



MorphoAccess™ 100 Series

Host System Interface Specification

Produced by Sagem Défense Sécurité

Copyright ©2006 Sagem Défense Sécurité

www.sagem.com

Table of Content

OVERVIEW	7
<hr/>	
REMOTE MANAGEMENT PROTOCOLS	8
<hr/>	
OVERVIEW	8
<hr/>	
STANDARD INTERFACE PRESENTATION	9
<hr/>	
PRESENTATION	9
ILV COMMAND	9
COMPATIBILITY WITH MORPHOACCESS FIRST GENERATION (MA2XX AND MA3XX) TERMINAL	11
<hr/>	
INITIALIZATION FUNCTIONS DESCRIPTION	13
<hr/>	
PING	14
DEFAULT INIT	15
GET VERSION	17
REBOOT	19
SET CONFIGURATION	20
GET CONFIGURATION	22
SET REGISTRY KEY	24
GET REGISTRY KEY	27
GET REGISTRY FILE	29
SET CONTACTLESS KEYS (OR CRYPTO WRITE) [MA110 OR MA120 ONLY]	31
DISTANT SESSION	34
UPGRADE SOFTWARE [DISTANT SESSION]	36
<hr/>	
BIOMETRIC FUNCTIONS DESCRIPTION	38
<hr/>	
VERIFY	39
ENROLL	42
IDENTIFY	47
<hr/>	
DATABASE FUNCTIONS DESCRIPTION	50
<hr/>	
CREATE DATABASE	51
ERASE BASE	54
ERASE ALL BASE	55
ADD BASE RECORD	56
REMOVE BASE RECORD	59
FIND USER BASE	61
GET DATA	63
GET PUBLIC FIELDS	65
UPDATE PUBLIC DATA	67
GET BASE CONFIG	69
GET ALL BASES CONFIG	72

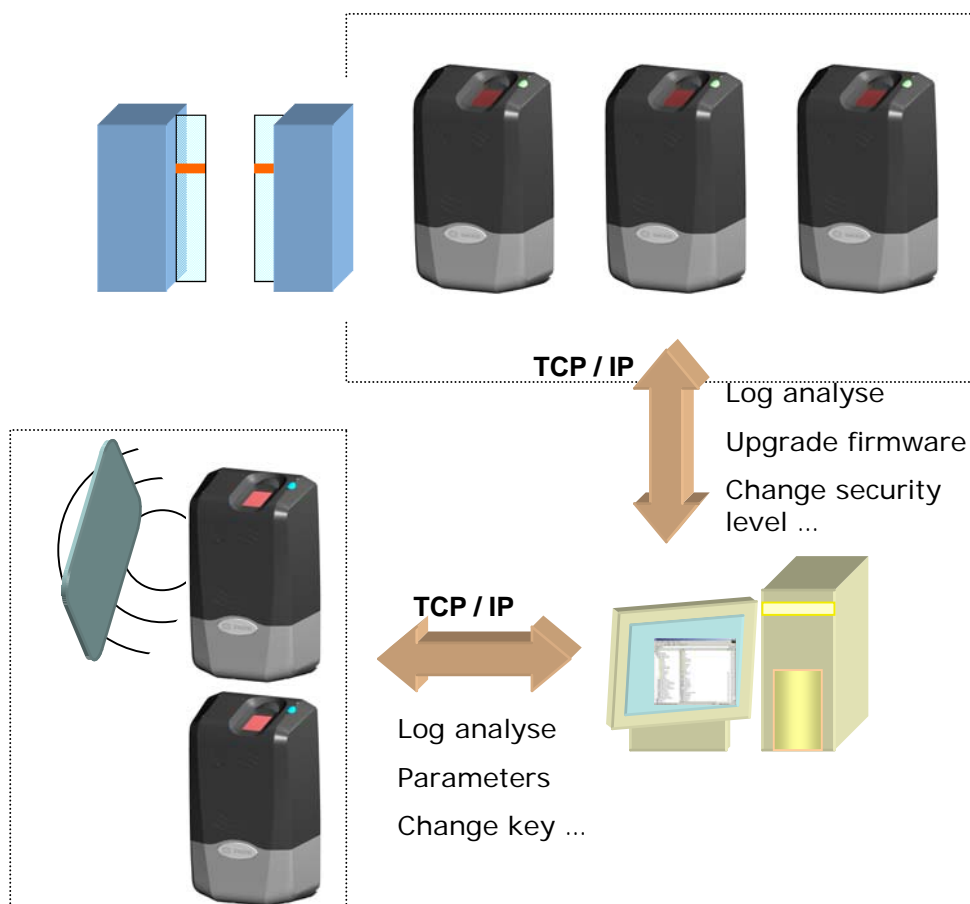
LOG FUNCTIONS DESCRIPTION	74
OVERVIEW	74
BASE STRUCTURE	75
LINE STRUCTURE	76
AUTHENTICATION RECORDS FORMAT	77
IDENTIFICATION RECORDS FORMAT	79
COMPATIBILITY NOTES	80
GET LOG STATUS	81
ERASE LOG	83
GET LOG	85
CONTACTLESS FUNCTIONS DESCRIPTION	88
CONTACTLESS VERIFICATION (OR CONTACTLESS AUTHENTICATION)	89
CONTACTLESS CARD FUNCTION	91
PROXY MODE FUNCTIONS DESCRIPTION	96
OVERVIEW	96
ACCESS AUTHORIZATION [PROXY MODE ONLY]	97
DATA SEND [PROXY MODE ONLY]	98
BIOMETRIC DATA DESCRIPTION	100
PK_COMP V2	101
PK_MAT	102
DATABASE DATA DESCRIPTION	103
USER INDEX	104
USER ID	105
PKBASE	106
BASE CONFIGURATION	107
ADDITIONAL DATA FIELD DESCRIPTION	109
FIELD CONTENT	110
PUBLIC FIELD	111
PRIVATE FIELD	112
ADDITIONAL USER DATA	113
NO CHECK ON TEMPLATE	114
WIEGAND DATACLOCK DATA DESCRIPTION	115
WIEGAND DATA	116
DATACLOCK DATA	117
CONTACTLESS DATA DESCRIPTION	118
ADDRESS BIOMETRIC DATA	119

<u>CONFIGURATION DATA DESCRIPTION</u>	<u>120</u>
DATE AND TIME CONFIGURATION	121
<u>MISCELLANEOUS DATA DESCRIPTION</u>	<u>122</u>
MATCHING SCORE	123
LATENT DETECTION	124
<u>CONSTANT VALUES</u>	<u>125</u>
IDENTIFIER - STATUS	125
ERROR CODES VALUE	125
<u>CONFIGURATION FILES</u>	<u>127</u>
APPLICATION PARAMETERS	127
REMOTE MANAGEMENT PARAMETERS	130
NETWORK PARAMETERS	130
SENSOR PARAMETERS PARAMETERS	130
<u>CONTACTLESS CARD MAPPING</u>	<u>131</u>
<u>SETTING UP THE MATCHING THRESHOLD</u>	<u>132</u>

OVERVIEW

The MorphoAccess™ provides remote management facilities. It is possible to change the terminal settings through Ethernet using a TCP-IP connection.

