MorphoAccess® SIGMA Family & MorphoWave® Compact

Host System Interface

Osny, France

Warning

Revision History

The table below contains the history of changes made to the present document.

Version	Date	Description
1	Oct. 2016	Update Script chapter USB Script supported on MASigma Previous version : 2015_2000012335_v2
2	March 2017	Replace 5G Series with SIGMA Family Add SIGMA Extreme Series in the MorphoAccess® SIGMA Family Change the contact information
3	December 2017	Update company name (IDEMIA)
4	July 2018	Add MorphoWave® Compact support

Table of Content

WARNING 2

REVISION HISTORY 3

SECTION 1 : INTRODUCTION 8

MorphoAccess® SIGMA Family and MorphoWave Compact Terminals 9

Scope of the Document 10

Host System Interface Overview 13

MorphoAccess® SIGMA Family Modes 14

MorphoAccess® 500 or J Series legacy mode 14
L-1 Bioscrypt 4G Series legacy mode 14
MorphoAccess® SIGMA Family native mode 15

Remote Management Protocol 16

TCP Remote Management 16
SSL Securing 18
RS422 Remote Management 21

SECTION 2 : DISTANT COMMANDS 22

About Thrift Commands 23

Serial Communication 24

Serial Packet Structure 24
Data chunking 26

SECTION 3 : LEGACY MPH DISTANT COMMANDS 27

Standard Interface Presentation 28

Presentation 28
ILV command 28

Initialize Function Description 30

Ping 30
Default Init 32
Get version 35
Get version (new prototype) 37
Reboot 41
Set Configuration 42
Get Configuration 44
Set Registry Key 46
Get Registry Key 49
Get Registry File 51

Distant session 53
Set Contactless Keys (or Crypto Write) [Contactless Terminal only] 55
Send CBM Command [IDEMIA Only, no support is given on this function] 58

***Wi-Fi™ Functions Description* 60**
Scan Wi-Fi™ networks 60
Configure Wi-Fi™ network 62
Get Wi-Fi™ network status 64

***Security Functions Description* 66**
Manage Crypto Object Functions Description 66
Load Security Content 72
Device Discovery Response 74

***Biometric Functions Description* 76**
Verify 76
Enroll 80
Identify 88

***Database Functions Description* 93**
Create Database 93
Erase Base 98
Erase All Base 100
Add Base Record 102
Remove Base Record 107
Get Data 109
Get Public Fields 111
Update Public Data 114
Get Base Config 117
Get All Bases Config 120
Destroy DB 122
Handle Several Records [DISTANT SESSION] 124

***Access Control Log Functions Description* 128**
Overview 128
Base Structure 129
Line Structure 130
Authentication Records format 131
Identification Records format 136
Get log status 140
Erase log 143
Get log 144

***Contactless Card Operations* 147**
Get Card Information 150
Contactless Read 153
Contactless Write 155
Contactless Erase 157

***Access Control Functions Description* 159**
Overview 159

Access Authorization 159
Display Screen [MA500 Series only, PROXY Mode Only] 161
Read from keyboard [Terminal with DISPLAY and keyboard] 163
Data Send 165

***Time Mask Functions Description* 167**
How to use time mask feature? 167
Set Time Mask 168
Get Time Mask 170

***Biometric Data Description* 172**
Biometric coder selection 172
Biometric Algorithm Parameter [0x38] 173
ISO and ANSI template formats 174
ISO_PK_PARAM [0x40] 175
ISO_PK_DATA_ANSI 378 [0x41] 175
ISO_PK_DATA_MINEX_A [0x6F] 176
ISO_PK_DATA_ISO_FMR [0x6E] 176
ISO_PK_DATA_ISO_FMC_NS ID = 0X6D 177
ISO_PK_DATA_ISO_FMC_CS ID = 0X6C 177
Security Level 178
Biometric Presence Detection Mode 179

***Database Description* 181**
User Index 181
User ID 182
PKBase 183
Base Configuration 184

***Additional Data Field Description* 186**
Field Content 186
Public field 187
Private Field 188
Additional User Data 189
No Check on Template 190
Time Mask 191

***Wiegand Data Description* 192**
Wiegand data 192
Wiegand raw data [IDEMIA Only, no support is given on this function] 193

***Clock & Data Protocol Data Description* 194**
Clock & Data 194
Clock & Data raw data [IDEMIA Only, no support is given on this function] 195

***Contactless Data Description* 196**
Address Biometric Data 196
MIFARE® Classic Contactless Profile 197
DESFire® Contactless Profile 201
DESFire® File description 208

Security Data Description **211**
SOM identifier 211
Terminal Version Data Description **212**
Terminal Identifier 212
Biometric sensor Identifier 213
Contactless Features 215
Man Machine Interfaces 216
Application Information 217
Terminal Description 218
Peripherals 219
Configuration Data Description **221**
Date and Time Configuration 221
Wi-Fi™ Data Description **222**
Wi-Fi™ Network Information 222
SSID 223
BSSID 223
Encryption key 224
Miscellaneous Data Description **225**
Matching Score 225
Latent Detection 226
Constant Values **227**
Identifier – status 227
Error codes table 228
Status codes table 232
Configuration File **233**
File Types 233
MIFARE® Contactless Card Mapping **239**
Setting up Matching Threshold **240**
Database Size Limits **241**
USB Scripts **243**
SECTION 4 : L1 BIOSCRYPT LEGACY MODE DISTANT COMMANDS 244
References **245**

Annex 1 : Bibliography **246**
Annex 2 : Support **249**

Section 1 : Introduction

MorphoAccess® SIGMA Family and MorphoWave Compact Terminals

Congratulations for choosing a MorphoAccess® SIGMA Family or/and MorphoWave® Compact Automatic Fingerprint or Hand Recognition Terminals.

MorphoAccess® SIGMA Family (also referred as MA5G) or/and MorphoWave® Compact terminals provides an innovative and effective solution for access control applications using Fingerprint or Hand Verification or/and Identification.

Among a range of alternative biometric technologies, the use of finger imaging has significant advantages: each finger constitutes an unalterable physical signature, developed before birth and preserved until death. Unlike DNA, a finger image is unique for each individual - even identical twins.

The MorphoAccess® SIGMA Family and MorphoWave® Compact terminals integrate Morpho image processing and feature matching algorithms. This technology is based on lessons learned during 25 years of experience in the field of biometric identification and the creation of literally millions of individual fingerprint identification records.

Designed for physical access control applications, MorphoAccess® SIGMA Family and MorphoWave® Compact terminals feature a compact, attractive design, coupled with high reliability and security. These 5th generation terminals are both robust and easy to use for a variety of applications, including office, headquarters and administrative building security, as well as protection of external access points.

To ensure the most effective use of your MorphoAccess® SIGMA Family & MorphoWave® Compact terminal using host system, you should read this User Guide completely

Scope of the Document

This document is intended to help developers on how to operate using host system interface for controlling MorphoAccess® SIGMA Family & MorphoWave® Compact terminal.

MorphoAccess® SIGMA Family & MorphoWave® Compact terminal supports various modes, viz. MA5G mode (Sigma Family), Legacy L1 (Bioscrypt terminals) and Legacy Morpho (500 Series terminals).

This guide is divided into three major sections, and covers each distant command used for controlling the terminal in all three modes.

Terminal Series	Terminal Name	Biometrics	Contactless smartcard reader			Hardware Fake Finger Detection	Outdoor
			iCLASS®	MIFARE® DESFire® NFC®	Prox®		
MorphoAccess® SIGMA Family & MorphoWave® Compact	MorphoAccess® Sigma	✓					
	MorphoAccess® Sigma WR	✓					✓
	MorphoAccess® Sigma iCLASS®	✓	✓				
	MorphoAccess® Sigma iCLASS® WR	✓	✓				✓
	MorphoAccess® Sigma Multi	✓		✓			
	MorphoAccess® Sigma Multi WR	✓		✓			✓
	MorphoAccess® Sigma Prox	✓			✓		
	MorphoAccess® Sigma Prox WR	✓			✓		✓
	MorphoAccess® Sigma Lite / Lite+	✓					
	MorphoAccess® Sigma Lite / Lite+ iCLASS®	✓	✓				

	MorphoAccess® Sigma Lite / Lite+ Multi	✓		✓			
	MorphoAccess® Sigma Lite / Lite+ Prox	✓			✓		
	MorphoAccess® Sigma Extreme iCLASS®	✓	✓				✓
	MorphoAccess® Sigma Extreme FFD iCLASS®	✓	✓			✓	✓
	MorphoAccess® Sigma Extreme Multi	✓		✓			✓
	MorphoAccess® Sigma Extreme FFD Multi	✓		✓		✓	✓
	MorphoAccess® Sigma Extreme Prox	✓			✓		✓
	MorphoAccess® Sigma Extreme FFD Prox	✓			✓	✓	✓

NOTE: WR indicates terminal is Weather Resistant.

FFD indicates terminal is with a prism with Fake Finger Detection

Terminal Series	Terminal Name	Biometrics	Contactless smartcard reader			Outdoor
			iCLASS®	MIFARE® DESFire®	Prox®	
MorphoWave® Compact	MorphoWave® Compact MD	Y		Y		
	MorphoWave® Compact MDPI	Y	Y	Y	Y	

NOTE: *MorphoWave® Compact includes following variants :*
- MorphoWave® Compact MIFARE® DESFire® (MD)
- MorphoWave® Compact MIFARE® DESFire® Prox® iCLASS® (MDPI)

Host System Interface Overview

The MorphoAccess® SIGMA Family & MorphoWave® Compact terminals provides remote management facilities, using which it is possible to change the terminal settings through Ethernet using a TCP-IP connection, or through serial link using a RS422 connection.

The MorphoAccess® SIGMA Family & MorphoWave® Compact terminal acts as a server and provides a unique socket, in case of TCP-IP connection.

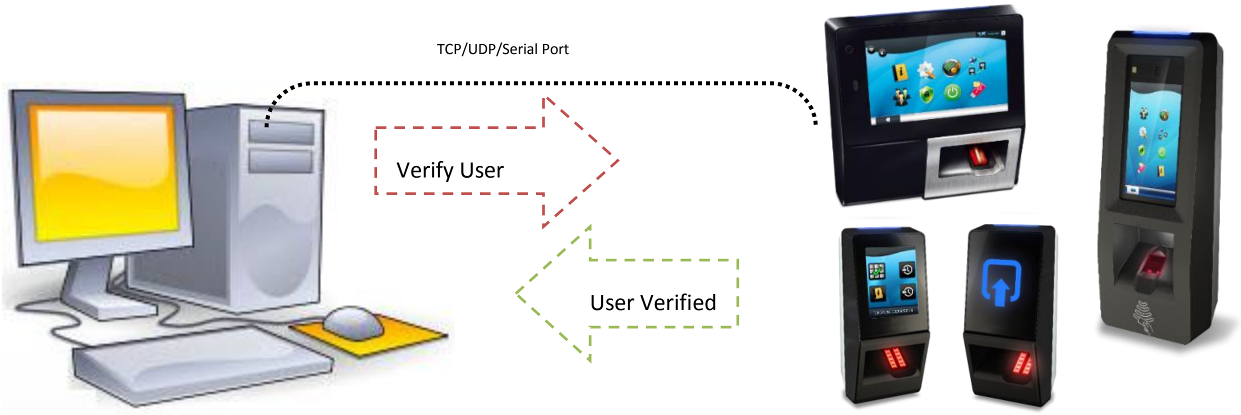


Figure 1: Operating terminal through Host system

The following operations are allowed:

- Changing system settings such as control type, control timeout…,
- Sending biometric requests,
- Firmware upgrade

It is also possible to receive information from the terminal. After a biometric control, the terminal sends information in TCP, UDP, SSL, RS485 or RS422. In those cases the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal is a client and the PC is the server.

This document details the syntax of each command.

Recommendation: If network intensive or database intensive operations are performed on terminal from host system using distant commands, then the terminal response time would be affected until this background operation is completed. Hence it is advisable to do such network or database intensive operation when terminal is in idle state.

MorphoAccess® SIGMA Family & MorphoWave® Compact Modes

MorphoAccess® SIGMA Family & MorphoWave® Compact (also referred as MA5G) terminals are standalone biometric access control terminals which offer advance features for access rights check of the users. MorphoAccess® SIGMA Family & MorphoWave® Compact terminals are equipped with a facility to emulate (partially) :

- MorphoAccess® SIGMA Family & MorphoWave® Compact terminal previous generation or
- L-1 Bioscrypt 4G Series terminals (except for MorphoAccess® SIGMA Extreme Series – this mode is not supported)

When MorphoAccess® SIGMA Family & MorphoWave® Compact is set in any of the legacy modes, it supports the database structures and configurations of the selected legacy terminal. When the terminal is booted for the first time, user can select any of the legacy mode described in the next sections.

NOTE: Legacy mode is not applicable for MorphoWave® Compact

MorphoAccess® 500 or J Series legacy mode

MorphoAccess® SIGMA Family & MorphoWave® Compact terminal can be operated in MA500 mode (also referred as Legacy Morpho). In this mode, the terminal will support configurations and operations of MA500 terminals. Terminal can authenticate users enrolled in the MA500 terminals, using biometric check as well as contactless card. New users can also be enrolled in MA500 mode.

Access Path

System Menu > First Boot Assistant > Protocol Settings > Legacy Morpho

L-1 Bioscrypt 4G Series legacy mode

MorphoAccess® SIGMA and SIGMA Lite Series terminal can be operated in Bioscrypt 4G mode (also referred as Legacy L1). In this mode, the terminal will support limited operations and configurations that are done using SecureAdmin application. The terminal in L1 mode is able to authenticate the users enrolled on 4G terminals and contactless cards. However **user enrolment** in legacy L1 mode on MorphoAccess® SIGMA and SIGMA Lite Series terminal is possible only when Secure Admin station is equipped with a MorphoSmart™ MSO biometric sensor.

In case, users are enrolled in MA5G mode, the user data cannot be exported when terminal is in L1 mode. There are certain other limitations, when MorphoAccess® SIGMA and SIGMA Lite Series terminal is run in L1 Legacy mode. For details about these limitations, refer to document MorphoAccess® 5G Series - ApplicationNotes **– L1 Legacy Mode Limitations.**

Access Path

System Menu > First Boot Assistant > Protocol Settings > Legacy L1

MorphoAccess® SIGMA Family & MorphoWave® Compact native mode

MorphoAccess® SIGMA Family & MorphoWave® Compact terminal is by default in native mode; this native mode is designed by MA5G, which means MorphoAccess® 5th generation. This mode supports new features and a remote management application called Webserver.

Access Path

System Menu > First Boot Assistant > Protocol Settings > MA5G

NOTE: When terminal mode is switched from MA5G to any of the legacy modes, the entire configuration and all databases are erased, except communication links and language settings.

The terminal is rebooted on mode change and factory settings are applicable.

Remote Management Protocol

TCP Remote Management

It is possible to administrate a MorphoAccess® SIGMA Family & MorphoWave® Compact terminal from a distant computer. In this case the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal works as a standard TCP server waiting for requests coming from a remote client.

A standard exchange follows always the following schema:

Typical PC – MorphoAccess® SIGMA Family & MorphoWave® Compact terminal TCP-IP communication		
PC (client)		MorphoAccess® SIGMA Family (server)
Connect on port 11010	→ ←	Accept on port 11010
Send distant command	→	Receive distant command
		Execute distant command
Receive distant command response	←	Send distant command response
…		…
Close socket	→	Close socket

Note: Maximum 10 input connections can be opened at the same time on MorphoAccess® SIGMA Family & MorphoWave® Compact terminal.

SSL Securing

Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) are cryptographic protocol designed to provide communication security over Ethernet or Wi-Fi™ channels.

These protocols are used to protect the communication between the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal and a distant system, such as a central access controller or a terminal configuration station.

The cryptographic protocols supported by the terminal are listed below:

- SSLv3 only (Only for compatibility purposes)
- SSLv23 (Accept TLS 1.0, 1.1, 1.2)
- TLS 1.0 only
- TLS 1.1 only
- TLS 1.2 only

The terminal supports the algorithms listed below for communication security:

- AES128-SHA Openssl ciphersuite
- AES256-SHA Openssl ciphersuite
- AES128-SHA256 Openssl ciphersuite
- AES256-SHA256 Openssl ciphersuite
- AES128-GCM-SHA256 Openssl ciphersuite
- ECDHE-ECDSA-AES256-SHA Openssl ciphersuite
- ECDHE-ECDSA-AES128-GCM-SHA256 Openssl ciphersuite
- ECDHE-ECDSA-AES128-SHA256 Openssl ciphersuite
- ECDHE-ECDSA-AES128-SHA Openssl ciphersuite

Note: The communication security is automatically configured during negotiation between the client and the server. The client specifies the security level requested, and the server accepts or proposes a lower level. The client accepts it or cancels its request. The final configuration corresponds to the higher security level common with the client and the server.

Compatibility of cipher algorithms with SSL protocol versions

Cipher Algorithm List	**Protocol Version**				
	sslv23	**sslv3**	**tlsv1**	**tlsv1.1**	**tlsv1.2**
AES128-SHA	Y	Y	Y	Y	Y
AES256-SHA	Y	Y	Y	Y	Y
AES128-SHA256	N	N	N	N	Y
AES256-SHA256	N	N	N	N	Y
AES128-GCM-SHA256	N	N	N	N	Y
ECDHE-ECDSA-AES256-SHA:ECDH-ECDSA-AES256-SHA	Y	Y	Y	Y	Y
ECDHE-ECDSA-AES128-GCM-SHA256:ECDH-ECDSA-AES128-GCM-SHA256	N	N	N	N	Y
ECDHE-ECDSA-AES128-SHA256:ECDH-ECDSA-AES128-SHA256	N	N	N	N	Y
ECDHE-ECDSA-AES128-SHA:ECDH-ECDSA-AES128-SHA	Y	Y	Y	Y	Y

Note: Cipher algorithm that ends with ‘SHA256’ supports only SSL protocol version tls1.2.

SSL Protocol Versions support for remote communication

	Client side (from PC application)					
		sslv23	sslv3	tlsv1	tlsv11	tlsv12
On Terminal	sslv23	Y	N	Y	Y	Y
	sslv3	Y	Y	N	N	N
	tlsv1	Y	N	Y	N	N
	tlsv11	Y	N	N	Y	N
	tlsv12	N	N	N	N	Y

The above table describes the protocol versions supported by client side application, when communication is started by terminal using specific protocol. For e.g. If terminal starts communication using sslv23 protocol, then client side application will be able to communicate using all the protocol versions. While if communication is initiated using sslv3 protocol, then client application will only support sslv23 and sslv3 protocol versions for communication.

References

- See "*SSL Configuration*" section under Security Menu, in MorphoAccess® SIGMA Family & MorphoWave® Compact **Admin User Guide**, for details on how to configure SSL communication port
- See SSL Solution for MorphoAccess® documentation for details on SSL securing

RS422 Remote Management

It is also possible to administrate the MorphoAccess® SIGMA Family & MorphoWave® Compact MorphoAccess® SIGMA Family & MorphoWave® Compact from a remote computer using a RS422 connection. In this case the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal works as a server waiting for requests coming from a remote client.

The terminal listens to the serial port. If there is data, then the terminal executes the requests, else it listens again to the serial port until there is something or the user asks to close the connection.

A standard exchange follows always the following schema:

Typical PC – MorphoAccess® SIGMA Family & MorphoWave® Compact RS422/RS485 communication		
 PC (client)		 MorphoAccess®SIGMA Family (server)
		Listen on serial port in a continuous loop
Send distant command	→	Receive distant command
		Execute distant command
Receive distant command response	←	Send distant command response
		Listen on serial port in a continuous loop
		…

Section 2 : Distant Commands

About Thrift Commands

MorphoAccess® SIGMA Family & MorphoWave® Compact terminal is by default in MA5G mode that is a native mode. MorphoAccess® SIGMA Family & MorphoWave® Compact terminals can communicate using Apache Thrift framework. Know more about Apache Thrift framework on http://thrift.apache.org/.

Distant commands used by host system for interacting with terminal in MA5G mode are available in document, "**MA5G_distant_commands**".

Serial Communication

The serial channel communication implements a particular protocol and applies it over the data to send or receive. The data transfer is accompanied by the appropriate data encapsulation and data chunking. Out of the two steps of the serial protocol, the packet encapsulation is applicable only to communication in MA5G mode.

Note that the serial channel communication protocol defined here is not dependent on the terminal mode, but on the mode in which the communication is occurring. For example, even though the terminal may be in MA5G mode, the communication with the controller or remote messaging involving serial channel shall occur in ILV protocol (as used in terminal's MA5xx mode).

Serial Packet Structure

The packet structure encapsulation shall only be applicable to for the communication in the MA5G mode.

- **MA5G communication mode:** Following is the packet structure that the serial channel shall implement:

Byte#	1	2	3 – 4	5 - 8	9 – 10	11 - <11 + Data size>
Field	0x4D	0x41	<Net ID>	<Checksum>	<Data size>	<Data>
Length	1	1	2	4	2	<Data size> (maximum 1024 bytes)
Endianness	-	-	Big	Big	Big	-

Checksum calculation

The checksum shall be calculated on the data part only.

Steps:

1. Take a sum all the bytes in the data.
2. Take 2's complement of the sum.

Example:

If the data is 0x0102, then checksum is ((~(0x01 + 0x02) + 1) => 0xFFFFFFFD (For C/C++, it's decimal (-3)).

