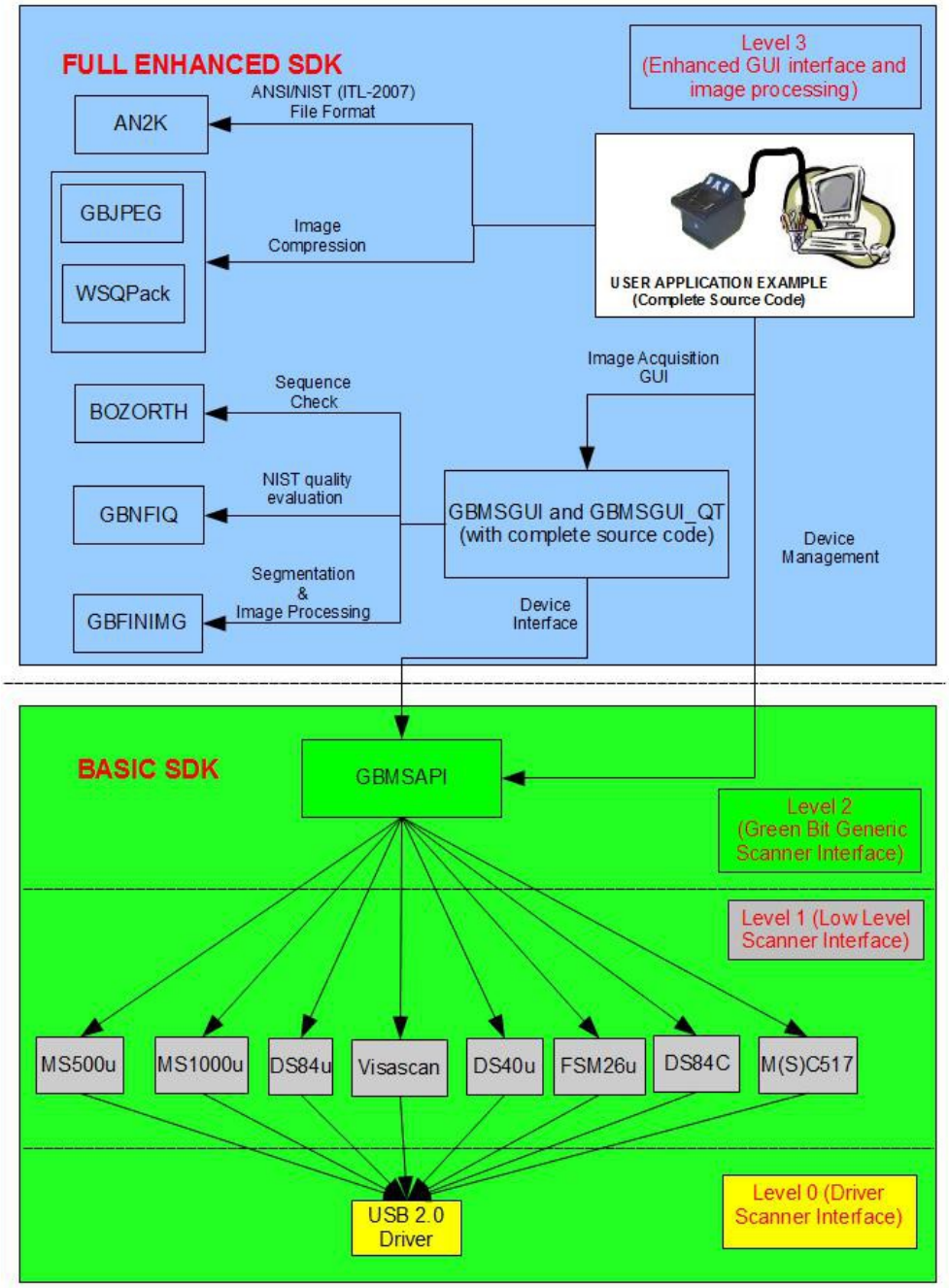
BASIC SDK includes 0,1 and 2 levels, while FULL ENHANCED SDK covers level 3.

By using MULTISCAN SDK, the interface with the devices is allowed only through 2 and 3 levels[2]. Green Bit recommends using FULL ENHANCED SDK to have a more increased development speed and to perform a correct image acquisition process.

---

[1] For properly working, FULL ENHANCED SDK needs BASIC SDK

[2] For special needs developers can ask for access to level 1: in this case, please apply for Green Bit support (support@greenbit.com)


FULL ENHANCED SDK
Level 3
(Enhanced GUI interface and image processing)
ANSI/NIST (ITL-2007)
File Format
AN2K
GBJPEG
WSQPack
Image
Compression
USER APPLICATION EXAMPLE
(Complete Source Code)
Image Acquisition
GUI
Sequence
Check
BOZORTH
NIST quality
evaluation
GBNFIQ
GBMSGUI and GBMSGUI_QT
(with complete source code)
Device
Management
Segmentation
&
Image Processing
GBFINIMG
Device
Interface
BASIC SDK
GBMSAPI
Level 2
(Green Bit Generic Scanner Interface)
Level 1 (Low Level Scanner Interface)
MS500u
MS1000u
DS84u
Visascan
DS40u
FSM26u
DS84C
M(S)C517
USB 2.0
Driver
Level 0 (Driver Scanner Interface)

2.3 DactyMatch SDK

DactyMatch SDK is provided in the FULL ENHANCED SDK as a trial version. Green Bit DactyMatch SDK is useful for subjects verification and identification by means of fingerprint images processing. It makes use of Green Bit proprietary GBFRSW algorithms.

DactyMatch SDK can be also used as an ISO/IEC 19794-2 and INCITS 378 (MINEX certified – encoder only) formats converter (to and from).

For further information see also the specific documentation inside the FULL ENHANCED directory.

2.4 Supported software license

Software licensing policy is hardware dependent. More exactly, it's related to the third letter of the scanner serial number, according to the following schema:

- "R": flat and roll scans, BASIC SDK + FULL ENCHANCED SDK support
- "F": only flat scans, BASIC SDK + FULL ENCHANCED SDK support
- "E": flat and roll scans, only BASIC SDK support (NO FULL ENCHANCED)
- "B": only flat scans, only BASIC SDK support (NO FULL ENCHANCED)

2.5 Advantages of using MultiScan SDK

MULTISCAN SDK offers a lot of advantages to application developers:

- **A single development**: by following Green Bit recommendations, the application can be developed once for a scanner and then can be reused for the other scanners supported by the SDK simply by a "plug and play", because of the multi-device support and the definition of a unique acquisition process for all supported objects to be scanned.

- **The image acquisition process is fully managed** by the libraries, so that the developer is sure of the procedure correctness.

- **A complete solution** is offered by means of FULL ENHANCED SDK, that provides the full acquisition process from image acquisition and processing to an ANSI/NIST ITL 1-2007 file creation with the acquired objects.

