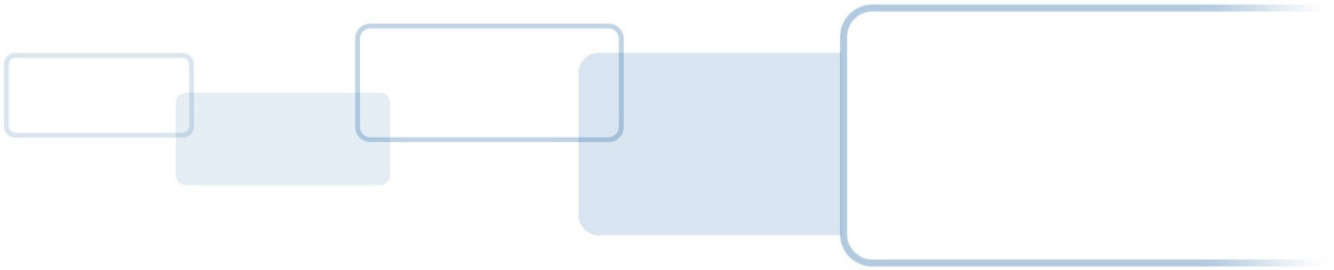
Lumidigm vCOM Integration Kit

PLT-02304, Rev. A.2
Software Version 6.00
August 2016



hidglobal.com

Copyright

Trademarks

HID GLOBAL, HID, and the HID logo are the trademarks or registered trademarks of HID Global Corporation, or its licensors, in the U.S. and other countries.

Lumidigm is a registered trademark of Lumidigm, Inc.

Revision History

Date	Description	Version
08/15/16	Software Version 6.00	A.2
07/24/15	V371 release.	A.1
12/16/14	First release as part of HID Global (1.6)	A.0

Contacts

For additional offices around the world, see **www.hidglobal.com** corporate offices.

North America & Corporate	**Lumidigm**
611 Center Ridge Drive Austin, TX 78753 USA Phone: 866-607-7339 Fax: 949 732 2120	For Lumidigm specific issues: Website: http://www.hidglobal.com/lumidigm-technical-support Email: Lumidigm@hidglobal.com Phone: 505 272 7057
HID Global Customer Support: support.hidglobal.com	

Contents

1 Overview

The vCOM Integration Kit is designed to assist software developers as they integrate their client applications with Lumidigm sensors. The Kit provides a sample library which implements the Lumidigm vCOM protocol along with set of examples of integrating the library with various client applications. One of these examples, the VCOMExampleApp, demonstrates multi-platform integration. Another example presents a solution for integrating with Java clients. The identification features of Lumidigm's M30x, M31x, V30x and V31x sensors are exhibited in the Identification Demo example. Finally, an example is provided on how to update firmware on a M30x sensor.

1.1 System Requirements

These requirements are general system requirements required for the vCOM Integration Kit as a whole. Some of the example applications are not currently supported on all platforms or may require additional requirements that will be described in the detailed documentation that is provided for the example.

All supported platforms require one or more of the following Lumidigm fingerprint sensors:

- V30x-x0 (embedded) sensor with 9532 firmware or higher
- Identification requires firmware higher than 9538
- M30x sensor with 16508 firmware or higher
- M31x sensor
- V31x sensor

Note: M31x and V31x series sensors are not currently supported on WinCE.

See *Appendix A: vCOM Supported Functionality* for a list of supported functionality for the Lumidigm sensor based on SKU.

1.1.1 Microsoft Windows XP, Microsoft Windows 7/8/10 x86/x64

1. Microsoft Visual Studio 2005
2. WINDDK (Driver Development Kit) v3790.1830 or v7600.16385.1 (USB only)
3. Appropriate hardware interfaces (USB 2.0/DB-9)
4. Lumidigm drivers installed

1.1.2 Microsoft Windows CE

1. Microsoft Visual Studio 2005
2. Microsoft Windows Embedded CE 5.0 and/or 6.0 SDK
3. Microsoft ActiveSync (or some other mechanism for file transfer and debugging)
4. Appropriate SDK for your specific CE device
5. Appropriate hardware interfaces (USB 1.1 or higher)

1.1.3 Linux

1. This version of the source code has been implemented and tested on Ubuntu 12.04 LTS (32 and 64 bit). The following configurations are also supported:
   - Any Linux distribution kernel version 2.6.23 or later (should work but may require minimal effort).
2. GCC version 4.1.2
3. Appropriate hardware interfaces (USB 2.0/DB-9)

1.2 Installation Contents

The following folders are created in the directory that is specified during installation. This section summarizes the contents of the installation directory.

1.2.1 Windows Installation Directory Structure

bin	
install	Contains built versions of example applications and DLLs for Microsoft Windows XP.
Build	
Win32	Contains VS2005 Solution and Project files to build VCOMExample library for Windows XP, 7 and 8
WinCE	Contains VS2005 Solution and Project files to build VCOMExample library for Windows CE
doc	Contains documentation for the vCOM Integration Kit and example projects
drivers	
WinCE/VX00DrvCE	Contains the required source code along with VS2005 Solution and Project files to build driver for Windows CE. Also contains a registry file to install the driver correctly.
SampleCode	
VCOMExampleApp	Contains source code for the VCOMExampleApp console application along with VS2005 Solution and Project files to build for Windows XP, 7, 8 and CE.
LumiEnroll	Contains source code and project files for a client application that is implemented in Java. Not distributed with the x64 installer.
IdentificationDemo	Contains source code and project files for the MFC client application that demonstrates the identification features of M30x, M31x, V30x and V31x sensors.
M30XFWUpdater	Contains source code and project files for an MFC client application that illustrates how to update firmware on a M30x sensor.

1.2.1.1 VCOMExample

Contains all the source code to a library which implements the vCOM protocol.

API	API Directory
bin	Output Directory for VComExample.dll (on Windows platforms)
include	Include directory
lib	Output Directory for VComExample.lib (on Windows platforms)
src	Source Files

1.2.2 Linux Installation Directory Structure

bin	Output directory for VCOMExampleApp
doc	Contains a udev rule to make Lumidigm sensors usable by any user and a script to install it.
driver	Contains a udev rule to make Lumidigm sensors usable by any user and a script to install it.
SampleCode	
VCOMExampleApp	Contains source code for the VCOMExampleApp console application along with a Makefile.
LumiEnroll	Contains source code for a Java GUI application along with project files for building the project within the NetBeans development environment.

1.2.2.1 VCOMExample

Contains all the source code and Makefile to build a library which implements the vCOM protocol. The library will be built into an archive (.a) and shared object (.so) library. The VCOMExampleApp will link against the archive statically.

API	API Directory
include	Include directory
src	Source Files
lib	libVCOMExample.a and .so

2 Example Applications

This section will provide an overview of the various example applications that are distributed with the Kit.

Note: None of the example code provided in this Kit is designed to be used in a production system. Rather, it was created to assist software developers in the design and implementation of their own production code.

2.1 vCOM Example Application

The VCOMExampleApp provides example code and projects for integrating vCOM over RS-232 and USB for the following platforms: Microsoft Windows XP, Microsoft Windows 7, Microsoft Windows 8, Microsoft Windows CE, and various Linux distributions.

Please refer to *the Lumidigm vCOM Example* document for a full description of the application. This document is installed in the *doc* folder.

2.2 Identification Demo

The Identification Demo is a MFC Dialog application that demonstrates Lumidigm's identification features on Lumidigm sensors that support Identification commands. See *Appendix A: vCOM Supported Functionality*. **Note:** This example application is supported only on Windows XP, Windows 7, and Windows 8.

Please refer to the *Lumidigm Identification Demo* document for a full description of the application. This document is installed in the *SampleCode/IdentificationDemo/doc* folder.

2.3 M30x Firmware Updater

The M30XFWUpdater application demonstrates how to update firmware on a M30x sensor. **Note:** This example application is supported only on Windows XP, Windows 7, and Windows 8.

Please refer to the *Lumidigm M30x FW Updater Example* document for a full description of the application. This document is installed in the *SampleCode/M30XFWUpdater/doc* folder.

2.4 CSharp vCOM Example

The CSharp vCOM Example is a C# .Net dialog application that demonstrates how to perform a fingerprint capture on Lumidigm V30x, M30x, M31x, and V31x sensors. The application also demonstrates how to perform the identification function on Lumidigm M30x, M31x and V31x sensors, and V30x sensors with firmware greater than 9538.

Please refer to the *Lumidigm CSharp vCOM Example* document for a full description of the application. This document is installed in the *SampleCode/CSharpVCOMExample/doc* folder.

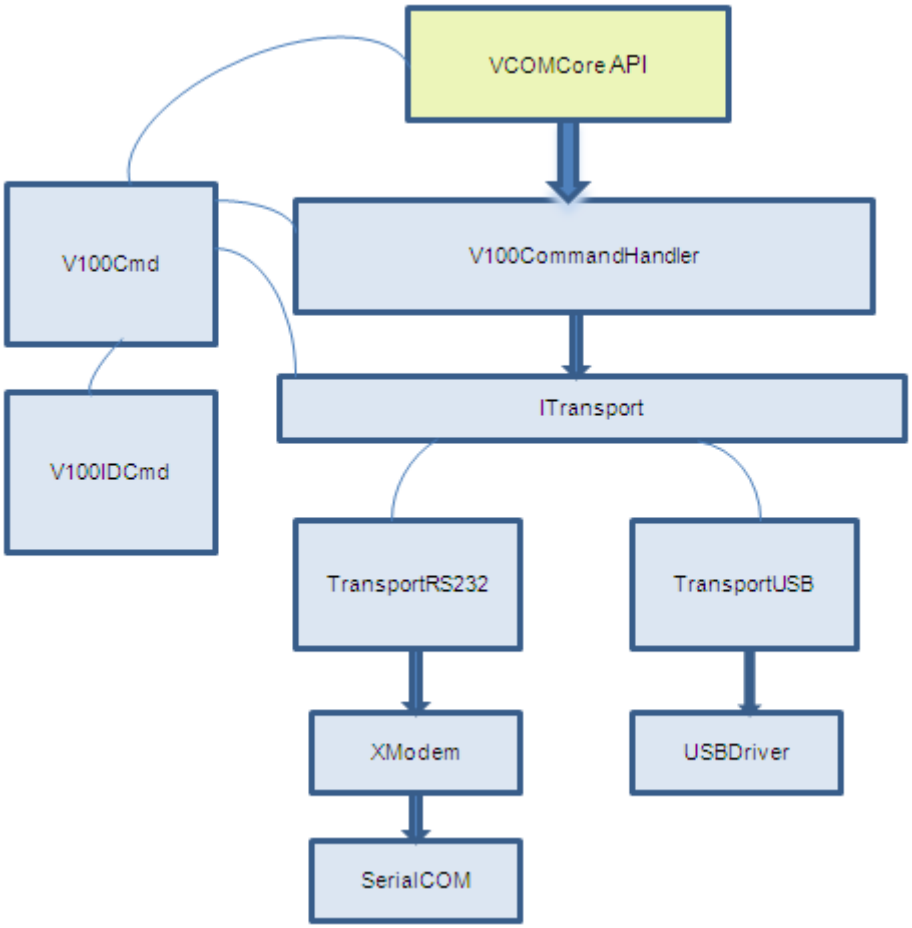
2.5 Java Example

The LumiEnroll application is an example of integrating a Java client using the VCOMExample library to capture, match, and verify fingerprint images from a Lumidigm sensor.