- L1 legacy / MA5xx legacy communication modes

  There is no packet structure defined, i.e., no data encapsulation done, at serial channel level for communication in these modes.

Data chunking

The maximum chunk size of the data that shall be sent on the serial channel is predefined as 1024 bytes (1 KB).

Before actually sending the data, based on the data size the chunk is produced. If the data size is more than the predefined maximum chunk size, then more than one chunk will be produced and packet encapsulation is applied to each chunk, if applicable. Then each chunk is sent to the physical channel sequentially (not writing all chunks in one go).

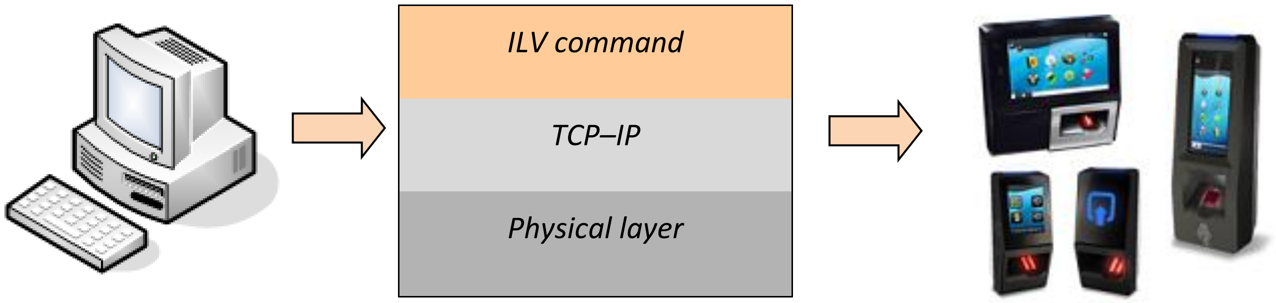
Similarly, after receiving the data, the process should be reversed, i.e., the encapsulated packet structure will be removed from each chunk received sequentially, and the data from each chunk will be combined into a single final data, which is then passed on to the above layers.

Section 3 : Legacy MPH Distant Commands

Standard Interface Presentation

Presentation

One connection can be established on the port 11010 (the port can be changed). The MorphoAccess® is able to receive one ILV command. The request is analyzed, the command is executed, the answer is sent on the established socket and the socket can be closed.



For TCP remote management, at the end of the transaction the client closes the socket. If the connection is not closed, the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal can automatically close the connection after a given period of inactivity.

ILV command

ILV commands have the following format:

ILV command		
Identifier	**Length**	**Value**
1 byte	2 bytes	Length bytes
Command identifier	*Command data length* *(little endian format)*	*Command data*

The application data has three fields:

- **Identifier** called **I**; this is the identifier of the command,

- **Length** called **L**; this is the length of the Value field in byte,
- **Value** called **V**; this is the data or parameters.

This data structure is variable. The Value field can contain optional ILV formatted data. Its length is variable.

If command length is greater than 65534 bytes (0xFFFE), length is coded with 4 bytes and ILV command has the following format:

ILV command			
Identifier	**Length**	**Length**	**Value**
1 byte	2 bytes	4 bytes	Length bytes
Command identifier	*0xFFFF* *(escape code)*	*Command data length* *(little endian format)*	*Command data*

Note about incomplete frame

Command execution is triggered when the complete frame has been received.

If an incomplete frame is sent, the terminal will wait during 1 minute the resting part of the frame.

Note about ILV reply

If the *Request Status* field in the reply is different from ILV_OK, the reply is necessary limited to this status and does not contain additional data.

Initialize Function Description

Ping

This function allows checking that the terminal responds correctly. As an option, a buffer with binary values can be included in the request; the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal returns the same value in the reply.

Command

Request

I	CMD_PING [0x08]	1 byte
L	0x0000 + *L*	2 bytes
V	Optional Value	L bytes

Optional Value
Binary data to be returned by the terminal without modification (for transmission test).

Reply

I	CMD_PING [0x08]	1 byte
L	0x0001 + *L*	2 bytes
V	Request status	1 byte
	Optional Value	L bytes

Request status	
ILV_OK [0x00]	The execution succeeded.

Optional Value
Binary data received within the command.

Note

To only check if a terminal is present on the network, the ICMP ping request is enough.

Compatibility Note

This command is fully compatible with first generation terminals.

Default Init

Description

This command resets MorphoAccess® SIGMA Family & MorphoWave® Compact terminal configuration parameters to default value (factory configuration).

❗ **All MIFARE®, DESFire® crypto keys and Ethernet settings can be reset to default value!**

- the network configuration is also reset when the *Erase Settings* mask is present and includes the NET_MASK flag,
- the MIFARE® authenticate keys are also reset when the *Erase Settings* mask is present and includes the MIFARE_KEYS_MASK flag,
- the application passwords are reset when the *Erase Settings* mask is present and includes the PASSWORDS_MASK flag.
- the DESFire® authenticate keys are also reset when the *Erase Settings* mask is present and includes the DESFIRE_KEYS_MASK flag
- In case of conflict between *Keep Settings* and *Erase Settings* mask, then the priority is given to the *Erase Settings* mask.

When the network configuration is reset to default value, a restart (using Reboot ILV command) is mandatory to apply the changes.

Command

Request

I	CMD_DEFAULT_INIT [0x0A]	1 byte
L	0x0000, 0x0001 or 0x0002	2 bytes
V	Keep Settings mask (optional, but mandatory when Erase Setting mask has to be present)	1 byte
	Erase Settings mask (optional)	1 byte

Keep settings

This optional parameter prevents some settings from being reset to default value. A mask of bits defines parameters to keep to current value:

- NET_MASK [0x01] [0b0000 0001] keeps Ethernet parameters,
- MIFARE_KEYS_MASK [0x02] [0b0000 0010] keeps MIFARE® keys.

It is recommended to use the Erase settings mask (see below) instead of the Keep setting mask. But for compatibility with previous software releases, the Keep settings mask remains available.

When the Erase setting parameter must be present, the keep setting parameter becomes mandatory.

Erase settings

Resets configuration keys (irrespective of keep settings mask and erase_settings mask). Configuration keys include keys starting by **app**, **adm**, **bio**, **gui**, **events**, **remotemsg**, **exe**, **mmi**, **enr**, and **log**.

This optional parameter defines the settings to be reset to default value, using a bit mask:

- NET_MASK [0x01] [0b0000 0001] resets Ethernet parameters,
- MIFARE_KEYS_MASK [0x02] [0b0000 0010] resets MIFARE® keys,
- PASSWORDS_MASK [0x80] [0b1000 0000] resets passwords,
- DESFIRE_KEYS_MASK [0x40] [0b0100 0000] resets DESFire® keys.

The MorphoAccess® SIGMA Family & MorphoWave® Compact terminal allows several formats for the Default_Init ILV command, and depending on the action to perform, the length field's value can be different, and the command parameters can be mandatory or not:

- MorphoAccess® 100 Series, J Series, VP Series and 500 Series format: L = 0x0002:
    - the *Keep Settings* parameter is mandatory (1st value byte),
    - the *Erase Settings* parameter is mandatory (2nd value byte).

Reply

I	CMD_DEFAULT_INIT [0x0A]	1 byte
L	0x0001	2 bytes
V	Request status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.

Compatibility Note

- The Keep Settings parameter replaces Keep Net Settings described in MorphoAccess® 200 and 300 Series Host Interface.
- MIFARE_KEYS_MASK [0x02] and PASSWORD_MASK [0x80] are new options of MA100/J/VP/500 Series.

Note about biometric database

The database is not erased. Use *Erase All Base* to erase the base.

Note about HID™ crypto keys

HID™ crypto keys (MorphoAccess® 110 terminal) are not reset. To reset HID™ crypto keys use directly "*Set Contactless Keys (or Crypto Write) [Contactless Terminal only]*" command.

Get version

Description

This function returns the serial number of the terminal, the revision of the software application and the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal type.

Command

Request

I	CMD_GET_VERSION [0x03]	1 byte
L	0	2 bytes

Reply

I	CMD_GET_VERSION [0x03]	1 byte
L	17	2 bytes
V	Request Status	1 byte
	Serial Number	9 bytes
	Software Revision	4 bytes
	Extended Memory	1 byte
	MorphoAccess® Type	2 bytes

Request status	
ILV_OK [0x00]	The execution succeeded.

The following parameters are returned only if *Request Status* is ILV_OK.

Serial Number

Terminal serial number in ASCII “123456789”.

Software Revision

Fixed value: 4 bytes: ‘S’, ‘A’, ‘0x0B’, ‘0’

Extended Memory

Memory dedicated to local database in Mbytes. This value is fixed ‘0’ in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal

MorphoAccess® Type (2 bytes in big endian)

100	MorphoAccess® 100 terminal (and MorphoAccess® J-Bio and VP-Bio)
110	MorphoAccess® 110 terminal
120	MorphoAccess® 120 or 120 D terminal (and MorphoAccess® J-Dual and VP-Dual)
500	MorphoAccess® 500 or 500+ terminal
520	MorphoAccess® 520, 520+ D, OMA 520 or OMA 520 D terminal
521	MorphoAccess® 521, 521+ D, OMA 521 or OMA 521 D terminal

Compatibility Note

- The *[MorphoAccess® Type] Type* and *Extended Memory* will logically differ from the first generation terminals.
- The database size depends on the license installed into the terminal.
- It is highly advised to use the new prototype of the GET VERSION request to know in details the terminal abilities.

Get version (new prototype)

Description

Calling "GET VERSION" with parameters will return a detailed description of terminal characteristics.

Command

Request

I	CMD_GET_VERSION [0x03]	1 byte
L	1 (+ 1) + … + (+ 1)	2 bytes
V	ID_TERMINAL_IDENTIFIER [0x01] (Optional)	1 byte
	ID_SENSOR_IDENTIFIER[0x02] (Optional)	1 byte
	ID_CLSS_IDENTIFIER [0x03] (Optional)	1 byte
	ID_MMI_IDENTIFIER [0x04] (Optional)	1 byte
	ID_APPLICATION_INFORMATION [0x05] (Optional)	1 byte
	ID_TERMINAL_DESCRIPTION [0x06] (Optional)	1 byte
	ID_BIN_PERIPH [0x07] (Optional)	1 byte

ID_TERMINAL_IDENTIFIER

If present, the *Terminal ILV* will be present in the reply.

ID_SENSOR_IDENTIFIER

If present, the *Biometric Sensor ILV* will be present in the reply.

ID_CLSS_IDENTIFIER

If present, the *Contactless ILV* will be present in the reply.

ID_MMI_IDENTIFIER

If present, the *Man Machine Interface ILV* will be present in the reply.

ID_APPLICATION_INFORMATION

If present, the *Application Information ILV* will be present in the reply.

ID_TERMINAL_DESCRIPTION

If present, the *Terminal Description ILV* will be present in the reply.

ID_BIN_PERIPH

If present, the *Peripherals ILV* will be present in the reply.

Reply

I	CMD_GET_VERSION [0x03]	1 byte
L	1 (+ A) (+ B) (+ C) (+ D) (+ E)	2 bytes
V	Request Status	1 byte
	Terminal ILV (Optional)	A bytes
	Biometric Sensor ILV (Optional)	B bytes
	Contactless ILV (Optional)	C bytes
	Man Machine Interface ILV (Optional)	D bytes
	Application Information ILV (Optional)	E bytes

	Terminal Description ILV (Optional)	F bytes
	Peripherals ILV (Optional)	G bytes

Request status

ILV_OK [0x00]	The execution succeeded.

Terminal ILV

See the Terminal Identifier structure for details.

Biometric Sensor ILV

See the Biometric sensor Identifier structure for details.

Contact-Less ILV

See the Contactless features structure for details.

Man Machine Interface ILV

See the Man Machine Interfaces structure for details.

Application Information ILV

See the Application Information structure for details.

Terminal Description ILV

See the Terminal Description structure for details.

Peripherals ILV

See the Peripherals structure for details.

Reboot

Description

This function allows rebooting the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal (software reset). This command can be required in order to apply some parameters.

Command

Request

I	CMD_REBOOT [0x04]	1 byte
L	0	2 bytes

Reply

I	CMD_REBOOT [0x04]	1 byte
L	1	2 bytes
V	Request status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.

Note

The reboot process can last for approx. 45 seconds. The terminal reboots without confirming that the reply has been sent.

Set Configuration

Description

This function modifies the system's configuration. For now, only the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal Real Time Clock configuration is customizable.

Command

Request

I	CMD_SET_CONFIGURATION [0x01]	1 byte
L	1 (+ Ld)	2 bytes
V	MorphoAccess® Mode	1 byte
	Date And Time Configuration	Ld bytes

MorphoAccess® Mode

Set to 0. This parameter is RFU.

For now, only Date and Time Configuration command is available.

Date And Time Configuration

This sub ILV allows setting terminal date.

See the *Date and Time Configuration* structure for details.

This is a 16 bytes ILV long (1 for I field, 2 for L field, and 13 for V field).

Reply

I	CMD_SET_CONFIGURATION [0x01]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_ERROR [0xFF]	Execution failed.

Compatibility Note

The following options are **no more available**:

- Serial Link Configuration,
- Wiegand/Dataclock Configuration,
- Wiegand/Dataclock Options,
- Graphical User Interface Parameters,
- Language Parameters,
- Contactless Parameters,
- Wiegand Anybit Parameters,
- Failure ID Parameters,
- Time Attendance Parameters,
- Graphic Mode Parameters,
- Relay,
- Application mode,
- Network parameters.

Use the *Set Registry Key* command to set these parameters.

Get Configuration

Description

This function retrieves the system's configuration. For now, only the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal Real Time clock configuration is available

Command

Request

I	CMD_GET_CONFIGURATION [0x06]	1 byte
L	N	2 bytes
V	Parameter Identifier 1	1 byte
	Parameter Identifier 2	1 byte
	…	…
	Parameter Identifier i	1 byte
	…	…
	Parameter Identifier N	1 byte

For now, only Date and Time Configuration command is available.

Parameter Identifier i

It is used to determine which parameter will be returned:

- ID_DATE [0x13]

Only the necessary parameters can be included.

Reply

I	CMD_GET_CONFIGURATION [0x06]	1 byte
L	2 (+ Ld)	2 bytes
V	Request status	1 byte

	MorphoAccess® Mode	1 byte
	Date and Time Configuration	Ld bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

The following parameters are returned only if *Request Status* is ILV_OK.

MorphoAccess® Mode

Must be 0.

Date And Time Configuration

See the *Date and Time Configuration* structure for details.

Compatibility note

See the *Set Configuration* command corresponding remark.

Set Registry Key

Description

Use this command to change one or more parameters' value. Parameters are stored in a ".cfg" file.

For example to modify the settings of the relay (stored in **app.cfg** file), change:

"*/app/relay/enabled/1*" to "*/app/relay/enabled/0*".

Command

Request

I	CMD_SET_REGISTRY_KEY [0x0B]	1 byte
L	$L_0 + L_1 + \ldots + L_N$	2 bytes
V	Registry path and value in ASCII [0]	L_0 bytes
	Registry path and value in ASCII [1]	L_1 bytes
	…	
	Registry path and value in ASCII [N]	L_N bytes

Several parameters may be included.

Each string must be "null terminated" (\0).

Registry path and value in ASCII [i].
"/file/section/parameter/new value". **Remark**: '/' and '|' symbols are accepted. '=', '[', ']', '\r' and '\n' symbols are forbidden. **Remark**: the "file" corresponds to "file.cfg". The ".cfg" extension must be omitted.

Reply

I	CMD_SET_REGISTRY_KEY [0x0B]	1 byte

L	1 + N + 1	2 bytes
V	Request status	1 byte
	Setting result [0]	1 byte
	Setting result [1]	1 byte
	…	
	Setting result [N]	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.

Setting result [i]	
ILV_OK [0x00]	New value has been written.
ILVERR_ERROR [0xFF]	An error occurred while writing the new parameter.
ILVERR_NO_SUCH_KEY [0xE1]	The entry does not exist.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters

Notes

When you set a value using the Set Registry Key command, the system checks the coherency of the value after replying to the request. An "ILV_OK" status does not mean that the parameters has been applied but only that the terminal took your change in account.

Use the Get Registry Key or Get Registry File commands to check that the changes have been done.

When requesting for a value stored in a file, the ".cfg" extension must be omitted.

Each parameter is stored in a section. A section can contain several parameters. A configuration file contains a list of sections.

File name: ***app.cfg***

```
[section1]
parameter_1 = value_1
parameter_2 = value_2
…
[relay]
enabled = 1
aperture time in 10 ms = 30
…
```

Get Registry Key

Description

Use this command to read a parameter's value.

Command

Request

I	CMD_GET_REGISTRY_KEY [0x0C]	1 byte
L	$L_0 + L_1 + … + L_N$	2 bytes
V	Registry path n ASCII [0]	L_0 bytes
	Registry path in ASCII [1]	L_1 bytes
	…	
	Registry path in ASCII [N]	L_N bytes

Several parameters may be included. Each string must be "null terminated" (\0).

Registry path in ASCII [i].

"/file/section/parameter" For example:

"/app/relay/aperture time in 10 ms".

Remark: '/' and '|' symbols are accepted. '=', '[', ']', '\r' and '\n' symbols are forbidden.

Reply

I	CMD_GET_REGISTRY_KEY [0x0C]	1 byte
L	$1 + L_0 + L_1 + … + L_N$	2 bytes
V	Request status	1 byte

	Registry value [0]	L_0 bytes
	Registry value [1]	L_1 bytes
	…	
	Registry value [N]	L_N bytes

Request status	
ILV_OK [0x00]	The execution succeeded.

Registry Value [i]
“/file/key/parameter/value” The string is “null terminated”. Example: *“/app/relay/aperture time in 10 ms/300”.*

ILVERR_ERROR [0xFF] Execution failed

ILVERR_BAD_PARAMETER [0xFE] Invalid parameters.

ILVERR_NO_SUCH_KEY [0xE1] The entry does not exist.

Get Registry File

Description

This command will return a configuration file. It is useful to create a configuration table.

Command

Request

I	CMD_GET_REGISTRY_FILE [0x0D]	1 byte
L	L	2 bytes
V	File path in ASCII	L bytes

File path in ASCII

It is possible to read two file types:

- "file_name.cfg" will contain values of a list of parameters stored into sections,

The following "registry files" can be returned:

- *"app.cfg"* - contains application settings,
- *"adm*.cfg" contains administration parameters,
- *"net.cfg"* – stores network parameters,
- *"bio*.cfg" – stores sensor parameters.

The file path must be followed by a "end of string terminator" (0x00 byte)

Files are described in section "*Configuration File*".

Reply

I	CMD_GET_REGISTRY_FILE [0x0D]	1 byte
L	1 (+ L)	2 bytes
V	Request status	1 byte

🟧	File Content	L bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_NO_SUCH_KEY [0xE1]	The file does not exist.

The following parameters are returned only if *Request Status* is ILV_OK.

File Content
The complete file formatted as below.

Notes

File contents are described in MorphoAccess® SIGMA Family & MorphoWave® Compact **Parameter Guide**.

Distant session

Description

This mode will allow sending commands requiring more resources and forbidden in standard mode.

Commands that require this mode are followed with the [DISTANT SESSION] mention.

In distant session mode the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal is "stopped" waiting for orders. Access control restarts only when the distant session is closed or after a customizable period of time. This mode is useful to increase speed when a lot of commands have to be sent.

Command

Request

I	CMD_DISTANT_SESSION [0x63]	1 byte
L	1	2 bytes
V	Session parameter	1 byte

Session parameter	
ID_OPEN_DISTANT_SESSION [0x01]	Terminal switches to "distant session".
ID_CLOSE_DISTANT_SESSION [0x02]	Terminal switches to standalone mode.

Reply

I	CMD_DISTANT_SESSION [0x63]	1 byte
L	1 (+ 1)	2 bytes
V	Request Status	1 byte
	Session Status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

The following parameters are returned only if *Request Status* is ILV_OK.

Session status	
DISTANT_SESSION_OPENED [0x01]	Terminal is now in "distant session".
DISTANT_SESSION_CLOSED [0x02]	Standalone mode.

Compatibility note

This command is fully compatible with first generation terminals.

Set Contactless Keys (or Crypto Write) [Contactless Terminal only]

Description

On MorphoAccess® terminals equipped with a MIFARE® contactless smartcard reader, this command allows changing the MIFARE® keys for one sector.

On MorphoAccess® terminals equipped with an iCLASS® contactless smartcard reader, this command allows changing the iCLASS® keys.

On MorphoAccess® terminals equipped with a DESFire® contactless smartcard reader, or for changing the MIFARE® keys for more sectors in one time, please refer to Manage Crypto Object command and *MorphoAccess® Contactless Card Specification*.

Command

Request (for MorphoAccess® terminals equipped with a MIFARE® contactless smart card reader)

I	CMD_SET_CONTACTLESS_KEYS [0x58]	1 byte
L	14	2 bytes
V	Key Number	1 byte
	Sector Number	1 byte
	Keys	12 bytes

Key Number

This parameter is ignored.

Sector Number

This value can be set from 0 to 39.

Keys

This value is divided in 6 bytes for keys A and 6 bytes for keys B.

Request (for MorphoAccess® terminals equipped with a iCLASS® contactless smart card reader)

I	CMD_SET_CONTACLTESS_KEYS [0x58]	1 byte
L	9	2 bytes
V	Key Number	1 byte
	Key	8 bytes

Key Number

This parameter defines the key position in the terminal iCLASS® reader.

Must be set to 6.