The MorphoAccess™ acts as a server and provides a unique socket.



The following operations are allowed:

- Changing system settings such as control type, control timeout...
- Sending biometric requests
- Software upgrade
- ...

It is also possible to receive information from the terminal. After a biometric control the terminal sends information in TCP-IP, UDP or RS485. In this case the MorphoAccess™ is a client and the PC is the server.

This document details the syntax of each command.

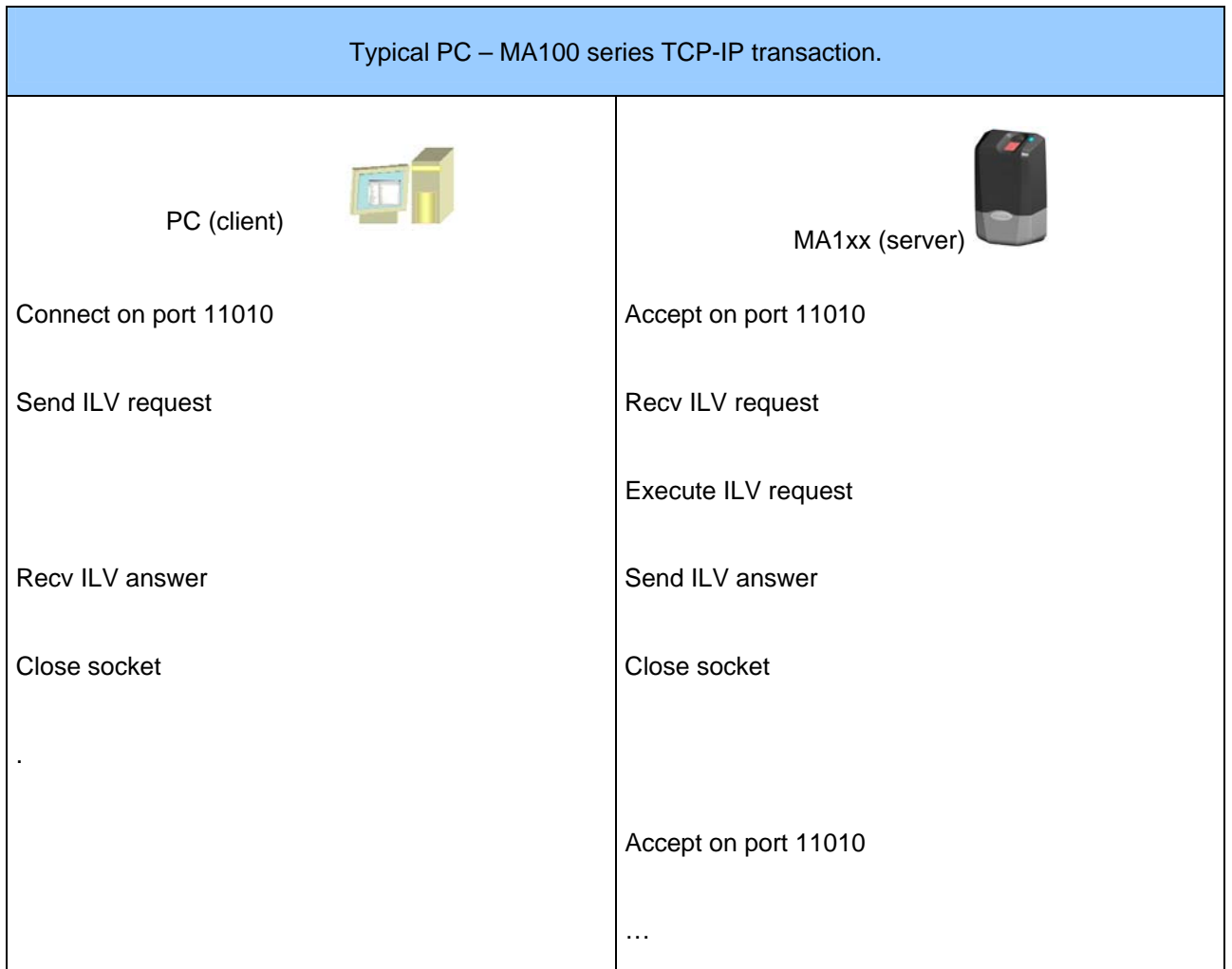
REMOTE MANAGEMENT PROTOCOLS

OVERVIEW

It is possible to administrate the MorphoAccess™ from a remote computer. In this case the MorphoAccess™ works as a standard TCP server waiting for requests coming from a remote client.

At the end of the transaction the client closes the socket. If the connection is not closed, the MorphoAccess™ can automatically close the connection after a given period of inactivity.

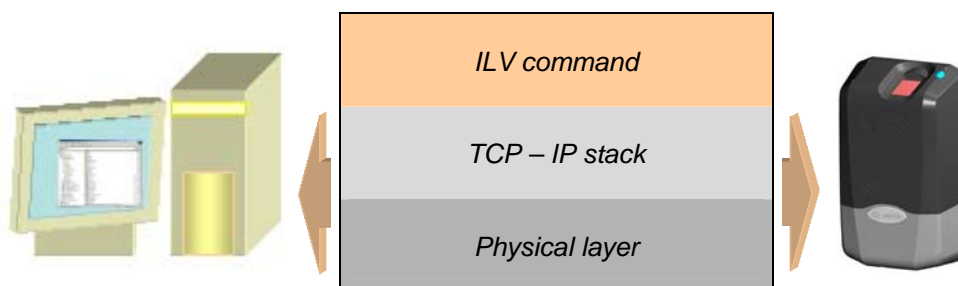
A standard exchange follows always the following schema:



STANDARD INTERFACE PRESENTATION

PRESENTATION

One connection can be established on the 11010 port (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.