Please refer to the *Java Example* document for a full description of the application. This document is installed in the *SampleCode/LumiEnroll/doc* folder.

3 VCOMExample Library API Overview

3.1 VCOMExample Source Diagram



Note: RS-232 is not currently supported in the vCOM Integration Kit on WinCE.

3.2 Component Descriptions

3.2.1 VCOMCore

VCOMCore.cpp represents the API layer of the example project. There is a one-to-one relationship between the `_V100_COMMAND_SET` and API calls to demonstrate how to properly utilize each command. The pattern used is similar for each call, which is to create the challenge packet using the V100CommandHandler, populate any input fields, then send off the transaction. Once a response packet returns, we can check to see if it is an error packet or a proper response packet. If it is a proper response packet, we then proceed to populate any outgoing parameters in the API and return.

The API also suggests some “Macro” commands that should be made available in any implementation. Calls to Capture or Verify illustrate this concept.

Remember that the sample implementations are not all-inclusive, and that more extensive error handling should be implemented in production versions of your code.

3.2.2 V100CommandHandler

The V100CommandHandler is responsible for creating the command classes for the VCOMCore level as well as for abstracting the transport layer from the rest of the library.

3.2.3 ITransport

Abstract Base Class design to abstract transport layer from any layers above.

3.2.4 TransportRS232

The TransportRS232 class is responsible for implementing the transport protocol for RS-232. See the function “TransmitCommand” for implementation details on how the TransportRS232 class communicates with the host, using standard XModem 128-CRC16.

3.2.5 TransportRS232_NX

The TransportRS232_NX class is responsible for implementing the transport protocol for RS-232. See the function “TransmitCommand” for implementation details on how the TransportRS232 class communicates with the host, using simple RS232 transport class.

3.2.6 Xmodem

A standard XModem-CRC implementation, with 128 byte packets.

3.2.7 SerialCom

Communicates with the physical layer. A good candidate to get ported immediately to the OS of choice. Basically implements WriteByte and ReadByte, with timeout.

3.2.8 SerialComPOSIX

The POSIX specific version of this component.

3.2.9 STranRS232

Simplified RS232 transport class.

3.2.10 TransportUSB

The TransportUSB class is responsible for implementing the transport protocol for USB. Check the implementation of the function “TransmitCommand” for protocol specifications. Notice that the maximum packet transfer size, for both read and write, is 64KB and all larger packets are split appropriately into multiple calls to Read/Write.

3.2.11 TransportUSB_CE

The Windows CE specific version of this component.

3.2.12 TransportSE

The TransportSE class is responsible for implementing the transport protocol for communicating with the SEngine component of the Lumidigm streaming sensors over USB. Check the implementation of the function “TransmitCommand” for protocol specifications.

3.2.13 TransportSEPOSIX

The POSIX specific version of this component using libusb 1.0.

3.2.14 USBDriver

A standard Win32 USB 2.0 Bulk transfer mode implementation for sending/receiving data up to 64KB.

3.2.15 V100Cmd

The V100Cmd classes implement the vCOM protocol. They should be candidates for cross-compilation on various platforms that support a variant of C++, saving the integrator a lot of the work associated with implementing such a granular protocol. The code for unpacking challenge commands and packing response commands has been left in these sets of classes for reference, though they should never be called from the client code.

3.2.16 V100IDCmd

This follows the same model as V100Cmd, but is limited to implementation of the 1:N identification commands. This is done for readability reasons only.

3.2.17 IMemMgr.h

A skeleton of the Lumidigm internal memory manager.

4 General API Commands

4.1 V100_Arm_Trigger

Starts presence detection and image stack acquisition.

```
V100_ERROR_CODE V100_Arm_Trigger (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_TRIGGER_MODE mode)
```

Parameters

pDev	Pointer to device handle
mode	Type of trigger mode to set

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See V100_TRIGGER_MODE for information on types of trigger modes.

TRIGGER_ON: Arms the device trigger, and starts the processing chain as described in the V100_INTERFACE_COMMAND_TYPE structure.

TRIGGER_OFF: If presence detection is running, this disarms the trigger and returns the device to an idle state. It returns GEN_OK or GEN_ERROR_APP_BUSY if the system is busy. In either case user must poll for completion using V100_Get_OP_Status for macro commands and V100_Get_Acq_Status for atomic commands.

TRIGGER_FINGER_DETECT: Arms the device trigger and starts finger detection mode. User can poll for status using CMD_GET_ACQ_STATUS for finger presence or not.

See also

V100_Get_OP_Status, V100_Get_Acq_Status

4.2 V100_Cancel_Operation

Cancels capture-related commands

```
VCOM_CORE_EXPORT V100_ERROR_CODE V100_Cancel_Operation (const
V100_DEVICE_TRANSPORT_INFO * pDev)
```

Parameters

pDev	Pointer to device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

You can cancel V100_Arm_Trigger, V100_ID_Identify, V100_ID_Verify_User_Record,

V100_ID_Enroll_User_Record are V100_ID_Verify_Many using this command. It returns GEN_OK or GEN_ERROR_APP_BUSY if the system is busy. In either case user must poll for completion using V100_Get_OP_Status for macro commands and V100_Get_Acq_Status for atomic commands.

See also

- V100_Arm_Trigger, V100_ID_Identify
- V100_ID_Verify_User_Record
- V100_ID_Enroll_User_Record
- V100_ID_Verify_Many
- V100_Get_OP_Status, V100_Get_Acq_Status

4.3 V100_Capture

Acquires images, returns composite image (if set), minutia template (if set) and spoof score (if supported).

`V100_ERROR_CODE V100_Capture (`V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar * *pCompositeImage*, uint & *nWidth*, uint & *nHeight*, uchar * *pTemplate*, uint & *nTemplateSize*, int & *Spoof*, int *getComposite*, int *getTemplate*)

Parameters

pDev	Pointer to device handle
pCompositeImage	Pointer to the composite image to be returned
nWidth	Composite image width (number of pixels per row)
nHeight	Composite image height (number of pixels per column)
pTemplate	Pointer to the minutia template (max 2048 bytes) to be returned
nTemplateSize	Size of returned template
Spoof	Returned spoof score. Returns -1 if spoof not supported.
getComposite	Get composite image, otherwise 0
getTemplate	Get template, otherwise 0

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful

Remarks

Arms device trigger, waits for user presence detection, acquires image and returns composite image and minutia template if set. Returns spoof score if supported.

See also

V100_Arm_Trigger, V100_Get_Acq_Status, V100_Get_Composite_Image, V100_Get_Template

4.4 V100_Close

Closes communication to device.

`V100_ERROR_CODE V100_Close (V100_DEVICE_TRANSPORT_INFO * pDev)`

Parameters

pDev	Pointer to Device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Applications must call this function to close communication with a device before exiting.

See also

None.

4.5 V100_Config_Comport

Used to change the baud rate, data bits and flow control settings of the serial communication channel.

`V100_ERROR_CODE V100_Config_Comport` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, uint *nBaudRate*)

Parameters

pDev	Pointer to device handle
nBaudRate	Baud rate to set. Selectable (9600, 19200, 38400, 57600, 115200).

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Only supported for RS232 communication.

See also

None.

4.6 V100_Get_Acq_Status

Returns status of current acquisition.

`V100_ERROR_CODE V100_Get_Acq_Status` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_ACQ_STATUS_TYPE * *pACQ_Status*)

Parameters

pDev	Pointer to device handle
pACQ_Status	Pointer to acquisition status to be returned

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _V100_ACQ_STATUS_TYPE for information on types of acquisition status.

See also

V100_Capture

4.7 V100_Get_Cmd

Returns command structure.

`V100_ERROR_CODE V100_Get_Cmd` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_INTERFACE_COMMAND_TYPE * *pCmd*)

Parameters

pDev	Pointer to device handle
pCmd	Pointer to Command structure to be returned

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Returns current settings of USER controllable features. See _V100_INTERFACE_COMMAND_TYPE for information on Command Structure

See also

None.

4.8 V100_Get_Composite_Image

Returns the latest composite image from composite image buffer and associated spoof value.

`V100_ERROR_CODE V100_Get_Composite_Image` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar * *pImage*, int * *SpoofValue*, uint * *nImageSize*).

Parameters

pDev	Pointer to device handle
pImage	Pointer to the Composite image to be returned
SpoofValue	Pointer to the returned spoof value. -1 if not supported
nImageSize	Pointer to the size of returned Composite image

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Returned Composite image is the last composite image processed. The dimensions and image format of the image returned can be found by issuing a call to V100_Get_Config, and apply as follows:

Width	Composite_Image_Size_X
Height	Composite_Image_Size_Y
Format	8-BPP monochrome

See also

V100_Get_Image

4.9 V100_Get_Config

Returns device configuration structure which includes definitions of all supported services.