Key

An 8 bytes binary key. The default key for MA110 is [01 23 45 67 89 AB CD EF].

Reply

I	CMD_SET_ CONTACLTESS_KEYS [0x58]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request Status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_TIMEOUT [0xFA]	Timeout

Note

The *Contactless card mapping* section describes the card structure.

Compatibility note

This command is fully compatible with first generation terminals *Crypto Write* command.

Send CBM Command [IDEMIA Only, no support is given on this function]

Description

This function allows sending commands directly to the sensor.

Command

Request

I	CMD_SEND_CBM [0x12]	1 byte
L	Lrq	2 bytes
V	[ILV Request]	Lrq bytes

[ILV Request]

The syntax of the request is detailed in the *MorphoSmart® Host System Interface specifications* document.

The Identifier, the Length and Data must be present.

Reply

I	CMD_SEND_CBM [0x12]	1 bytes
L	Lrp	2 bytes
V	[ILV Reply]	Lrp bytes

[ILV Reply]

The syntax of the reply is detailed in the *MorphoSmart® Host System Interface specifications* document.

The Identifier, the Length and Data are present.

Note

! Asynchronous ILV are not returned.

Examples

Verify request

I	CMD_SEND_CBM [0x12]	1 byte
L	8 + L_1 … (+ L_N) (+ 4)	2 bytes
V	CMD_VERIFY [0x20]	1 byte
	5 + L_1 … (+ L_N) (+ 4)	2 bytes
	Timeout	2 bytes
	Matching Threshold	2 bytes
	Acquisition quality threshold	1 byte
	Reference Template 1	L_1 bytes
	Reference Template 2 (optional)	*L_2 bytes*
	Reference Template N (optional) (N < 10)	*L_N bytes*
	Matching Score (optional)	*4 bytes*

Wi-Fi™ Functions Description

Scan Wi-Fi™ networks

Description

This function scans and retrieves available Wi-Fi™ networks information.

That command is new regarding MorphoAccess® 500, J Series.

Command

Request

I	CMD_SCAN_WIFI_NETWORKS [0xE0]	1 byte
L	0	2 bytes

Reply

I	CMD_SCAN_WIFI_NETWORKS [0xE0]	1 byte
L	1+(L1+L2+L3+…+LN)	2 bytes
V	Request status	1 byte
	Wi-Fi™ network information no 1	L1
	Wi-Fi™ network information no 2	L2
	Wi-Fi™ network information no 3	L3
	…..	
	Wi-Fi™ network information no N	LN

Request status

ILV_OK [0x00] The execution succeeded.

ILVERR_ERROR [0xFF] Execution failed

Wi-Fi™ network information

See Wi-Fi™ network information for detail.

Configure Wi-Fi™ network

Description

This function configure Wi-Fi™ network with provided information.

That command is new regarding MorphoAccess® 500, J Series.

Command

Request

I L V	CMD_CONFIGURE_WIFI_NETWORKS [0xE1]	1 byte
	L1+1+L2	2 bytes
	SSID	L1 bytes
	Encryption Type	1 byte
	Encryption key	L2 bytes

SSID

See SSID for detail.

Encryption Type

Encryption type of network:-

1. 0 = open network
2. 1 = WEP
3. 2 = WPA Personal
4. 3 = WPA2 Personal

Encryption key

See Encryption key for detail.

Reply

I	CMD_CONFIGURE_WIFI_NETWORKS [0xE1]	1 byte
L	1	2 bytes
V	Request status	1 byte

Request status

ILV_OK [0x00] The execution succeeded.

ILVERR_ERROR [0xFF] Execution failed

ILVERR_BAD_PARAMETER [0xFE] Invalid parameters.

Get Wi-Fi™ network status

Description

This function retrieves currently configured Wi-Fi™ network information.

That command is new regarding MorphoAccess® 500, J Series.

Command

Request

I	CMD_GET_WIFI_NETWORK_STATUS [0xE2]	1 byte
L	0	2 bytes

Reply

I	CMD_GET_WIFI_NETWORK_STATUS [0xE2]	1 byte
L	1+L1+L2+1+1	2 bytes
V	Request status	1 byte
	SSID	L1 bytes
	BSSID	L2 bytes
	Status	1 bytes
	Encryption Type	1 byte

Request status

ILV_OK [0x00] The execution succeeded.

ILVERR_ERROR [0xFF] Execution failed

SSID

See SSID for detail.

BSSID

See BSSID for detail.

Status

Current status with network (i.e. CONNECTED 0x01, DISCONNECTED 0x00)

Encryption Type

Encryption type of network:-

1. 0 = open network
2. 1 = WEP
3. 2 = WPA Personal
4. 3 = WPA2 Personal

Security Functions Description

Manage Crypto Object Functions Description

Description

This command manages the cryptographic objects in the terminal's secured key storage component.

The objects are managed as binary data. They are composed with several unitary objects as:

- cryptographic keys (AES, DES3…),
- certificates (public keys…),
- text passwords,
- validity date,
- time to live durations,
- version.

This command should be used in a confident environment or using a secured communication channel as SSL.

Command

Request

		Size in bytes	**Value**
I	CMD_MANAGE_CRYPTO_OBJECT [0x67]	1 byte	
L	Length	2 bytes	N$_1$ + […] + [N$_N$]
V	SUB_CMD_MANAGE_CRYPTO_OBJECT$_1$	N$_1$ bytes	
	…		
	SUB_CMD_MANAGE_CRYPTO_OBJECT$_N$	N$_N$ bytes	

SUB_CMD_MANAGE_CRYPTO_OBJECT

Sub ILV command that contains the specific action command to execute.

The available sub ILV commands are:

- [0x00] SUB_CMD_SET_OBJECT,
- [0x01] SUB_CMD_GET_OBJECT *(RFU for DESFire® project)*,
- [0x02] SUB_CMD_VERIFY_PWD *(RFU for DESFire® project)*,
- All the other values are reserved for a future use.

These sub ILV commands are described below.

Reply

		Size in bytes	Value
I	CMD_MANAGE_CRYPTO_OBJECT [0x67]	1 byte	
L	Length	2 bytes	
V	Request Status	1 byte	
	SUB_CMD_MANAGE_CRYPTO_OBJECT$_1$_RESPONSE	N_1 bytes	
	…		
	SUB_CMD_MANAGE_CRYPTO_OBJECT$_N$_ RESPONSE	N_N bytes	

If Request Status does not equal to ILV_OK, then L equals to 1, and the SUB_CMD_MANAGE_CRYPTO_OBJECT$_n$_RESPONSE is not present.

Request Status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Set Secured Object

Description

The purpose of this sub-command is to create or modify a new or existing secure object.

Request

		Size in bytes	Value
I	**SUB_CMD_SET_OBJECT [0x00]**	1 byte	0x 00
L	**Length**	2 bytes	
V	**ID** of the key that is contained in this command **Note:** This field is big-endian formatted (unlike the ILV length that is in little-endian).	4 bytes	
	Bit field Describes the following data: - 0x00000001: access rights are contained - 0x00000002: start validity date is contained - 0x00000004: the time to live is contained - 0x00000008: the version is contained - 0x80000000: the data are contained - all the other values are reserved for a future use **Note:** For a key object, the user can only change either the access rights, or the data/ttl/start date/version, or both. It means that the bit field can only have 3 values. (0x00000001, 0x8000000E, or 0x8000000F)	4 bytes	
	Access Rights (*Mandatory if Bit field & 0x00000001 only)* Describes the access rights for the object:	12 bytes	

		Size in bytes	**Value**
	- Read access rights: 4 first bytes - Write access rights: 4 following bytes - Execute access rights: 4 last bytes The right "always" is performed by setting the corresponding access rights to 0xFFFFFFFF. The right "never" is performed by setting the corresponding access rights to 0x00000000. The "read" access right is checked before reading the data or the access rights of an object. The "write" access right is checked before writing the data or the access rights of an object. The "execute" access right is checked before executing an object (verify a password for example). **Note:** This field is big-endian formatted (unlike the ILV length that is in little-endian).		
	Start validity date (*Mandatory if Bit field & 0x00000002)* Format: number of days since 01/01/1970. ***The current date must always be used***	4 bytes	
	Time to live in days *(Mandatory if Bit field & 0x00000004)* If 0, then infinite time to live duration ***Inifinite TTL should be used***	4 bytes	
	Version *(Mandatory if Bit field & 0x00000008)* For a key, if the set version is different from 0, the version for a new value to set must be superior to the previous one ***A version equal to 0 should always be used***	4 bytes	
	Data length *(Mandatory if Bit field & 0x80000000)*	4 bytes	N_{data}

		Size in bytes	Value
	Data *(Mandatory if Bit field & 0x80000000)*	N_{data}	

Note: To access an object, the access rights must be set. By default, no access rights are set. So the first operation must set the access rights for the objects that the administrator would manage.

Reply

		Size in bytes	Value
I	**SUB_CMD_SET_OBJECT [0x00]**	1 byte	0x00
L	**Length**	2 bytes	0x04
V	**Status**	4 bytes	

Status	
SOM_SUCCESS [0x00]	The execution succeeded.
SOM_ERR_FS [-2830301]	Error in Flash memory.
SOM_KEY_VERSION_INVALID [-2830305]	Bad version number. Check the version field
SOM_INVALID_PARAMETER [-2830306]	Check the ILV parameters
SOM_KEY_NOT_FOUND [-2830307]	Object ID is not handled by the MorphoAccess® terminal. Check the object ID.
SOM_ERR_INTERNAL [-2830308]	Error during initialization
SOM_NOT_OPENED [-2830310]	Secured Object Management Library not opened.

SOM_ERR_ACCESS_DENIED [-2830311]	Operation refused due to object access rights policy.
SOM_ERR_ACCESS_FILE_DOES_NOT_EXIST [-2830312]	Access rights file does not exist. The first operation on an object must set the access rights (because no access rights are set the first time)
SOM_ERR_KEY_DOES_NOT_EXIST [-2830313]	Object's access rights do not exist. The first operation on an object must set the access rights (because no access rights are set the first time)
SOM_ERR_CAN_NOT_OPEN_ACCESS_FILE [-2830315]	Error in Flash memory, access rights file cannot be opened.
SOM_ERR_CAN_NOT_WRITE_ACCESS_FILE [-2830316]	Error in Flash memory, access rights file cannot be updated.
SOM_ERR_CAN_NOT_READ_ACCESS_FILE [-2830317]	Error in Flash memory, access rights file cannot be read.

Load Security Content

Description

This command will load the security content to device.

That command is new regarding MorphoAccess® 500, J Series.

Request

I	CMD_LOAD_SECURITY_CONTENT [0xE3]	1 byte
L	(L1) + (L2)	2 bytes
V	SSL Security Data (optional)	L1 bytes
	SSL Security Passphrase (optional)	L2 bytes

SSL Security Data

See SSL Security Data structure

SSL Security Passphrase

See SSL Security Passphrase structure

Reply

I	CMD_LOAD_SECURITY_CONTENT[0xE3]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request Status

ILV_OK [0x00] the execution succeeded.

ILVERR_ERROR [0xFF] Execution failed

ILVERR_BAD_PARAMETER [0xFE] Invalid parameters.

SSL Security Data

This structure is for SSL security data.

Data structure

I	ID_SSL_SECURITY_DATA [0x01]	1 byte
L	1 + L	2 bytes
V	Profile Number	1 bytes
	Security Data	L bytes

Profile Number

Profile number 0 or 1.

Security Data

Binary security data(P12 file content).

SSL Security Passphrase

This structure is for SSL security passphrase.

Data structure

I	ID_SSL_SECURITY_PASSPHRASE[0x02]	1 byte
L	1 + L	2 bytes
V	Profile Number	1 bytes
	Passphrase	L bytes

Profile Number

SSL Profile number 0 or 1.

Passphrase

It will contain passphrase data.

Device Discovery Response

Description

This is not command but it will be received when device discovery client is started.

That command is new regarding MorphoAccess® 500, J Series.

Request

I	CMD_DEVICE_DISCOVERY_RESP [0x17]	1 byte
L	(L1) + (L2)	2 bytes
V	Serial Number	L1 bytes
	Protocol Mode	L2 bytes

Serial Number

See Serial number structure

Protocol Mode

See Protocol mode structure

Serial Number

Device serial number.

Data structure

I	DDP_SERIAL_NUM [0x18]	1 byte
L	L	2 bytes
V	Device Serial Number	L bytes

Device serial number

Device's serial number is (14 bytes).

Protocol Mode

Protocol mode in which device is running.

Data structure

I	DDP_DEVICE_MODE[0x19]	1 byte
L	1	2 bytes
V	Device protocol mode	1 bytes

Device protocol mode

For device in ILV mode value for protocol mode is 2.

For device in MA5G mode value for protocol mode is 3.

Biometric Functions Description

Verify

Description

This command captures a fingerprint image, converts it into a fingerprint template, and then compares it with all the reference fingerprint templates.

The reference templates are either provided in the command (from 1 to 20), or to be found in one specified record in the specified database.

The maximum number of reference fingerprint templates is 20.

Command

Request

I	CMD_VERIFY [0x20]	1 byte
L	5 + L_1 … (+ L_N) (+ 4) (+ 4) (+7) (+7)	2 bytes
V	Timeout	2 bytes
	Matching threshold	2 bytes
	Acquisition quality threshold	1 byte
	Reference Template 1	L_1 bytes
	Reference Template 2 (optional)	*L_2 bytes*
	…	
	Reference Template N (optional) (N < 10)	*L_N bytes*
	Matching Score (optional)	*4 bytes*
	Biometric Coder Selection (optional)	*4 bytes*
	Security Level (optional)	*7 bytes*
	Biometric Presence Detection Mode (optional)	*7 bytes*

Timeout

Finger detection timeout in seconds. A value of 0 corresponds to an infinite timeout.

Matching Threshold

This parameter can be set to values from 0 to 10 (IDEMIA recommends 5). This parameter specifies how tight the matching threshold is.

See section *Setting Up the Matching Threshold* for more information about this parameter.

Acquisition quality threshold

Ignored. Must be set to 0.

Reference Template

This is a list of ILV that contains the referenced templates. The maximum number of referenced templates is 20. The biometric data can be one of the following ILV:

PK_COMP V2, *PK_MAT* or ID_PK_FVP,

PKBase: Verify the captured finger against the templates of a database record (one or two templates depending on the database format). It is not possible to process verification against more than one database record,

ANSI or ISO formatted templates.

Matching Score

This ILV is optional. If it is not present, the matching score will not be returned.

See the *Matching Score* structure for details.

Biometric Coder Selection
This ILV is optional. If it is not present, the default biometric coder will be used. See the Biometric Coder structure for details.

Security Level
This ILV is optional. If it is not present, the default security level will be used. See the Security Level structure for details.

Biometric Presence Detection mode
This ILV is optional. If it is not present, the default presence detection will be used. See the Biometric Presence Detection Mode structure for details.

Reply

I	CMD_VERIFY [0x20]	1 byte
L	2 (+ 7)	2 bytes
V	Request status	1 byte
	Matching result	1 byte
	Matching Score (optional)	7 bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Other values	Please refer to Biometric functions error codes table

The following parameters are returned only if *Request Status* is ILV_OK.

Matching result	
All value	Please refer to Biometric functions status codes table

Matching Score

This ILV is optional. It is returned on request.

See the *Matching Score* structure for details.

Compatibility note

An optional *User ID* field used to be present in the command syntax. This field was required to display the user identifier on the screen. This field is no more expected.

The *Matching Score* sub ILV is optional.

Enroll

Description

This function captures or enrols live fingers and extracts their templates.

The template is calculated after one or three fingerprint image acquisitions (the user has to put each finger 1 or 3 times on the sensor). We strongly recommend getting 3 images for enrolment purpose, and 1 image for verification purpose. An enrolment based on 3 images will increase the system accuracy.

The number of fingers can be specified.

The calculated minutiae can be exported to the host as well as the reference image.

To obtain the best accuracy, it is strongly recommended to use the fore, the thumb or the middle finger.

Command

Request

I	CMD_ENROLL [0x21]	1 byte
L	8 (+ LUID) (+ LData1) + … + (+ Ldatai) (+ 4) (+ 4) (+7) (+7) (+7)	2 bytes
V	Database identifier	1 byte
	Timeout	2 bytes
	Acquisition quality threshold	1 byte
	Enrolment type	1 byte
	Number of fingers	1 byte
	Save record	1 byte
	Export Minutiae Size	1 byte
	User ID (optional, depending on enrol type)	LUID bytes
	Additional user data field 1 (optional)	LData1 bytes
	…	

Additional user data field I (optional)	LDatai bytes
Biometric Coder Selection (optional)	4 bytes
Biometric Algorithm Parameter	4 bytes
Fingerprint Latent Detection (optional)	7 bytes
Security Level (optional)	7 bytes
Biometric Presence Detection Mode (optional)	7 bytes

Database identifier

Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

Timeout

Finger detection timeout in seconds. A value of 0 corresponds to an infinite timeout.

Acquisition quality threshold

Not used. Set to 0.

Enrolment Type

Specifies the number of fingerprint image acquisitions. Allowed values are 0, 1 and 3.

We strongly recommend setting this value to 0 (default value) or 3 for enrolment purpose to increase the system performances: in this case, the template is generated from a consolidation calculation of three consecutive acquisitions of the same fingerprint.

It is also possible to set this value to 1 for verification purpose. In this case, it is not possible to save the record in the internal database: the template is generated from one single fingerprint acquisition.

Number of fingers

The number of fingers to enrol. This function can enrol 1 or 2 fingers.

Set this value to 0x01 to enrol 1 finger.

Set this value to 0x02 to enrol 2 fingers.

Save Record

Set this Boolean to TRUE (0x01) to store calculated minutiae into the local database. Otherwise set it to FALSE (0x00).

When Enrolment Type is set to 1, the Save Record flag must be set to FALSE.

If the Save Record flag is set to TRUE, the User ID parameter is mandatory.

Export Minutiae Size

Defines the format of the exported minutiae.

Set this value to 0x00 to exclude the calculated minutiae from the reply.

Set this value to 0x01 to export the minutiae with its default size.

For PK_COMP only, this value can be set from 170 (0xAA) to 255 (0xFF) to limit the maximum size of the template to be returned. It means that the exported template size will less or equal to the specified value: if the template size is higher than the limit then it is compressed to fit in the required size. This feature is especially useful when the template has to be stored to a smart card with a limited size for storage.

User ID

ILV formatted data, which contains the unique user identifier. It must be filled if the *Save Record* value is set to TRUE. The User ID can be retrieved by the *Identify* function under Hit condition.

Refer to *User ID* structure for further details.

❗ **This field is managed as a byte array.**

Additional User Data field j

These are additional user data. There must be filled if the *Save Record* value is set to TRUE. All the additional user data fields defined in the *Create Database* command have to be filled.

See the *Additional User Data* structure for details.

Biometric Coder Selection

This ILV is optional. If it is not present, the default biometric coder will be used.

See the Biometric Coder structure for details.

Biometric Algorithm Parameter

When *Export Minutiae Size* field is equal to 0x01, this optional ILV formatted data allows selecting the format of the template to be exported. If not present, the templates will be exported (if required) in the PK_COMP V2 format.

Fingerprint Latent Detection

This ILV is optional. It is used to enable the fingerprint latent detection.

The latent detection mechanism should be enabled when capturing the template for verification performed by the host system (i.e. when the parameter **Enrolment Type** is set to 1).

See the Fingerprint Latent Detection structure for details.

Security Level

This ILV is optional. If it is not present, the default security level will be used.

See the Security Level structure for details.

Biometric Presence Detection Mode

This ILV is optional. If it is not present, the default presence detection will be used.

See the Biometric Presence Detection Mode structure for details.

Reply

I	CMD_ENROLL [0x21]	1 byte
L	6 (+ Lfp1) (+ Lfp2) (+ Lmm1) (+Lmm2)	2 bytes
V	Request status	1 byte
	Enrol status	1 byte
	User database index	4 bytes
	Fingerprint template 1 (optional)	Lfp1 bytes
	Fingerprint template 2 (optional)	Lfp2 bytes
	Multimodal Template 1 (optional)	Lmm1 bytes
	Multimodal Template 2 (optional)	Lmm2 bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other values	Please refer to Biometric functions error codes table

The following parameters are returned only if *Request Status* is ILV_OK.

Enrol status	
ILVSTS_OK [0x00]	The enrolment succeeded.
Other values	Please refer to Biometric functions status codes table

User database index

If enrolment is successful, the user database index value is returned as '0x00000000'. As MorphoAccess® SIGMA Family & MorphoWave® Compact terminal's database does not contain 'user's database index'.

If *Save record* field in the request was set to FALSE the returned value is 0xFFFFFFFF.

If the *Request Status* is not ILV_OK or the enrol status differs from ILVSTS_OK the *User Database Index* is not returned.

Fingerprint template i

ILV formatted data which include the template generated for acquired fingerprint # i. This template complies with the PK_Comp v2 format (which is the compressed version of the PK_MAT format).

When the *Export Minutiae Size* field value is different from 0, there is one biometric data per captured finger, in accordance to the value specified in the *Number of fingers* field of the ENROLL command. The biometric data are retuned in the same order as the fingerprint capture: 1st biometric data is related to the 1st fingerprint acquired.

When the *Export Minutiae Size* field value is equal to 0, there is none biometric data returned.

See the PK_COMP V2 structure for details.

Multimodal template i

Resulting multimodal template generated by the sensor if the **Export Minutiae Size** parameter is different from 0.

Only available on Multimodal finger vein / fingerprint terminals.