ILV COMMAND

ILV commands have the following format:

<i>ILV command</i>		
Identifier	Length	Value
1 byte	2 bytes	Length bytes
<i>Command identifier</i>	<i>Command data length (little endian format)</i>	<i>Command data</i>

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 (0xFFFFE), length is coded with 4 bytes and ILV command has the following format:

<i>ILV command</i>			
Identifier	“Length”	Length	Value
1 byte	2 bytes	4 bytes	Length bytes
<i>Command identifier</i>	<i>0xFFFF</i> <i>(escape code)</i>	<i>Command data length</i> <i>(little endian format)</i>	<i>Command data</i>

COMPATIBILITY WITH MORPHOACCESS FIRST GENERATION (MA2XX AND MA3XX) TERMINAL

This interface has been designed with a constant view of compatibility. However some commands have been removed or have changed for technical reasons. The following table details the compatibility status of each ILV command.

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.

Maintained commands

Get Version	This function will return MorphoAccess™ features.
Reboot	Fully compatible.
Ping	Fully compatible.
Get Log Status	Fully compatible – 8000 log lines.
Get Log	Fully compatible. The whole event base can be downloaded with one command in distant session.
Erase Log	Fully compatible.
Set Configuration	Limited compatibility – date only.
Get Configuration	Limited compatibility – date only.
Default Init	This command can erase Mifare™ crypto keys or not.
Distant Session	Fully compatible.
Verify	Limited compatibility.
Enroll	Limited to one base, and limited compatibility : returned PK format is changed.
Identify	Fully compatible but limited to one base.
Create Database	Compatible syntax but limited to one base.
Erase Base	Compatible syntax but limited to one base.
Erase All Base	Compatible syntax but limited to one base.
Add Base Record	Compatible syntax but limited to one base.

Remove Base record	Compatible syntax but limited to one base.
Get Base Config	Compatible syntax but limited to one base.
Access Authorization	Fully compatible.

Removed commands

Set Time Mask	
Get Time Mask	
Set User Message	
Contactless Authent	
Garbage Collector	
Erase User Table	
Print Screen Message	
Read From Keyboard	
Set Public Key	
Wiegand Dataclock Read	
Contactless Read	This function has been replaced by the <i>Contactless Card Function</i> command [SAGEM only].

New commands

Set Registry Key	This command allows changing a parameter value.
Get Registry Key	This command allows retrieving a parameter value.
Get Registry File	This command allows retrieving a configuration file.
Contactless Card Function	This command provides Mifare™ card functionality such as data or serial number reading [SAGEM only].
Wiegand Dataclock Send	This command allows sending data over the Wiegand / Dataclock link [SAGEM only].

INITIALIZATION FUNCTIONS DESCRIPTION

PING

Description

This function allows checking that the terminal responds correctly.

Command

Request

Identifier value	CMD_PING [0x08]	1 byte
Length value	0	2 bytes

Reply

Identifier value	CMD_PING [0x08]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request status	1 byte

Request status

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

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 function allows initializing the MorphoAccess™ with the factory configuration.

Sent without argument (length is NULL), *Default Init* does not reset Ethernet settings and Mifare™ Keys.

An additional argument (*Keep Settings*) allows erasing Mifare™ Keys and resetting Ethernet parameters.



All Mifare™ crypto keys and Ethernet settings can be erased!

Command

Request

Identifier value	CMD_DEFAULT_INIT [0x0A]	1 byte
Length value	0 or 1	2 bytes
Value (Parameters)	<i>Keep Settings (optional)</i>	1 byte

Keep settings (optional)

This optional parameter prevents some settings from being reset. A mask of bits define option to keep.

- NET_MASK [0x01] [0b0000 0001] keeps Ethernet parameters.
- MIFARE_KEYS_MASK [0x02] [0b0000 0010] keeps Mifare™ keys.
- DATABASE_MASK [0x04] [0b0000 0100] keeps the local database [SAGEM only].
- TRACE_MASK [0x08] [0b0000 1000] keeps the trace file [SAGEM only].

Reply

Identifier value	CMD_DEFAULT_INIT [0x0A]	1 byte
Length value	0x01	2 bytes
Value (Parameters)	Request status	1 byte

Request status

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

Compatibility note

The *Keep Settings* parameter replace *Keep Net Settings* presented in *MorphoAccess 1G Host Interface*.

MIFARE_KEYS_MASK [0x02] is a new option.

Note

The database is not erased. Use [Erase All Base](#) to erase the base.

GET VERSION

Description

This function returns the serial number of the terminal, the revision of the software application and the MorphoAccess terminal type.

Command

Request

Identifier value	CMD_GET_VERSION [0x03]	1 byte
Length value	0	2 bytes

Reply

Identifier value	CMD_GET_VERSION [0x03]	1 byte
Length value	17	2 bytes
Value (Parameters)	Request Status	1 byte
	<i>Serial Number</i>	9 bytes
	<i>Software Revision</i>	4 bytes
	<i>Extended Memory</i>	1 byte
	<i>MorphoAccess Type</i>	2 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.

Serial Number

Terminal serial number in ASCII "123456789".

Software Revision

1 byte	1 byte	1 byte	1 byte	Software revision.
RFU	RFU	Major	Minor	

Extended Memory

The returned value is 1.

MorphoAccess Type

1 byte	1 byte	500 users database	Contactless reader
RFU	100	Y	N
RFU	120	Y	Y (Mifare™ cards)
RFU	110	Y	Y (iCLASS™ cards)

Compatibility note

The *MorphoAccess Type* and *Extended Memory* will logically differ from the first generation terminals.

REBOOT

Description

This function allows rebooting the MorphoAccess™. This command can be required in order to apply some parameters.

Command

Request

Identifier value	CMD_REBOOT [0x04]	1 byte
Length value	0	2 bytes

Reply (in case of failure)

Identifier value	CMD_REBOOT [0x04]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request status	1 byte

Request status

ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Note

The reboot process can last up to 15 seconds.



In case of success, no reply must be expected.

SET CONFIGURATION

Description

This function initializes the MorphoAccess™ Real Time Clock..

Command

Request

Identifier value	CMD_SET_CONFIGURATION [0x01]	1 byte
Length value	1 (+ Ln) (+ Lr) (+ Ld) (+ La)	2 bytes
Value (Parameters)	MorphoAccess Mode	1 byte
	<i>Date And Time Configuration</i>	<i>Ld bytes</i>

MorphoAccess mode

Set to 0. This parameter is RFU.

Date And Time Configuration

This sub ILV allows setting terminal date.

See the [Date and Time Configuration](#) structure for details.

Reply

Identifier value	CMD_SET_CONFIGURATION [0x01]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request Status	1 byte


Request status

ILV_OK [0x00]

The execution succeeded.

ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.

Note

 This command is available only for compatibility reasons. If you begin with MorphoAccess™ products, or if you want to use MorphoAccess™ 100 series specific features, prefer the [Set Registry Key](#) command.