- GBMSGUI library **complete source code** (that provides a user-friendly device graphical interface, built on .NET and QT frameworks) and a lot of examples (including a complete acquisition process), in order to allow developers to fully customize their application.

- **Multi-language support**: all libraries are provided with the standard C interface and wrappers for .NET and Java frameworks. Therefore the libraries are ready to be used with the all languages supported by the framework (C#, C++, Visual Basic .NET. Java and so on)

- **Complete access to device functionality.**

3 Supported Devices Technical Features

In this chapter a description of each supported scanner will be given, focusing on technical features. Supported scanner features can be seen synoptically in the Scanner Featured.pdf document.

3.1 Specific scanner libraries

See the Modification List document for more detailed information.

3.2 MultiScan1000

MultiScan1000 is based on an optical scanner, capable to acquire the following objects:

- slap fingerprint images
- two thumbs fingerprint images
- single flat fingerprint images
- rolled fingerprint images
- Lower, Upper and Writer's Palm

The above mentioned optical scanner doesn't have any mechanical part movements.
The device auto-detects whenever cleaning is necessary. If cleaning of the surface is recommended by a message on the operator's monitor time between enrollments will be less than 30 seconds.
The functioning of the scanner is not affected by external light sources.
The Multiscan1000 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 3s (depending on skin conditions).

3.2.1 Scan Areas

MultiScan1000 scanning windows are the followings:

- The Flat Area corresponds to the whole scanning area
- The Roll Area corresponds to a portion of the Flat Area (marked by strips on the device)

3.2.2 Image characteristics

Image resolution	1000 dpi ±0.7%; 500 dpi ±0.7%
Dynamic range	8-bit, 256 gray tone
Acquisition area	5" x 5" (in "Flat Area") and 1,6" x 1,6" (in "Roll Area")

Image dimension (compliant with ANSI/NIST-ITL-1-2007)	5000 x 5000 pixel (832 x 832 for preview) in "Flat Area" and 1600 x 1600 pixel in "Roll Area" (both for preview and acquisition phases)
Acquisition speed	"Flat Area": - 7.0 fps in preview - 0.5 s for grab and 1.2 s for transport for final frame (1000 dpi) "Roll Area" - up to 16 fps (for Rolled objects)

3.2.3 Certificates

MultiScan1000 is subject to IAFIS-IQS Appendix F certification at 1000dpi and 500 dpi

3.3 MultiScan500

MultiScan500 is based on an optical scanner, capable to acquire the following objects:

- slap fingerprint images
- two thumbs fingerprint images
- single flat fingerprint images
- rolled fingerprint images
- Lower, Upper and Writer's Palm

The above mentioned optical scanner doesn't have any mechanical part movements.
The functioning of the scanner is not affected by external light sources.
The MultiScan500 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 3s (depending on skin conditions).

NOTE: for this device a .ini file (MS500.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (MS500u.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.3.1 Scan Areas

MultiScan500 scanning areas are the followings:

- The Flat Area corresponds to the whole scanning area
- The Roll Area corresponds to a portion of the Flat Area (marked on the device)

3.3.2 Image characteristics

Image resolution	500 dpi ±1%

Dynamic range	8-bit, 256 gray tone
Acquisition area	5” x 5” (in “Flat Area”) and 1,6” x 1,6” (in “Roll Area”)
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	2500 x 2500 pixel (832 x 832 for preview) in “Flat Area” and 800 x 800 pixel in “Roll Area” (both for preview and acquisition phases)
Acquisition speed	• 1.8 fps for Full resolution frames in “Flat Area” • (4.4 fps for preview) in “Flat Area” • 9.2 fps in “Roll Area”

3.3.3 Certificates

Scanner is compliant to the following standards:

- FBI certifies that MultiScan500 is compliant with FBI CJIS Division’s Integrated Automated Fingerprint Identification System Image Quality Specification (IQS) Appendix F for the following categories:

  - live-scan fingerprint and palm-print systems

- GA 625/626/864/865/866

3.4 MC500

MC500 device is based on the MultiScan500 one, but has some differences, hereby listed:

1) No LCD display and no pedal

2) Rolled fingerprint area dimensions: 1.28' x 1.28'

3) Rolled fingerprint framerate: 15 fps

4) High Speed feature for full resolution preview images (2,5 fps)

NOTE: for this device a .ini file (MS500.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (MS500u.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.5 MSC500

MSC500 device is based on the MultiScan500 one, but has some differences, hereby listed:

1) Rolled fingerprint area dimensions: 1.28' x 1.28'

2) Rolled fingerprint framerate: 15 fpsHigh Speed feature for full resolution preview images (2,5 fps)

NOTE: for this device a .ini file (MS500.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (MS500u.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.6 DactyScan84(n)

DactyScan84 is based on an optical scanner, capable to acquire the following objects:

- slap fingerprint images
- two thumbs fingerprint images
- single flat fingerprint images
- rolled fingerprint images

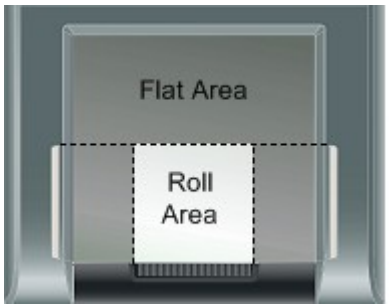
The above mentioned optical scanner doesn't have any mechanical part movements.
The device is water and dust ingress protected and complies with IP-54 standard.
The device auto-detects whenever cleaning is necessary. If cleaning of the surface is recommended by a message on the operator's monitor time between enrollments will be less than 30 seconds.
The functioning of the scanner is not affected by external light sources.
The DactyScan84 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 3s (depending on skin conditions).

NOTE: for this device a .ini file (DS84U.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (DS84U.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.6.1 Scan Areas

For DactyScan84 the scanning areas are shown in the picture below, more exactly:

- The Flat Area corresponds to the whole scanning area (including the "Roll Area" surface)
- Roll Area corresponds to the white window (refer to picture)



3.6.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	3,2" x 3,0" (in "Flat Area") and 1,6" x 1,6" (in "Roll Area")
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	1600 x 1500 pixel (in "Flat Area") and 800 x 800 pixel (in "Roll Area")
Acquisition speed	15 fps (in "Roll Area") and 5 fps (in "Flat Area")

3.6.3 Certificates

Scanner is compliant to the following standards:

- FBI certifies that DactyScan84 is compliant with FBI CJIS Division's Integrated Automated Fingerprint Identification System Image Quality Specification (IQS) Appendix F in the following categories:

  - live-scan ten-print systems
  - Identification flats systems

- GA 625/626/864/865/866

3.7 DactyScan84(c)

DactyScan84c is based on an optical scanner, capable to acquire the following objects:

- slap fingerprint images
- two thumbs fingerprint images
- single flat fingerprint images
- rolled fingerprint images

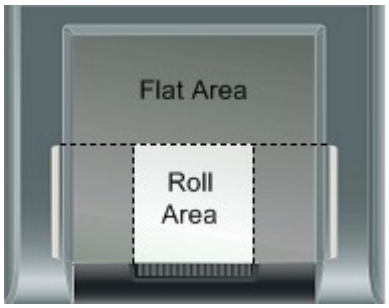
The above mentioned optical scanner doesn't have any mechanical part movements.
The device is water and dust ingress protected and complies with IP-54 standard.
The device auto-detects whenever cleaning is necessary. If cleaning of the surface is recommended by a message on the operator's monitor time between enrollments will be less than 30 seconds.
The functioning of the scanner is not affected by external light sources.
The DactyScan84 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 3s (depending on skin conditions).