See the PK_FVP structure for details.

Compatibility note

For additional fields, ID_C_DATA, ID_S_DATA, ID_L_DATA identifiers do not exist anymore.

As a difference with previous generation of MorphoAccess® (MA200 and 300 Series) the template exported, using PK_Comp V2 format, is not fulfilled to 256 bytes with 0x00. Then if the exported template has to be sent to a terminal of previous generation, this template must be fulfilled to 256 bytes with 0x00 before sending.

Notes about additional fields

Size and order of additional data fields must be consistent with database structure.

The CREATE DATABASE function provides more information about additional fields.

Identify

Description

This function identifies a live finger against the local database. If the database is empty, the function will return immediately.

Command

Request

I	CMD_IDENTIFY [0x22]	1 byte
L	6 (+ 4) (+ 4)	2 bytes
V	Database Identifier	1 byte
	Timeout	2 bytes
	Matching Threshold	2 bytes
	RFU	1 byte
	Matching Score (optional)	4 bytes
	Biometric Coder Selection (optional)	4 bytes
	Security Level (optional)	7 bytes
	Biometric Presence Detection Mode (optional)	7 bytes

Database Identifier
Number of available databases depends on terminal type and license. Value of this command is ‘0’, as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

Timeout

Finger detection timeout in seconds. A value of 0 corresponds to an infinite timeout.

Matching threshold

This parameter can be set to values from 0 to 10 (IDEMIA recommends 5). This parameter specifies how tight the matching threshold is.

See section *Setting Up the Matching Threshold* for more information about this parameter.

RFU

Not used, set to 0.

Matching Score

This ILV is optional. If it is not present, the matching score will not be returned.

See the *Matching Score* structure for details.

Biometric Coder Selection

This ILV is optional. If it is not present, the default biometric coder will be used.

See the Biometric Coder structure for details.

Security Level

This ILV is optional. If it is not present, the default security level will be used.

See the Security Level structure for details.

Biometric Presence Detection Mode
This ILV is optional. If it is not present, the default presence detection will be used. See the Biometric Presence Detection Mode structure for details.

Reply

I	CMD_IDENTIFY [0x22]	1 byte
L	2 + 4 + LUID (+ L_1) + … + (+ L_N) (+ 7)	2 bytes
V	Request Status	1 byte
	Matching Result	1 byte
	User Database Index	4 bytes
	User ID (Field 0)	LUID bytes
	Additional user data field 1	L_1 bytes
	…	
	Additional user data field N	L_N bytes
	Matching score (optional)	7 bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other values	Please refer to Biometric functions error codes table

The following parameters are returned only if *Request Status* is ILV_OK.

Matching result

All value	Please refer to Biometric functions status codes table

User Database Index

This is the index database of the record (person). If the request status is not ILV_OK, or the matching status is not ILVSTS_HIT, the User Database index is not returned.

User ID

ILV formatted data which contain the unique user identifier of the record that matches the fingerprint. These data are returned only if the request status is ILV_OK, and if the matching status is ILVSTS_HIT.

Refer to *User ID* structure for further details.

This field is managed as a byte array.

Additional user data field i

One ILV per additional field. All the additional user data fields defined in the *Create Database* command are present (public, private, empty fields).

If the request status is not ILV_OK, or if the matching status is not ILVSTS_HIT, those fields are not returned.

Matching Score

This ILV is optional. It is returned on request.

See the *Matching Score* structure for details.

Compatibility note

The *Matching Score* sub ILV can be omitted in the request.

Additional user data fields are returned under hit condition.

Database Functions Description

Create Database

Description

This function creates a biometric database in flash memory.

Command

Request

I	CMD_CREATE_DB [0x30]	1 byte
L	5 (+ L_1 + L_2 + … + L_i)	2 bytes
V	Database identifier	1 byte
	RFU	1 byte
	Person number in database	2 bytes
	Finger number per person	1 byte
	Additional data Field 1	L_1 bytes
	Additional data Field 2	L_2 bytes
	…	
	Additional data Field i	L_i bytes

Database identifier
Number of available databases depends on terminal type and license. Value of this command is ‘0’, as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal, there is only ONE database present. See Database Size Limits to known the maximum number of bases.

RFU

Not used, set to 0.

Person number in database

Number of user depends on terminal type and license.

See Database Size Limits to known the maximum number of persons.

Finger number per person

Number of fingers per person in the database must be 1 or 2.

Additional data Fields

A list of Public or Private Fields describing the database format.

See the *Private Field* and *Public Field* structures for details.

See Database Size Limits to known the maximum number and the maximum size of additional user's fields per user.

Reply

I	CMD_CREATE_DB [0x30]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request status

ILV_OK [0x00]	The execution of the function succeeded.

ILVERR_ERROR [0xFF]	An error occurred during the execution of the function.
Other error codes	Please refer to Biometric functions error codes table

Compatibility note

❗ The *Public Fields* name cannot exceed 5 characters on all MorphoAccess® SIGMA Family & MorphoWave® Compact terminals (100 Series, J Series, VP Series, 500 Series) instead of 7 on MorphoAccess® 200 and 300 Series.

The *Public Field* structure corresponds to a standard additional field.

Particular fields

These fields can be interpreted by the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal “access control” application:

- ADMIN to activate biometric login,
- TMSK to activate time mask verification,
- NAME to display user name,
- FNAME to display user first name.

Admin field:

To provide biometric administrator login it is possible to specify if a user has administration rights. When creating a database, the Name of field must be ‘ADMIN’ for additional field to be used for biometric login, have following structure:

Name	Length
ADMIN	3 bytes

When enrolling a user or adding a record, the corresponding additional field must contain « 1 » (i.e. 0x31 0x00 x00) to provide administration rights or « 0 » (i.e. 0x30 0x00 x00) if the user is not an administrator. Please see sections 5-39 and 5-52 for more information about Enrolment and Add Base Record functions.

Managed by enrolment application and access control application on MorphoAccess® 500 Series.

Time mask field:

In the way to enable time mask features, the additional field must have the following structure:

Name	Length
TMSK	84 bytes

When enrolling a user or adding a record the corresponding additional field will contain the time mask descriptor. See paragraph Time Mask Functions Description for more information about time mask.

On MorphoAccess® 500 Series, when enrolling a user through the enrolment application, the Time mask field is set to "always accepted".

This field is processed by the access control application of MorphoAccess® 100 Series, J Series, VP Series and 500 Series

Name field:

Managed by enrolment application (MorphoAccess® 500 Series only) and by access control application of the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal.

Name	Length
NAME	20 bytes

First name field:

Managed by enrolment application (MorphoAccess® 500 Series only) and by access control application of the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal.

Name	Length
FNAME	20 bytes

Warning

It is not possible to change the size and/or the number of additional fields of an already created base. First, the existing base must be deleted, and then a new base must be created. It means that all the biometric records, of the original base, are deleted. To avoid the lost of valuable data, it is recommended to keep a copy of the biometric data in the host system. As, for safety reason, the terminal does not allow to extract the content of the biometric database, the host must capture the minutiae data during the enrolment of the user. By selecting the "export minutiae" option of the ENROLL command, the host will receive the minutiae data built from the captured fingerprints of the user. Other data, such as the user ID and the content of the additional data field, are provided by the host during the enrolment process, then all the data required to register the user in the data base using the ADD_BASE_RECORD command will be available in the host system.

Erase Base

Description

This function erases all records in the local database. Base structure is not erased.

Command

Request

I	CMD_ERASE_DB [0x32]	1 byte
L	1	2 bytes
V	Database identifier	1 byte

Database identifier
Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal, there is only ONE database present. See Database Size Limits to known the maximum number of bases.

Reply

I	CMD_ERASE_DB [0x32]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request status	
ILV_OK [0x00]	The execution of the function succeeded.
ILVERR_ERROR [0xFF]	An error occurred during the execution of the function.

Other error codes	Please refer to Biometric functions error codes table

Compatibility note

The *Garbage Collector* command is not required any more.

Erase All Base

Description

This function deletes the local database in flash memory. Base structure is erased.

Command

Request

I	CMD_ERASE_ALL_DB [0x33]	1 byte
L	1 [0x0001]	2 bytes
V	RFU [0x00]	1 byte

RFU	
	Not used, must be set to 0.

Reply

I	CMD_ERASE_ALL_DB [0x33]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request status	
ILV_OK [0x00]	The execution of the function succeeded.
ILVERR_ERROR [0xFF]	An error occurred during the execution of the function.
ILVERR_BADPARAMETER [0xFE]	Invalid ILV format
Other error codes	Please refer to Biometric functions error codes table

Note

A new database can be created after using this command.

Add Base Record

Description

This function adds a record to the local database.

Command

Request

I	CMD_ADD_DB_RECORD [0x35]	1 byte
L	1 + L_1 (+ L_2) + LUID (+ LData1) + … (+ LDatai) (+ 4)	2 bytes
V	Database Identifier	1 byte
	Reference Template 1	L_1 bytes
	Reference Template 2 (optional)	L_2 bytes
	User ID	LUID bytes
	Additional User Data Field 1 (optional)	LData1 bytes
	…	
	Additional User Data Field i (optional)	LDatai bytes
	No check on template (optional)	4 bytes

Database Identifier

Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

See Database Size Limits to known the maximum number of bases.

Reference Templates i

The following templates format can be used:

- IDEMIA fingerprint template private formats: PK_COMP V2, PK_MAT,
- International fingerprint minutiae record: ANSI or ISO formatted templates,
- IDEMIA multimodal private format: PK_FVP.

The maximum number of referenced fingers depends on the number of fingers supported by the database.

Fingerprint and multimodal templates cannot be mixed.

User ID

ILV formatted data that contain the unique user identifier of the record (the user) in the database. If the specified User ID value is already in the database, the command is rejected with a specific error code.

Refer to *User ID* structure for further details.

❗ **This field is managed as a byte array.**

Additional User Data Field j

These are additional user data. All the additional user data fields defined in the *Create Database* command have to be filled.

See the *Additional User Data* structure for details.

No check on template (optional)

This ILV is optional. If it is not present, checks on reference templates are performed: same finger cannot be used twice, and the person must not be already enrolled. This option is useful to reduce the time taken to fill large databases. In this case, the database coherence must be previously checked.

See the *No Check On Template* structure for details.

Reply

I	CMD_ADD_DB_RECORD [0x35]	1 byte
L	1 (+ 5)	2 bytes
V	Request status	1 byte
	Base status	1 byte
	User database index	4 bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other error codes	Please refer to Biometric functions error codes table

The following parameters are returned only if *Request Status* is ILV_OK.

Base Status	
ILVSTS_OK [0x00]	The enrolment succeeded.
Other status codes	Please refer to Biometric functions status codes table

User Database Index
This is the record index of the person. If the request status is not ILV_OK, the User Database Index is not returned.

Notes about additional fields:

Size and order of additional data fields must be consistent with database structure.

The CREATE DATABASE function provides more information about additional fields.

Note

The *No Check on Template* option is useful to decrease templates download time.

Compatibility note

ⓘ The *Time Stamp* sub ILV is no more supported.

The *No Check On Template* option is not required but useful.

Remove Base Record

Description

This function removes a record from the local database.

Command

Request

I	CMD_REMOVE_DB_RECORD [0x36]	1 byte
L	1 + L	2 bytes
V	Database identifier	1 byte
	User ID ILV or User Index ILV	L bytes

Database identifier

Number of available databases depends on terminal type and license. Value of this command is ‘0’, as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

See Database Size Limits to known the maximum number of bases.

User ID

ILV formatted data that contain the unique user identifier of the record to remove, as it was provided with by the *Enrol* command or by the *Add Base Record* command at record creation.

Refer to *User ID* structure for further details.

❗ **This field is managed as a byte array.**

User Index ILV
ILV formatted data that contain the index of the record in the database, returned by the *Enrol* command or by the *Add Base Record* command at record creation.

Reply

I	CMD_REMOVE_DB_RECORD [0x36]	1 byte
L	1 [0x0001]	2 bytes
V	Request status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other error codes	Please refer to Biometric functions error codes table

Compatibility note

The *Time Stamp* sub ILV is ignored.

Get Data

Description

This function reads a database public field.

Command

Request

I	CMD_GET_DATA [0x3F]	1 byte
L	5 + L	2 bytes
V	Database identifier	1 byte
	Field Index	4 bytes
	ILV User ID or User Index	L bytes

Database identifier

Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

See Database Size Limits to known the maximum number of bases.

Field Index

Field index on which the search is performed. Field 0 is dedicated to User ID.

User ID or User Index
User ID: ILV formatted data that contain the unique user identifier of the record, as it was provided with by the Enrol command or by the Add Base Record command at record creation. User Index: ILV formatted data that contain the index of the record in the database, returned by the Enrol command or by the Add Base Record command at record creation.

Reply

I	CMD_GET_DATA [0x3F]	1 byte
L	1 + L	2 bytes
V	Request Status	1 byte
	Field	L bytes

Request Status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other error codes	Please refer to Biometric functions error codes table

The following parameters are returned only if *Request Status* is ILV_OK.

Field
The field content.

Compatibility note

This command is a MorphoAccess® 100 Series, J Series, VP Series and 500 Series new functionality.

Get Public Fields

Description

This function reads a given public field from all users. For example, this function can be used to return a list of users present in the local database.

Command

Request

I	CMD_GET_PUBLIC_FIELDS [0x3E]	1 byte
L	5	2 bytes
	Database Identifier	1 byte
	Field Index	4 bytes

Database Identifier

Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

See Database Size Limits to known the maximum number of bases.

Field Index

Field index on which the search is performed. Field 0 is dedicated to User ID

Reply

I	CMD_GET_PUBLIC_FIELDS [0x3E]	1 byte
L	5 + L_1 + … + L_k	2 bytes
V	Request Status	1 byte
	User Number	4 bytes
	Field 1	L_1 bytes
	…	
	Field k	L_k bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other error codes	Please refer to Biometric functions error codes table

The following parameters are returned only if *Request Status* is ILV_OK.

User Number

Field that contains the number of users found in the database. This value is equal to **k**, the number of fields included in the reply.

Field I

A *Field Content* ILV formatted data packet that contains field content. There is one field per record. Content from all records is returned, even empty records. This allows the record index to correspond properly to the records.

See the *Field Content* structure for details.

Compatibility note

This command is a MorphoAccess® 100 Series, J Series, VP Series and 500 Series new functionality.

Update Public Data

Description

This function modifies one or more public fields from one record.

Command

Request

I	CMD_UPDATE_PUBLIC_DATA [0x3C]	1 byte
L	2 + L_1 + L_2 + … + L_N	2 bytes
V	Database identifier	1 byte
	Field number = N+1	1 byte
	User ID or User Index	L_1
	Field0	L_2
	Field1	L_3
	…	
	FieldN	L_N

Database identifier

Number of available databases depends on terminal type and license. Value of this command is ‘0’, as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

See Database Size Limits to known the maximum number of bases.

Field number

Indicate number of fields to update.

User ID

ILV formatted data that contain the unique user identifier of the record, as it was provided with by the *Enrol* command or by the *Add Base Record* command at record creation.

User Index *ILV*

ILV formatted data that contain the index of the record in the database, returned by the *Enrol* command or by the *Add Base Record* command at record creation.

Field

An ILV formatted data packet that contains field content. There is one field per record. Content from all records is returned, even empty records. This allows the record index to correspond properly to the records.

Reply

I	CMD_UPDATE_PUBLIC_DATA [0x3C]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request Status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Other error codes	Please refer to Biometric functions error codes table

Compatibility note

This command is a MorphoAccess® 100 Series, J Series, VP Series and 500 Series new functionality.

Get Base Config

Description

This function retrieves the configuration of the local database.

Command

Request

I	CMD_GET_DB_CONFIG [0x07]	1 byte
L	1	2 bytes
V	Database identifier	1 byte

Database identifier
Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present. See Database Size Limits to known the maximum number of bases.

Reply

If the requested base exists, the reply has the format described below.

I	CMD_GET_DB_CONFIG [0x07]	1 byte
L	18 + L_{ts} + L_1 + … + L_i	2 bytes
V	Request Status = ILV_OK [0x00]	1 byte
	Number fingers/person	1 byte
	Max record number	4 bytes
	Current Record Number	4 bytes
	Free Record Number	4 bytes

Number of Fields	4 bytes
ILV Timestamp	L_{ts} bytes
Additional data Field 1	L_1 bytes
…	
Additional data Field i	L_i bytes

Request status	
ILV_OK [0x00]	The execution succeeded: specified database found.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
Other error codes	Please refer to Biometric functions error codes table

The fields described here after are not returned when the Request status is different from ILV_OK.

Number fingers/person

Number of fingers saved per person.

Max record number

Maximum number of records in database.

Current Record Number

Number of records currently saved in Database.

Free Record Number

Number of records currently available in Database.

Fields Number

Number of fields used in Database.

ILV Timestamp (not implemented)

I	ID_TIMESTAMP [0x11]	1 byte
L	L	2 bytes
V	ASCII string with Day, Month, Years, Hours, minutes and Seconds each encoded with 2 bytes: “DDMMYYHHmmSS”.	L bytes

Additional data Fields

A list of Public or private fields describing the database format.

The maximum number of additional fields is 8 only irrespective of number of database records.

Compatibility note

❗ The *Time Stamp* sub ILV is ignored.

Get All Bases Config

Description

This function retrieves the configuration of all local databases.

Command

Request

I	CMD_GET_ALL_DB_CONFIG [0x31]	1 byte
L	0	2 bytes

Reply

I	CMD_GET_ALL_DB_CONFIG [0x31]	1 byte
L	$2 + L_1 + … + L_n$	2 bytes
V	Request Status	1 byte
	Nb Returned Bases	1 byte
	Base 1 configuration	L_1 bytes
	…	
	Base i configuration	L_i bytes
	…	
	Base n configuration	L_n bytes

Request status	
ILV_OK [0x00]	The execution succeeded.

Nb Returned Bases

Current release of MorphoAccess® 100 Series, J Series and VP Series do not support management of several databases.

Base i configuration

This ILV is returned for each created base (1 in practice).

See the *Base Configuration* ILV for more details.

Compatibility note

! The *Time Stamp* sub ILV is ignored.

Destroy DB

Description

This command deletes the specified local database in the flash memory: all the records of the database and the structure of the database are removed.

Command

Request

I	CMD_DESTROY_DB [0x3B]	1 byte
L	0*0001	2 bytes
	Database Identifier	

Database Identifier

Identifier of the database to be deleted. Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Reply (Success)

I	CMD_DESTROY_DB [0x3B]	1 byte
L	0*0001	2 bytes
V	Request Status = ILV_OK	1 byte

Reply (Failed)

I	CMD_DESTROY_DB [0x3B]	1 byte
L	0*0005	2 bytes
V	Request Status	1 byte
	Internal Error Code	4 bytes

Request status	
ILV_OK [0x00]	The execution succeeded.

Nb Returned Bases

Current release of MorphoAccess® 100 Series, J Series and VP Series do not support management of several databases.

Base i configuration

This ILV is returned for each created base (1 in practice).

See the *Base Configuration* ILV for more details.

Compatibility note

❗ The Time Stamp sub ILV is ignored.

Request Status:

Result of the execution of the command (error code)

ILVERR_ERROR: An unexpected error occurred during the execution of the command

ILVERR_BASE_NOT_FOUND: The specified database does not exist.

ILVERR_BADPARAMETER: ILV Format incorrect.

ILVERR_CMD_INPROGRESS: Command received while another command is running.

ILVERR_OPERATION_NOT_SUPPORTED: Operation not supported by the firmware loaded in the MorphoSmart™ device.

Internal error code:

4 bytes little endian numeric value. Reserved for IDEMIA internal use (mainly for development purpose).

Handle Several Records [DISTANT SESSION]

Description

Because database size can exceed the amount of data that can be transmitted in one exchange, a dedicated syntax of the command allows sending, deleting or getting several database records.

To know the maximum size of data that can be transmitted in one request, use the Get Version command with Terminal Identifier parameter (*[ILV receipt buffer size]* field).

A record contains the user template, the user identifier, and additional data.

Each Add Record, or Del Record request is analyzed by the terminal.

Depending on the Add Record, or Del Record result, the download continues or is stopped:

If the sub request is a success or if the record is already present in the database (same ID) the download continues.

The base must have been previously created using the Create Database command.

Command (fragmented database)

Request (extended ILV if base size if greater than 65535 bytes)

I	ID_HANDLE_SEVERAL_RECORDS [0x72]	1 Byte
L	N	2 or 6 Bytes
V	<ILV Add/Del Record 1>	
	…	
	<ILV Add/Del Record N>	

Reply

I	ID_HANDLE_SEVERAL_RECORDS [0x72]	1 byte
L	1 + N	2 bytes

V	Request Status	1 byte
	<ILV Add/Del Record Reply 1>	
	…	
	<ILV Add/Del Record Reply N>	

Request Status	
ILV_OK [0x00]	The execution succeeded (at least one record has been added.)
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_BAD_SESSION_STS[0xDF]	Distant session not opened.
Other error codes	Please refer to Biometric functions error codes table

Example

First ILV request

I	ID_HANDLE_SEVERAL_RECORDS [0x72]
L	N bytes
V	[ILV Add Record 1]
	[ILV Add Record 2]
	…
	…
	[ILV Add Record i]

Following ILV request

I	ID_HANDLE_SEVERAL_RECORDS [0x72]
L	N bytes
V	[ILV Add Record i+1]
	[ILV Add Record i+2]
	…
	[ILV Add Record N]

Status for first ILV request: all records have been loaded

I	ID_HANDLE_SEVERAL_RECORDS [0x72]
L	N + 1 bytes
V	Request Status: [0x00]
	[Reply ILV Add Record 1]
	[Reply ILV Add Record 2]
	…
	[Reply ILV Add Record i]

Status for second ILV request: all records have been loaded

I	ID_HANDLE_SEVERAL_RECORDS [0x72]
L	N + 1 bytes
V	Request Status: [0x00]
	[Reply ILV Add Record i+1]
	[Reply ILV Add Record i+2]

…

[Reply ILV Add Record n]

Remarks

A *Get Base Config* command will return the base state at the end of the downloading.