`V100_ERROR_CODE V100_Get_Config` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_INTERFACE_CONFIGURATION_TYPE * *ICT*)

Parameters

pDev	Pointer to device handle
ICT	Pointer to configuration structure to be returned.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _V100_INTERFACE_CONFIGURATION_TYPE structure for information on configuration structure

See also

None.

4.10 V100_Get_FIR_Image

Returns the latest composite image in FIR (finger image record) format

`V100_ERROR_CODE V100_Get_FIR_Image` (const V100_DEVICE_TRANSPORT_INFO **pDev*, _V100_FIR_RECORD_TYPE *FIRType*, _V100_FINGER_PALM_POSITION *FingerType*, uchar* *pFIRImage*, uint* *nFIRImageSize*)

Parameters

pDev	Pointer to device handle
FIRType	Type of FIR record to be returned
FingerType	Type of finger to be set
pFIRImage	Pointer to the FIR image to be returned
nFIRImageSize	Pointer to the size of FIRImage allocated/returned

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Returned FIR image is the last composite image processed and packed in FIR format. Client must allocate the pFIRImage buffer with size as follows.

For LUMI_FIR_ANSI:	nFIRImageSize = ANSI_381_HDR_SIZE + composite image size
For LUMI_FIR_ISO:	nFIRImageSize = ISO_19794_4_HDR_SIZE + composite image size

The dimensions of the composite image can be found by issuing a call to V100_Get_Config, and apply as follows:

Width	Composite_Image_Size_X
Height	Composite_Image_Size_Y
Format	8-BPP monochrome

FIR types supported are:

LUMI_FIR_ISO	ISO/IEC 19794-4:2005
LUMI_FIR_ANSI	ANSI/INCITS 381-2004

See also

V100_Get_Config

4.11 V100_Get_GPIO

Gets GPIO mask

```
VCOM_CORE_EXPORT V100_ERROR_CODE V100_Get_GPIO
```
(const V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar & *mask*)

Parameters

pDev	Pointer to device handle
mask	If successful returns GPIO mask

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Note: This functionality is not currently supported on V30x sensors.

See also

V100_Set_GPIO

4.12 V100_Get_Image

Returns current image from _V100_IMAGE_TYPE buffer.

`V100_ERROR_CODE V100_Get_Image` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_IMAGE_TYPE *type*, uchar * *pImage*, uint & *nImageSize*)

Parameters

pDev	Pointer to device handle
type	Type of image to be returned
pImage	Pointer to the image to be returned
nImageSize	Size of returned image

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _V100_IMAGE_TYPE for information on types of images. Valid types of images to get for general use are IMAGE_COMPOSITE, IMAGE_VID_STREAM and IMAGE_WSQ. The dimensions and image format of the image returned can be found by issuing a call to V100_Get_Config, and apply as follows:

IMAGE_COMPOSITE

Width	Composite_Image_Size_X
Height	Composite_Image_Size_Y
Format	8-BPP monochrome

IMAGE_VID_STREAM

Width	Native_Image_Size_X
Height	Native_Image_Size_Y

Format Bayer-pattern BGGR

IMAGE_WSQ

The size of the image buffer client needed to allocate is Composite_Image_Size_X*Composite_Image_Size_Y bytes. The actual size of the WSQ image returned will be nImageSize. Client can set the WSQ compression ratio using V100_Set_Option call with OPTION_SET_WSQ_COMPRESSION_RATIO. Default Compression ratio used is 11:1 (bit rate of 0.7273).

See also

V100_Get_Composite_Image

4.13 V100_Get_Num_USB_Devices

Returns the number of devices attached to the system.

`V100_ERROR_CODE V100_Get_Num_USB_Devices` (int * *nNumDevices*)

Parameters

nNumDevices	Number of devices attached to system

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None.

See also

None.

4.14 V100_Get_OP_Status

Retrieves the current status of Macro operation.

`V100_ERROR_CODE V100_Get_OP_Status` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_OP_STATUS* *opStatus*)

4.14.1.1 Parameters

pDev	Pointer to device handle
opStatus	OP Status structure returned upon success. See table in this section.

4.14.1.2 Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

4.14.1.3 Remarks

None.

4.14.1.4 See also

- V100_Enroll_User
- V100_Verify_User
- V100_Format_DB
- V100_ID_Create_DB
- V100_ID_Set_Working_DB
- V100_ID_Identify
- V100_ID_Identify_378
- V100_ID_Verify_User_Record
- V100_ID_Enroll_User_Record
- V100_ID_Delete_DB

Verification		
Command	**Mode**	**Parameter**
V100_ENROLL_USER	IN_PROGRESS	Describes which finger is currently being enrolled.
	ERROR	Error code related to enrolment.
	COMPLETE	None.
V100_VERIFY_USER	IN_PROGRESS	None.
	ERROR	Error code related to verification
	COMPLETE	Match/No Match
V100_FORMAT_DB	IN_PROGRESS	Number of Records deleted so far.
	ERROR	Error code related to formatting DB
	COMPLETE	Number of Records deleted.

Identification		
Command	**Mode**	**Parameter**
V100_ID_CREATE_DB	IN_PROGRESS	% of completion of DB Creation
	ERROR	Error code related to DB Creation
	COMPLETE	None.
V100_ID_SET_WORKING_GROUP	IN_PROGRESS	% complete
	ERROR	Error code related to setting working DB
	COMPLETE	None.
V100_ID_ENROLL_USER_RECORD	IN_PROGRESS	Describes which finger is currently being enrolled.
	ERROR	Error code related to enrollment
	COMPLETE	None.
V100_ID_IDENTIFY	IN_PROGRESS	% completion of identification
	ERROR	Error code related to identification
	COMPLETE	_V100_OP_ERROR code: STATUS_ID_USER_FOUND - User Found STATUS_ID_USER_NOT_FOUND - User not Found
V100_ID_IDENTIFY_378	IN_PROGRESS	% completion of identification
	ERROR	Error code related to identification
	COMPLETE	_V100_OP_ERROR code: STATUS_ID_USER_FOUND - User Found STATUS_ID_USER_NOT_FOUND - User not Found
V100_ID_VERIFY_USER_RECORD	IN_PROGRESS	% completion of verification
	ERROR	Error code related to verification
	COMPLETE	_V100_OP_ERROR code: STATUS_ID_MATCH - Match STATUS_ID_NO_MATCH - No Match
V100_ID_VERIFY_378	IN_PROGRESS	% completion of verification
	ERROR	Error code related to verification
	COMPLETE	_V100_OP_ERROR code: STATUS_ID_MATCH - Match STATUS_ID_NO_MATCH - No Match
V100_ID_VERIFY_MANY	IN_PROGRESS	% completion of verification
	ERROR	Error code related to verification
	COMPLETE	_V100_OP_ERROR code: STATUS_ID_MATCH - Match STATUS_ID_NO_MATCH - No Match
V100_ID_DELETE_DB	IN_PROGRESS	% complete
	ERROR	Error code related to deleting DB
	COMPLETE	None.

Error Codes		
Command	**Error Code**	**Description**
V100_ID_CREATE_DB	ERROR_ID_PARAMETER	Passing parameters in using the MX00_DB_INIT_STRUCT which are out of the constraints
V100_ID_CREATE_DB	ERROR_ID_DB_TOO_LARGE	DB too large to create. Number of templates calculated exceeds maximum limit.
V100_ID_CREATE_DB	ERROR_ID_DB_EXISTS	DB Already exists.
V100_ID_CREATE_DB	ERROR_ID_NOT_ENOUGH_SPACE	Not enough space on device to create the DB
V100_ID_SET_WORKING_DB	ERROR_DB_DOES_NOT_EXIST	DB passed in does not exist or the required DB files are missing
V100_ID_ENROLL_USER_RECORD	ERROR_DB_DOES_NOT_EXIST	nGroupID passed in using _MX00_ID_USER_RECORD does not exist or the required DB files are missing
V100_ID_ENROLL_USER_RECORD	ERROR_ID_DB_NOT_LOADED	nGroupID passed in using _MX00_ID_USER_RECORD is not loaded into memory. This error code can be returned with FLAG_FAIL_ENROLL_ON_DUPLCATE is set and nGroupID passed in is an identification capable DB
V100_ID_ENROLL_USER_RECORD	ERROR_ENROLLMENT_QUALIFICATION	Captured prints for enrollment didn't match.
V100_ID_ENROLL_USER_RECORD	ERROR_ID_USER_EXISTS	User-finger passed using _MX00_ID_USER_RECORD already exists in Database
V100_ID_ENROLL_USER_RECORD	ERROR_ID_DUPLICATE	Captured prints matched with other user-finger in database
V100_ID_ENROLL_USER_RECORD	ERROR_ID_DB_FULL	Database is full and cannot accept further user-fingers
V100_ID_ENROLL_USER_RECORD	ERROR_ID_USER_FINGERS_FULL	User already enrolled all fingers in the database. Number of fingers each user can enroll is specified during V100_ID_CREATE_DB call using _MX00_DB_INIT_STRUCT. Call V100_ID_DB_METRICS to get information on DB metrics.
V100_ID_ENROLL_USER_RECORD	ERROR_ID_USERS_FULL	Database is full with users and cannot accept new users. Number of users you may enroll is specified during V100_ID_CREATE_DB call using _MX00_DB_INIT_STRUCT. Call V100_ID_GET_DB_METRICS to get information on DB metrics.
V100_ID_ENROLL_USER_RECORD	ERROR_CAPTURE_TIMEOUT	Timeout occurred during capture
V100_ID_ENROLL_USER_RECORD	ERROR_CAPTURE_LATENT	Device detected latent
V100_ID_ENROLL_USER_RECORD	ERROR_CAPTURE_CANCELLED	User canceled capture using V100_CANCEL_OPERATION call
V100_ID_ENROLL_USER_RECORD	ERROR_CAPTURE_INTERNAL	Internal error occurred during capture