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 configuration.

Command

Request

Identifier value	CMD_GET_CONFIGURATION [0x06]	1 byte
Length value	N	2 bytes
Value (Parameters)	Parameter Identifier 1	1 byte
	<i>Parameter Identifier 2</i>	<i>1 byte</i>

	<i>Parameter Identifier i</i>	<i>1 byte</i>

	<i>Parameter Identifier N</i>	<i>1 byte</i>

Parameter Identifier i

It is used to determine which parameter will be returned:

- ID_DATE [0x13]

Only the necessary parameters can be included.

Reply

Identifier value	CMD_GET_CONFIGURATION [0x06]	1 byte
Length value	2 (+ Ln) (+ Lr) (+ Ld) (+ La)	2 bytes
Value (Parameters)	Request status	1 byte
	MorphoAccess Mode	1 byte
	<i>Date and Time Configuration</i>	<i>Ld bytes</i>

Request status

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

MorphoAccess Mode

Must be 0.

Date And Time Configuration

See the [Date and Time Configuration](#) structure for details.

Notes



This command is available only for compatibility reasons. If you begin with MorphoAccess products, or if you want to use MorphoAccess™ 100 series specific features, prefer the *Get Registry Key* command.

Only requested sub ILV are returned.

See the [Set Configuration](#) command to get the details of each sub ILV.

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 relay settings (stored in **app.cfg** file).

Change “/app/relay/enabled/1” to “/app/relay/enabled/0”.

Command

Request

Identifier value	CMD_SET_REGISTRY_KEY [0x0B]	1 byte
Length value	$L_0 + L_1 + \dots + L_N$	2 bytes
Value (Parameters)	Registry path and value in ASCII [0]	L_0 bytes
	<i>Registry path and value in ASCII [1]</i>	L_1 bytes
	...	
	<i>Registry path and value in ASCII [N]</i>	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: the “file” parameter may be (or not) preceded by the “/” symbol. “/” and “[” symbols are accepted.

Remark: the “file” corresponds to “file.cfg”. The “.cfg” extension must be omitted.

Reply

Identifier value	CMD_SET_REGISTRY_KEY [0x0B]	1 byte
Length value	$N + 2$	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Setting result [0]</i>	1 byte
	<i>Setting result [1]</i>	1 byte

...
 Setting result [N] 1 byte


Request status

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


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.

Notes

 When you set a value using the [Set Registry Key](#) command, the system checks the value coherency *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 verify 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 configuration key value.

Command

Request.

Identifier value	CMD_GET_REGISTRY_KEY [0x0C]	1 byte
Length value	$L_0 + L_1 + \dots + L_N$	2 bytes
Value (Parameters)	Registry path n ASCII [0]	L_0 bytes
	<i>Registry path in ASCII [1]</i>	L_1 bytes
	...	
	<i>Registry path in ASCII [N]</i>	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:

"/application/biometric control/matching threshold".

Remark: the "file" parameter may be (or not) preceded by the "/" symbol. "/" and "\" symbols are accepted.

Reply

Identifier value	CMD_GET_REGISTRY_KEY [0x0C]	1 byte
Length value	$1 + L_0 + L_1 + \dots + L_N$	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Registry value [0]</i>	L_0 bytes
	<i>Registry value [1]</i>	L_1 bytes
	...	
	<i>Registry value [N]</i>	L_N bytes

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_NO_SUCH_KEY [0xE1]	The entry does not exist.

Registry Value [i]

"file/key/parameter/value" The string is "null terminated". Example:

"/app/biometric control/matching threshold/3".

GET REGISTRY FILE

Description

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

Command

Request.

Identifier value	CMD_GET_REGISTRY_FILE [0x0D]	1 byte
Length value	L	2 bytes
Value (Parameters)	File path in ASCII	L bytes

File path in ASCII

"file_name.cfg".

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)

File are described in section [Configuration files](#).

Reply

Identifier value	CMD_GET_REGISTRY_KEY [0x0D]	1 byte
Length value	1 + L	2 bytes
Value (Parameters)	Request status	1 byte
	<i>File Content</i>	<i>L bytes</i>

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_NO_SUCH_KEY [0xE1]	The file does not exist.

File Content

The complete file formatted as below.

Notes

File contents are described in *MA100 Series User Guide*.

The “*MA1XX Configuration Tool.exe*” illustrates the use [Set Registry Key](#), [Get Registry Key](#) and [Get Registry File](#) commands.

SET CONTACTLESS KEYS (OR CRYPTO WRITE) [MA110 OR MA120 ONLY]

Description

On MA12x terminal this command allows changing the Mifare™ keys.

On MA11x terminal this command allows changing the *iCLASS*™ keys.

Command

Request (for MA12x terminals)

Identifier value	CMD_SET_CONTACTLESS_KEYS [0x58]	1 byte
Length value	14	2 bytes
Value (Parameters)	Key Number	1 byte
	Sector Number	1 byte
	Keys Sector	12 bytes

Key Number

This parameter is ignored.

Sector Number

This value can be set from 0 to 39.

Keys Sector

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

Request (for MA11x terminals)

Identifier value	CMD_SET_CONTACTLESS_KEYS [0x58]	1 byte
Length value	9	2 bytes
Value (Parameters)	Key Number	1 byte
	Key	8 bytes

Key Number

This parameter defines the key position in the terminal *iCLASS*TM reader.

Must be set to 6.

Key

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

Reply

Identifier value	CMD_SET_MIFARE_KEYS [0x58]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request Status	1 byte

Request Status

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

Note

The [Contactless card mapping](#) section describes the card structure.

Compatibility note

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

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 terminal is “stopped” waiting for orders. Biometric control restarts only when the distant session is closed. This mode is useful to increase speed when a lot of commands have to be sent.

Command

Request

Identifier value	CMD_DISTANT_SESSION [0x63]	1 byte
Length value	1	2 bytes
Value (Parameters)	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

Identifier value	CMD_DISTANT_SESSION [0x63]	1 byte
Length value	2	2 bytes
Value (Parameters)	Request Status	1 byte
	Session Status	1 byte

Request status

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

ILVERR_BAD_PARAMETER [0xFE] Invalid parameters.

Session status (*if Request Status is ILV_OK*)

DISTANT_SESSION_OPENED [0x01] Terminal is now in “distant session”.

DISTANT_SESSION_MEMORY_ERROR [0xE0] Resources cannot be allocated.

DISTANT_SESSION_CLOSED [0x02] Standalone mode.

Compatibility note

This command is fully compatible with first generation terminals.

UPGRADE SOFTWARE [DISTANT SESSION]

Description

This command will upgrade the software. SAGEM SA numerically signs the software.