There is no “ending ILV request”.

Access Control Log Functions Description

Overview

The MorphoAccess® SIGMA Family & MorphoWave® Compact terminal is able to log all its biometric activities. This function can be used to get information about control (like duration, matching score…) or for time and attendance purposes.

Logged data are:

- event date and time,
- type of event (identification, authentication),
- ID of the person concerned by the event,
- Biometric information (minutiae quality or matching score).

A log occupies 64 bytes. It is possible to log either *8000* or *65000* events depending on the terminal configuration.

Download the binary log file using the *Get Log* command.

Erase the log file using the *Erase Log* command.

The number of written lines is returned by the *Get Log Status* command.

❗ When the base is full, the log file must be downloaded and erased. Records are **not cyclic.**

Base Structure

Log Base: 240 Kbytes.		Line 1: 64 bytes.	###
		Line 2: 64 bytes.	###
		…	
		Line k-1: 64 bytes.	###
Record pointer	**>**	Line k: 64 bytes.	First free record.
		Line k+1: 64 bytes.	
		…	
		Line 3840: 64 bytes.	

A line: 64 bytes.

#####	: Written record.
🟩	: Free record.

Line Structure

Each record is written in line that has the following generic form:

Line 1	Record Date (12 bytes)	Action type (1 byte)	Binary data	Padding (with 0x00)
	64 bytes			
…				
Line k-2	Record Date (12 bytes)	Action type (1 byte)	Binary data	Padding (with 0x00)
	64 bytes			
Line k-1	Record Date (12 bytes)	Action type (1 byte)	Binary data	Padding (with 0x00)
	64 bytes			

- **Record Date** field is coded in ASCII in the following format: DDMMYYHHMMSS. The following value corresponds to 25/07/02 15:35:25:

Record Date (12 bytes)
323530373032313533353235

- **Action type** byte describes the nature of the event logged. There are two major log families: Authenticate records and Identification records.
- **Binary data** depends on the Action type. Refer to *Authent Records* structure and *Identification Records* structure to know how binary data are organized.

Authentication Records format

Structure

Data format is big endian.

```
typedef struct
{
        unsigned char         m_auc_Date[LOG_DATE_SIZE];
        unsigned char         m_uc_ActionType;
        unsigned char                m_uc_AcquisitionTh;
        unsigned short               m_us_Timeout;
        unsigned short               m_us_MatchingTh;
        unsigned char         m_puc_UserId[LOG_USER_ID_SIZE];
        unsigned short               m_us_Duration;
        unsigned char         m_uc_Result;
        unsigned char         m_uc_NbFinger;
        unsigned short               m_us_not_used;
        unsigned char         m_uc_MatchFinger;
        unsigned char         m_uc_TA_Ext_PressedKey;
        unsigned char         m_uc_Pad[18];
} T_LOG_AUTHENT;
```

Definitions

```
#define LOG_DATE_SIZE             12
#define LOG_USER_ID_SIZE  20
```

Description

Member	Description	Size
Date and time	m_puc_Date[12]	12
	Format: DDMMYYHHMMSS	
Action type	m_uc_Action	1
	Authentication with database requested (Keyboard authentication for example): TLOG_AUTH_CTL_WDB_ID 0x25 **Authentication without database, Only ID is requested:** TLOG_AUTH_CTL_WODB_ID_ONLY_TERMINAL (terminal choice)0x22 TLOG_AUTH_CTL_WODB_ID_ONLY_CARD (Card Choice) 0x2A **Authentication without database, PIN and Fingerprints are requested:** TLOG_AUTH_CTL_WODB_PIN_THEN_PKS_TERMINAL 0x28 TLOG_AUTH_CTL_WODB_PIN_THEN_PKS_CARD 0x2E **Authentication without database, Fingerprints are requested:** TLOG_AUTH_CTL_WODB_PKS_TERMINAL 0x24 TLOG_AUTH_CTL_WODB_PKS_CARD 0x2B	

	Authentication without database, PIN and BIOPIN are requested: TLOG_AUTH_CTL_WODB_PIN_THEN_BIOPIN_TERMINAL 0x29 TLOG_AUTH_CTL_WODB_PIN_THEN_BIOPIN_CARD 0x2F **Authentication without database, PIN is requested:** TLOG_AUTH_CTL_WODB_PIN_TERMINAL 0x27 TLOG_AUTH_CTL_WODB_PIN_CARD 0x2D **Authentication without database, BIOPIN is requested:** TLOG_AUTH_CTL_WODB_BIOPIN_TERMINAL 0x26 TLOG_AUTH_CTL_WODB_BIOPIN_CARD 0x2C **Specific to Time and Attendance (a bit is added to the action code):** Local authentication when arriving TA_LOG_IN +0x00 Local authentication when going on duty again TA_LOG_IN_FROM_DUTY +0x80 Local authentication when going off duty TA_LOG_OUT_FROM_DUTY +0xC0 Local authentication when leaving TA_LOG_OUT +0x40	
Acquisition threshold	m_uc_AcquisitionTh	1
	0	

Member	Description	Size
Timeout	m_us_Timeout	2
	Timeout for operation	
Matching threshold	m_us_MatchingTh	2
	Terminal threshold	
User Id	m_puc_UserId[20]	20
	“98756” for example	
Operation duration	m_us_Duration	2
	(in tenth of second)	
Result	m_uc_Result	1
	Success LOG_AUTH_OK 0x00 **Failure** LOG_AUTH_FAILED 0x01 **“Timeout”** LOG_AUTH_TIMEOUT 0x19 **Not on time (specific to Time mask)** LOG_BIO_NOT_ON_TIME 0x02 **False Finger detected (specific to MorphoAccess® 521)** LOG_FAKE_FINGER_DETECTED 0x30 **Invalid Minutiae** LOG_AUTH_PK 0x12 **Generic error** LOG_BIO_ERROR 0xFF	

Number of finger	m_uc_NbFinger	1
	2	
RFU	m_us_not_used	2
	0	
Matched finger number	m_uc_MatchFinger	1
	Set to 0, must be ignored.	
Extended Time and Attendance pressed key	m_uc_TA_Ext_PressedKey	1
	0x30 for key '0' – 0x31 for key '1' 0x32 for key '2' 0x33 for key '3' 0x34 for key '4' 0x35 for key '5' 0x36 for key '6' 0x37 for key '7' – 0x38 for key '8' – 0x39 for key '9' – 0x43 for 'dot' key – 0x59 for 'yellow' key 0x47 for 'green' key 0x55 for 'up' key 0x44 for 'down" key	

Identification Records format

Structure

Data format is **big endian**.

```
typedef struct
{
        unsigned char       m_puc_Date[LOG_DATE_SIZE];
        unsigned char       m_uc_ActionType;
        unsigned char       m_uc_FlashType;
        unsigned char       m_puc_BaseName[LOG_BASE_NAME_SIZE];
        unsigned char       m_puc_UserId[LOG_USER_ID_SIZE];
        unsigned char       m_uc_not_used;
        unsigned short              m_us_Timeout;
        unsigned short              m_us_Duration;
        short                       m_s_PersonIndex;
        unsigned char       m_uc_Result;
        unsigned char       m_uc_TA_Ext_PressedKey;
        unsigned char       m_uc_Pad[6];
} T_LOG_IDENT;
```

Definitions

```
#define LOG_DATE_SIZE               12
#define LOG_BASE_NAME_SIZE          15
#define LOG_USER_ID_SIZE    20
```

Description

Field	Description	Size
Date and time	m_puc_Date[12]	12
	Format: DDMMYYHHMMSS	
Action type	m_uc_Action	1
	Local identification TLOG_IDENT 0x30 Specific to Time and Attendance (a bit is added to the action code): Local identification when arriving TA_LOG_IN +0x00 Local identification when going on duty again TA_LOG_IN_FROM_DUTY +0x80 Local identification when going off duty TA_LOG_OUT_FROM_DUTY +0xC0 Local identification when leaving TA_LOG_OUT +0x40	
Base support	m_uc_FlashType	1
	0	
Base name	m_puc_BaseName[15]	15
	“bioa”	
User Id	m_puc_UserId[20]	20
	“98756” for example	

RFU	m_uc_not_used	1
	0	
Timeout	m_us_Timeout	2
Operation duration	m_us_Duration	2
	(in tenth of second)	
User index	m_s_PersonIndex	2
	-1 if failed	

Field	Description		Size
Result	m_uc_Result		1
	Success		
	LOG_BIO_OK	0x00	
	Not recognized		
	LOG_BIO_FAILED	0x01	
	Not on time (specific to Time mask)		
	LOG_BIO_NOT_ON_TIME	0x02	
	False Finger detected (for terminal with FFD)		
	LOG_FAKE_FINGER_DETECTED	0x30	
	Timeout		
	LOG_BIO_TIMEOUT	0x19	
	Invalid Minutiae		
	LOG_AUTH_PK	0x12	
	Generic error		
	LOG_BIO_ERROR	0xFF	
Extended Time and Attendance pressed key	m_uc_TA_Ext_PressedKey		1
	0x30 for key ‘0’ – 0x31 for key ‘1’ 0x32 for key ‘2’ 0x33 for key ‘3’ 0x34 for key ‘4’ 0x35 for key ‘5’ 0x36 for key ‘6’ 0x37 for key ‘7’ – 0x38 for key ‘8’ – 0x39 for key ‘9’ – 0x43 for ‘dot’ key – 0x59 for ‘yellow’ key 0x47 for ‘green’ key 0x55 for ‘up’ key 0x44 for ‘down” key		

Compatibility notes

❗ **Enrolment** operations are no more logged.

❗ **Distant operations** (ILV commands) are no more logged.

Get log status

Description

This command returns the state of the log file.

Command

Request

I	CMD_GET_LOG_STATUS [0x5D]	1 byte
L	1	2 bytes
V	Log Base ID	1 byte

Log Base ID

Base number. Must be 0. (RFU)

Reply

I	CMD_GET_LOG_STATUS [0x5D]	1 byte
L	1 (+ 9)	2 bytes
V	Request Status	1 byte
	Log Enabled	1 byte
	Current Line	2 bytes
	Max Number of Lines	2 bytes
	Line Size	2 bytes
	Base State	1 byte
	Version	1 byte

Request Status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.

The following parameters are returned only if *Request Status* is ILV_OK.

Log Enabled (if Request status is ILV_OK)	
0	Biometric events are not logged.
1	Biometric events are logged.

Current Line (if Request status is ILV_OK)

First free record index.

Max Number of Lines (if Request status is ILV_OK)

Number of records.

Line Size (if Request status is ILV_OK)

Record size in bytes.

Line Size (if Request status is ILV_OK)

Record size in bytes.

Base State (if Request status is ILV_OK)	
0	Base is corrupted.
1	Base is OK.

Version (if Request status is ILV_OK)

Log files revision. 0xXY for « X.Y ».

Erase log

Description

This command erases all the records written in the log file.

Command

Request

I	CMD_ERASE_LOG [0x5C]	1 byte
L	1	2 bytes
V	Log Base ID	1 byte

Log Base ID
Base number. Must be 0. (RFU)

Reply

I	CMD_ERASE_LOG [0x5C]	1 byte
L	1 (+ 1)	2 bytes
V	Request status	1 byte
	Erase status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.

The following parameters are returned only if *Request Status* is ILV_OK.

Erase status (if Request status is ILV_OK)	
ILVSTS_OK [0x00]	Base successfully erased.

Get log

Description

Use this command to download the log database.

Command

Request

I	CMD_GET_LOG [0x5B]	1 byte
L	5	2 bytes
V	Log Base ID	1 byte
	Offset from current line	2 bytes
	Number of lines	2 bytes

Line 1: 64 bytes.	
Line 2: 64 bytes.	
…	
[First downloaded line	*Number Of Lines* **Downloaded lines**
…	
Last downloaded line]	
…	*Offset From Current Line*
Line k-1: 64 bytes.	
Line k: 64 bytes.	
…	
Line n: 64 bytes.	

Log Base ID
Base number. Must be 0. (RFU)

Offset from current line
Offset from the last written line. Represents the line in the base from where the upload starts. “0” for the first line.

Number of lines
This parameter is the number of line to upload from *“Offset from current line”*. This parameter must be ranged between 1 and 8000. To retrieve more than 8000 log entries, you should send several CMD_GET_LOG.

Reply

I	CMD_GET_LOG [0x5B]	1 byte
L	1+ 2 + L	2 bytes
V	Request status	1 byte
	Number of lines	2 bytes
	Data	L bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

The following parameters are returned only if *Request Status* is ILV_OK.

Number of lines

Number of lines really uploaded. This number is equal to 0 in case of error.

Data

Lines in binary format.

Contactless Card Operations

Description

That command allows the distant host to perform several contactless operations using the terminal:

- Read the serial number of a contactless smartcard,
- Read a supported contactless smartcard,
- Encode a supported contactless smartcard,
- Erase a supported contactless smartcard.

Command

Request

I	ILV_CLS_TREATMENT (0x5F)	1 byte
L	2+L1 (+...+Li)	2 bytes
V	Timeout	2 bytes
	SUB_COMMAND #1 Request	L1 bytes
	...	...
	SUB_COMMAND #i Request (optional)	Li bytes

Timeout

Maximum delay in seconds for each command. Special value 0 means no timeout.

SUB_COMMAND

Sub ILV that contains a specific operation to be performed. The available commands are:

- SUB_ILV_CLS_GET_CARD_INFO (0x03) to retrieve card information,
- SUB_ILV_CLS_READ (0x01) to read a card,
- SUB_ILV_CLS_WRITE (0x00) to write a card,
- SUB_ILV_CLS_ERASE (0x02) to erase a card.

Note

Each sub-command is executed whether the previous sub-command has succeeded or not. Using several sub-commands in a single ILV is functionally identical to calling each sub-command in one separate ILV.

Reply

I	ILV_CLS_TREATMENT (0x5F)	1 byte
L	1 (+L1+...+Li)	2 bytes
V	Global return status	1 byte
	SUB_COMMAND #1 Reply (optional)	L1 bytes
	...	...
	SUB_COMMAND #i Reply (optional)	Li bytes

Global return status

ILV_OK (0x00): All sub-commands have succeeded.

Note If no sub-command is specified, the global return status will be ILV_OK. You may use this behaviour to check that this ILV is supported by a terminal.

ILVERR_ERROR (0xFF): At least one of the sub-commands has failed. You should check the status of each sub-command to find out what happened.

ILVERR_BADPARAMETER (0xFE): The ILV is incorrect and no sub-command has been called.

ILVERR_OPERATION_NOT_SUPPORTED (0xC7): The terminal is not equipped with a contactless card reader.

SUB_COMMAND Reply

It contains a sub ILV command reply.

Note

If a sub-command is unknown, the global return status will be ILVERR_ERROR (0xFF), and the sub-command return status will be ILVERR_NOT_IMPLEMENTED (0x9D).

Get Card Information

Description

That command allows the distant host to retrieve some information about a contactless card.

It activates the contactless reader and waits for either a contactless card is detected, either the timeout defined in ILV_CLS_TREATMENT occurs. If a contactless card is detected, its serial number is written into the ILV reply.

Request

I	SUB_ILV_CLS_GET_CARD_INFO (0x03)	1 byte
L	0	2 bytes

Reply (in case of success)

I	SUB_ILV_CLS_GET_CARD_INFO (0x03)	1 byte
L	1+L1+L2	2 bytes
V	Return status = ILV_OK	1 byte
	Card type	L1 bytes
	Serial number	L2 bytes

Return status
Sub-command return status ILV_OK (0x00) : A card has been detected, and its serial number is present in the ILV reply

Reply (in case of failure)

Card type

ILV-formatted data (ID_CARD_TYPE) containing the card type.

I	ID_CARD_TYPE (0x01)	1 byte
L	2	2 bytes
V	Card type	2 bytes

Card type

0x0000	ISO 14443-3
0x0100	ISO 14443-4 type A
0x0101	ISO 14443-4 type B

Serial number

ILV-formatted data (ID_CARD_SN) containing the card serial number.

I	ID_CARD_SN (0x02)	1 byte
L	L	2 bytes
V	Card SN	L bytes

Card SN

Card serial number in byte array format.

I	SUB_ILV_CLS_GET_CARD_INFO (0x03)	1 byte
L	5	2 bytes
V	Return status	1 byte
	Error code (optional)	4 bytes

Return status

Sub-command return status

ILVERR_TIMEOUT (0xFA): No card has been detected in the field.

ILVERR_BADPARAMETER (0xFE): The sub ILV command is incorrect.

ILVERR_NOT_IMPLEMENTED (0x9D): The sub ILV command is not implemented by the terminal.

Error code

This optional field may indicate an internal error code.

Contactless Read

Description

That command reads data on a contactless smartcard according to a given list of profile.

Request

I	SUB_ILV_CLS_READ (0x01)	1 byte
L	L1 (+L2+...+Li)	2 bytes
V	Contactless profile #1	L1 bytes
	Contactless profile #2 (optional)	L2 bytes
	...	...
	Contactless profile #i (optional)	Li bytes

Contactless profile

ILV-formatted parameter that specifies a contactless card profile to use during the read process:

- ID_CLS_PRF_MIF (0x02)
- ID_CLS_PRF_DESF (0x03)

The terminal will try to read a card with every profile in the list, and the ILV will return as soon as a card has been read using one profile.

Reply

I	SUB_ILV_CLS_READ (0x01)	1 byte
L	1+L1 (+L2+...+Li)	2 bytes
V	Return status	1 byte
	Contactless profile reply #1	L1 bytes
	Contactless profile reply #2 (optional)	L2 bytes

	...	...
	Contactless profile reply #i (optional)	Li bytes

Return status

Sub-command return status

ILV_OK (0x00): A card has been read successfully.

ILVERR_TIMEOUT (0xFA): No card has been detected in the field.

ILVERR_BADPARAMETER (0xFE): The sub ILV command is incorrect.

ILVERR_NOT_IMPLEMENTED (0x9D): The sub ILV command is not implemented by the terminal.

Contactless profile reply

Profile read reply, only if r**eturn status** is ILV_OK (0x00).

Contactless Write

Description

That command encodes data on a contactless smartcard according to one or more given profiles.

Request

I	SUB_ILV_CLS_WRITE (0x00)	1 byte
L	L1 (+L2+...+Li)	2 bytes
V	Contactless profile #1	L1 bytes
	Contactless profile #2 (optional)	L2 bytes
	...	...
	Contactless profile #i (optional)	Li bytes

Contactless profile

ILV-formatted parameter that specifies a contactless card profile to use during the write process. The terminal will try to write a card with every profile in the list, and the ILV will return as soon as a card has been written using one profile.

Reply

I	SUB_ILV_CLS_WRITE (0x00)	1 byte
L	1+L1 (+L2+...+Li)	2 bytes
V	Return status	1 byte
	Contactless profile reply #1	L1 bytes
	Contactless profile reply #2 (optional)	L2 bytes
	...	...
	Contactless profile reply #i (optional)	Li bytes

Return status

Sub-command return status

ILV_OK (0x00): A card has been successfully written.

ILVERR_TIMEOUT (0xFA): No card has been detected in the field.

ILVERR_BADPARAMETER (0xFE): The sub ILV command is incorrect.

ILVERR_NOT_IMPLEMENTED (0x9D): The sub ILV command is not implemented by the terminal.

Contactless profile reply

Only if **Return status** is ILV_OK (0x00).

Contactless Erase

Description

That command erases data on a contactless smartcard according to one or more given profiles.

Request

I	SUB_ILV_CLS_ERASE (0x02)	1 byte
L	L1 (+L2+...+Li)	2 bytes
V	Contactless profile #1	L1 bytes
	Contactless profile #2 (optional)	L2 bytes
	...	...
	Contactless profile #i (optional)	Li bytes

Contactless profile

ILV-formatted parameter that specifies a contactless card profile to use during the erase process. The terminal will try to erase a card with every profile in the list, and the ILV will return as soon as a card has been erased using one profile.

Reply

I	SUB_ILV_CLS_ERASE (0x02)	1 byte
L	1+L1 (+L2+...+Li)	2 bytes
V	Return status	1 byte
	Contactless profile reply #1	L1 bytes
	Contactless profile reply #2 (optional)	L2 bytes
	...	...
	Contactless profile reply #i (optional)	Li bytes

Return status

ILV_OK (0x00): A card has been successfully erased.

ILVERR_TIMEOUT (0xFA): No card has been detected in the field.

ILVERR_BADPARAMETER (0xFE): The sub ILV command is incorrect.

ILVERR_NOT_IMPLEMENTED (0x9D): The sub ILV command is not implemented by the terminal.

Contactless profile reply

Only if **Return status** is ILV_OK.

Access Control Functions Description

Overview

This set of function allows controlling the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal Man Machine Interface and other peripherals.

Access Authorization

Description

This function is a way to open access or to refuse it.