Error Codes		
Command	**Error Code**	**Description**
V100_ID_ENROLL_USER_RECORD	ERROR_SPOOF_DETECTED	Device detected spoof. This error code can be returned if FLAG_FAIL_ENROLL_ON_SPOOF is set and the device supports spoof.
V100_ID_IDENTIFY	ERROR_ID_DB_NOT_LOADED	No DB loaded into memory. Call V100_ID_SET_WORKING_DB to load a database into memory
V100_ID_IDENTIFY	ERROR_ID_OPERATION_NOT_SUPPORTED	Currently loaded DB is not capable of identification
V100_ID_IDENTIFY	ERROR_CAPTURE_TIMEOUT	Timeout occurred during capture
V100_ID_IDENTIFY	ERROR_CAPTURE_LATENT	Device detected latent
V100_ID_IDENTIFY	ERROR_CAPTURE_CANCELLED	User canceled capture using V100_CANCEL_OPERATION call
V100_ID_IDENTIFY	ERROR_CAPTURE_INTERNAL	Internal error occurred during capture
V100_ID_IDENTIFY	ERROR_SPOOF_DETECTED	Device detected spoof. This error code can be returned if FLAG_FAIL_IDENTIFY_ON_SPOOF is set and the device supports spoof.
V100_ID_IDENTIFY	ERROR_UID_DOES_NOT_EXIST	User ID does not exist. This error code can also appear if the loaded DB is empty.
V100_ID_IDENTIFY_378	ERROR_ID_DB_NOT_LOADED	No DB loaded into memory. Call V100_ID_SET_WORKING_DB to load a database into memory
V100_ID_IDENTIFY_378	ERROR_ID_OPERATION_NOT_SUPPORTED	Currently loaded DB is not capable of identification
V100_ID_IDENTIFY_378	ERROR_ID_PARAMETER	Template provided is invalid
V100_ID_IDENTIFY_378	ERROR_UID_DOES_NOT_EXIST	User ID does not exist. This error code can also appear if the loaded DB is empty.
V100_ID_VERIFY_USER_RECORD	ERROR_DB_DOES_NOT_EXIST	nGroupID passed in using _MX00_ID_USER_RECORD does not exist or the required DB files are missing
V100_ID_VERIFY_USER_RECORD	ERROR_ID_USER_NOT_FOUND	User not found in the Database
V100_ID_VERIFY_USER_RECORD	ERROR_CAPTURE_TIMEOUT	Timeout occurred during capture
V100_ID_VERIFY_USER_RECORD	ERROR_CAPTURE_LATENT	Device detected latent
V100_ID_VERIFY_USER_RECORD	ERROR_CAPTURE_CANCELLED	User canceled capture using V100_CANCEL_OPERATION call
V100_ID_VERIFY_USER_RECORD	ERROR_CAPTURE_INTERNAL	Internal error occurred during capture
V100_ID_VERIFY_USER_RECORD	ERROR_SPOOF_DETECTED	Device detected spoof. This error code can be returned if FLAG_FAIL_VERIFY_ON_SPOOF is set and the device supports spoof.
V100_ID_VERIFY_378	ERROR_DB_DOES_NOT_EXIST	nGroupID passed in using _MX00_ID_USER_RECORD does not exist or the required DB files are missing
V100_ID_VERIFY_378	ERROR_ID_USER_NOT_FOUND	User not found in the Database
V100_ID_VERIFY_378	ERROR_ID_PARAMETER	Template provided is invalid
V100_ID_VERIFY_MANY	ERROR_CAPTURE_TIMEOUT	Timeout occurred during capture

Error Codes		
Command	**Error Code**	**Description**
V100_ID_VERIFY_MANY	ERROR_CAPTURE_LATENT	Device detected latent
V100_ID_VERIFY_MANY	ERROR_CAPTURE_CANCELLED	User canceled capture using V100_CANCEL_OPERATION call
V100_ID_VERIFY_MANY	ERROR_CAPTURE_INTERNAL	Internal error occurred during capture
V100_ID_VERIFY_MANY	ERROR_SPOOF_DETECTED	Device detected spoof. This error code can be returned if FLAG_FAIL_VERIFY_ON_SPOOF is set and the device supports spoof.

4.15 V100_Get_Serial

Returns device serial number

`V100_ERROR_CODE V100_Get_Serial`(const V100_DEVICE_TRANSPORT_INFO *pDev, uint* pSerialNumber)

Parameters

pDev	Pointer to device handle
pSerialNumber	pointer to serial number to be returned

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Overrides the method of obtaining the serial number from the fields 'Device_Serial_Number' and 'Device_Serial_Number_Ex' in the _V100_INTERFACE_CONFIGURATION_TYPE structure. This method of obtaining the serial number is preferred, but they are equivalent.

See also

_V100_INTERFACE_CONFIGURATION_TYPE structure for information on Configuration structure.

4.16 V100_Get_Status

Returns device Status structure which contains all device error codes, conditions and health monitoring data.

`V100_ERROR_CODE V100_Get_Status` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_INTERFACE_STATUS_TYPE * *pIST*)

Parameters

pDev	Pointer to device handle
pIST	Pointer to Status structure to be returned

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _V100_INTERFACE_STATUS_TYPE structure for information on Status structure

See also

None.

4.17 V100_Get_Tag

Retrieves tag data set by V100_Set_Tag

`V100_ERROR_CODE V100_Get_Tag` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, char* *pTag*, ushort& *nTagLength*)

Parameters

pDev	Pointer to device handle
pTag	Data buffer which stores the tag, upon success
nTagLength	Size of data returned in bytes

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

If tag size is unknown, create output buffer with maximum length of 255 bytes.

See also

V100_Set_Tag

4.18 V100_Get_Template

Returns the minutia Template in the current Template buffer. This was the last template processed.

`V100_ERROR_CODE V100_Get_Template` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar * *pTemplate*, uint * *nTemplateSize*)

Parameters

pDev	Pointer to device handle
pTemplate	Pointer to the minutia template (max 2048 bytes) to be returned
nTemplateSize	pointer to the size of returned template

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Gets the latest extracted template from the device from probe template buffer. The format of the template returned corresponds to the template mode set, during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

See also

None.

4.19 V100_Match

Performs similarity match between two minutia templates and returns the score.

`V100_ERROR_CODE V100_Match` (V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar * *pProbeTemplate*, uint *nProbeTemplateSize*, uchar * *pGalleryTemplate*, uint *nGalleryTemplateSize*, uint & *MatchScore*)

Parameters

pDev	Pointer to device handle
pProbeTemplate	Pointer to minutia template1 to be matched
nProbeTemplateSize	Size of template1 to be matched
pGalleryTemplate	Pointer to minutia template2 to be matched
nGalleryTemplateSize	Size of template2 to be matched
MatchScore	Returned similarity score ranging from 0 - 100000

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

The format of the input templates must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

See also

V100_Verify, V100_Match_Ex

4.20 V100_Match_Ex

Matches one or two templates, returns match score and spoof score.

`V100_ERROR_CODE V100_Match_Ex` (V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar * *Tpl1*, uint *nSzT1*, uchar * *Tpl2*, uint *nSzT2*, int & *nMatchScore*, int & *nSpoofScore*)

Parameters

pDev	Pointer to device handle
Tpl1	Pointer to template1 to be matched
nSzT1	Size of template1 to be matched
Tpl2	Pointer to template2 to be matched
nSzT2	Size of template2 to be matched
nMatchScore	Returned similarity score ranging from 0 – 100000
nSpoofScore	Spoof not supported; returns -1.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

The format of the input templates must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

See also

V100_Match

4.21 V100_Open

Opens a connection to the device using USB or RS232 interface.

`V100_ERROR_CODE V100_Open` (V100_DEVICE_TRANSPORT_INFO * *pDev*)

Parameters

pDev	Pointer to Device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

An application must call this function to begin communication with a device.

For M31x and V31x devices it may return `GEN_ERROR_DEVICE_NOT_READY` if the call is made after a V100_Reset. If this happens wait for up to a second and try again (these devices may take up to 1.5 seconds to fully restart).

V30x sensors with firmware revisions 9538 and higher have new watchdogs that force the device to reboot itself in the event of a terminal USB communication event or an unrecoverable system event. Client applications must handle these reboot events appropriately to maintain communication with the device once it has recovered from the reboot and make this call again.

See also

None.

4.22 V100_Reset

Reboots the device.

`V100_ERROR_CODE V100_Reset` (const V100_DEVICE_TRANSPORT_INFO * *pDev*)

Parameters

pDev	Pointer to device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Issues a System Reset (Reboot) command to the device.

See also

None.

4.23 V100_Save_Last_Capture

Saves last transaction on the Micro-SD card

`V100_ERROR_CODE V100_Save_Last_Capture` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _MX00_SAVE_CAPTURE *Record*)

Parameters

pDev	Pointer to device handle
Record	Record to save

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

To check completion of this operation, poll using V100_Get_OP_Status.

Note: This functionality is not currently supported on Lumidigm sensors.

See Also

None

4.24 V100_Set_Cmd

Sets Command Structure.

`V100_ERROR_CODE V100_Set_Cmd` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_INTERFACE_COMMAND_TYPE * *pCmd*)

Parameters

pDev	Pointer to device handle
pCmd	Pointer to Command structure

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Sets current settings of USER controllable features. See _V100_INTERFACE_COMMAND_TYPE for information on Command Structure.

See also

None.

4.25 V100_Set_Composite_Image

Sets the current composite image buffer.

`V100_ERROR_CODE V100_Set_Composite_Image` (const V100_DEVICE_TRANSPORT_INFO * pDev, uchar * pImage, uint nImageSize)

Parameters

pDev	Pointer to device handle
pImage	Pointer to the composite image to write to composite image buffer
nImageSize	Size of composite image

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

The dimensions and image format of the image set **must** be the same as the image size returned by V100_Get_Config, and apply as follows:

Width	Composite_Image_Size_X
Height	Composite_Image_Size_Y
Format	8-BPP monochrome

Note: This functionality is not currently supported on M30x, M31x and V31x sensors, or V30x sensors with firmware greater than 9538.

See also

V100_Set_ Image

4.26 V100_Set_GPIO

Sets GPIO mask

`VCOM_CORE_EXPORT V100_ERROR_CODE V100_Set_GPIO`
(const V100_DEVICE_TRANSPORT_INFO * *pDev*, uchar *mask*)

Parameters

pDev	Pointer to device handle
mask	GPIO mask to set

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Note: This functionality is not currently supported on V-Series V3xx sensors.

See also

V100_Get_GPIO

4.27 V100_Set_LED

Sets user feedback LED's on/off.