NOTE: for this device a .ini file (DS84C.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (DS84c.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.7.1 Scan Areas

For DactyScan84c the scanning areas are shown in the picture below, more exactly:

- The Flat Area corresponds to the whole scanning area (including the "Roll Area" surface)
- Roll Area corresponds to the white window (refer to picture)



3.7.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	3,2" x 3,0" (in "Flat Area") and 1,6" x 1,6" (in "Roll Area")
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	1600 x 1500 pixel (in "Flat Area") and 800 x 800 pixel (in "Roll Area")
Acquisition speed (max)	8.6 fps (in "Flat Area", for FULL RESOLUTION mode), 27.0 fps (in "Flat Area", for preview mode) and 25 fps (in "Roll Area")

3.7.3 Certificates

Scanner is compliant to the following standards:

- FBI certifies that DactyScan84C is compliant with FBI CJIS Division's Integrated Automated Fingerprint Identification System Image Quality Specification (IQS) Appendix F in the following categories:

- - live-scan ten-print systems
- - Identification flats systems

3.8 PoliScan2

The main Poliscan2 components are:

- Scanner for palmprint and simultaneous flat fingerprints (Palm Scanner).
- Scanner for single flat and rolled fingerprint (MRO40 – Roll Scanner).

The above mentioned optical scanner doesn't have any mechanical part movements.

PoliScan2 device is capable to acquire the following objects:

- slap fingerprint images
- two thumbs fingerprint images
- single flat fingerprint images
- rolled fingerprint images
- Lower, Upper and Writer's Palm

NOTE: for this device a .ini file (Visascan.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (Visascan.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.8.1 Scan Areas

PoliScan2 scanning areas are the following:

- The Flat Area corresponds to Palm Scanner
- The Roll Area corresponds to Roll Scanner

3.8.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	5” x 5” (in “Flat Area”) and 1,6” x 1,6” (in “Roll Area”)
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	2500 x 2500 pixel (832 x 832 for preview) in “Flat Area” and 800 x 800 pixel in “Roll Area” (both for preview and acquisition phases)
Acquisition speed	1.8 fps (4.4 fps for preview) in “Flat Area” and 15.8 fps in “Roll Area”

3.8.3 Certificates

Both scanners are certified for ELECTRONIC FINGERPRINT TRANSMISSION SPECIFICATION, May 2, 2005 (IAFIS-DOC-01078-7.1), Appendix F: IAFIS Image Quality Specifications.

Very Important Note:
POLISCAN2 development will be dismissed starting from sdk versions greater than 2.8. Its components will be enclosed in the SDK for backward compatibility.

3.9 VisaScan3

VisaScan3 device is like PoliScan2 device, and has as a further component that is a scanner for images acquisition from paper (photo, signature), referred to as "Photo Scanner".
For Photo objects, the only available scanning area is the Flat Area, corresponding to the whole Photo Scanner acquisition surface.
NOTE: for this device a .ini file (Visascan.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (Visascan.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.9.1 Image characteristics (Photo Scanner only)

Image resolution	300 dpi ±1%
Dynamic range	24 bit colour
Acquisition area	4" x 2"
Image dimension	1280 x 648 pixel

Very Important Note:
VISASCAN3 development will be dismissed starting from sdk versions greater than 2.8. Its components will be enclosed in the SDK for backward compatibility.

3.10 DactyScan26 and DactyScan26i

NOTE: both devices DactyScan26 and DactyScan26i shall hereinafter referred to as DactyScan26(i), unless otherwise is specified.

DactyScan26(i) is based on an optical scanner, capable to acquire the following objects:

- single flat fingerprint images

The above mentioned optical scanner is without any mechanical part movements.
The functioning of the scanner is not affected by external light sources.
The DactyScan26 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 2s (depending on skin conditions).

3.10.1 Scan Areas

For DactyScan26(i) the Flat area only is provided.

3.10.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	1" x 1" (in "Flat Area")
Image dimension (compliant with FBI-PIV standard)	500 x 500 pixel (in "Flat Area")
Acquisition speed	From 2,5 to 10,0 fps

3.10.3 Certificates

DactyScan26(i) conforms to ELECTRONIC FINGERPRINT TRANSMISSION SPECIFICATION, May 2, 2005 (IAFIS-DOC-01078-7.1), Appendix F: IAFIS Image Quality Specifications.
The device is also certified by BSI German Institute (Bundesamt für Sicherheit in der Informationstechnik) when working with FSM26u.dll ver. 2.2.0.0 or higher.
DactyScan26 is based on optical scanner FSM26U and is for a single flat fingerprint image acquisition. FSM26U scanner (working with FSM26U.dll ver. 2.1.0.1 or higher) has been certified by FBI as compliant to "Personal Identity Verification (PIV) Single Finger Capture Device Specification" of the FBI CJIS Division's IAFIS Image Quality Specifications.

3.11 DactyScan40

The main DactyScan40 components are:

- Scanner for single flat and rolled fingerprint (MRO40 – Roll Scanner).

The above mentioned optical scanner doesn't have any mechanical part movements.
The functioning of the scanner is not affected by external light sources.
The DactyScan40 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 2s (depending on skin conditions).

DactyScan40 device is capable to acquire the following objects:

- Slap (2 fingers) fingerprint images (Index and middle both right and left; two indexes in any combination)
- single flat fingerprint images
- rolled fingerprint images

NOTE: for this device a .ini file (Visascan.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (Visascan.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.11.1 Scan Areas

For DactyScan40 device scanning areas correspond both to the Roll Scanner.

3.11.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	1,6” x 1,6” (both “Flat” and “Roll Area”)
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	800 x 800 pixel (both for preview and acquisition phases)
Acquisition speed	15.8 fps

3.11.3 Certificates

Scanner is certified for ELECTRONIC FINGERPRINT TRANSMISSION SPECIFICATION, May 2, 2005 (IAFIS-DOC-01078-7.1), Appendix F: IAFIS Image Quality Specifications.