Command

Request

I	CMD_ACCESS_AUTHORIZATION [0x56]	1 byte
L	1 (+ 2)	2 bytes
V	Access Authorization	1 byte
	Relay Duration (optional)	2 bytes

Access Authorization
0x00: the message of "access refused" is displayed; Played audio of access refused. 0x01: access granted; the message of "access granted" is displayed, Played audio of "access granted" and the relay switches. The duration is defined in the relay parameters. 0x02: turns the terminal into an emergency state; the access is granted and kept granted indefinitely. On receiving of the same request containing the parameter 0x01, the terminal leaves the emergency state then executes the following action: opens the access.

Relay Duration
This optional parameter defines the relay activation in 100ms. 0 means the relay is never activated. The function returns immediately. This parameter is ignored if *Access Authorization* is set to 2.

Reply

I	CMD_ACCESS_AUTHORIZATION [0x56]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Compatibility note

This command controls the led and the buzzer.

Display Screen [MA500 Series only, PROXY Mode Only]

Description

This function replaces the "PRINT SCREEN MESSAGE" command available on MorphoAccess® 200 and 300 Series.

This function allows displaying a text message in the terminal screen while in Proxy Mode (none local access control mode allowed).

If the request is valid, the terminal returns immediately the reply, without expecting the end of the display time-out: it means that the host application must handle its own timer, to avoid unwanted overwrite of the current message.

Command

Request

I	CMD_DISPLAY_SCREEN [0x54]	1 byte
L	3 + L	2 bytes
V	X position	1 byte
	Y position	1 byte
	Timeout	1 byte
	Message	L bytes

X position: horizontal position of the 1st character of the message (in number of pixels)

The value range is 0 (left side of the screen) to 136.

Y position: vertical position of the 1st character of the message (in number of pixels)

The value range is 0 (up side of the screen) to 34.

Timeout: duration of the message display. At the end, the terminal restores the default access control mode message, i.e. “proxy mode”.

The value range is 0 (up side of the screen) to 255 seconds. When set to 0, the timeout process is disabled: the message is displayed until a command requires another display (i.e. Access Authorization, Identify, Verify, Print Screen,).

The field value must be set to zero. Other values are not yet supported.

Message: text to be displayed on the screen

L (L < 91) ASCII characters (the end-of-string character 0x00 is not required).

If the message is too long for one line, the terminal automatically displays the extra characters on the next lines.

Reply

I	CMD_DISPLAY_SCREEN [0x54]	1 byte
L	1 [0x0001]	2 bytes
V	Request Status	1 byte

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Read from keyboard [Terminal with DISPLAY and keyboard]

Description

This function allows displaying 1 line, and an input zone. The function returns after validation or after a given timeout.

Command

Request

I	READ FROM KEYBOARD [0x55]	1 byte
L	2 + L	2 bytes
V	Secret mode	1 byte
	Time of input	1 byte
	Prompt	L byte

Secret mode

Not yet implemented. Must be set to 0x01. By default it is configured in secret mode.

Timeout

Defines the time in seconds during which the user is invited to enter data. This value must be set between 1 second and 255 seconds.

Prompt

This string defines the input prompt printed on the screen. This parameter is mandatory and the message length cannot exceed 29 characters.

Reply

I	READ FROM KEYBOARD [0x55]	1 byte
L	1 + L	2 bytes
V	Request Status	1 byte
	Input content	L bytes

Request status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

The following parameters are returned only if *Request Status* is ILV_OK.

Input content

The text entered in the input zone, NULL terminated. For now, the input cannot exceed 20 bytes.

Data Send

Description

This function performs a write on Wiegand, Dataclock or Serial output.

Command

Request

I	CMD_DATA_SEND [0x51]	1 byte
L	L	2 bytes
V	Data To Send On Selected Output	L bytes

Data To Send On Selected Output

Wiegand Data, *Dataclock Data*, are data to send on the associated outputs.

Wiegand Raw Data, *Dataclock Raw Data* are data to send on the associated outputs [IDEMIA only, not support is given on this function].

These data are sub ILVs described in chapter *Wiegand Dataclock Data Description*.

Only one of these sub ILVs can be used at the same time.

The following outputs will be defined later:

- Serial Data,
- Serial Raw Data.

Note

The host application must configure the terminal to use the convenient protocol.

Reply

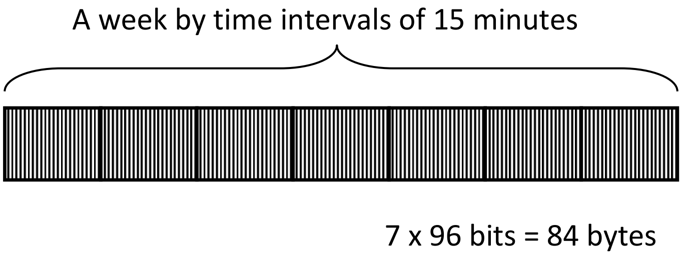
I	CMD_DATA_SEND [0x51]	1 byte
L	1	2 bytes
V	Request Status	1 byte

Request Status	
ILV_OK [0x00]	The execution succeeded.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Time Mask Functions Description

The MorphoAccess® is able to grant access to a user according to its time mask.

A user time mask is a bit array as follows:



Time mask structure

The week is starting on Sunday as described in ANSI standards.

How to use time mask feature?

During base creation, a “time mask” additional field must be created.

When adding a record or enrolling a user this field must be filled with the corresponding data.

Set Time Mask

Description

This function allows setting a user time mask.

Command

Request

I	CMD_SET_TIME_MASK [0x42]	1 byte
L	88 + L	2 bytes
V	Base Id	1 byte
	User ID	L bytes
	Time mask	87 bytes

Base Id

ID of the database in which the user is registered.

User ID

This is the user ID whose time mask is set.

See the User ID structure for details.

Time mask

This is the time mask to set.

See the Time mask structure for details.

Reply (new syntax)

I	CMD_SET_TIME_MASK [0x42]	1 byte
L	1	2 byte
V	Request Status	1 byte

Request status	
ILV_OK [0x00]	The execution of the function succeeded.
ILVERR_ERROR [0xFF]	An error occurred during the execution of the function.
ILVERR_BADPARAMETER [0xFE]	ILV Format incorrect.
Other error codes	Refer to Biometric error code table

Get Time Mask

Description

This function allows getting a user time mask.

Command

Request

I	CMD_GET_TIME_MASK [0x43]	1 byte
L	1 + L	2 bytes
V	Base Id	1 byte
	User ID	L bytes

Base Id

ID of the database in which the user is registered.

User ID

This is the user ID whose time mask is set.

See the User ID structure for details.

Reply (new syntax)

I	CMD_GET_TIME_MASK [0x43]	1 byte
L	88	2 bytes
V	Request Status	1 byte
	Time mask	87 bytes

Request status

ILV_OK [0x00]	The execution of the function succeeded.
ILVERR_ERROR [0xFF]	An error occurred during the execution of the function.
ILVERR_BADPARAMETER [0xFE]	ILV Format incorrect.
Other error codes	Refer to Biometric error code table

The following parameters are returned only if *Request Status* is ILV_OK.

Time mask

This is the time mask to set.

See the Time mask structure for details.

Biometric Data Description

Biometric coder selection

Description

This ILV is optional. It can be used to select the type of biometric coder process suited to your application. The best coder strongly depends on the population you plan to enrol using the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal.

Data structure

I	ID_CODER [0X43]	1 byte
L	0x0004	2 bytes
V	Coder Algorithm Choice	4 bytes

Coder Algorithm Choice

Default value: 0 (standard process: best for average to large fingers).

Enables juvenile mode: 8 (improve thin finger acquisition).

Note

This provides a solution with reliable performance to the automatic processing of finger images with small to normal ridges, extending the capability of the terminal by providing access control to a wider range of population.

Since the encoding time is longer when the juvenile option is turned on, we offer the possibility to the user to set up the terminal either in "Juvenile" mode or in standard mode.

Except a better ability to encode small fingerprints (and a longer encoding time), there is no difference with standard function. It is recommended to use this function:

- in ENROL command, when a small finger encoded with standard function fails matching,
- in IDENTIFY command, when only small fingers are placed on the sensor,
- in VERIFY command, when only small fingers are placed on the sensor.

Biometric Algorithm Parameter [0x38]

Description

When used in ENROL command, these optional ILV formatted data specify the format of the biometric templates to be output: default template format is PK_COMP V2.

Data structure

I	ID_BIO_ALGO_PARAM [0x38]	1 byte
L	0x0001	2 bytes
V	Template format selection	1 byte

Template format selection:

0	PK_COMP format [0x00], default value
2	PK_MAT format [0x02]
65	ANSI 378 format [0x41] : ISO_PK_DATA_ANSI_378
108	ISO FMC CS format [0x6C] : ISO_PK_DATA_ISO_FMC_CS
109	ISO FMC NS format [0x6D] : ISO_PK_DATA_ISO_FMC_NS
110	ISO FMR format [0x6E] : ISO_PK_DATA_ISO_FMR
111	MINEX A format [0x6F] : ISO_PK_DATA_MINEX_A

See also *Enrol* ILV command.

Compatibility Note

The Template Formats “PK_COMP_NORM” and “PK_COMP_MAT_NORM” are not supported in MA SIGMA running in Legacy Morpho (500 series) mode.

ISO and ANSI template formats

Description

The terminal supports several template formats defined by the ISO and by the ANSI standardization organizations:

- ANSI INCITS 378-2004 template format,
- MINEX A Specification (Restricted ANSI INCITS 378-2004 template format),
- ISO/IEC 19794-2 Finger Minutiae Record,
- ISO/IEC 19794-2 Finger Minutiae Card Record, Normal Size,
- ISO/IEC 19794-2 Finger Minutiae Card Record, Compact Size.

The terminal generates ISO and ANSI templates with only one fingerprint inside, but is able to manage ISO/ANSI templates with several fingerprints inside, generated by other terminals.

ISO_PK [0x3F]

This ILV formatted data are used to specify an ISO or an ANSI formatted template as an input parameter. It includes the template itself, and the fingerprint selection (one or all of the fingerprints included in the ISO or ANSI formatted template).

I	ID_ISO_PK [0x3F]		1 byte
L	5 + L		2 bytes
V	Data		5+ L bytes
	ILV	ISO_PK_PARAM	5 bytes
	ILV	ISO_PK_DATA_x	L bytes

ISO_PK_PARAM: ILV formatted data that select the fingerprint to be used (refer to ISO_PK_PARAM ILV description).

ISO_PK_DATA_x: ILV formatted data that contain an ISO or an ANSI formatted template such as:

- ISO_PK_DATA_ANSI_378,
- ISO_PK_DATA_MINEX_A,

- ISO_PK_DATA_ISO_FMR,
- ISO_PK_DATA_ISO_FMC_NS,
- ISO_PK_DATA_ISO_FMC_CS.

See also *Verify* and *Add Base Record*.

ISO_PK_PARAM [0x40]

These ILV formatted data are used, inside ISO_PK ILV data, to specify which fingerprint of an ISO or of an ANSI formatted template has to be used. This is useful as several ISO and ANSI formatted templates can include several fingerprints.

I	ID_ISO_PK_PARAM [0x40]	1 byte
L	0x0002	2 bytes
V	Finger selection (select only one fingerprint in the template file)	1 byte
	Select all fingers (selects all fingerprints found in the template file)	1 byte

Finger Selection: this field is used to select only one of the fingerprints included in an ISO or in an ANSI formatted template file. The value 0 selects the first fingerprint, 1 selects the 2nd fingerprint, and so on…. The value in this field is ignored when the "**Select all fingers**" field is set to 0x01.

Select all fingers:

- set to 1 (0x01) to use all the templates stored in the ISO or ANSI formatted template,
- set to 0 (0x00) to use only the template selected by the Finger Selection field.

ISO_PK_DATA_ANSI 378 [0x41]

ILV formatted data that contain an ANSI INCITS 378 formatted template.

I	ID_ISO_PK_DATA_ANSI_378 [0x41]	1 byte
L	Lx	2 bytes
V	ANSI 378 template	Lx bytes

ANSI 378 template: biometric template that complies with the Finger Minutiae Record format described in ANSI INCITS 378 standard.

Note:

This template format can contains more than one fingerprint per template, but the template generated by the terminal includes only one fingerprint. However, the terminal is able to accept, as input parameter, templates with several fingerprints inside, see ISO_PK and ISO_PK_PARAM ILV description.

See also *Verify*, *Enrol*, *Add Base Record*.

ISO_PK_DATA_MINEX_A [0x6F]

ILV formatted data that contain a MINEX_A formatted template.

I	ID_ISO_PK_DATA_MINEX_A [0x6F]	1 byte
L	Lx	2 bytes
V	MINEX A template	Lx bytes

MINEX A template: biometric template that complies with the Finger Minutiae Record format described in Minutiae Interoperability Exchange Test 2004 (MINEX04 API) document.

Note: this template format is a based on the ANSI INCITS 378 template format, but with only one fingerprint.

See also *Verify*, *Enrol*, *Add Base Record*.

ISO_PK_DATA_ISO_FMR [0x6E]

ILV formatted data that contain an ISO/IEC 19794-2 chapter 7 formatted templates.

I	ID_ISO_PK_DATA_ISO_FMR [0x6E]	1 byte
L	Lx	2 bytes
V	ISO FMR template	Lx bytes

ISO FMR template: biometric template that complies with the Finger Minutiae Record format described in chapter 7 of ISO/IEC 19794-2 standard.

Note: this template format can contains more than one fingerprint per template, but the template generated by the terminal includes only one fingerprint. However, the terminal is able to accept, as input parameter, templates with several fingerprints inside, see ISO_PK and ISO_PK_PARAM ILV description.

See also *Verify*, *Enrol*, *Add Base Record*.

ISO_PK_DATA_ISO_FMC_NS ID = 0X6D

ILV formatted data that contain an ISO/IEC 19794-2 chapter 8.1 formatted templates.

I	ID_ISO_PK_DATA_FMC_NS [0x6D]	1 byte
L	Lx	2 bytes
V	ISO FMC NS template	Lx bytes

ISO FMC NS template: biometric template that complies with the Finger Minutiae Card Record, Normal Size format described in chapter 8.1 of ISO/IEC 19794-2 standard.

See also *Verify*, *Enrol*, *Add Base Record*.

ISO_PK_DATA_ISO_FMC_CS ID = 0X6C

ILV formatted data that contain an ISO/IEC 19794-2 chapter 8.2 formatted templates.

I	ID_ISO_PK_DATA_FMC_CS [0x6C]	1 byte
L	Lx	2 bytes
V	ISO FMC CS template	Lx bytes

ISO FMC CS template: biometric template that complies with the Finger Minutiae Card Record, Compact Size format described in chapter 8.2 of ISO/IEC 19794-2 standard.

See also *Verify*, *Enrol*, *Add Base Record*.

Security Level

Description

This ILV formatted parameter is available for all biometric commands with finger acquisition to set the security level. The offered security levels depend on the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal and on the biometric command.

Data structure

I	ID_FFD_LEVEL [0X42]	1 byte
L	0x0004	2 bytes
V	Security level	4 bytes

Parameters

Security level:

	Enrolment	Authentication/Identification
Multimodal finger vein / fingerprint terminals	Useless	**0x00000000**: Standard **0x00000200**: Medium **0x00000100**: High
FFD terminals	Useless	**0x00000000**: Low **0x00000001**: Medium **0x00000002**: High
Other terminals	Useless	Useless

Note

By default, the security level is always set to the lowest value.

Depending on product variant configuration, the “STANDARD” value may be not available. In that case the default value is “MEDIUM”. See also*Verify*, *Enrol*, *Identify*.

Biometric Presence Detection Mode

Description

This ILV formatted data is used to select the presence detection mode suited to the application. It is used to fine-tune the system, especially if the user faces problems with dry fingers.

Data structure

I	ID_MODE_DETECT [0X44]	1 byte
L	0x0004	2 bytes
V	Detection Mode	4 bytes

Parameters

Detection Mode: Bitmask of the following:

- **0x00000000**: (MORPHO_DEFAULT_DETECT_MODE) The detection mode is set by the MorphoAccess® itself, which amounts not to sending the ID_MODE_DETECT optional parameter.
- **0x00000001**: (MORPHO_VERIF_DETECT_MODE) More permissive detection mode. In this mode, the MorphoAccess® detects more easily finger presence, but might issue lower quality templates.
- **0x00000002**: (MORPHO_ENROLL_DETECT_MODE) Strongest detection mode.
- **0x00000004**: (MORPHO_WAKEUP_LED_OFF) Impedance detection mode, only available on FFD terminals. In this mode, LED is turned off while waiting for a finger.
- **0x00000010**: (MORPHO_FORCE_FINGER_ON_TOP_DETECT_MODE) Force the finger to cover an area starting at the top of the sensor window.
- **0x00000040**: (MORPHO_ WAKEUP_LED_ON) only available on multimodal finger vein / fingerprint terminals. In this mode, LED is turned on while waiting for a finger.

Note

MORPHO_VERIF_DETECT_MODE and MORPHO_ENROLL_DETECT_MODE cannot be used together.

MORPHO_FORCE_FINGER_ON_TOP_DETECT_MODE can only be used with MORPHO_ENROLL_DETECT_MODE.

See also*Verify*, *Enrol*, *Identify*.

Database Description

User Index

Description

ILV formatted data that contain the unique internal database index for a record. It is managed by the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal database. The value is returned by ENROL or ADD_BASE_RECORD.

Data structure

I	ID_USER_INDEX [0X36]	1 byte
L	4	2 bytes
V	Index	4 bytes

Note

See also: *Remove Record*, *Get Data*, *and Update Public Data*.

User ID

Description

ILV formatted data that contain the User ID saved into the database. The size of the data User ID is 24 bytes at the most.

The User ID must be unique in the database. **This field is managed as a byte array**. If you need to use string, do not forget to manage the ending '\0'.

For an enrolment, the User ID field can be automatically filled with the user database index if the length is set to 0.

'User ID' can be retrieved by the *Identify* function when a Hit (match) occurs.

Data structure

I	ID_USER_ID [0x04]	1 byte
L	L	2 bytes
V	ID User	L bytes

Note

See also: *Enrol*, *Identify*, *Add Base Record*, *Remove Record, Set time mask, Get time mask*.

PKBase

Description

ILV formatted data that contain the database identifier and the record identifier to match with.

Data structure

I	PKBASE [0x3A]	1 byte
L	1+ L_{UID}	2 bytes
V	Database identifier	1 byte
	ILV User ID or User Index	L_{UID}

Database identifier
Number of available databases depends on terminal type and license. Value of this command is ‘0’, as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

Note

See also *Verify*.

Base Configuration

Description

This ILV contains the database configuration.

Data structure

I	ID_BASE_CONF [0x1F]	1 byte
L	19	2 bytes
V	Database Identifier	1 byte
	Base Valid	1 byte
	NB Finger/person	1 byte
	Max Record Number	4 bytes
	Current Record Number	4 bytes
	Free Record Number	4 bytes
	Fields Number	4 bytes

Database Identifier

Number of available databases depends on terminal type and license. Value of this command is '0', as in MorphoAccess® SIGMA Family & MorphoWave® Compact terminal; there is only ONE database present.

Base Valid

This flag is set to 1 if the base is valid.

Number Finger/Person

Number of fingers saved per person.

Max Record Number

Maximum number of records in database.

Current Record Number

Number of records currently saved in Database.

Free Record Number

Number of records currently available in Database.

Fields Number

Number of fields used in Database.

Additional Data Field Description

Field Content

Description

ILV formatted data packet that contains field content.

Data structure

I	ID_FIELD_CONTENT [0x32]	1 byte
L	4 + 4 + L	2 bytes
V	Data Index	4 bytes
	Data Length	4 bytes
	Data	L bytes

Data Index

Field index that is consistent with database structure.

Data Length

Field data size.

Data

Buffer of field data.

Note

See also: *Get Public Field*, Update Public Data.

Public field

Description

ILV formatted data packet that contains the public field structure definition.

Data structure

I	ID_PUBLIC_FIELD or ID_FIELD [0x0F]	1 byte
L	2 + L	2 bytes
V	Field Size	2 bytes
	Field Name	L bytes

Field Size

Defines the maximum size (in bytes) of a record. It cannot exceed 128 bytes.

Field Name

String specifying the field name. The size of this string must be equal to 6.

Note

The size of the public field has not to exceed 32 bytes when the number of database records is superior to 100.

See also: *Create Database*, *Base Config*.

Private Field

Description

ILV formatted data packet that contains the private field structure definition.

Data structure

I	ID_PRIVATE_FIELD [0x31]	1 byte
L	2 + L	2 bytes
V	Field Size	2 bytes
	Field Name	L bytes

Field Size

Defines the maximum size (in bytes) of a record. It cannot exceed 256 bytes.

Field Name

String specifying the field name. The size of this string must be equal to 6 including NULL ('\0').

See also: *Create Database*, *Get Base Config*.

Additional User Data

Description

ILV formatted data containing personal user data to be saved into the database if the Save Record value is set to TRUE. The content of the data is not interpreted by the system. The user must define the content of the buffer. All the additional user data fields defined in the *Create Database* command have to be set in the *Enrol* command. Size and order of additional data fields must be consistent with database structure.

Data structure

I	ID_PUC_DATA [0x14]	1 byte
L	L	2 bytes
V	User Data	L bytes

User Data

User Data: Buffer containing the data of the database field. The size of the data can be up to 256 bytes.

If a field is empty, its length is 0.

Note

The entire 'Additional User Data field' defined in *Create Database* have to be set in the *Add Base Record* command.

The number of fingers per person must be the same as defined by *Create Database*.