`V100_ERROR_CODE V100_Set_LED` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_LED_CONTROL *Control*)

Parameters

pDev	Pointer to device handle
Control	LED to control

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Permits manual on/off control of the user feedback LEDs. See _V100_LED_CONTROL for information on LED controls.

Note: This functionality is not currently supported on M-Series (M30x, M31x) sensors.

See also

None.

4.28 V100_Set_Option

Sets system options.

`V100_ERROR_CODE V100_Set_Option` (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _V100_OPTION_TYPE *OptType*, uchar * *pData*, uint *nDataSize*)

Parameters

pDev	Pointer to device handle
OptType	Type of Option that will be set
pData	Pointer to Option data
nDataSize	Size of Option data

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _V100_OPTION_TYPE for information on types of options. Supported options include OPTION_PD_LEVEL, OPTION_SET_TEMPLATE_MODE, OPTION_SET_WSQ_COMPRESSION_LEVEL, OPTION_SET_LATENT_DETECTION_MODE and OPTION_SET_FORCE_FINGER_LIFT_MODE (M30x sensors and V30x sensors with firmware greater than 9538).

Template types supported using OPTION_SET_TEMPLATE_MODE are as follows:

TEMPLATE_ANSI_378	ANSI/INCITS 378-2004
	ANSI 378+ V30x-40-S (FW 29428 or higher)
TEMPLATE_ISO_NORMAL	ISO/IEC 19794-2:2005
	ISO 19794:2011 (FW 29428 or higher)

The format of the input/output template for the following commands corresponds to template mode set using V100_Set_Option call. The default template mode will be TEMPLATE_ANSI_378.

CMD_GET_TEMPLATE

CMD_MATCH

CMD_MATCH_EX

CMD_SET_TEMPLATE

CMD_TRUNCATE_378

CMD_ID_GET_USER_RECORD

CMD_ID_IDENTIFY_378

CMD_ID_SET_USER_RECORD

CMD_ID_VERIFY_378

CMD_ID_VERIFY_MANY

Using OPTION_SET_WSQ_COMPRESSION_RATIO, the WSQ compression level can be set from 2 to the value set in MAX_WSQ_COMPRESSION_RATIO. Compression level should be provided in unsigned integer value. The default compression ratio is 11:1.

User can turn on/off latent detection using OPTION_SET_LATENT_DETECTION_MODE as follows:

LATENT_DETECTION_OFF Turns off latent detection

LATENT_DETECTION_ON Turns on latent detection

For M30x sensors and V30x sensors with firmware greater than 9538, the user can turn on/off the force finger lift mode for enrollment using V100_Set_Option call with OPTION_SET_FORCE_FINGER_LIFT_MODE as follows:

FORCE_FINGER_LIFT_MODE_ON Turns on force finger lift mode. User required to lift finger after each capture during enrollment.

FORCE_FINGER_LIFT_MODE_OFF Turns off force finger lift mode. This is the default behavior for M30x sensors and for V30x sensors with firmware > 9538.

See also

None.

4.29 V100_Set_Tag

Sets a tag which persists on the file system, to be retrieved using V100_Get_Tag

```
V100_ERROR_CODE V100_Set_Tag (const V100_DEVICE_TRANSPORT_INFO * pDev, char*
pTag, ushort nTagLength)
```

Parameters

pDev	Pointer to device handle
pTag	Opaque data array of tag to store
nTagLength	Size of tag in bytes

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Maximum length of tag is 255 bytes

See also

V100_Get_Tag

4.30 V100_Set_Template

Downloads the minutia Template to current template buffer.

```
V100_ERROR_CODE V100_Set_Template (const V100_DEVICE_TRANSPORT_INFO * pDev,
uchar * pTemplate, uint nTemplateSize)
```

Parameters

pDev	Pointer to device handle
pTemplate	Pointer to the minutia template to write to template buffer
nTemplateSize	Size of the template to write to template buffer

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Sets the gallery template buffer. The format of the input template must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

See also

None.

4.31 V100_Truncate_378

Truncates a template obtained from a call to V100_Get_Template.

```
V100_ERROR_CODE V100_Truncate_378 (const V100_DEVICE_TRANSPORT_INFO * pDev,
uint nMaxTemplateSize, const uchar* pInTemplate, uint nTplSize, uchar*
pOutTemplate, uint& nActualSize)
```

Parameters

pDev	Pointer to device handle
nMaxTemplateSize	Size to be truncated to
pInTemplate	Template to be truncated
nTplSize	Size of template to be truncated
pOutTemplate	Output template buffer (must be allocated)
nActualSize	Size of output template

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

The format of the input template must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

See also

V100_Get_Template

4.32 V100_Update_Firmware

Allows user to update firmware on the unit

```
V100_ERROR_CODE V100_Update_Firmware (const V100_DEVICE_TRANSPORT_INFO * pDev,
uchar * pFirmwareStream, uint nFWStreamSize)
```

Parameters

pDev	Pointer to device handle
pFirmwareStream	Firmware stream
nFWStreamSize	Size of Firmware stream in bytes

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

To check completion of this operation, poll using V100_Get_OP_Status.

See Also

None

4.33 V100_Verify

Captures image, extracts template, verifies against input minutia template, returns match score and spoof score (if supported).

```
V100_ERROR_CODE V100_Verify (V100_DEVICE_TRANSPORT_INFO * pDev, uchar *
pTemplate, uint nTemplateSize, int & Spoof, uint & MatchScore)
```

Parameters

pDev	Pointer to device handle
pTemplate	Pointer to the minutia template to be verified
nTemplateSize	Size of the template to be verified
Spoof	Returned spoof score. -1 if not supported
MatchScore	Returned similarity score ranging from 0 - 100000

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Arms device, captures an image upon presence detection, extracts template, verifies against the input template, then returns a match score and spoof score, if supported.

See also

- V100_Arm_Trigger
- V100_Get_Acq_Status
- V100_Get_Template
- V100_Match
- V100_Match_Ex

4.34 V100_Verify_378

Captures image, extracts minutia, verifies against input template.

```
V100_ERROR_CODE V100_Verify_378 (const V100_DEVICE_TRANSPORT_INFO * pDev,
uchar * pTemplate, uint nTemplateSize)
```

Parameters

pDev	Pointer to device handle
pTemplate	Pointer to the minutia template to write to template buffer
nTemplateSize	Size of the template to write to template buffer

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

V100_Verify_378 is a macro operation, thus one must poll for completion using V100_Get_OP_Status. After V100_Get_OP_Status returns completion, V100_ID_Get_Result must be used to get the result of the verification.

See also

- V100_Get_OP_Status
- V100_Get_Template
- V100_ID_Get_Result

4.35 V100_Vid_Stream

Sets Video stream mode.

```
V100_ERROR_CODE V100_Vid_Stream (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_VID_STREAM_MODE mode)
```

Parameters

pDev	Pointer to device handle
mode	Turn on/off Video stream mode

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Set the Video Stream Mode on or off. When video stream is on, one can retrieve the current raw image using V100_Get_Image. It is highly recommended that the only vCOM calls made to the system between modes V100_Vid_Stream(on) and V100_Vid_Stream(off) is V100_Get_Image. Recommended for high-bandwidth transport modes only. The dimensions and image format of the image returned using IMAGE_VID_STREAM as image type with V100_Get_Image can be found by issuing a call to V100_Get_Config and applying as follows:

Width	Native_Image_Size_X
Height	Native_Image_Size_Y
Format	Bayer-pattern BGGR

See also

None.

4.36 V100_WaitForFingerClear

Arms device trigger to finger detection mode and waits for finger clear.

```
V100_ERROR_CODE V100_WaitForFingerClear (const V100_DEVICE_TRANSPORT_INFO *
pDev)
```

Parameters

pDev	Pointer to device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

This call can be used with V100_ID_Identify, V100_ID_Verify_User_Record, V100_Capture, V100_Verify to make sure there is no valid finger placement on device platen.

See also

- V100_Arm_Trigger
- V100_Get_Acq_Status
- V100_ID_Identify
- V100_ID_Verify_User_Record
- V100_Capture
- V100_Verify

5 1:1 Verification-Specific API Commands

This section describes the 1:1-specific API commands provided by the vCOM protocol **for V30x sensors with 9538 FW only (V30x-20)**. All of the functions listed in this section are currently not supported on V30x-30, V30x-40, M30x, M31x, or V31x sensors. Verification commands are no longer supported starting with release v6.00 firmware on V30x sensors.

5.1 V100_Add_User

Adds a user to a device, user having been previously retrieved using V100_Get_User_By_Index, or V100_Get_User calls.

```
V100_ERROR_CODE V100_Add_User (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_USER_RECORD UserRecord, char* pRecordData)
```

Parameters

pDev	Pointer to device handle
UserRecord	User Record to add
pRecordData	Opaque user record data to add

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

The user is committed to the database by the time the call returns.

See also

None.

5.2 V100_Delete_User

Deletes a user record from the database

```
V100_ERROR_CODE V100_Delete_User (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_USER_RECORD UserRecord)
```

Parameters

pDev	Pointer to device handle
UserRecord	User Record describing which record to delete

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None

See also

V100_Get_OP_Status, V100_DB_Metrics

5.3 V100_Enroll_User

Requests that a user described in UserRecord is enrolled.

```
V100_ERROR_CODE V100_Enroll_User (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_USER_RECORD UserRecord)
```

Parameters

pDev	Pointer to device handle
UserRecord	User Record describing Enrollment ID and Metadata

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Client must poll for completion using V100_Get_OP_Status

See also

- V100_Get_OP_Status
- V100_Verify_User

5.4 V100_Format_DB

Erases all of the User Records in the database. Does not erase Tags generated by calls to V100_Set_Tag.

```
V100_ERROR_CODE V100_Format_DB (const V100_DEVICE_TRANSPORT_INFO * pDev)
```

Parameters

pDev	Pointer to device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Client must poll for completion using V100_Get_OP_Status. nParameter in V100_Get_OP_Status determines percentage completion of formatting command. May take up to 30 seconds to complete on high-capacity units.