Very Important Note:
Dactyscan40 development will be dismissed starting from sdk versions greater than 2.8. Its components will be enclosed in the SDK for backward compatibility.

3.12 DactyScan40i

The main DactyScan40i components are:

- Scanner for single flat and rolled fingerprint (MRO40 – Roll Scanner).

The above mentioned optical scanner doesn't have any mechanical part movements.

DactyScan40i device is capable to acquire the following objects:

- Slap (2 fingers) fingerprint images (Index and middle both right and left; two fingers in any combination)
- single flat fingerprint images
- rolled fingerprint images

Average scanning time is < 3s (depending on skin conditions).

The functioning of the scanner is not affected by external light sources.

The DactyScan40i acquires high quality fingerprints regardless of skin condition (humid/dry).

NOTE: for this device a .ini file (DS40U.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (DS40U .dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.12.1 Scan Areas

For DactyScan40i device scanning areas correspond both to the Roll Scanner.

3.12.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	1,6” x 1,6” (both “Flat” and “Roll Area”)
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	800 x 800 pixel (both for preview and acquisition phases)
Acquisition speed	16 fps

3.12.3 Certificates

Scanner is compliant to the following standards:

- ELECTRONIC FINGERPRINT TRANSMISSION SPECIFICATION, May 2, 2005 (IAFIS-DOC-01078-7.1), Appendix F: IAFIS Image Quality Specifications.

- GA 625/626/864/865/866

3.13 DactyScan32

The main DactyScan32 components are:

- Scanner for single flat and rolled fingerprint (MRO40 – Roll Scanner).

The above mentioned optical scanner doesn't have any mechanical part movements.

DactyScan32 device is capable to acquire the following objects:

- Slap (2 fingers) fingerprint images (Index and middle both right and left; two fingers in any combination)
- single flat fingerprint images
- rolled fingerprint images

Average scanning time is < 3s (depending on skin conditions).

The functioning of the scanner is not affected by external light sources.

The DactyScan32 acquires high quality fingerprints regardless of skin condition (humid/dry).

NOTE: for this device a .ini file (DS40U.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (DS40U .dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.13.1 Scan Areas

For DactyScan32 device scanning areas correspond both to the Roll Scanner.

3.13.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area	1,28” x 1,28” (both “Flat” and “Roll Area”)
Image dimension (compliant with ANSI/NIST-ITL-1-2007)	640 x 640 pixel (both for preview and acquisition phases)
Acquisition speed	16 fps

3.13.3 Certificates

Scanner is compliant to the following standards:

– GA 625/626/864/865/866

3.14 MC517

MC517 is based on an optical scanner, capable to acquire the following objects:

- slap fingerprint images
- two thumbs fingerprint images
- single flat fingerprint images
- rolled fingerprint images
- Lower, Upper and Writer's Palm

The above mentioned optical scanner doesn't have any mechanical part movements.
The functioning of the scanner is not affected by external light sources.
The MC517 acquires high quality fingerprints regardless of skin condition (humid/dry).
Average scanning time is < 3s (depending on skin conditions).

NOTE: for this device a .ini file (MC517.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (MC517.dll). Range of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

3.14.1 Scan Areas

MC517 scanning areas are the followings:

- The Flat Area corresponds to the whole scanning area
- The Roll Area corresponds to a portion of the Flat Area (marked on the device)

3.14.2 Image characteristics

Image resolution	500 dpi ±1%
Dynamic range	8-bit, 256 gray tone
Acquisition area max dimensions (compatible with GA standard for PALM-PRINT and ROLL-PRINT, and with IAFIS-IQS for FLAT SLAP PRINT)	– 4.6” x 4.6” (FLAT SCAN AREA) – 1.28” x 1.28” (ROLL SCAN AREA, GA STANDARD) – 1.6” x 1.6” (ROLL SCAN AREA, IQS STANDARD)
Image max dimension (compatible with GA standard for PALM-PRINT and ROLL-PRINT, and with IAFIS-IQS for FLAT SLAP PRINT)	– 2304 x 2304 pixels (FLAT SCAN AREA) – 640 x 640 pixels (ROLL SCAN AREA, GA STANDARD) – 800 x 800 pixels (ROLL SCAN AREA, IQS STANDARD)
Acquisition speed	– 2.1 fps, programmable in the range 0.2-4.2 fps (FLAT SCAN AREA, full-resolution mode) – 4.7 fps, programmable in the range 1.0-9.4 fps (FLAT SCAN AREA, low-resolution preview mode) – 17.5 fps, programmable in the range 1.0-35.0 fps (ROLL SCAN AREA, GA STANDARD) – 12.7 fps, programmable in the range 1.0-25.4 fps (ROLL SCAN AREA, IQS STANDARD)

3.14.3 Certificates

Scanner is compliant to the following standards:

– IAFIS- IQS (EBTS 9.3, appendix F)

– GA 625/626/864/865/866

3.15 MSC517

MSC517 device is equal to the MC517 one regarding the features, certifications and images characteristics, but it's provided with a RGB touchscreen display that allows the operator to interact with it in an easier way.

4 BASIC SDK

BASIC SDK covers the 0,1 and 2 levels of abstraction for device interfacing.

4.1 BASIC SDK: C Interface Libraries

4.1.1 Level 0: Serial BUS Drivers

The interfacing between devices and computer is made through USB 2.0 serial connection. The first thing to be done before connecting a device to the PC is to install its own drivers.
Only a driver (GreenBit Universal Driver) is needed for all the devices.
A setup executable file is provided for each system architecture (32 or 64 bit), in the DRIVERS SETUP folder of the BASIC SDK CD. In order to install drivers simply launch the corresponding SETUP file.
For USB drivers, also the corresponding .sys and .inf files are as well provided in the same folder of the CD (again, they are cataloged by system architecture).

NOTE 1: Level 0 is not directly accessible to application

NOTE 2: In order to avoid blue screen deaths on Windows XP 64 bit, it's warmly recommended to install the hot fix recommended in the KB973618 Article (freely downloadable from Microsoft support), without caring that it is written that it applies to Windows server2003 operating system.

NOTE 3: for Linux environment the driver is composed by the gb-transport and usb libs. No setup executable is provided. See the 4.5 BASIC SDK: Application deployment paragraph for further information.