- A **Reboot** command must follow the upgrade.

Command

Request (extended ILV if package size if greater than 65532 bytes)

Identifier value	CMD_UPGRADE_SOFTWARE [0x0E]	1 byte
Length value	0xFFFF (escape)	2 bytes
Length value	2 + L	4 bytes
Value (Parameters)	CRC_1	1 byte
	CRC_2	1 byte
	Software Package	L bytes

CRC_1, CRC_2

Ignored (RFU).

The new firmware is contained in a file (.img) and is numerically signed. Before writing the file, its signature is checked.

Software Package

The content of the file furnished by SAGEM DS.

The content of the file furnished by SAGEM DS.

Software package is a list of [Software Data](#) sub ILV [SAGEM only]

Reply

Identifier value	CMD_UPGRADE_SOFTWARE [0x0E]
Length value	1

Value (Parameters)	Request status	1 byte
--------------------	----------------	--------

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_BAD_SIGNATURE [0xF0]	File signature does not match.
ILVERR_BAD_CRC [0xE0]	Invalid CRC (RFU).
ILVERR_BAD_SESSION_STS [0xDF]	Terminal is not in "distant session mode".

BIOMETRIC FUNCTIONS DESCRIPTION

VERIFY

Description

This function captures a finger and checks if it matches either with the minutiae file sent to the terminal or with a database record.

The maximum number of reference templates is 10.

Command

Request

Identifier value	CMD_VERIFY [0x20]	1 byte
Length value	5 + L1 ... (+ Ln) (+ 4)	2 bytes
Value (Parameters)	Timeout	2 bytes
	Matching threshold	2 bytes
	Acquisition quality threshold	1 byte
	Reference Template 1	L1 bytes
	<i>Reference Template 2 (optional)</i>	<i>L2 bytes</i>
	...	
	<i>Reference Template n (optional) (n < 10)</i>	<i>Ln bytes</i>
	<i>Matching Score (optional)</i>	<i>4 bytes</i>

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 (SAGEM DS 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 i

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

- [PK_COMP_V2](#) or [PK_MAT](#),
- [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 a verification against more than one database record.

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.

Reply

Identifier value	CMD_VERIFY [0x20]	1 byte
Length value	2 (+ 7)	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Matching result</i>	<i>1 byte</i>
	<i>Matching Score (optional)</i>	<i>7 bytes</i>

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_INVALID_MINUTIAE [0xFD]	The reference ILV minutiae is not valid: bad identifier, corrupted minutiae.
ILVERR_TIMEOUT [0xFA]	The finger detection timeout has expired.

Matching result

ILVSTS_HIT [0x01]	The comparison succeeded.
ILVSTS_NO_HIT [0x02]	It is not the same finger.
ILVSTS_DB_EMPTY [0x05]	The database is empty.

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 the [Matching Score](#) sub ILV is optional.

ENROLL

Description

This function captures or enrolls 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 to get 3 images for enrollment purpose, and 1 image for verification purpose. An enrollment 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 fingers.

Command

Request

Identifier value	CMD_ENROLL [0x21]	1 byte
Length value	8 + LUID + LData1 + ... + LData _i bytes	2 bytes
Value (Parameters)	Database identifier	1 byte
	Timeout	2 bytes
	Acquisition quality threshold	1 byte
	Enrollment type	1 byte
	Number of fingers	1 byte
	Save record	1 byte
	Export Minutiae Size	1 byte
	<i>User ID (optional, depending on enroll type)</i>	<i>LUID bytes</i>
<i>Additional user data field 1 (optional)</i>	<i>LData1 bytes</i>	
...		
<i>Additional user data field i (optional)</i>	<i>LData_i bytes</i>	

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Timeout

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

Acquisition quality threshold

Not used. Set to 0.

Enrollment Type

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

We strongly recommend to set this value to 0 (default value) or 3 for enrollment 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 enroll. This function can enroll 1 or 2 fingers.

Set this value to 0x01 to enroll 1 finger.

Set this value to 0x02 to enroll 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 Enrollment 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 compress the template. The buffer exported will have the size you have chosen at the most, but could be less.

User ID

This is 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.

The User Id is a 24-characters length (max) ASCII string. "8516757" or "John Smith".



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.

Reply

Identifier value	CMD_ENROLL [0x21]	1 byte
Length value	6 (+ LBIO1) (+ LBIO2)	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Enroll status</i>	1 byte
	<i>User database index</i>	4 bytes
	<i>Biometric data 1</i>	LBIO1 bytes
	<i>Biometric data 2 (optional)</i>	LBIO2 bytes

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_TIMEOUT [0xFA]	The finger detection timeout has expired.
ILVERR_INVALID_USER_DATA [0xFB]	The input ILV user data is not valid: bad identifier or wrong size.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_SAME_FINGER [0xE4]	User used the same finger twice.
ILVERR_FIELD_NOT_FOUND [0xE9]	Invalid field number.
ILVERR_INVALID_USER_ID [0xFC]	The specified User ID is already used in the database.
ILVERR_ALREADY_ENROLLED [0xF8]	The person is already in this database

Enroll status

ILVSTS_OK [0x00]	The enrollment succeeded.
ILVSTS_DB_FULL [0x04]	This status can be returned if the Save Record is TRUE. It means that the maximum number of users that can be stored in the local database has been reached.

User database index

This is the database index of the record (person).

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 enroll status differs from ILVSTS_OK the *User Database Index* is not returned.

Biometric data i

The resulting templates returned by the terminal if the *Export Minutiae Size* is different than 0. The returned template is a compressed PK.

See the *PK_COMP V2* structure for details.

Compatibility note

Only the database 0 is available.

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

Compressed PK are not padded with 0x00.

IDENTIFY

Description

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

Command

Request

Identifier value	CMD_IDENTIFY [0x22]	1 byte
Length value	6 (+4)	2 bytes
Value (Parameters)	Database Identifier	1 byte
	Timeout	2 bytes
	Matching Threshold	2 bytes
	RFU	1 byte
	<i>Matching Score (optional)</i>	<i>4 bytes</i>

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

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 (SAGEM DS 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.

Reply

Identifier value	CMD_IDENTIFY [0x22]	1 byte
Length value	2 (+ 7)	2 bytes
Value (Parameters)	Request Status	1 byte
	<i>Matching Result</i>	<i>1 byte</i>
	<i>User Database Index</i>	<i>4 bytes</i>
	<i>User ID (Field 0)</i>	<i>LUID bytes</i>
	<i>Additional user data field 1</i>	<i>L1 bytes</i>
	...	
	<i>Additional user data field n</i>	<i>Ln bytes</i>
	<i>Matching score (optional)</i>	<i>7 bytes</i>

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_TIMEOUT [0xFA]	The finger detection timeout has expired.