'User ID' can be retrieved by the *Identify* function when a Hit (match) occurs.

Size and order of additional data fields must be consistent with database structure.

See also: *Enrol*, *Identify*, *Add Base Record*, *Create Database*.

Compatibility note

ID_C_DATA, ID_S_DATA, ID_L_DATA identifiers do not exist anymore.

No Check on Template

Description

ILV formatted data used with ADD_BASE_RECORD to suppress checks on templates.

Data structure

I	ID_NO_CHECK_ON_TEMPLATE [0x60]	1 byte
L	1	2 bytes
V	Value	1 byte

Value

If this ILV is present, the value must be set to 0x01 to perform add record without checks on templates, 0x00 otherwise.

Other values: bad parameter.

Note

See also: *Add Record*.

Time Mask

Description

This is the time mask additional field.

Data structure

I	ID_TMSK [0x24]	1 byte
L	84	2 bytes
V	Time mask	84 bytes

Time mask

Time mask field.

Note

See also: *Set time mask* and *Get time mask*.

Wiegand Data Description

Wiegand data

Description

This sub ILV describes the data to send on Wiegand output. Data are defined in a high level format; data sent on Wiegand port depend on the Wiegand configuration stored in the terminal.

Data structure

I	ID_WIEGAND_DATA [0x11]	1 byte
L	12	2 bytes
V	Site	4 bytes
	ID	4 bytes
	Custom	4 bytes

Note

Site, ID and Custom data are *little endian*. Data type is hexadecimal.

Wiegand raw data [IDEMIA Only, no support is given on this function]

Description

This sub ILV describes the data to send on Wiegand output. Data are defined in a raw format; data sent on Wiegand port are exactly the same as data described in this ILV, without any encapsulation.

Data structure

I	ID_WIEGAND_RAW_DATA [0x10]	1 byte
L	4 + N	2 bytes
V	Number of bits in raw data buffer	4 bytes
	Raw data buffer	N bytes

Note

Example, for standard Wiegand 26 bits, number of bits in raw data buffer would be 26 and raw data buffer would be 4 bytes length, with bits organized as follows:

X	X	X	X	X	X	E	Site code (8 bits)	User ID (16 bits)	O

raw_buffer[0] raw_buffer[1] raw_buffer[2] raw_buffer[3]

Clock & Data Protocol Data Description

Clock & Data

Description

This sub ILV describes the data to send on Clock & Data output. Data are encapsulated according to ISO2 format. In ISO2 format only numerical characters are allowed (0-9).

Data structure

I	ID_DATACLOCK_DATA [0x21]	1 byte
L	L	2 bytes
V	ID	L bytes

Note

ID is a null terminated ASCII string. Null character is not sent on Clock & Data output.

For now, ID’s length is limited to 32 bytes.

Clock & Data raw data [IDEMIA Only, no support is given on this function]

Description

This sub ILV describes the data to send on Clock & Data output. Data are defined in a raw format; data sent on Clock & Data port are exactly the same as data described in this ILV, without any encapsulation.

Data structure

I	ID_DATACLOCK_RAW_DATA [0x20]	1 byte
L	4 + L	2 bytes
V	Number of bits in raw data buffer	4 bytes
	Raw data buffer	L bytes

Note

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
raw_data_buffer[0]								raw_data_buffer[1]								raw_data_buffer[2]								raw_data_buffer[3]							

The above format is used to send the 26 bits of a 4 bytes raw buffer.

Contactless Data Description

Address Biometric Data

Description

These parameters give addresses containing the reference minutiae of fingerprint. The reference minutiae are compressed with algorithm PKCOMP (with MorphoKit^{TM} for example.)

These addresses define the exact location of the minutiae raw data (So, the PK TAG structure as defined is not supported in this function).

Data structure

I	ID_ADDRPK [0x18]	1 byte
L	3	2 bytes
V	Address	3 bytes

Address

An address of card parameter has this structure:

<Address xxx> is composed of 3 bytes: <B><N><C>.

<B>: First block number to read (1 to 48).

<N>: Number of blocks to read (1 to 48).

<C>: Key number for the read (1or 2).

Blocks are numbered in an absolute way, 1 for block 0 sector 0, then 3 blocks for each sector.

MIFARE® Classic Contactless Profile

Description

ILV-formatted data representing a MIFARE® profile.

Data structure

			Reading	Writing	Erasing
I	ID_CLS_PRF_MIF (0x02)	1 byte	Mandatory	Mandatory	Mandatory
L	L1+L2+L3+LD	2 bytes	Mandatory	Mandatory	Mandatory
V	Start block	L1 bytes	Mandatory	Mandatory	Mandatory
	Blocks number	L2 bytes	Optional	Optional	Optional
	Security policy	L3 bytes	Mandatory	Mandatory	Mandatory
	Data	LD bytes	N/A	Mandatory	N/A

Start block

ILV-formatted data that contains the index of the block where the operation will start from.

I	ID_MIFARE_START_BLOCK (0x0B)	1 byte
L	0x01	2 bytes
V	Block index	1 byte

Block index

Index of the block, starting from 0. Refer to MorphoAccess® Contactle Specification.

Blocks number

ILV-formatted data that contains the number of blocks.

I	ID_MIFARE_BLOCKS_NUMBER (0x0D)	1 byte
L	0x01	2 bytes
V	Blocks number	1 byte

Blocks number
Number of blocks. Refer to MorphoAccess™ Contactless Card Specification.

Security policy

ILV-formatted data that contains the security policy to use during the contactless operation.

For an encoding operation, that parameter specifies whether to use Keys A or Keys B as the protection key (according to MIFARE® Card IC MF1 IC S50 functional specifications).

For a reading operation, that parameter specifies whether to use Keys A only, or Keys B only, or Keys A then Keys B to read the card.

I	ID_MIFARE_SECURITY_POLICY (0x0C)	1 byte
L	0x01	2 bytes
V	Used keys	1 byte

Used keys
0 means Keys A, 1 means Keys B, 2 means Keys A then Keys B (only for reading operations).

Data

Mandatory for encoding purpose. ILV-formatted data that contains the data buffer to encode.

I	ID_PUC_GENERIC_DATA (0xA5)	1 byte
L	L	2 bytes

V	Data (byte array)	L bytes

Reading

When this profile is used inside SUB_ILV_CLS_READ, it instructs the terminal to read the given number of blocks at given offset, with given security policy.

Note

If **Blocks number** is not specified, the maximum number of blocks will be read.

Writing

When this profile is used inside SUB_ILV_CLS_WRITE, it instructs the terminal to write the given data at given offset in the card, with given security policy.

Note

If **Blocks number** is not specified, it will be automatically determined according to **Data** size. If **Blocks number** is specified but does not correspond to **Data** size, the **Data** will be either truncated, either padded with zeros.

Erasing

When this profile is used inside SUB_ILV_CLS_ERASE, it instructs the terminal to overwrite the blocks specified by offset and size with zeros. In addition, the NXP® default factory keys (ff…ff) will be restored on these blocks.

Reply (in case of success)

I	ID_CLS_PRF_MIF (0x02)	1 byte
L	1(+LD)	2 bytes
V	Return status = ILV_OK	1 byte
	Data (optional)	LD bytes

Return status

ILV_OK [0x00]: Successful read, write or erase

Data

Only present when reading. ILV-formatted data that contains the read data.

I	ID_PUC_GENERIC_DATA (0xA5)	1 byte
L	L	2 bytes
V	Data (byte array)	L bytes

Reply (in case of failure)

I	ID_CLS_PRF_MIF (0x02)	1 byte
L	5	2 bytes
V	Return status	1 byte
	Error code (optional)	4 bytes

Return status

ILVERR_ERROR[0xff] : Unspecified error

Error code

Internal error code for debugging purpose

See also

SUB_ILV_CLS_READ, SUB_ILV_CLS_WRITE, SUB_ILV_CLS_ERASE.

DESFire® Contactless Profile

ILV-formatted data representing a DESFire® profile.

Parameters

			Reading	Writing	Erasing
I	ID_CLS_PRF_DESF (0x03)	1 byte	mandatory	mandatory	mandatory
L	Value length	2 bytes	mandatory	mandatory	mandatory
V	Special parameters	5 bytes	mandatory	mandatory	mandatory
	Application AID	6 bytes	mandatory	mandatory	optional(1)
	Card Masterkey ID	7 bytes	mandatory	mandatory	mandatory
	Card Key Settings	4 bytes	mandatory	mandatory	N/A
	Application cryptographic algorithm	4 bytes	mandatory	mandatory	N/A
	Application Masterkey ID	7 bytes	mandatory	mandatory	N/A
	Application Key Settings	4 bytes	mandatory	mandatory	N/A
	Access Key ID #1	7 bytes	optional	optional	N/A
	…	…	…	…	…

	Access Key ID #i	7 bytes	optional	optional	N/A
	File description #1	XX bytes	optional(2)	optional(3)	N/A
	…	…	…	…	…
	File description #i	XX bytes	optional	optional	N/A

Note

(1) If Application AID is not specified or equal to 0 during a card erase, all the applications will be erased.

(2) You may specify no file for a reading in order to check application presence.

(3) You may specify no file for writing; an empty application will be created.

Special parameters

ILV-formatted data that contains a bit mask of the special commands being used during the process.

I	ID_SPECIAL_PARAMETERS (0x15)	1 byte
L	2	2 bytes
V	Parameters	k bytes

Parameters

Bits field (16 bits little-endian integer), used to activate special parameters for MIFARE® Classic™ and DESFire® profiles.

bit	Description

0	[DESFire® only][writing only] Do not format the card before writing
1	[DESFire® only][reading only] Do not change the keys on the fly when reading a card
2...3	RFU
4	[DESFire® only][writing only or rotation only] Do not diversify MASTER_PICC key
5...15	RFU

Application AID

ILV-formatted data that contains the DESFire® application AID number.

Warning During a write operation, if an application with the same AID already exists in the card, it will be overwritten. Any existing file inside the application will be lost.

I	**ID_DESFIRE_APP_AID (0x04)**	1 byte
L	**0x03**	2 bytes
V	**Application Identifier**	3 bytes

Application Identifier

An application is identified by an "Application Identifier", which is implemented as a 24 bit number.

Warning Application Identifier 0x00 00 00 is reserved as a reference to the card level (PICC) itself and cannot be used.

Card Masterkey ID

ILV formatted data that contains the SOM identifier (ID_SOM_OBJ_ID) of the master key at card (PICC) level (according to DESFire® MF3ICD84 functional specification).

That key is contained in the terminal.

Note The card master key is the access key ID #0 at card level.

Card Key Settings

ILV-formatted key settings for the Change Key Settings DESFire® function; should be set to 0x0F.

I	ID_DESFIRE_KEY_SETTINGS (0x14)	1 byte
L	0x01	2 bytes
V	Key Settings	1 byte

Cf. Change Key Settings function in DESFire® MF3ICD84 functional specification.

Application cryptographic algorithm

ILV formatted data that contains the cryptographic method of the application (according to DESFire® MF3ICD84 functional specification).

I	ID_DESFIRE_APP_CRYPTO_ALGO (0x11)	1 byte
L	0x01	2 bytes
V	Cryptographic algorithm	1 byte

Cryptographic algorithm

0x02 for 3DES,

0x04 for AES128

Application Masterkey ID

ILV-formatted data that contains the SOM identifier (ID_SOM_OBJ_ID) of the master key at application level (according to DESFire® MF3ICD84 functional specification).

That key is contained in the terminal.

Note The application master key is the access key ID #0 at application level.

Application Key Settings

ILV-formatted key settings for the Change Key Settings DESFire® function; should be set to 0x0B.

I	ID_DESFIRE_KEY_SETTINGS (0x14)	1 byte
L	0x01	2 bytes
V	Key Settings	1 byte

Cf. Change Key Settings function in DESFire® MF3ICD84 functional specification.

Access Key ID #1 to #i

ILV-formatted data that contains the SOM identifier (ID_SOM_OBJ_ID) of each key stored within the application for cryptographic purposes (according to DESFire® MF3ICD84 functional specification).

Those keys are contained in the terminal.

Note A DESFire® application can contain up to 14 access keys.

File description #1 to #i

ILV-formatted data (ID_DESFIRE_APP_FILE_DESC) that contains parameters about a DESFire® application file (File ID, communication settings ...).

Note A DESFire® application can contain up to 32 files.

Reading

When this profile is used inside SUB_ILV_CLS_READ, it instructs the terminal to read one or more DESFire® files.

Writing

When this profile is used inside SUB_ILV_CLS_WRITE, it instructs the terminal to create a new application (or to overwrite the existing one) and to write a list of files.

> **Warning**
>
> A DESFire® profile contains only one DESFire® application.
>
> To proceed with several DESFire® applications, several SUB_ILV_CLS_WRITE have to be used.
> A DESFire® card can contain up to 28 applications.

Erasing

When this profile is used inside SUB_ILV_CLS_ERASE, it instructs the terminal to erase either a specified application, either all the applications. If all applications are erased, the card master key is reset to default DESFire® value.

Reply (in case of success)

I	ID_CLS_PRF_DESF (0x03)	1 byte
L	1(+L1+...+Li)	2 bytes
V	Return status = ILV_OK	1 byte
	Data #1 (optional)	L1 bytes
	...	...
	Data #i (optional)	Li bytes

Return status
ILV_OK [0x00]: Successful read, write or erase.

Data #1 to #3

Only present when reading. ILV-formatted data that contains each requested file.

Reply (in case of failure)

I	ID_CLS_PRF_DESF (0x03)	1 byte
L	5	2 bytes
V	Return status	1 byte
	Error code (optional)	4 bytes

Return status

ILVERR_ERROR[0xff]: Unspecified error.

Error code

Internal error code for debugging purpose.

See also

SUB_ILV_CLS_READ, SUB_ILV_CLS_WRITE, SUB_ILV_CLS_ERASE

DESFire® File description

Description

ILV-formatted data that contains parameters about a DESFire® application file (File ID, communication settings ...).

Data structure

I	ID_DESFIRE_APP_FILE_DESC (0x07)	1 byte
L	4+4+5+LD	2 bytes
V	Identifier	4 bytes
	Communication settings	4 bytes
	Access rights	5 bytes
	Data (optional)	LD bytes

Identifier

ILV-formatted data that contains a DESFire® application file identifier.

I	ID_DESFIRE_APP_FILE_ID (0x08)	1 byte
L	0x01	2 bytes
V	Identifier	1 byte

Identifier

Number between 0x00 and 0x1F.

Communication settings

ILV-formatted data that defines the level of security for the communication with the reader.

I	ID_DESFIRE_APP_FILE_COMMODE (0x09)	1 byte
L	0x01	2 bytes
V	Settings	1 byte

Settings
0x00 or 0x02: Plain communication, 0x01: Plain communication secured by MAC, 0x03: Fully enciphered communication.

Access rights

ILV-formatted data that defines access rights for the specified DESFire® file (according to DESFire® MF3ICD81 functional specification).

I	ID_DESFIRE_APP_FILE_ACCESS_RIGHTS (0x0A)	1 byte
L	0x02	2 bytes
V	Access rights	2 bytes

Access rights
Read Access, Write Access, Read & Write Access, Change Access Rights.

15	12	11	8	7	4	3	0

Read Access	Write Access	Read & Write Access	Change Access Rights
MSbit			LSbit

Data

Only for encoding purpose. ILV-formatted data that contains the data buffer to encode.

I	ID_PUC_GENERIC_DATA (0xA5)	1 byte
L	L	2 bytes
V	Data (byte array)	L bytes

Security Data Description

SOM identifier

Description

ILV-formatted data representing the ID of a SOM managed object, such as cryptographic keys, files, etc.

Data structure

I	ID_SOM_OBJ_ID (0x10)	1 byte
L	0x0004	2 bytes
V	SOM Object identifier	4 bytes

SOM Object identifier

Identifier number of the SOM managed object.

Please refer to the description of the command: ILV_CLS_TREATMENT, ILV_MANAGE_SECURED_OBJECT.

Terminal Version Data Description

Terminal Identifier

Description

These data return terminal characteristics.

Data structure

I	ID_TERMINAL_IDENTIFIER [0x01]	1 byte
L	24	2 bytes
V	Extended Terminal serial number in ASCII	11 bytes
	Plate-form Type: 0x00: PLATFORM_UNKNOWN 0x01: PLATFORM_EFT30_V2 0x02: PLATFORM_EFT30_V3 0x03: PLATFORM_EFT930 0x05: PLATFORM_TWIN 0x06: PLATFORM_UCM 0x07: PLATFORM_RADIO_BASE 0x08: PLATFORM_EFT930S 0x09: PLATFORM_MA2G 0x0C: PLATFORM_MA500	1 byte
	Embedded MACI Version	4 bytes
	ILV receipt buffer size	4 bytes
	ILV sender buffer size	4 bytes

Biometric sensor Identifier

Description

These data return biometric sensor version.

Data structure

I		ID_BIO_SENSOR_IDENTIFIER [0x02]		1 byte
L		$29 + L_1 + L_2 + L_3$		2 bytes
V		Bio Sensor state - 0x01: Bio Sensor detected - 0xFF: Bio Sensor not found. The sub ILV 0x29 / 0x2B / 0x2A are not present.		1 byte
		Product Information ILV: Product information data structure definition.		L_1+ 3 bytes
	I	ID_DESC_PRODUCT [0x29]		
	L	L_1		
	V	String	L_1	
		Sensor Information ILV: Sensor information data structure definition.		L_2 + 3 bytes
	I	ID_DESC_SENSOR [0x2B]		
	L	L_2		
	V	String	L_2	
		Software Information ILV: Software information data structure definition.		L_3 + 3 bytes
	I	ID_DESC_SOFTWARE [0x2A]		
	L	L_3		
	V	String	L_3	
		Software Version ILV: ILV formatted data with software release identification.		10 bytes
	I	ID_FORMAT_BIN_VERSION [0x74]		
	L	7		
	V	Major release number	2 ASCII digits	
		Separator	1 ASCII dot character (0x2E)	
		Minor release number	2 ASCII digits	

Separator	1 ASCII dot character (0x2E)
Internal release letter	1 ASCII alphabetic low case character

Number of Bases. 4 bytes

I	ID_FORMAT_BIN_NB_BASE [0x76]
L	1
V	Number of base available.

Number of users per base. 5 bytes

I	ID_FORMAT_BIN_MAX_USER [0x75]
L	2
V	Maximum number of records allowed

Contactless Features

Description

These data return the contactless reader capabilities.

Data structure

I	ID_CLSS_IDENTIFIER [0x03]	1 byte
L	5	2 bytes
V	Reader type: 0x00: NONE 0x01: MIFARE 0x02: HID	1 byte
	Version	4 bytes

Man Machine Interfaces

Data structure

I	ID_MMI_IDENTIFIER [0x04]	1 byte
L	3	2 bytes
V	Keyboard Type:	1 byte
	0x00: NONE	
	0x10: 16 keys	
	0x12: 18 keys	
	0x18: 24 keys	
	Screen Type:	1 byte
	0x00: NONE	
	0x01: TERM_DISPLAY_2L_ALPHA	
	0x02: TERM_DISPLAY_2L_GRAPHIC	
	0x03: TERM_DISPLAY_4L_GRAPHIC	
	0x04: TERM_DISPLAY_8L_GRAPHIC	
	0x05: TERM_DISPLAY_GRAPHIC_128x64	
	0x06: TERM_DISPLAY_GRAPHIC_128x128	
	Buzzer Type:	1 byte
	0x00: NONE	
	0x01: BUZZER_TYPE1	

Application Information

Description

These data return application information.

Data structure

I	ID_BIO_APPLICATION_INFORMATION [0x05]			1 byte
L	$9 + L_1 + L_2 + L_3$			2 bytes
V	Name			L_1+ 3 bytes
	I	ID_APP_NAME [0x01]		
	L	L_1		
	V	String	L_1	
	Version			L_2 + 3 bytes
	I	ID_APP_VERSION [0x02]		
	L	L_2		
	V	String	L_2	
	Comment			L_3 + 3 bytes
	I	ID_APP_COMMENT [0x03]		
	L	L_3		
	V	String	L_3	

Terminal Description

Description

These data return information about the terminal.

Data structure

I	ID_BIO_TERMINAL_DESCRIPTION [0x06]	1 byte
L	$L_1 + L_2 + L_3$	2 bytes
V	Response	L_1

Response

Null terminated string containing several fields separated by a ';'.

- FW_VERSION= Version of the terminal firmware .The format is XX.YY.ZZ where XX is the major number, YY the minor number and ZZ the release one.
- TX_NAME= Terminal Commercial name (J-Dual for example).
- TX_PN= Terminal Product Number.
- TX_SN= Terminal Serial Number

Example of response:

FW_VERSION=XX.YY.ZZ;TX_NAME=aaa;TX_PN=bbb;TX_SN=sss

Please note that the order of the fields can change from one terminal to another.

Peripherals

Description

This data returns the list of present peripherals.

Data structure

I	ID_BIN_PERIPH [0x07]	1 byte
L	4	2 bytes
V	Response	4 bytes

Response

32 bits value. Bits set to 0 represent the absent peripherals, and bits set to 1 represent the present peripherals.