4.1.2 Level 1: Individual Scanner SDKs (Low level SDK)

To use Green Bit scanners, the corresponding dlls (or ".so" for Linux) should be copied into the application or into SYSTEM32 folder. They can be found in SDK\Device Dlls folder of the BASIC SDK CD. More precisely:

- VSRoll for all the scanners
- MS1000U for MultiScan1000 device (Windows only)
- MS500U for MultiScan500 device (Windows only)
- DS84U for DactyScan84 device

- DS84C for DactyScan84C device

- FSM26U for DactyScan26(i) device

- DS40U for DactyScan40i device

- Visascan and VSQuality for Poliscan2, DactyScan40 and Visascan3 devices (Windows only)

NOTE: Level 1 normally is not directly accessible to application

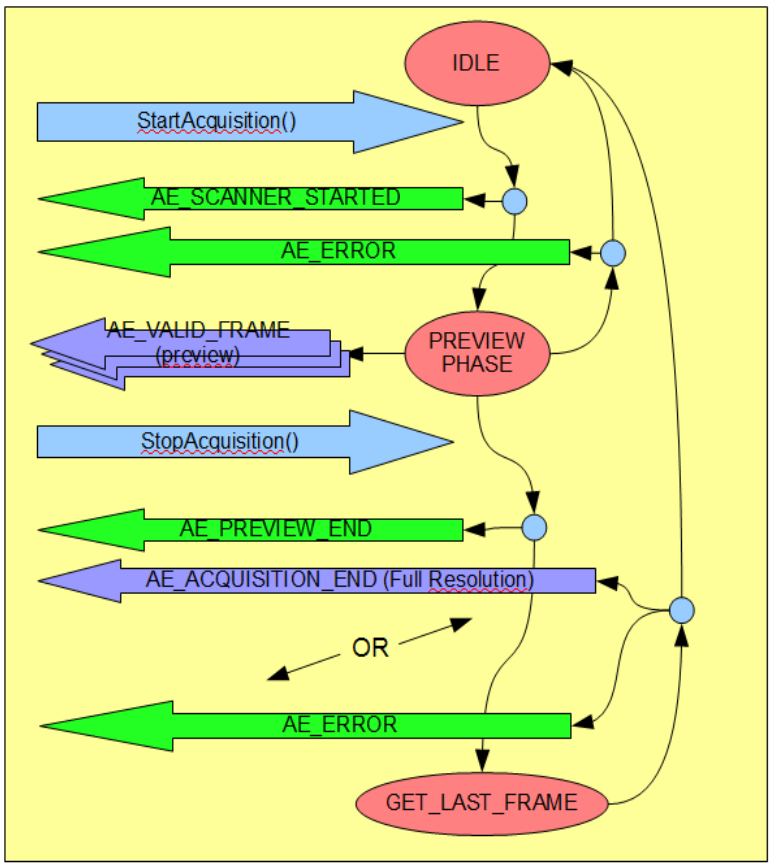
4.1.3 Level 2: GBMSAPI SDK

This is the first level which the developer has direct access to.
It allows the device management and the image acquisition.
For "device management" are intended all the functions allowing the application to retrieve information about the scanner features (objects that can be acquired, image size, supported scanner functionalities and so on) and the identification (serial number, library and so on) and to setup the device behaviour (calibration tools, acquisition options and so on). In order to develop applications making use GBMSAPI SDK, please refer to the comments put in the library header files (GBMSAPI_Defines.h and GBMSAPI_Library.h) .
The main software features of BASIC SDK are (depending on the device):

- Latent print elimination
- Rolling artefact evaluation
- Rolling process correctness check
- Flat objects sliding detection
- Flat objects auto-capture
- Automatic evaluation of some parameters like contrast, size and lower half palm completeness
- Automatic clipping region evaluation

NOTE: for the GBMSAPI library a .ini file (gbmsapi.ini) is provided, that allows to change in a very fast way some settings of the corresponding library (gbmsapi.dll). Range and meaning of parameters and how to set them is well explained in the comments of the .ini file. GreenBit recommends to use this file only when strictly necessary. For further information please contact the GreenBit support (support@greenbit.com).

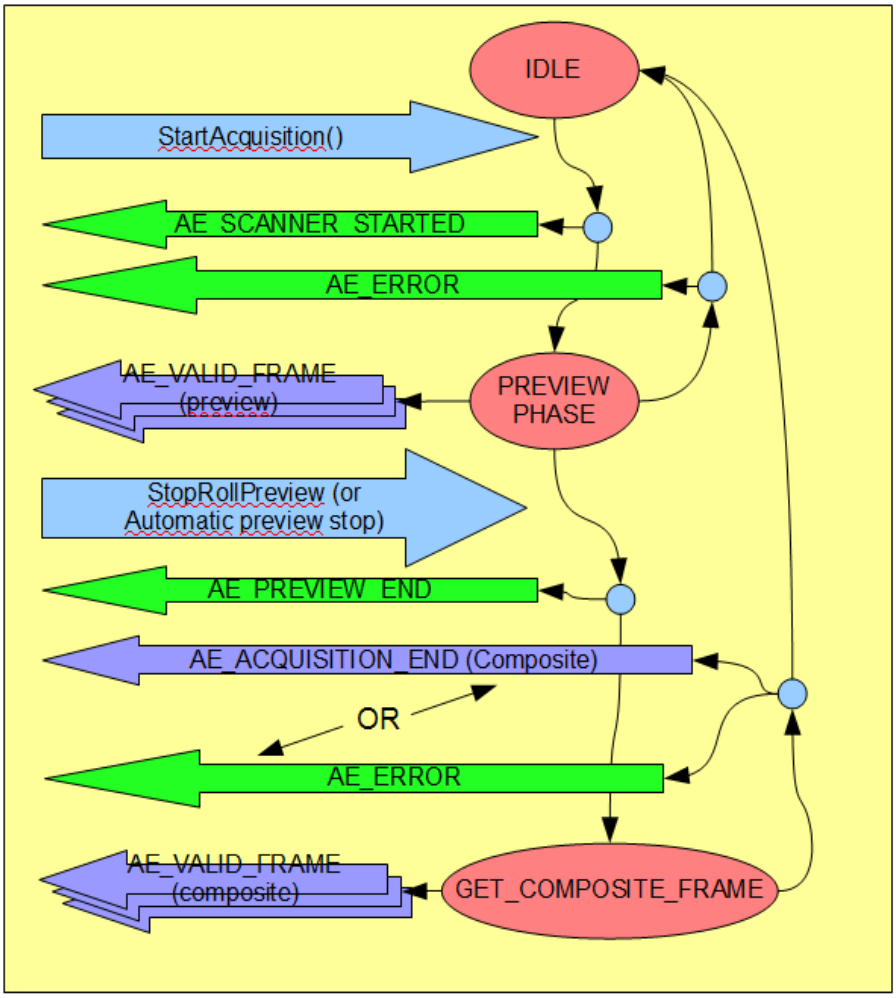
GBMSAPI Acquisition Thread States for flat capture



LEGENDA:

Command	Application commands (that is, GBMSAPI functions called by the application) that force the state switch
GBMSAPI state	GBMSAPI acquisition thread state
Acquisition event	Events fired by the GBMSAPI thread that must be processed by the application: such events carry Information only
Acquisition event	Events fired by the GBMSAPI thread that must be processed by the application; such events carry Information and a frame.