See also

V100_Get_OP_Status

5.5 V100_Get_DB_Metrics

Retrieves database metrics structure, which includes size of the database, maximum number of records in database, last spoof score, and last match score.

```
V100_ERROR_CODE V100_Get_DB_Metrics (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_DB_METRICS* dbMetrics)
```

Parameters

pDev	Pointer to device handle
dbMetrics	DB Metrics structure

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Useful for database management, and for determining index range for command V100_Get_User_By_Index

See also

V100_Get_User_By_Index

5.6 V100_Get_User

Gets a user from the database, based on UserRecord identifier.

```
V100_ERROR_CODE V100_Get_User (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_USER_RECORD* pUserRecord, char* pRecordData)
```

Parameters

pDev	Pointer to device handle
pUserRecord	Pointer to a User Record structure, IN/OUT parameter
pRecordData	Pointer to opaque record data structure, IN/OUT parameter

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

UID member must be populated. All other members in UserRecord are ignored.

See also

None.

5.7 V100_Get_User_By_Index

Gets an existing user record from the database, based on zero-based index.

```
V100_ERROR_CODE V100_Get_User_By_Index (const V100_DEVICE_TRANSPORT_INFO *
pDev, uint nIndex, _V100_USER_RECORD* pUserRecord, char* pRecordData)
```

Parameters

pDev	Pointer to device handle
nIndex	Index of the user to get
pUserRecord	Contains User Record upon success
pRecordData	Contains User Record data upon success

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None.

See also

V100_Get_User_By_Index

5.8 V100_Get_Verification_Rules

Gets rules for enrollment and verification

```
V100_ERROR_CODE V100_Get_Verification_Rules (const V100_DEVICE_TRANSPORT_INFO
* pDev, _V100_VERIFICATION_RULES& verRules)
```

Parameters

pDev	Pointer to device handle
verRules	Verification Rules structure

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None.

See also

None.

5.9 V100_Set_Verification_Rules

Sets rules for enrollment and verification

```
V100_ERROR_CODE V100_Set_Verification_Rules (const V100_DEVICE_TRANSPORT_INFO
* pDev, _V100_VERIFICATION_RULES verRules)
```

Parameters

pDev	Pointer to device handle
verRules	Verification Rules structure

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None.

See also

- V100_Spoof_Get_Template
- V100_Spoof_Get_Template_From_Image

5.10 V100_Verify_User

Verifies against user described in User Record.

```
V100_ERROR_CODE V100_Verify_User (const V100_DEVICE_TRANSPORT_INFO * pDev,
_V100_USER_RECORD UserRecord)
```

Parameters

pDev	Pointer to device handle
UserRecord	User Record describing Enrollment ID and Metadata

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

Client must poll for completion using V100_Get_OP_Status

See also

- V100_Get_OP_Status
- V100_Enroll

6 1:N Identification/Verification Specific API

This section describes the 1:N-specific API commands provided by the vCOM protocol for M30x, M31x,V31x sensors, and V30x sensors with 24087 firmware (V30x-30).

With Lumidigm software build v6.00, a new Minex III certified Extractor and Matcher is utilized by the Lumidigm V31x sensor.

1:N Identification database groups created and enrolled with v5.30.53 and earlier are supported in 6.00 in a legacy mode, but are not compatible with the new Minex III certified Extractor and Matcher which is considered a proprietary 1:N database group. To migrate to the Minex III certified proprietary 1:N database, a new proprietary database group will need to be created and users will need to be re-enrolled.

For regular 1:1 operation such V100_Match (VCOM), the Minex III certified Extractor and Matcher will be utilized.

Please review the *Lumidigm Biometric Performance - V-Series* document for details and matching thresholds.

6.1 V100_ID_Create_DB

Creates a new database

```
V100_ERROR_CODE V100_ID_Create_DB (const V100_DEVICE_TRANSPORT_INFO * pDev,
_MX00_DB_INIT_STRUCT dbInitStructIn)
```

Parameters

pDev	Pointer to device handle
dbInitStructIn	DB Initialization structure

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _MX00_DB_INIT_STRUCT for information on DB initialization structure. Identification capable databases require enrolling 3 instances per finger for maximum identification performance. One must use V100_Get_OP_Status to poll for operation completion and/or error codes that occurred during the database creation process.

See also

V100_ID_Delete_DB, V100_Get_OP_Status

6.2 V100_ID_Delete_DB

Deletes specified database/group

```
V100_ERROR_CODE V100_ID_Delete_DB (const V100_DEVICE_TRANSPORT_INFO * pDev,
uint nDbNo)
```

Parameters

pDev	Pointer to device handle
nDbNo	Database/Group no. to delete.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful

Remarks

To check completion of this operation, poll using V100_Get_OP_Status. If nDbNo specified is currently set using V100_ID_Set_Working_DB, the database is unloaded from memory.

See also

- V100_Get_OP_Status
- V100_ID_Create_DB

6.3 V100_ID_Delete_User_Record

Deletes a User or User Record from a database

Because one can enroll multiple fingers per user, V100_ID_Delete_User_Record allows the caller to choose whether to remove a user completely or whether to remove user-finger record.

```
VCOM_CORE_EXPORT V100_ERROR_CODE V100_ID_Delete_User_Record (const
V100_DEVICE_TRANSPORT_INFO * pDev, const _MX00_ID_USER_RECORD rec, bool
nDeleteAllFingers)
```

Parameters

pDev	Pointer to device handle
rec	User record to delete
nDeleteAllFingers	If true deletes all user records corresponding to the user field in _MX00_ID_USER_RECORD. If false deletes user-finger specified in _MX00_ID_USER_RECORD.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

If the database specified in _MX00_ID_USER_RECORD is currently loaded then it is unloaded from memory during this call.

If desired client has to call V100_ID_Set_Working_DB to load the database into memory after this call.

See also

6.4 V100_ID_Enroll_User_Record

Begins enrollment process.

Users are enrolled in the system using the "rules" set using the V100_ID_Set_Parameters call.

```
V100_ERROR_CODE V100_ID_Enroll_User_Record (const V100_DEVICE_TRANSPORT_INFO *
pDev, _MX00_ID_USER_RECORD UserRecord)
```

Parameters

pDev	Pointer to device handle
UserRecord	User record to enroll

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

You must call V100_ID_ENROLL_USER_RECORD for each User-Finger you wish to enroll. However, this command does automatically enroll multiple instances of said User-Finger.

To check progress/completion of this operation, poll using V100_Get_OP_Status.

If the database specified in the nGroupID member of the _MX00_ID_USER_RECORD has multiple instances per user finger specified, V100_ID_Enroll_User_Record will attempt to enroll that many instances of the user's finger. If the nFlags set during V100_ID_Set_Parameters call contains the FLAG_ENROLLMENT_QUALIFICATION flag, each instance of the captured print is checked against each other captured print in order to ensure enrollment quality. The FLAG_ENROLLMENT_QUALIFICATION is a requirement when enrolling into any database which is capable of identification. This is a requirement to ensure good identification performance.

If the FLAG_FAIL_ENROLL_ON_DUPLICATE flag is set, for identification capable databases check for duplicate is performed. If the captured prints match with a user-finger in the database, V100_Get_OP_Status returns "ERROR_ID_DUPLICATE" _V100_OP_ERROR code in nParameter member of _V100_OP_STATUS structure. You may call V100_ID_Get_Result to get the status/user information.

If the FLAG_FAIL_ENROLL_ON_SPOOF flag is set, a check for spoof is performed for each instance of the captured print.

For M31x and V31x sensors, a check for finger clear is done after each capture to make sure there is no valid finger placement on device platen before capturing next print. User is required to lift the finger after each capture during enrollment. For M30x sensors, and V30x sensors with firmware greater than 9538, user can set the force finger lift for enrollment using V100_Set_Option call with OPTION_SET_FORCE_FINGER_LIFT_MODE as the _V100_OPTION_TYPE and FORCE_FINGER_LIFT_MODE_ON as the option data.

See also

- V100_Get_OP_Status
- V100_ID_Set_Parameters
- V100_ID_Set_User_Record

6.5 V100_ID_Get_DB_Metrics

Gets metrics and parameters of a database

```
V100_ERROR_CODE V100_ID_Get_DB_Metrics (const V100_DEVICE_TRANSPORT_INFO *
pDev, _MX00_DB_METRICS * dbMetrics, bool bGetCurrent)
```

Parameters

pDev	Pointer to device handle
dbMetrics	Input: If bGetCurrent is false then nGroupID member is used to retrieve the metrics. Output: Contains Database metrics upon success
bGetCurrent	If true retrieves the metrics of the currently loaded database. If false the nGroupID member of the _MX00_DB_METRICS structure that is sent in is used in order to determine which database metrics to retrieve.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

If bGetCurrent is true and no group is currently loaded, an error will be returned.

See also

V100_ID_Set_Working_DB

6.6 V100_ID_Get_Parameters

Retrieves the current ID parameters

```
V100_ERROR_CODE V100_ID_Get_Parameters (const V100_DEVICE_TRANSPORT_INFO *
pDev, _MX00_ID_PARAMETERS & param)
```

Parameters

pDev	Pointer to device handle
param	Contains Identification parameters upon success

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _MX00_ID_PARAMETERS for information on ID parameters.

See also

V100_ID_Set_Parameters

6.7 V100_ID_Get_Result

Retrieves the result of last successful identify executed. This includes completion of the commands V100_ID_Identify_378, V100_ID_Identify, V100_ID_Verify_User_378 and V100_ID_Verify_User_Record as polled by V100_Get_OP_Status

```
V100_ERROR_CODE V100_ID_Get_Result (const V100_DEVICE_TRANSPORT_INFO * pDev,
_MX00_ID_RESULT & res)
```

Parameters

pDev	Pointer to device handle
res	The results structure

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None.

See also

- V100_Get_OP_Status
- V100_ID_Identify_378
- V100_ID_Identify
- V100_ID_Verify_User_378
- V100_ID_Verify_User_Record

6.8 V100_ID_Get_System_Metrics

Provides information on database groups currently found on the system

```
V100_ERROR_CODE V100_ID_Get_System_Metrics (const V100_DEVICE_TRANSPORT_INFO *
pDev, _MX00_DB_METRICS ** dbMetrics, uint & nNumDBsFound)
```

Parameters

pDev	Pointer to device handle
dbMetrics	Database metrics records found
nNumDBsFound	Contains number of groups, thus number of _MX00_DB_METRICS to follow

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

The nCurEnrolledUserFingers and nCurEnrolledUsers members of the _MX00_DB_METRICS structure returned by this call should be ignored since these are not valid values. To get this information call V100_ID_Get_DB_Metrics.