Matching Result.

ILVSTS_HIT [0x01]	The comparison succeeded
ILVSTS_NO_HIT [0x02]	It is not the same finger
ILVSTS_DB_EMPTY [0x05]	The database is empty.

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

This is the unique user identifier saved in the database. The User ID is only returned under Hit condition.

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.

DATABASE FUNCTIONS DESCRIPTION

CREATE DATABASE

Description

This function creates a biometric database in flash memory.

Command

Request

Identifier value	CMD_CREATE_DB [0x30]	1 byte
Length value	5 (+ L1 + L2 + ... + Li)	2 bytes
Value (Parameters)	Database identifier	1 byte
	RFU	1 byte
	Person number in database	2 bytes
	Finger number per person	1 byte
	<i>Additional data Field 1</i>	<i>L1 bytes</i>
	<i>Additional data Field 2</i>	<i>L2 bytes</i>
	<i>Additional data Field i</i>	<i>Li bytes</i>

Database identifier

Current release of MorphoAccess™ series does not support management of multiple databases. Set this parameter to 0.

RFU

Not used, set to 0.

Person number in database

Must not exceed 500.

Finger number per person

Number of finger per person in the database (1 or 2).

Additional data Fields

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

The maximum number of additional fields is 16 when the number of database records is inferior or equal to 100.

The maximum number of additional fields is 2 when the number of database records is superior to 100.

See the [Private Field](#) and [Public Field](#) structures for details.

Reply

Identifier value	CMD_CREATE_DB [0x30]	1 byte
Length value	1	2 bytes
Value (Parameters)	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_BASE_ALREADY_EXISTS [0xF6]	The <i>Database Identifier</i> is wrong, or this database already exists
ILVERR_NO_SPACE_LEFT [0xF2]	The Database can not be created because there is not enough memory
ILVERR_BADPARAMETER [0xFE]	Wrong number of finger, ILV Format incorrect or invalid database configuration.
ILVERR_OUT_OF_FIELD [0xEB]	The number of additional fields is superior to 16 or 2 (depending of the number of database records)
ILVERR_FIELD_INVALID [0xE8]	Additional field name length is more than 6

Compatibility note

The [Public Field](#) structure correspond to a standard additional field.

The *Admin Field* and *Time Mask Field* are no more managed.

ERASE BASE

Description

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

Command

Request

Identifier value	CMD_ERASE_DB [0x32]	1 byte
Length value	1	2 bytes
Value (Parameters)	Database identifier	1 byte

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Reply

Identifier value	CMD_ERASE_DB [0x32]	1 byte
Length value	1	2 bytes
Value (Parameters)	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_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_BADPARAMETER [0xFE]	Wrong number of finger, ILV Format incorrect or invalid database configuration.

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

Identifier value	CMD_ERASE_ALL_DB [0x33]	1 byte
Length value	1	2 bytes
Value (Parameters)	RFU	1 byte

RFU

Not used, set to 0.

Reply

Identifier value	CMD_ERASE_ALL_DB [0x33]	1 byte
Length value	1	2 bytes
Value (Parameters)	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]	Wrong number of finger, ILV Format incorrect or invalid database configuration.

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

Identifier value	CMD_ADD_DB_RECORD [0x35]	1 byte
Length value	1 + LUID + LData1 + ... + LData _i bytes	2 bytes
Value (Parameters)	Database identifier	1 byte
	Reference template 1	L1 bytes
	<i>Reference template 2 (optional)</i>	<i>L2 bytes</i>
	User ID	LUID bytes
	<i>Additional User Data Field 1 (optional)</i>	<i>LData1 bytes</i>
	...	
	<i>Additional User Data Field i (optional)</i>	<i>LData_i bytes</i>
	<i>No check on template (optional)</i>	<i>4 bytes</i>

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Reference Templates i

The following templates format can be used: PK_COMP V2, PK_MAT or X984 biometric token. The maximum number of referenced finger depends on the number of finger supported by the database.

User ID

This is the unique user identifier. The User ID can be retrieved by the [Identify](#) function under Hit condition. The user ID length is 24 characters max, in ASCII format. “73259” or “John Smith” for example.



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 can not 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

Identifier value	CMD_ADD_DB_RECORD [0x35]	1 byte
Length value	2 (+ 7)	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Base status</i>	1 byte
	<i>User database index</i>	4 bytes

Request status

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

ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_INVALID_USER_DATA [0xFB]	The input ILV user data is not valid: bad identifier or wrong size.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_SAME_FINGER [0xE4]	User used the same finger twice.
ILVERR_FIELD_NOT_FOUND [0xE9]	Invalid field number.
ILVERR_INVALID_USER_ID [0xFC]	The specified User ID is already used in the database.
ILVERR_INVALID_MINUTIAE [0xFD]	The reference ILV minutiae is not valid: bad identifier, corrupted minutiae.
ILVERR_ALREADY_ENROLLED [0xF8]	The person is already in this database

Base Status

ILVSTS_OK [0x00]	The enrollment succeeded.
ILVSTS_DB_FULL [0x04]	The maximum number of users that can be stored in the database has been reached.

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.

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

Identifier value	CMD_REMOVE_DB_RECORD [0x36]	1 byte
Length value	1 + L1	2 bytes
Value (Parameters)	Database identifier	1 byte
	ILV User ID or User Index	L1

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

User ID or User Index

User ID: This is the unique user identifier saved in the database.



This field is managed as a byte array.

User Index: This is the unique database index returned by [Enroll](#) or [Add Base Record](#).

Reply

Identifier value	CMD_REMOVE_DB_RECORD [0x36]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request status	1 byte

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_INVALID_USER_ID [0xFC]	The specified User ID is already used in the database.

Compatibility note

 The [Time Stamp](#) sub ILV is ignored.

FIND USER BASE

Description

This function searches the first record which has a field equal to a given field (same length, same content). Search is processed only on public fields.

Command

Request

Identifier value	CMD_FIND_USER_BASE [0x38]	1 byte
Length value	9 + L	2 bytes
Value (Parameters)	Database identifier	1 byte
	Field Index	4 bytes
	Record Offset	4 bytes
	Data to Find	L bytes

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Field Index

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

Record Offset

Search starts from this record offset.

Data to Find

An [Additional User Data](#) structure that contains the data to find.

See the [Additional User Data](#) structure for details.