GBMSAPI Acquisition Thread States for roll capture



LEGENDA:

Command	Application commands (that is, GBMSAPI functions called by the application) that force the state switch
GBMSAPI state	GBMSAPI acquisition thread state
Acquisition event	Events fired by the GBMSAPI thread that must be processed by the application: such events carry Information only
Acquisition event	Events fired by the GBMSAPI thread that must be processed by the application; such events carry Information and a frame.

4.2 BASIC SDK: .NET-MONO WRAPPER

For each library belonging to level 2 (in this case the GBMSAPI library only) the corresponding wrapper for the .NET/MONO framework (in most cases the .dll files generated by Visual Studio compiler for .NET framework can be copied "as they are" in a MONO environment) is provided.
.NET Wrapper is given also in source code, and there its documentation can be found (as comments in the code).
Make sure that version 2.0 of the framework is installed (both in Linux and in Windows).

4.3 BASIC SDK: Java Wrapper

In order to easily integrate MULTISCAN SDK in a JAVA application, a wrapper has been provided for each SDK library. To be more precise, the wrapper has been written for JAVA 1.6 JDK version. Where needed, Readme files are provided for documentation. Anyway, Java wrapper is given in source code.

4.4 BASIC SDK: Programming Examples

BASIC SDK is provided with several examples to show how developers should use the SDK functions, both for C libraries and .NET/MONO wrapper libraries.

Their folder are named by the following template: Name_<UserInterfaceType>_<ProgrammingLanguage>.

The ones with "GettingStarted" prefix are to be considered as not complete, in the sense that only the strictly needed functions for performing an acquisition are used.

All the examples are provided with their source code (projects are created by using the Visual Studio 2008 IDE, except for the GettingStarted_Cpp_Example under Linux, developed with Eclipse).

NOTE1: For their functioning, all examples, except the C++ "GettingStarted", require a .NET framework (ver 2.0 at least) installed on the PC where they should run.

NOTE2: Furthermore, for Windows, C++ examples require also the CRT libraries for Visual C++ 2008 (included in the example directory).

Note:

Green Bit recommends application developers to follow as close as possible the way the examples are written in order to ensure both the independency of the application from the type of Green Bit used device (if it is among those supported by the SDK) and the correctness of the acquisition process.

4.4.1 GettingStarted C++ Example

This is a very simple console example that shows the use of the acquisition callback and some simple settings for the device. It also shows the acquisition of different objects (flat fingerprints, slaps, rolled fingerprints, palm): these objects are captured only if scanner supports them.

Green Bit recommends to use this example in order to focus the developer's attention to the acquisition process and/or as a base for tests.
For Windows, it's given as a Visual Studio project, while for Linux it's given with a makefile in order to be easily compiled by command line.

4.4.2 GettingStarted C# Example

This example is the C# translation of the GettingStarted C++ example. For Linux, the Program.cs file only is given, since the project was developed with Visual Studio. Anyway this file can be easily included into a Mono project.

4.4.3 BASIC C++ Example (for Windows only)

This is a complete GUI example that shows all the features of the scanner.
Green Bit recommends to use this example for testing all the scanner features.
Under Linux, this example is not provided.

4.4.4 BASIC C# Example

This example is the C# translation of the Basic C++ example. Its source code can be compiled with MonoDevelop IDE under Linux, though at the state of art MonoDevelop does not provide a graphic window design.

4.5 BASIC SDK: Application deployment

In order to deploy their applications, integrators must include the following components of the MultiScan SDK:

- GB Universal USB drivers (for Windows)
- libgb-transport (for Linux)
- Device specific dll (windows)/so (linux)
- VSRoll library for those scanners supporting rolled objects
- DSBeep library
- GBMSApi library
- GBMSApi_NET.dll for those applications using the GBMSApi .NET/MONO wrapper
- .ini files for libraries, where needed (see notes for scanners and GBMSAPI library)

The above mentioned libraries support the following operating systems (both 32-bit and 64-bit architectures): Windows XP, Windows 2000, Windows Vista, Windows 7.

The Linux version has been tested on Ubuntu 10.10 and Fedora 14 (Kernel 2.6)

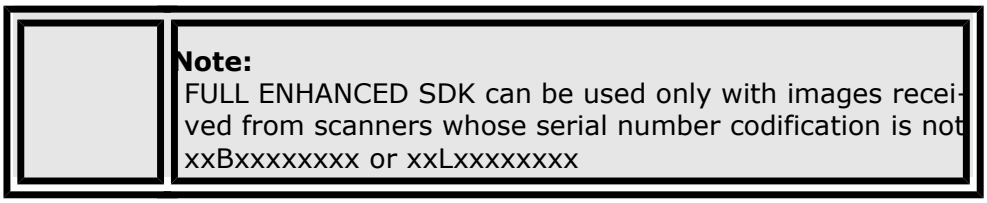
Integrators should also care that in the user PC the .NET/MONO framework (where required) and the CRT libraries for Visual C++ 2008 (in Windows) are present.

5 FULL ENHANCED SDK (Level 3)

FULL ENHANCED SDK includes the Level 3 of abstraction. The libraries which this SDK is composed of can be divided in two main categories:

- Graphical Interface libraries
- Image Processing Tools

For both of them a more detailed description will be given in the next paragraphs.



Note:
FULL ENHANCED SDK can be used only with images received from scanners whose serial number codification is not xxBxxxxxxxx or xxLxxxxxxxx

5.1 FULL ENHANCED SDK: C Interface Libraries

5.1.1 Graphical Interface libraries (GBMSGUI, GBMSGUI_QT)

These tools provide the developer with a graphical interface for image(s) acquisition. As options, they perform the segmentation, the fingerprint image quality assessment, the image enhancement and the sequence check. A lot of parameters are settable in order to fulfil the needs of any integrator: if these parameters have not been changed, they have the values recommended by Green Bit.
ISO characters for every language are supported.
In case that some problems appear during the acquisition, the user is always advised through short and meaningful messages.
GBMSGUI_NET library (of which the GBMSGUI is the C wrapper) is written in C#, and lies on the .NET framework; GBMSGUI_QT one is written in C++ and lies on the QT graphical framework (so that its code can be reused for Linux environment with little changes).
A more detailed description of these interface is given in the related document (GBMSGUI_UserGuide.pdf) that can be found in DOC\GBMSGUI folder inside the FULL ENHANCED SDK CD.
Developers' user guide is given through the libraries header files (GBMSGUI.h, GBMSGUI_QT.h) that can be found in the SDK\<GUI_FOLDER>\DLL (where <GUI_FOLDER> is respectively "GBMSGUI" and "GBMSGUI_QT") folder inside the FULL ENHANCED SDK CD.
For customization GBMSGUI, GBMSGUI_NET and GBMSGUI_QT are given together with their full source code (a complete project written in C# and C++ language, depending on the environment). They can be found in SDK\GBMSGUI\PROJECT SOURCE CODE folder inside the FULL ENHANCED SDK CD.