0	PERIPH_SCREEN_BIT	16	PERIPH_ETHERNET_BIT
1	PERIPH_KEYBOARD_BIT	17	PERIPH_USB_HOST_BIT
2	PERIPH_RC531_BIT	18	PERIPH_USB_SLAVE_BIT
3	PERIPH_ICLASS_BIT	19	PERIPH_CBM_BIT
4	PERIPH_LED_BIT	20	PERIPH_MSO_BIT
5	PERIPH_RELAY_BIT	21	PERIPH_FFD_BIT
6	PERIPH_WIEGAND_OUT_BIT	22	PERIPH_PLATFORM_MA2G_BIT
7	PERIPH_DATACLK_OUT_BIT	23	PERIPH_PLATFORM_MA500_BIT
8	PERIPH_RS485_OUT_BIT	24	PERIPH_FVP_BIT
9	PERIPH_RS422_OUT_BIT	25	Reserved for future use
10	PERIPH_TAMPER_SWITCH_BIT	26	Reserved for future use
11	PERIPH_LEDIN_BIT	27	Reserved for future use
12	PERIPH_LEDOUT_BIT	28	Reserved for future use
13	PERIPH_WIEGAND_IN_BIT	29	Reserved for future use
14	PERIPH_DATACLK_IN_BIT	30	Reserved for future use
15	PERIPH_RS422_IN_BIT	31	Reserved for future use

Configuration Data Description

Date and Time Configuration

Description

This packet configures the time and the date of the system.

Data structure

I	ID_DATE [0x13]	1 byte
L	13	2 bytes
V	Time/Date system	13 bytes

Time/Date system

A string of ASCII characters with Day, Month, Years, Hours, Minutes and Seconds, each encoded with 2 bytes: “DDMMYYHHmmSS”. For example the string “010601120000” indicates the 06/01/01 at 12:00:00 clock.

The string is NULL terminated.

Wi-Fi™ Data Description

Wi-Fi™ Network Information

Description

This packet contains Wi-Fi™ network information.

Data structure

I	ID_WIFI_NETWORK_INFO [0x01]	1 byte
L	(L1+L2)+2+2+1	2 bytes
V	SSID	L1 bytes
	BSSID	L2 bytes
	Frequency	2 bytes
	Signal level	2 bytes
	Encryption Type	1 byte

SSID

See SSID for detail.

BSSID

See BSSID for detail.

Frequency

Frequency level of SSID

Encryption Type

Encryption type of network:-

1. 0 = open network
2. 1 = WEP
3. 2 = WPA Personal
4. 3 = WPA2 Personal

SSID

Description

This packet contains SSID of Wi-Fi™ network.

Data structure

I	ID_SSID [0x02]	1 byte
L	L1	2 bytes
V	Value	L1 bytes

Value

SSID of network

BSSID

Description

This packet contains BSSID of Wi-Fi™ network.

Data structure

I	ID_BSSID [0x03]	1 byte
L	L1	2 bytes
V	Value	L1 bytes

Value

BSSID of network

Encryption key

Description

This packet contains encryption key of Wi-Fi™ network.

Data structure

I	ID_ENCRYPTION_KEY [0x04]	1 byte
L	L1	2 bytes
V	Value	L1 bytes

Value

Value indicates the value of Encryption key of network.

Miscellaneous Data Description

Matching Score

Description

This ILV is optional. It can be used in a request to force the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal to return the matching score. In a reply, it contains the matching score value.

Data structure

I	ID_MATCHING_SCORE [0x56]	1 byte
L	1 or 4	2 bytes
V	Value or Score	1 or 4 bytes

Value or Score

Value: Set to a value different from 0 to force the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal to return the resulting matching score.

Score: Result matching score on 4 bytes.

Note

See also: *Verify*, *Identify*.

Latent Detection

Description

This ILV is optional. It can be used in a request to force the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal to return the matching score. In a reply, it contains the matching score value.

Data structure

I	ID_LATENT_SETTING [0x39]	1 byte
L	1	2 bytes
V	Value	1 byte

Value

0x00: the fingerprint latent detection is disabled.

0x01: the fingerprint latent detection is enabled.

Note

See also: *Enrol*.

Constant Values

Identifier – status

Remote management

Identifier	Description
[0x01] DISTANT_SESSION_OPENED	Distant session opened
[0x02] DISTANT_SESSION_CLOSED	Distant session closed

Error codes table

Error codes	Description
[0xFF] ILVERR_ERROR	An error occurred
[0xFE] ILVERR_BAD_PARAMETER	Input parameter(s) is(are) not valid
[0xFD] ILVERR_INVALID_MINUTIAE	The minutiae is not valid
[0xFC] ILVERR_INVALID_USER_ID	The User ID is not valid.
[0xFB] ILVERR_INVALID_USER_DATA	The user data are not valid
[0xFA] ILVERR_TIMEOUT	No response after defined time
[0xF9] ILVERR_INVALID_ID_PROTOCOL	The protocol used is not valid
[0xF8] ILVERR_ALREADY_ENROLLED	The person is already in this database
[0xF7] ILVERR_BASE_NOT_FOUND	The specified database does not exist
[0xF6] ILVERR_BASE_ALREADY_EXISTS	The specified database already exists
[0xF5] ILVERR_BIO_IN_PROGRESS	Command received during biometric processing
[0xF4] ILVERR_CMD_INPROGRESS	Command received while another command is running
[0xF3] ILVERR_FLASH_INVALID	Flash type invalid
[0xF2] ILVERR_NO_SPACE_LEFT	Not enough memory for the creation of a database
[0xF1] ILVERR_ADMIN_EXECUTING	Administrator Mode in use.
[0xF0] ILVERR_BAD_SIGNATURE	Invalid signature
[0xEF] ILVERR_CARD_PROBLEM	An error occurred while reading the card
[0xEE] ILVERR_ERROR_SL	An error occurred during the execution of Serial Link Configuration

[0xED]	ILVERR_ERROR_WIEGAND	An error occurred during the execution of the output Wiegand/Dataclock Configuration
[0xEC]	ILVERR_ERROR_NETWORK	An error occurred during the execution of Network Configuration
[0xEB]	ILVERR_OUT_OF_FIELD	Number of the additional fields is greater than allowed or an additional field length is greater than allowed
[0xEA]	ILVERR_INVALID_ACCESS	Access cannot be granted because time does not agree with time zone
[0xE9]	ILVERR_FIELD_NOT_FOUND	The required field does not exist in the database
[0xE8]	ILVERR_FIELD_INVALID	Field size or field name is invalid
[0xE7]	ILVERR_SECURITY_MODE	Security keys cannot be changed
[0xE6]	ILVERR_USER_NOT_FOUND	The required User ID has not been found in the database
[0xE5]	ILVERR_CMDE_ABORTED	Command has been cancelled
[0xE4]	ILVERR_SAME_FINGER	There are two templates of the same finger
[0xE3]	ILVERR_NO_HIT	Presented finger does not match
[0xE2]	ILVERR_SECU_CERTIF_NOT_EXIST	The required certificate does not exist
[0xE1]	ILVERR_SECU_BAD_STATE	Invalid security state
[0xE0]	ILVERR_SECU_ANTIREPLAY	An anti-replay error occurred
[0xDF]	ILVERR_SECU_ASN1	Error while decoding an ASN1 object
[0xDE]	ILVERR_SECU	Cryptographic error
[0xDD]	ILVERR_SECU_AUTHENTICATION	Mutual authentication error

[0xDC]	ILVERR_NO_LIC_MKIT	No licence MorphoKit™. Also used on MorphoSmart™ MSO when a license is missing. See also ILVERR_NO_LICENSE.
[0xDB]	ILVERR_FFD	False Finger Detected
[0xDA]	ILVERR_MOIST_FINGER	The finger can be too moist or the scanner is wet
[0xD9]	ILVERR_OTP_NOT_INITIALIZED	All OTP parameters are not initialized
[0xD8]	ILVERR_NO_MORE_OTP	No more OTP (sequence number = 0)
[0xD7]	ILVERR_OTP_ENROLL_NEEDED	OTP database is empty: enrol is needed
[0xD6]	ILVERR_OTP_NO_HIT	No Hit (or latent or FFD)
[0xD5]	ILVERR_OTP_REENROLL_NOT_ALLOWED	Re-enrolment not allowed
[0xD4]	ILVERR_OTP_ENROLL_FAILED	Enrolment failed (status != ILV_OK)
[0xD3]	ILVERR_OTP_IDENT_FAILED	Identification failed (status != ILV_OK except DB errors)
[0xD2]	ILVERR_OTP_PIN_NEEDED	Pin needed for first enrolment
[0xD1]	ILVERR_BAD_SUB_ILV	An unexpected or invalid sub-ILV has been found
[0xD0]	ILVERR_SOFTWARE_ACTIVATE_FAILED	The software upgrade has failed.
[0xCF]	ILVERR_UEPS_SECU_ERROR	Error security UEPS
[0xCE]	ILVERR_NO_UEPS_KEYS	No UEPS keys injected
[0xCD]	ILVERR_UEPS_ERROR_FILE	Error in UEPS key file
[0xCC]	ILVERR_UEPS_BAD_KEY_ID	Bad key ID

[0xCB]	ILVERR_UEPS_SECURE_SESSION_NOT_CREATED	Secure session not created
[0xCA]	ILVERR_UEPS_KEY_ID_NOT_FOUND	Key ID not found
[0xC9]	ILVERR_UEPS_DEFAULT_SESSION_NOT_CREATED	Default load key session not created
[0xC8]	ILVERR_FVP_MINUTIAE_SECURITY_MISMATCH	Reference template(s) is(are) not usable for matching operations in the required security level
[0xC7]	ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported by the firmware loaded in the MorphoSmart™ terminal. For example: database related commands are not supported by the MSO_WSQ firmware.
[0xC6]	ILVERR_FVP_MINUTIAE_MIX	When sending one multimodal reference template in ILV_ADD_RECORD command, all reference templates must be multimodal. Adding a user with both fingerprint and multimodal templates is not supported.
[0xC5]	ILVERR_FINGER_MISPLACED_OR_WITHDRAWN	The finger was misplaced or has been withdrawn during finger acquisition performed on MorphoSmart™ FINGER VP (only on ILV_VERIFY, ILV_IDENTIFY or ILV_UPDATE_PRIVATE_DATA commands).
[0xC4]	ILVERR_CBM_KO	No answer received from biometric sensor
[0xC3]	ILVERR_NO_LICENSE	A required license is missing. Only used on MorphoSmart™ FINGER VP, see ILVERR_NO_LIC_MKIT for MorphoSmart™ MSO.
[0xC2]	ILVERR_ADVANCED_SECURITY_LEVEL_MISMATCH	Failed to enrol finger compatible with advanced security levels (MorphoSmart™ FINGER VP only)

[0xC1] ILVERR_INCONSISTENT_PARAMETERS	Some parameters are not consistent (MorphoSmart™ FINGER VP only)
[0xBF] ILVERR_OTP_LOCK_SET_PARAM	After 5 false Pin presentations, the OtpSetParameters ILV is locked
[0xBE] ILVERR_OTP_LOCK_ENROLL	After 5 failed enrolments with pin or 5 failed identifications, the OTP_Enrol_User ILV is locked.
[0xBD] ILVERR_OTP_LOCK_GEN_OTP	After 10 failed identifications, the OtpGenerate ILV is locked.
[0xBC] ILVERR_APPLI_LOCKED	The MSO is locked
[0xBB] ILVERR_TERMINAL_NOT_LOCK	MorphoSmart™ is not locked
[0x9D] ILV_NOT_IMPLEMENTED	The request is not yet implemented

Status codes table

Status codes	Description
[0x01] ILVSTS_HIT	Authentication or Identification succeeded
[0x02] ILVSTS_NO_HIT	Authentication or Identification failed
[0x04] ILVSTS_DB_FULL	The database is full
[0x05] ILVSTS_DB_EMPTY	The database is empty
[0x22] ILVSTS_FFD	False finger detected
[0x23] ILVSTS_MOIST_FINGER	The finger can be too moist or the scanner is wet

Configuration File

File Types

Two types of files exist in the terminal to manage the configuration:

- "CFG" files which contain the values of the keys,
- "DSC" files which contain the description of each key (types, limits, possible values…). The "DSC" is "a copy" of the "CFG" file but specifying a type for each parameter.

The host can interpret various parameters types and associate a specific input style to each parameter.

It means IP addresses can be typed into an "IP field", binary choices entered into a check box …

Following types are available:

- String,
- IPV4,
- "read only",
- integer,
- range of integers,
- boolean,
- enumeration

st (string)

type=st (length)

The key whose type is string accepts any string value. The key's value can be an empty string.

length

Maximum length of the string (default: CFG_INT_VALUE_MAX_LENGTH).

Key values

This key can take any string value.

Graphic interpretation

An editable field can be used.

Example

terminal name = st (32)

ip

type=imp (attribute)

The key accepts any string that represents IP v4 IP. The key cannot be empty.

attribute

Type of IP version (default value 4):

4 = IP v4

6 = IP v6 (does not exist, reserved for future use)

Key values

This key can take any string value that matches an IP address.

Graphic interpretation

An editable field as "IP" can be used.

Example

terminal ip = ip (4)

ro (read only)

type=ro (length)

This key designed by ro can be modified from the application but not from a remote host (via Ethernet or USB). The key cannot be empty.

length

Maximum length of the string (default: CFG_INT_VALUE_MAX_LENGTH).

Key values

This key can take any string value.

Graphic interpretation

This field cannot be edited by user interface.

Example

software revision = ro (32)

in (integer)

type=in (min, max)

Integer value. This value must be accepted by the C type int.

min

Minimum value that can be taken by the key (default –MAX_INT).

max

Maximum value that can be taken by the key (default +MAX_INT).

Key values

This key can take any value between min and max.

Graphic interpretation

A numerical editable field can be used.

Example

reserved identifier = in(-5000, 5000)

ra (range of integer)

type=ra(min, max)

Integer value. This value must be accepted by the C type int.

min

Minimum value that can be taken by the key.

max

Maximum value that can be taken by the key.

Values

This key can take any value between min and max.

Graphic interpretation

A “progress bar” can be used.

Example

buzzer volume = ra(0,10)

Note

There is no difference between “integer” and “range of integer” types: only the graphical interpretation differs.

bo(boolean)

type=bo(attribute)

This key can have only 2 values but can be interpreted using several user controls.

attribute

Interpretation type of the Boolean (Default type: t)

y = yes / no

e = enabled / disabled

t = true / false

Key values

1 = TRUE

0 = FALSE

Graphic interpretation

The interpretation of the value can be different according to the user interface use (y, e, and t). It can be for example a “check box”.

Example

bypass authentication = bo(y)

authent PK contactless = bo(e)

EN(ENUMERATION)

type=en(enumeration definition)

This key contains a list of string values with associated integer values.

enumeration definition

The enumeration is defined by a list of strings with an associated integer value. If the integer value isn’t present, it is implicitly equal to the previous value + 1. The string values can contain only alphanumerical characters plus “underscore” (_) characters.

Key values

This key can take any string value that is present in the enumeration definition.

Graphic interpretation

The interpretation of the value can be a “combo box”.

Example

Security level = en(LOW:0;MEDIUM;MEDIUM_HIGH:4;HIGH)

In this example:

Possible values are LOW, MEDIUM, MEDIUM_HIGH and HIGH.

- The integer value associated to LOW is 0.
- The integer value associated to MEDIUM is 1.
- The integer value associated to MEDIUM_HIGH is 4.
- The integer value associated to HIGH is 5.

MIFARE® Contactless Card Mapping

Sector	Block	Block	Block	Block	Block	Block	Block	Size	Real size
0	*1*	*2*	*3*	***Key***				64	48
1	4	5	6	***Key***				128	96
2	7	8	9	***Key***				192	144
…									
14	43	44	45	***Key***				960	720
15	46	47	48	***Key***				**1024**	768
16	49	50	51	***Key***				1088	816
17	52	53	54	***Key***				1152	864
…									
30	91	92	93	***Key***				1984	1488
31	94	95	96	***Key***				2048	1536
32	97	98	99	100	101	111	***Key***	2304	1776
33	112	113	114	115	116	126	***Key***	2560	2016
…									
39	202	203	204	205	206	216	***Key***	**4096**	3456

In green: 1 K card.

Only “data block” are counted. Block 1, 2, 3 contain card serial number.

Setting up Matching Threshold

This parameter can be set to values from 0 to 10. This parameter specifies how tight the matching threshold is. Threshold scoring values are identified hereafter. It determines the False Acceptance Rate.

Value	Description	FAR value
0	Low threshold for test purpose only	There are few rejections, but many recognitions
1	Very few persons rejected	FAR < 1%
2		FAR < 0.3%
3	(Default value) Recommended value	FAR < 0.1%
4		FAR < 0.03%
5	Intermediate threshold	FAR < 0.01%
6		FAR < 0.001%
7		FAR < 0.0001%
8		FAR < 0.00001%
9	Very high threshold (few false acceptances) Secure application	FAR < 0.0000001%
10	High threshold for test purpose only	There are very few recognitions, and many rejections

Database Size Limits

The table below indicates the limits for the size of the Database.

The size limits are:

- the maximum number of users at the creation of the Database,
- the maximum number of additional user's fields per user,
- Additional fields size.

Terminal Series Name	Modes	Maximum User per Base	Maximum Number of Field	Maximum size available for all fields (Bytes)
MorphoAccess® Lite Series	MA5G	500	-	-
	MA2G	500	8	256
MorphoAccess® Lite Series with MA_3K_USERS licence	MA5G	3,000	-	-
	MA2G	3,000	8	256
MorphoAccess® Lite Series with MA_10K_USERS licence	MA5G	10,000	-	-
	MA2G	10,000	8	256
MorphoAccess® Sigma Series	MA5G	5,000	-	-
	MA2G	5,000	8	256
MorphoAccess® Sigma Series with MA_10K_USERS licence	MA5G	10,000	-	-
	MA2G	10,000	8	256
MorphoAccess® Sigma Series with MA_50K_USERS licence	MA5G	50,000	-	-
	MA2G	50,000	8	256
	MA5G	100,000	-	-

MorphoAccess® Sigma Series with MA_100K_USERS licence	MA2G	50,000	8	256
MorphoAccess® Sigma Extreme Series	MA5G	5,000	-	-
	MA2G	5,000	8	256
MorphoAccess® Sigma Extreme Series with MA_10K_USERS licence	MA5G	10,000	-	-
	MA2G	10,000	8	256
MorphoAccess® Sigma Extreme Series with MA_50K_USERS licence	MA5G	50,000	-	-
	MA2G	50,000	8	256
MorphoAccess® Sigma Extreme Series with MA_100K_USERS licence	MA5G	100,000	-	-
	MA2G	50,000	8	256
MorphoWave® Compact	MA5G	5,000	-	-
MorphoWave® Compact with MA_20K_USERS licence	MA5G	20,000	-	-
MorphoWave® Compact with MA_40K_USERS licence	MA5G	40,000	-	-

USB Scripts

MorphoAccess® SIGMA Family & MorphoWave® Compact terminal can be partially configured using encrypted USB scripts. These scripts can be created from MorphoBioToolBox. When user connects the USB drive which contains the USB scripts to the terminal, the intended operations will be performed and corresponding results will be written into the USB drive. User can check script execution's result with the help of MorphoBioToolBox. this feature, user can change the configuration of those terminals which are not connected to the network. Please refer to the **MorphoBioToolBox User guide** for more details.

Note: User can use the same USB Scripts to configure one or more terminals.

In such cases the USB Script execution's result is overwrite by the new one.

This result can be read using MorphoBioToolBox (see "Read response" section of "USB Script" chapter in user guide).

User can create the following scripts, using MBTB:

Get/Set IP Configuration

Get/Set Wi-Fi Configuration

Firmware Upgrade

Error Log Configuration

Retrieve Error Log

Reset Configuration

SSL Configuration

Protocol Switch

Section 4 : L1 Bioscrypt Legacy Mode Distant Commands

References

MorphoAccess® SIGMA Family & MorphoWave® Compact and SIGMA Lite Series terminal in L1 legacy mode supports distant commands in 32-bit word format. Details about all the commands supported in L1 terminals are available in **Serial Command Manual**.

Also refer to section "MA Sigma L1 Legacy Mode Command Support Matrix" in **Command Support Matrix** to see the commands supported by MorphoAccess® SIGMA and SIGMA Lite Series terminal in L1 legacy mode.

Note: MorphoAccess® SIGMA Extreme Series terminal doesn't support L1 Legacy mode.

Annex 1 : Bibliography

How to get latest version of the documents?

The last version of the documents can be downloaded from our web site at the address below:

www.biometric-terminals.com

(Login and password required).

To request a login, please send us an email to the address below:

support.bioterminals@idemia.com

Documents concerning the MorphoAccess® SIGMA Family & MorphoWave® Compact terminal

Bibliography

MorphoAccess® SIGMA Family Bibliography,

Ref. 2016_2000022505_v2 - MorphoAccess® SIGMA Family - Bibliography

This document gives document's references for MorphoAccess® SIGMA Family terminals. This document is in English.

Annex 2 : Support

Troubleshooting

Customer service

IDEMIA

SAV Terminaux Biométriques

Boulevard Lénine - BP428

76805 Saint Etienne du Rouvray

FRANCE

Phone: +33 2 35 64 53 52

Hotline

IDEMIA

Support Terminaux Biométriques

18, Chaussée Jules César

95520 Osny

FRANCE

support.bioterminals@idemia.com

Phone: + 33 1 30 20 30 40

(9H00am to 5H30pm French Time, Monday to Friday)

http://www.biometric-terminals.com/

A login and password are required to access the full site content. If you don't have one, please send us an email to the address above to request one.

Contact by email is preferred.

Head office:

IDEMIA

11, boulevard Gallieni

92130 Issy-les-Moulineaux – France

www.idemia.com