Reply

Identifier value	CMD_FIND_USER_BASE [0x38]	1 byte
Length value	5	2 bytes
Value (Parameters)	Request Status	1 byte
	<i>Record Index</i>	4 bytes

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_USER_NOT_FOUND [0xE6]	The required User ID has not been found in the database.

Record Index

Record of the found index.

Compatibility note

This command is a MorphoAccess™ 100 series new functionality.

GET DATA

Description

This function reads a database public field.

Command

Request

Identifier value	CMD_GET_DATA [0x3F]	1 byte
Length value	5 + L	2 bytes
Value (Parameters)	Database identifier	1 byte
	Field index	4 bytes
	ILV User ID or User Index	L

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Field Index

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

User ID or User Index

User ID: This is the unique user identifier saved in the database.

User Index: This is the unique database index returned by *Enroll* or *Add Base Record*.

Reply

Identifier value	CMD_GET_DATA [0x3F]	1 byte
Length value	1 + L	2 bytes

Value (Parameters)	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.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_USER_NOT_FOUND [0xE6]	The required User ID has not been found in the database.
ILVERR_FIELD_NOT_FOUND [0xE9]	The required field doesn't exist in the database.

Field

The field content.

Compatibility note

This command is a MorphoAccess™ 100 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

Identifier value	CMD_GET_PUBLIC_FIELDS [0x3E]	1 byte
Length value	5 + L	2 bytes
Value (Parameters)	Database identifier	1 byte
	Field index	4 bytes

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Field Index

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

Reply

Identifier value	CMD_GET_PUBLIC_FIELDS [0x3E]	1 byte
Length value	5 + N1 + ... + Nk	2 bytes
Value (Parameters)	Request Status	1 byte
	User Number	4 bytes
	Field 1	N1 bytes
	Field 2	N2 bytes
	...	

 Field k Nk bytes

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.
ILVERR_USER_NOT_FOUND [0xE6]	The required User ID has not been found in the database.
ILVERR_FIELD_NOT_FOUND [0xE9]	The required field doesn't exist in the database.

User Number

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

Field I

An *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 new functionality.

UPDATE PUBLIC DATA

Description

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

Command

Request

Identifier value	CMD_UPDATE_PUBLIC_DATA [0x3C]	1 byte
Length value	0x0002 + L ₁ + L ₂ + ... + L _n	2 bytes
Value (Parameters)	Database identifier	1 byte
	Field number = n+1	1 byte
	User ID or User Index	L1
	Field0	L2
	Field1	L3
	Fieldn	Ln

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Field number

Indicate number of fields to update.

User ID

This is the unique user identifier saved in the database.

User Index

This is the unique database index returned by [Enroll](#) or [Add Base Record](#).

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

Identifier value	CMD_UPDATE_PUBLIC_DATA [0x3C]	1 byte
Length value	0x0001	2 bytes
Value (Parameters)	Request Status	1 byte

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.

Compatibility note

This command is a MorphoAccess™ 100 series new functionality.

GET BASE CONFIG

Description

This function retrieves the configuration of the local database.

Command

Request

Identifier value	CMD_GET_DB_CONFIG [0x07]	1 byte
Length value	1	2 bytes
Value (Parameters)	Database identifier	1 byte

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Reply

Identifier value	CMD_GET_DB_CONFIG [0x07]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request Status	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	Lts bytes
	Additional data Field 1	L1 bytes
	Additional data Field 2	L2 bytes
	...	

	Additional data Field i	Li bytes
Request status		
ILV_OK [0x00]	The execution succeeded.	
ILVERR_ERROR [0xFF]	Execution failed.	
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.	
ILVERR_BASE_NOT_FOUND [0xF7]	The specified database does not exist.	
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)		
Identifier value	ID_TIMESTAMP [0x11]	1 byte

Length value	L	2 bytes
Value (Parameters)	ASCII string with Day, Month, Years, Hours, L bytes minutes and Seconds each encoded with 2 bytes: "DDMMYYHHmmSS".	


Additional data Fields

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

The maximum number of additional fields is 16 when the number of database records is less or equal to 100.

The maximum number of additional fields is 2 when the number of database records is superior to 100.

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

Identifier value	CMD_GET_ALL_DB_CONFIG [0x31]	1 byte
Length value	0	2 bytes

Reply

Identifier value	CMD_GET_ALL_DB_CONFIG [0x31]	1 byte
Length value	2 + L1 + ... + Ln	2 bytes
Value (Parameters)	Request Status	1 byte
	Nb Returned Bases	1 byte
	Base 1 configuration	L1 bytes
	...	
	Base i configuration	Li bytes
	...	
	Base n configuration	Ln bytes

Request status

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

NB Returned Bases

Current release of MorphoAccess™ 100 series does not support management of multiple 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.

LOG FUNCTIONS DESCRIPTION

OVERVIEW

The MorphoAccess™ 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 Information is:

- 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 8000 events.

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:

Record Date (12 bytes)
323530373032313533353235

will correspond to 25/07/02 15:35:25.

- **Action type** byte describes the nature of the event logged. There are two major log families : Authent 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 organised.

AUTHENTICATION 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_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 char    m_puc_not_used;
    unsigned char    m_uc_MatchFinger;
    unsigned char    m_uc_Pad[32];
} T_LOG_AUTHENT;
```

Definitions

```
#define LOG_DATE_SIZE        12
#define LOG_BASE_NAME_SIZE  15
#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
	Contactless authentication with database (templates are stored on the database).	
	AUTH_CTL_WDB_ID 0x25	
	Contactless authentication without database	
	AUTH_CTL_ID_ONLY (bypass mode) 0x2A	
	AUTH_CTL_PKS 0x2B	
Acquisition threshold	m_uc_AcquisitionTh	1
	0	
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	
Number of finger	m_uc_NbFinger	1
	2	
Matched finger number	m_uc_MatchFinger	1

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_puc_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_Pad[20];
} 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] Format : DDMMYYHHMMSS.	12
Action type	m_uc_Action Local ident TLOG_IDENT 0x30	1
Base support	m_uc_FlashType 0	1
Base name	m_puc_BaseName[15] "bioa"	15
User Id	m_puc_UserId[20] "98756" for example	20
Timeout	m_us_Timeout	2
Operation duration	m_us_Duration (in tenth of second)	2
User index	m_s_PersonIndex -1 if failed	2
Number of finger	m_uc_Result Success LOG_IDENT_OK 0x00 Not recognized LOG_IDENT_FAILED 0x01 Timeout LOG_IDENT_TIMEOUT 0x19 Generic error LOG_IDENT_ERROR 0xFF	1

COMPATIBILITY NOTES



Enrolment operations are no more logged.



Distant operations (ILV commands) are no more logged.

GET LOG STATUS

Description

This command will return the state of the log file.