Very Important Note:
GBMSGUI_VCL development will be dismissed starting from sdk versions greater than 2.7. This is why its source code cannot be distributed (being based on licensed Borland libraries).

5.1.2 Image Processing Tools Libraries

The Green Bit "Image Processing Tool SDK" is for those developers who want to customize their application but at the same time want to have some tools helping them with the fingerprint image processing. Since these tools perform only the image processing, they must be used together with the BASIC SDK libraries (provided for each Green Bit live scanner) in order to acquire fingerprint images.
Developers' user guide is given through the header file provided with each library.

1) GBFINIMG library

This tool performs:

- Segmentation from a slap (as for EFTS Appendix N standard)
- Provision of the segments' bounding boxes coordinates (EFTS Appendix N)
- Provision of single fingers image inside a slap
- Automatic segment naming (index, middle, ring, little)
- Detection of ambiguous situations in the segmentation, such as mixing-up right and left hands
- Possibility of manual segment naming performed by the operator
- Possibility of automatic segment naming even in case of missing fingers (providing that information about missing fingers is given before segmentation process)
- Possibility of automatic segment naming even in case of missing fingers (providing that a correct information about the hand – left or right – is provided before segmentation process)
- Possibility of hand orientation up to 45º
- Detection of finger rotation by 180º
- Green Bit proprietary quality assessment
- Invalid pattern detection (i.e. the distinction between a real fingerprint and a leather glove, a rubber or other objects)

- Incomplete pattern detection (i.e. it checks that the finger core is present and the basic/distal flow is complete)
- detection of ambiguous situations in the segment naming (such as Left/Right hands mixing up)
- Halo effect removal
- Optional faster segmentation process, with or without provision of segment data (in this case segments number only is provided), useful for real time segments quality evaluation

  NOTE: in order to perform the segment rotation when the hand orientation is different from 0 degrees, the GBImgTran.dll library is needed as well. It can be found in the same folder where the GbFinImg.dll is placed.

2) GBNFIQ library

This tool performs fingerprint quality assessment following the NISTIR 7151 standard. In addition, it is capable of getting minutiae from the image and storing them into a simple structure.

3) LFS library

This tool allows getting minutiae from a fingerprint image and its output is ready to be used as an input of the "BOZORTH" library. It's used for sequence check.

4) BOZORTH library

This tool makes a comparison between two sets of minutiae (got from the output of the "LFS" library) and returns a matching score: the higher is this matching score, the higher is the probability that the two fingerprints are from the same finger. This tool is based on the standard FBI "Bozorth3" algorithm. It's used for the sequence check.

5) AN2K_LIB library

This tool allows the developer to:

- Create a buffer compatible with ANSI/NIST ITL-1 2007 standard (i.e. to create a type-1 record)
- Create and manage type-2 records in order to store information about the scanning equipment and about amputated or bandaged fingers
- Create and manage type-4 records where single fingerprint images are stored

- Create and manage type-14 records where single/slap fingerprint images are stored, together with information about amputated or bandaged fingers, segments bounding boxes, NFIQ and the proprietary quality assessment
- Create and manage type-15 records where palmprint images are stored, together with information about proprietary quality assessment
- Parse any buffer compatible with the ANSI/NIST ITL-1 2007 standard

In the examples a tool is given so that the user can print in a human readable form the contents of a buffer compatible with ANSI/NIST ITL-1 2000 standard (very close to ANSI/NIST ITL-1 2007 standard). This tool, called an2ktool.exe, is provided in source code by FBI. Here the compiled version is provided, that needs "cygwin.dll" library for properly working.

6) WSQ_Pack library

This tool allows the developer to compress the raw images taken from the scanner in FBI WSQ (Wavelet Scalar Quantization) standard. Some options, such as the compression rate, are configurable by the developer. This tool is certified by FBI (for compression rate values 5:1 and 15:1). Compression level is programmable (up to 15:1).

WSQ library is certified to be compliant to the "Wavelet Scalar Quantization (WSQ) Gray-Scale Fingerprint Image Compression Specification Version 3.1"

7) GBJPEG library

This tool allows the developer to compress the raw images taken from the scanner in the JPEG or JPEG2000 standard. Some options, such as the compression rate or the JPEG quality, are configurable by the developer. Compression level is programmable (up to 14:1).

8) GBFIR library

This tool allows the developer to store images inside ISO/IEC 19794-4 or ANSI-INCITS 381 finger images record and to read image and fields from such records.

5.2 FULL ENHANCED SDK: .NET-MONO WRAPPER

For each library belonging to level 3 the corresponding wrapper for the .NET/MONO framework (in most cases the .dll files generated by Visual Studio compiler for .NET framework can be copied "as they are" in a MONO environment) is provided.
.NET/MONO Wrapper is given also in source code, and there its documentation can be found (as comments in the code).
Make sure that version 2.0 of the framework is installed (both in Linux and in Windows).

5.3 FULL ENHANCED SDK: Java Wrapper

In order to easily integrate MULTISCAN SDK in a JAVA application, a wrapper has been provided for each SDK library. To be more precise, the wrapper has been written for JAVA 1.6 JDK version. Where needed, Readme files are provided for documentation. Anyway, Java wrapper is given in source code.

5.4 FULL ENHANCED SDK: Programming examples

FULL ENHANCED SDK provided with several examples to show how developers should use the SDK functions, both for C libraries and .NET/MONO wrapper libraries.
Their folder are named by the following template: Name_<UserInterfaceType>_<ProgrammingLanguage>.

All the examples are provided with their source code (projects are created by using the Visual Studio 2005/2008 IDE).

NOTE1: For their functioning, most of examples require a .NET/MONO framework (ver 2.0 at least) installed on the PC where they should run.
NOTE2: Furthermore, C++ examples require also the CRT libraries for Visual C++ 2008 (for Windows).

5.4.1 Image Processing Tool C++ Example

This is a complete console example that shows how the functionalities of the image processing tool can be used within a C++ program. A 4-4-2 acquisition is simulated by loading images from the "SAMPLES" folder, faking a missing finger inside a hand. Also the sequence check and a NIST file building are performed.
Green Bit recommends to use this example as a base for other applications involving the image processing tool.

5.4.2 Image Processing Tool C# Example

This example is the C# translation of the Image Processing Tool C++ Example

5.4.3 Gbfinimg C# Example

This is a complete GUI example that shows all the functionalities of the GBFINIMG library.

Green Bit recommends to use this example as a base for applications using the GBFINIMG library.

5.4.4 GBFIR C++ Example

This is a complete GUI example that shows all the functionalities of the GBFIR library.
Green Bit recommends to use this example as a base for applications using the GBFIR library.