See also

V100_ID_Get_DB_Metrics

6.9 V100_ID_Get_User_Record

Retrieves the User record from a database

```
V100_ERROR_CODE V100_ID_Get_User_Record (const V100_DEVICE_TRANSPORT_INFO *
pDev, short nIndex, _MX00_ID_USER_RECORD & rec, _MX00_TEMPLATE_INSTANCE
instanceRecords[])
```

Parameters

pDev	Pointer to device handle
nIndex	Index of User Record to retrieve. If nIndex = -1, the structure the rec field contains is used to indicate which user-finger to retrieve. If nIndex = n, this will retrieve the user-finger stored at 0-based index n, where n is between 0 to nCurEnrolledUserFingers field retrieved during a call to V100_ID_Get_DB_Metrics
rec	Input: nInstances member is ignored. If nIndex is positive then szUserID, nFinger is ignored. Output: Contains User Record header structure upon success. nInstances member of this structure defines how many _MX00_TEMPLATE_INSTANCE structures follow this structure
instanceRecords	Contains nInstances of _MX00_TEMPLATE_INSTANCE structures which hold the templates upon success. The format of the templates returned corresponds to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

This command is useful for retrieving user-finger entries in the database sequentially, when user-finger information is unknown.

See also

V100_ID_Get_DB_Metrics

6.10 V100_ID_Get_User_Record_Header

Retrieves the User record header from working database

```
V100_ERROR_CODE V100_ID_Get_User_Record_Header (const
V100_DEVICE_TRANSPORT_INFO * pDev, short nIndex, _MX00_ID_USER_RECORD & rec)
```

Parameters

pDev	Pointer to device handle
nIndex	Index of User Record to retrieve. This will retrieve the user-finger stored at 0-based index n, where n is between 0 to nCurEnrolledUserFingers field retrieved during a call to V100_ID_Get_DB_Metrics
rec	Contains User Record header structure upon success

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

This command is useful for retrieving user-finger entries in the database sequentially, when user-finger information is unknown.

See also

V100_ID_Get_DB_Metrics

6.11 V100_ID_Identify

Identifies a user from capture. Captures a fingerprint and identifies the fingerprint. A database must be loaded using V100_ID_Set_Working_DB in order for this command to succeed.

```
V100_ERROR_CODE V100_ID_Identify (const V100_DEVICE_TRANSPORT_INFO * pDev)
```

Parameters

pDev	Pointer to device handle

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

V100_ID_Identify is a macro operation, thus one must poll for completion using V100_Get_OP_Status. After V100_Get_OP_Status returns completion, V100_ID_Get_Result must be used to get the result of the identification.

V100_ID_Identify works in two stages. Stage 1 is capturing the fingerprint. Polling V100_GET_OP_STATUS during this stage will allow you to get status on the capture, and will also return error conditions in case of time-outs, latent prints detected etc. If the FALG_FAIL_IDENTIFY_ON_SPOOF flag is set, a check for spoof is performed.

After a successful capture, V100_ID_Identify will begin the identification phase. After Identification is complete, you may call V100_ID_Get_Result to get the status information related to the last identification performed. If the operation completes successfully V100_Get_OP_Status returns following _V100_OP_ERROR codes in nParameter member of _V100_OP_STATUS structure

STATUS_ID_USER_FOUND	User found
STATUS_ID_USER_NOT_FOUND	User not found

See also

- V100_Get_OP_Status
- V100_ID_Get_Result
- V100_ID_Identify_378

6.12 V100_ID_Identify_378

Identifies a user from a template. A database must be loaded using V100_ID_Set_Working_DB in order for this command to succeed.

```
V100_ERROR_CODE V100_ID_Identify_378 (const V100_DEVICE_TRANSPORT_INFO * pDev,
const uchar * pTemplate, uint nSizeTemplate)
```

Parameters

pDev	Pointer to device handle
pTemplate	Template to be identified. The format of the template must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.
nSizeTemplate	Size of the template

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

V100_ID_Identify_378 is a macro operation, thus one must poll for completion using V100_Get_OP_Status. After V100_Get_OP_Status returns completion, V100_ID_Get_Result must be used to get the result of the identification. If the operation completes successfully V100_Get_OP_Status returns following _V100_OP_ERROR codes in nParameter member of _V100_OP_STATUS structure

STATUS_ID_USER_FOUND	User found
STATUS_ID_USER_NOT_FOUND	User not found

See also

- V100_Get_OP_Status
- V100_ID_Get_Result
- V100_ID_Identify

6.13 V100_ID_Release_System_Metrics

Release the memory created with call to V100_ID_Get_System_Metrics

```
V100_ERROR_CODE V100_ID_Release_System_Metrics (const
V100_DEVICE_TRANSPORT_INFO * pDev, _MX00_DB_METRICS * pDBMetrics)
```

Parameters

pDev	Pointer to device handle
pDBMetrics	Database metrics records returned from V100_ID_Get_System_Metrics

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

None.

See also

V100_ID_Get_System_Metrics

6.14 V100_ID_Set_Parameters

Sets the parameters which the identification engine uses to operate

```
V100_ERROR_CODE V100_ID_Set_Parameters (const V100_DEVICE_TRANSPORT_INFO *
pDev, _MX00_ID_PARAMETERS param)
```

Parameters

pDev	Pointer to device handle
param	ID parameters to set

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

See _MX00_ID_PARAMETERS for information on ID parameters.

See also

None.

6.15 V100_ID_Set_User_Record

Adds an existing User Record to database.

```
V100_ERROR_CODE V100_ID_Set_User_Record (const V100_DEVICE_TRANSPORT_INFO *
pDev, const _MX00_ID_USER_RECORD rec, const _MX00_TEMPLATE_INSTANCE
instanceRecords[])
```

Parameters

pDev	Pointer to device handle
rec	User Record header structure to add. The nInstances member of this structure defines how many _MX00_TEMPLATE_INSTANCE structures follow this structure.
instanceRecords	nInstances of _MX00_TEMPLATE_INSTANCE structures which hold the templates. The format of the input templates must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

This command is useful for adding existing user records enrolled from another sensor. If the database specified in nGroupID member of _MX00_ID_USER_RECORD has multiple instances per user-finger and the nFlags set during the V100_ID_Set_Parameters call contains the FLAG_ENROLLMENT_QUALIFICATION flag, each template instance is checked against each other template in order to ensure enrollment quality. The FLAG_ENROLLMENT_QUALIFICATION is a requirement while adding user record into any database which is capable of identification. This is a requirement to ensure good identification performance. **After adding all the user records using this command you must call V100_ID_Set_Working_DB in order to commit all the records to the device.**

See also

- V100_ID_Enroll_User_Record
- V100_ID_Set_Parameters
- V100_ID_Set_Working_DB

6.16 V100_ID_Set_Working_DB

Sets the working database. Database must have been created using V100_ID_Create_DB for this command to succeed.

```
V100_ERROR_CODE V100_ID_Set_Working_DB (const V100_DEVICE_TRANSPORT_INFO *
pDev, uint nDB)
```

Parameters

pDev	Pointer to device handle
nDB	Working Database/Group number to set

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

If database is empty this command will allow to enroll users into database using V100_ID_Enroll_User_Record or will allow to add existing records using V100_ID_Set_User_Record. If this database is populated, it will load in to active memory. Once database is in active memory, V100_ID_Identify, or V100_ID_Identify_378, may be used to identify. To check progress/completion of this operation, poll using V100_Get_OP_Status.

See also

- V100_ID_Create_DB
- V100_ID_Enroll_User_Record
- V100_ID_Set_User_Record
- V100_ID_Identify
- V100_ID_Identify_378
- V100_Get_OP_Status

6.17 V100_ID_Verify_378

Verifies a user from a template. This command should be used to verify an existing template against a user record present in the database.

V100_ERROR_CODE `V100`_ID_Verify_378 (const V100_DEVICE_TRANSPORT_INFO * *pDev*, _MX00_ID_USER_RECORD *UserRecord*, unsigned char * *pTemplate*, uint *nSizeTemplate,* short *nConsiderFinger*)

Parameters

pDev	Pointer to device handle
UserRecord	User record to verify. All members of the _MX00_ID_USER_RECORD structure should be populated except for nInstances member, which is ignored
pTemplate	Template to be verified. The format of the template must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.
nSizeTemplate	Size of the template
nConsiderFinger	If 0, verifies the user specified in _MX00_ID_USER_RECORD ignoring the nFinger member. All fingers enrolled for the user are considered. If 1, verifies the user-finger specified in _MX00_ID_USER_RECORD.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

V100_ID_Verify_378 is a macro operation, thus one must poll for completion using V100_Get_OP_Status. After V100_Get_OP_Status returns completion, V100_ID_Get_Result must be used to get the result of the verification. If the operation completes successfully V100_Get_OP_Status returns following _V100_OP_ERROR codes in nParameter member of _V100_OP_STATUS structure

STATUS_ID_MATCH	Match
STATUS_ID_NO_MATCH	No match

We recommend you to set working DB to verify a user. The command latency is much lower if working DB is set.

See also

- V100_Get_OP_Status
- V100_ID_Get_Result
- V100_ID_Verify_User_Record

6.18 V100_ID_Verify_Many

Captures image, extracts minutia template, verifies against input templates.

```
V100_ERROR_CODE V100_ID_Verify_Many (const V100_DEVICE_TRANSPORT_INFO * pDev,
uchar ** pTemplates, uint * pSizeTemplates, uint nNumTemplates)
```

Parameters

pDev	Pointer to device handle
pTemplates	Pointer to array of input templates. The format of the template must correspond to the template mode set during V100_Set_Option call using OPTION_SET_TEMPLATE_MODE.
pSizeTemplates	Array of input templates size.
nNumTemplates	Number of input templates to verify. Max 30 templates are allowed.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

V100_ID_Verify_Many is a macro operation, thus one must poll for completion using V100_Get_OP_Status. After V100_Get_OP_Status returns completion, V100_ID_Get_Result must be used to get the result of the verification.