5.4.5 GBMSDemo C# Example

This is a complete GUI example that uses the GBMSGUI and the image processing tool libraries. A guide to this demo ("GBMSDemo_UserGuide.pdf" file) can be found in the "DOCS" folder of the FULL ENHANCED SDK cdrom.
Green Bit recommends to use this example as a base for complete solutions involving fingerprints acquisition.

5.4.6 GBMSGUI_QT Test C++ Example

This is a complete GUI example that uses the GBMSGUI_QT and the image processing tool libraries.
Green Bit recommends to use this example as a base for other applications involving the GBMSGUI_QT library.

5.4.7 Libraries source code (GBMSGUI_NET and GBMSGUI_QT)

As told before, the GBMSGUI_NET, its C wrapper (GBMSGUI) and the GBMSGUI_QT are provided with their source code. Visual Studio projects of these 2 libraries are provided into the examples folder.
Green Bit recommends the integrators to use this source code as a base for developing their own GUI (if required, otherwise these libraries can be used "as they are").

5.5 FULL ENHANCED SDK: Application Deployment

In order to deploy their applications, integrators must include the following components of the MultiScan SDK:

- All the components specified in the paragraph 4.5 (BASIC SDK: Application deployment), included the GBMSAPI_Net dll
- The GBMSGUI_NET component (if the GBMSGUI is used only in the .NET/MONO form)
- Both the GBMSGUI_NET and the GBMSGUI dlls for those applications using the standard C GBMSGUI library for Windows. Furthermore, for Windows, the GBMSGUI dll also needs the CRT libraries for Visual C++ 2005 (included in the GBMSGUI package)
- The GBMSGUI_QT for those applications using this component, together with the QT libraries which the gui lays on (QtCore4, QtGui4), the CRT libraries for Visual C++ 2005 and the Redistributable CRT Service Pack 1 for Visual C++ 2005 (for Windows).
- The Image Processing Tools libraries and their .NET wrappers (for those applications written for the .NET framework). The list of components can be found in the paragraph 5.1.2 (Image Processing Tools Libraries).
- DSBeep library (needed by GBMSGUI)
- Furthermore, the CRT libraries for Visual C++ 2008 (included in the GBMSGUI package) for Windows,

The above mentioned libraries support the follwing operating systems (both 32-bit and 64-bit architetcures): Windows XP, Windows 2000, Windows Vista, Windows 7.

The Linux version has been tested on Ubuntu 10.10 and Fedora 14 (Kernel 2.6)

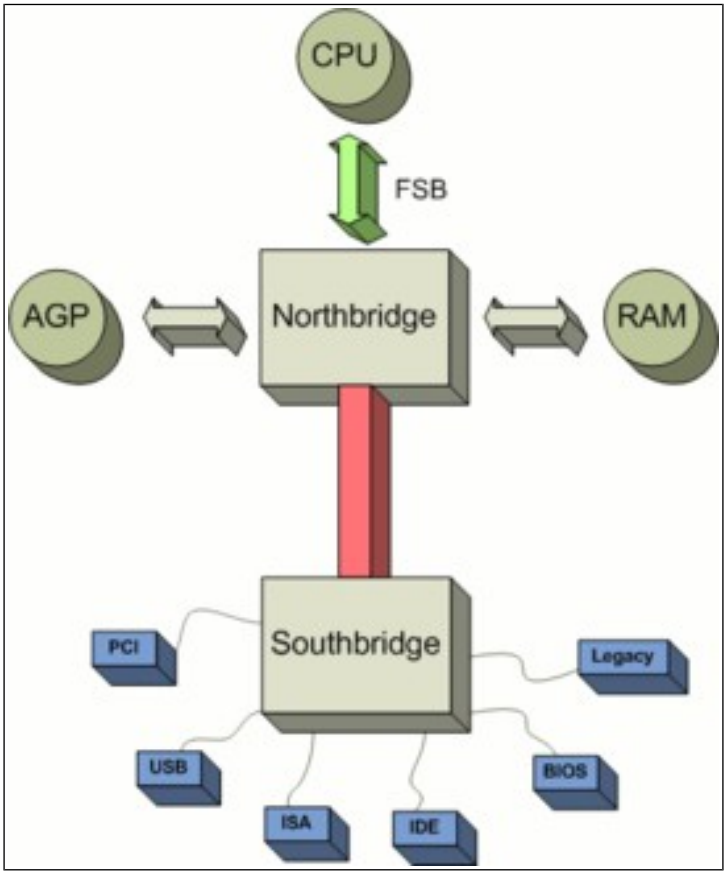
6 Troubleshooting

Sometimes, especially with older generation PCs/Notebooks, when acquiring large images at a very high speed, some troubles can happen with the USB connection. Therefore users can experience "SCANNER COMMUNICATION" or "DEVICE NOT RESPONDING" errors from the lower level libraries. These errors mean that the scanner is not responding to PC commands (no data from scanner) or that several bytes are lost during image flow.
Sometimes may happen that scanner is not working properly, but in the most of cases this behavior is due to low PC resources.
In order to understand which resources are to be considered and what "low" means, we should make a premise.
Data coming from the scanner must pass from the USB controller to the CPU, and here they must be processed. During data processing CPU makes a strong use of the memory, therefore a third data flow (from CPU to memory) must be considered. So we have three critical resources, whose power must be taken in count:

- Data flow (USB → CPU)
- Data flow (CPU → RAM and viceversa)
- CPU speed (and RAM capacity)

CPU speed and RAM capacity normally are provided with PC documentation, so they are easy to be found. The two resources left, linked to the data flow, depend on the chipset, whose parameters normally are not provided together with the PC documentation. They are at least as important as CPU power and RAM size. Infact in case of wide USB trafic, they are the bottle-neck for data flow.

A schema representing the chipset is the following:



Therefore what must be measured in order to understand whether an efficient communication is performed is the FSB (Front Side Bus) speed and the Memory Bus (the bus linking the Northbridge to the RAM) speed.

In order to measure these parameters, the "dmidecode" tool can be used (usage is better described in the "TroubleShooting_CheckList.xls" file, where the parameter values can be also collected, compared to the minimal requirements for the scanner and eventually delivered to the Green Bit technical support).

Sometimes particular attention should be put on the graphic board (indicated with "AGP"), that could interfere with the FSB traffic (the northbridge, as it can be seen, acts as a collector for data interchange with peripherals, memory and graphic card).

NOTE: for problems with particular Intel chipsets (Intel 5 / 3400 Series) see "TroubleShooting intel 5 series.pdf" document.

Green Bit S.p.A.
Via Rivalta, 9
10095 Grugliasco (TO)
Tel: +39 011 7703811
Fax: +39 011 7703880
info@greenbit.com