V100_ID_Verify_Many works in two stages. Stage 1 is capturing the fingerprint. Polling V100_GET_OP_STATUS during this stage will allow you to get status on the capture, and will also return error conditions in case of time-outs, latent prints detected etc. If the FALG_FAIL_VERIFY_ON_SPOOF flag is set, a check for spoof is performed.

After a successful capture, V100_ID_Verify_Many will begin the verification phase. If the operation completes successfully V100_Get_OP_Status returns following _V100_OP_ERROR codes in nParameter member of _V100_OP_STATUS structure

STATUS_ID_MATCH	Match
STATUS_ID_NO_MATCH	No match

After the operation completes successfully V100_ID_Get_Result returns (0-based) index of the input template which has highest match score in szUserID member of _MX00_ID_RESULT structure.

See also

- V100_Get_OP_Status
- V100_ID_Get_Result
- V100_ID_Verify_378
- V100_ID_Verify_User_Record
- V100_Verify

6.19 V100_ID_Verify_User_Record

Verifies a user from capture. Captures a fingerprint and verifies the fingerprint. This command should be used to verify against a user record present in the database.

```
V100_ERROR_CODE V100_ID_Verify_User_Record (const V100_DEVICE_TRANSPORT_INFO *
pDev, _MX00_ID_USER_RECORD UserRecord, short nConsiderFinger)
```

Parameters

pDev	Pointer to device handle
UserRecord	User record to verify. All members of the _MX00_ID_USER_RECORD structure should be populated except for nInstances member, which is ignored
nConsiderFinger	If 0, verifies the user specified in _MX00_ID_USER_RECORD ignoring the nFinger member. All fingers enrolled for the user are considered. If 1, verifies the user-finger specified in _MX00_ID_USER_RECORD.

Returns

V100_ERROR_CODE	Refer to error code documentation in the vCOM Command Reference document for detailed description of possible return values.
GEN_OK	Indicates operation was successful.

Remarks

V100_ID_Verify_User_Record is a macro operation, thus one must poll for completion using V100_Get_OP_Status. After V100_Get_OP_Status returns completion, V100_ID_Get_Result must be used to get the result of the verification. We recommend you to set working DB to verify a user. The command latency is much lower if working DB is set.

V100_ID_Verify_User_Record works in two stages. Stage 1 is capturing the fingerprint. Polling V100_GET_OP_STATUS during this stage will allow you to get status on the capture, and will also return error conditions in case of time-outs, latent prints detected etc. If the FALG_FAIL_VERIFY_ON_SPOOF flag is set, a check for spoof is performed.

After a successful capture, V100_ID_Verify_User_Record will begin the verification phase. After verification is complete, you may call V100_ID_Get_Result to get the status information related to the last verification performed. If the operation completes successfully V100_Get_OP_Status returns following _V100_OP_ERROR codes in nParameter member of _V100_OP_STATUS structure

STATUS_ID_MATCH	Match
STATUS_ID_NO_MATCH	No match

See also

- V100_Get_OP_Status
- V100_ID_Get_Result
- V100_ID_Verify_378

Appendix A: vCOM Supported Functionality

Use the tables below to determine what VCOM functionality is supported for a Lumidigm sensor. The label on the sensor contains the SKU (i.e. V300-02-S-USB01), cross-reference the first part of the SKU value with the items referenced in the table to determine what vCOM functions can be used with the sensor. **Note:** Some functionality can depend on specific firmware versions.

IMPORTANT: If a V30x sensor is upgraded to release v6.00 firmware 29428, the unit should be considered a V30x-40-S for the purposes of this table. The V30X-40 firmware is only available for an additional fee. Contact your HID Sales representative for details.

A.1 Supported vCOM Commands for Lumidigm Sensors (by SKU)

vCOM Command	Lumidigm sensors with SKUs starting with:								
	Vxxx-0x-x	Vxxx-10-X	Vxxx-20-X	V30x-30-S	V30x-40-S	M30x-00	M31x-00	V31x-00	
V100_Arm_Trigger		✓	✓	✓	✓	✓	✓	✓	
V100_Arm_Trigger (CANCEL_VERIFICATION)		✓	✓	✓	✓			✓	
V100_Arm_Trigger (TRIGGER_FINGER_DETECT)				✓	✓	✓*	✓	✓	
V100_Cancel_Operation				✓	✓	✓	✓	✓	
V100_Capture			✓	✓	✓	✓	✓	✓	
V100_Close		✓	✓	✓	✓	✓	✓	✓	
V100_Config_Comport		✓	✓	✓	✓	✓			
V100_Get_Acq_Status		✓	✓	✓	✓	✓	✓	✓	
V100_Get_Cmd		✓	✓	✓	✓	✓	✓	✓	
V100_Get_Composite_Image		✓	✓	✓	✓	✓	✓	✓	
V100_Get_Config		✓	✓	✓	✓	✓	✓	✓	
V100_Get_GPIO						✓			
V100_Get_FIR_Image				✓	✓	✓	✓	✓	
V100_Get_Image		✓	✓	✓	✓	✓	✓	✓	
V100_Get_Image (IMAGE_WSQ)				✓	✓	✓*	✓	✓	
V100_Get_Num_USB_Devices		✓	✓	✓	✓	✓	✓	✓	
V100_Get_OP_Status		✓	✓	✓	✓	✓	✓	✓	
V100_Get_Serial				✓	✓		✓	✓	
V100_Get_Status		✓	✓	✓	✓	✓			
V100_Get_Tag				✓	✓	✓	✓	✓	

vCOM Command	Lumidigm sensors with SKUs starting with:								
	Vxxx-0x-x	Vxxx-10-X	Vxxx-20-X	V30x-30-S	V30x-40-S	M30x-00	M31x-00	V31x-00	
V100_Get_Template			✓	✓	✓	✓	✓	✓	
V100_Match			✓	✓	✓	✓	✓	✓	
V100_Match_Ex			✓	✓	✓	✓	✓	✓	
V100_Open		✓	✓	✓	✓	✓	✓	✓	
V100_Reset		✓	✓	✓	✓	✓	✓	✓	
V100_Save_Last_Capture									
V100_Set_Cmd		✓	✓	✓	✓	✓	✓	✓	
V100_Set_Composite_Image		✓	✓						
V100_Set_GPIO						✓	✓		
V100_Set_LED		✓	✓	✓	✓			✓	
V100_Set_Option		✓	✓	✓	✓	✓	✓	✓	
V100_Set_Option (OPTION_SET_TEMPLATE_MODE)				✓	✓	✓	✓	✓	
V100_Set_Option (OPTION_SET_WSQ_COMPRESSION_LEVEL)				✓	✓	✓	✓	✓	
V100_Set_Option (OPTION_SET_FORCE_FINGER_LIFT_MODE)				✓	✓	✓*			
V100_Set_Tag				✓	✓	✓	✓	✓	
V100_Set_Template		✓	✓	✓	✓	✓	✓	✓	
V100_Truncate_378		✓	✓	✓	✓	✓	✓	✓	
V100_Update_Firmware				✓	✓	✓			
V100_Verify		✓	✓	✓	✓	✓	✓	✓	
V100_Vid_Stream		✓	✓	✓	✓	✓	✓	✓	
V100_WaitForFingerClear				✓	✓	✓*	✓	✓	
V100_Add_User		✓	✓						
V100_Delete_User		✓	✓						
V100_Enroll_User		✓	✓						
V100_Format_DB		✓	✓						
V100_Get_DB_Metrics		✓	✓	✓	✓	✓	✓	✓	
V100_Get_User		✓	✓						
V100_Get_User_By_Index		✓	✓						
V100_Get_Verification_Rules		✓	✓						

vCOM Command	Lumidigm sensors with SKUs starting with:								
	Vxxx-0x-x	Vxxx-10-X	Vxxx-20-X	V30x-30-S	V30x-40-S	M30x-00	M31x-00	V31x-00	
V100_Set_Verification_Rules		✓	✓						
V100_Verify_User		✓	✓						
V100_ID_Create_DB				✓		✓	✓	✓	
V100_ID_Delete_DB				✓		✓	✓	✓	
V100_ID_Delete_User_Record				√		√	√	√	
V100_ID_Enroll_User_Record				√		√	√	√	
V100_ID_Get_DB_Metrics				√		√	√	√	
V100_ID_Get_Parameters				√		√	√	√	
V100_ID_Get_Result				✓		✓	✓	✓	
V100_ID_Get_System_Metrics				✓		✓	✓	✓	
V100_ID_Get_User_Record				✓		✓	✓	✓	
V100_ID_Get_User_Record_Header				✓		✓	✓	✓	
V100_ID_Identify				✓		✓	✓	✓	
V100_ID_Identify_378				✓		✓	✓	✓	
V100_ID_Release_System_Metrics				✓		✓	✓	✓	
V100_ID_Set_Parameters				✓		✓	✓	✓	
V100_ID_Set_User_Record				✓		✓	✓	✓	
V100_ID_Set_Working_DB				✓		✓	✓	✓	
V100_ID_Verify_378				✓		✓	✓	✓	
V100_ID_Verify_Many				✓		✓	✓	✓	
V100_ID_Verify_User_Record				✓		✓	✓	✓	
V100_Verify_378				✓	✓	✓*			

*Supported for M30x sensors with firmware revision 18647 or later

A.2 Supported Features for Lumidigm Sensors (by SKU)

Feature	Lumidigm sensors with SKUs starting with:											
	127xxx-02	Vxxx-01-N	Vxxx-01-S	Vxxx-02-N	Vxxx-02-S	Vxxx-10-N	Vxxx-10-S	Vxxx-20-N	Vxxx-20-S Vxxx-30-S Vxxx-40-S	M30x-00	M31x-00	V31x-00
Processing	T	T	T	T	T	E	E	E	E	E	T	T
Policy	60	0	10	50	60	1	11	51	61	51	N/A	N/A
Image Out	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Template Out	No	No	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Spoof Out	Yes	No	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes
Crop Level	None	None	None	None	None	272x400	272x400	272x400	272x400	N/A	N/A	N/A

*Processing: T = Tethered; E = Embedded