Command

Request.

Identifier value	CMD_GET_LOG_STATUS [0x5D]	1 byte
Length value	1	2 bytes
Value (Parameters)	Log Base ID	1 byte

Log Base ID

Base number. Must be 0. (RFU)

Reply

Identifier value	CMD_GET_LOG_STATUS [0x5D]	1 byte
Length value	10	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Log Enabled</i>	1 byte
	<i>Current Line</i>	2 bytes
	<i>Max Number of Lines</i>	2 bytes
	<i>Line Size</i>	2 bytes
	<i>Base State</i>	1 byte
	<i>Version</i>	1 byte

Request status

ILV_OK [0x00]

The execution succeeded.

ILVERR_ERROR [0xFF] Execution failed.
ILVERR_BAD_PARAMETER [0xFE] Invalid parameters.

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 file revision. 0xXY for « X.Y ».

ERASE LOG

Description

This command will erase all the record written in the log file.

Command

Request.

Identifier value	CMD_ERASE_LOG [0x5C]	1 byte
Length value	1	2 bytes
Value (Parameters)	Log Base ID	1 byte

Log Base ID

Base number. Must be 0. (RFU)

Reply

Identifier value	CMD_ERASE_LOG [0x5C]	1 byte
Length value	2	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Erase status</i>	1 byte

Request status

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

Erase status (if Request status is ILV_OK)

ILVSTS_OK [0x00]	Base successfully erased.
------------------	---------------------------

ILVSTS_FLASH_ERROR [0x0A]

The execution failed.

GET LOG

Description

Use this command to download the database.

Command

Request.

Identifier value	CMD_GET_LOG [0x5B]	1 byte
Length value	5	2 bytes
Value (Parameters)	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	<i>Number Of Lines</i> Downloaded lines
...	
Last downloaded line]	
...	<i>Offset From Current Line</i>
Line k-1 : 64 bytes.	
Line k : 64 bytes.	
...	
Line 3840 : 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 "*Max Number of Lines*".

Reply

Identifier value	CMD_GET_LOG [0x5B]	1 byte
Length value	1+ 2+ L	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Number of lines</i>	<i>2 bytes</i>
	<i>Data</i>	<i>L bytes</i>

Request status

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

Number of lines

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

Data

Lines in binary format.

Compatibility note

In [Distant Session Mode](#), the complete database can be downloaded in 1 request.

CONTACTLESS FUNCTIONS DESCRIPTION

CONTACTLESS VERIFICATION (OR CONTACTLESS AUTHENTICATION)

Description

This function reads templates on a Mifare™ card then captures a finger and checks if it matches.

Command

Request

Identifier value	CMD_CONTACTLESS_VERIFY [0x59]	1 byte
Length value	5 + 6 ... (+6)	2 bytes
Value (Parameters)	Timeout	2 bytes
	Matching threshold	2 bytes
	Acquisition quality threshold	1 byte
	Address of reference Template 1	6 bytes
	Address of reference Template 2 (optional)	6 bytes
	Address of data for hashing (ignored and optional)	6 bytes
	Address of signature (ignored and optional)	6 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 (SAGEM DS 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.

Address of reference Template i

This parameters give addresses containing the reference minutiae of fingerprint.

See the [Address Biometric Data](#) structure for details.

Reply

Identifier value	CMD_CONTACTLESS_VERIFY [0x59]	1 byte
Length value	2	2 bytes
Value (Parameters)	Request status	1 byte
	<i>Matching result</i>	<i>1 byte</i>

Request status

ILV_OK [0x00]	The execution succeeded.
ILVERR_ERROR [0xFF]	Execution failed.
ILVERR_BAD_PARAMETER [0xFE]	Invalid parameters.
ILVERR_INVALID_MINUTIAE [0xFD]	The reference ILV minutiae is not valid: bad identifier, corrupted minutiae.
ILVERR_TIMEOUT [0xFA]	The finger detection timeout has expired.

Matching result

ILVSTS_HIT [0x01]	The comparison succeeded.
ILVSTS_NO_HIT [0x02]	It is not the same finger.
ILVSTS_DB_EMPTY [0x05]	The database is empty.

Note

See also [Set Contactless Keys](#) function to change the contactless keys.

The [Contactless card mapping](#) section describes the card structure.

CONTACTLESS CARD FUNCTION

Description

This function is an interface to perform read operations on Mifare™ cards.

Command

Request

Identifier value	CMD_CLESS_CARD_FUNCTION [0x5F]	1 byte
Length value	1+L	2 bytes
Value (Parameters)	Function ID	1 byte
	Function	3 or 5 bytes

Function ID

ID of the function to perform.

- AC_READ_SERIAL_NUMBER [0x1B]: Read serial number of a Mifare™ card. In this case the Function parameter is ignored. Non-blocking.
- AC_READ_DATA_CARD [0x1C]: Read data on a Mifare™ card. Non-blocking.
- AC_READ_SERIAL_NUMBER_TIMEOUT [0x2B]: Read serial number of a Mifare™ card. In this case the Function parameter is ignored. A timeout can be defined.
- AC_READ_DATA_CARD_TIMEOUT [0x2C]: Read data on a Mifare™ card. A timeout can be defined.

Function

Function ID = AC_READ_SERIAL_NUMBER	3 bytes
: Ignored	1 bytes
<N>: Ignored	1 bytes
<C>: Ignored	1 bytes

Function

Function ID = AC_READ_DATA_CARD	3 bytes
: First block number to read (1 to 48)	1 byte
<N>: Number of blocks to read (1 to 48)	1 byte
<C>: Key number for the read:	1 byte
1 Mifare™ security key A then B is selected	
2 Mifare™ security key A is selected.	
3 Mifare™ security key B is selected.	

Function

Function ID = AC_READ_SERIAL_NUMBER_TIMEOUT	5 bytes
: Ignored	1 byte
<N>: Ignored	1 byte
<C>: Ignored	1 byte
Card detection timeout in seconds. A value of 0 corresponds to an infinite timeout.	2 bytes

Function

Function ID = AC_READ_DATA_CARD_TIMEOUT	5 bytes
: First block number to read (1 to 48)	1 byte
<N>: Number of blocks to read (1 to 48)	1 byte
<C>: Key number for the read:	1 byte
1 Mifare™ security key A then B is selected	
2 Mifare™ security key A is selected.	
3 Mifare™ security key B is selected.	
Card detection timeout in seconds. A value of 0 corresponds to an infinite timeout.	2 bytes

Example

I	L	Function ID	B	N	C	Timeout		
0x5F	0x06	0x00	0x2c	0x04	0x08	0x01	0x05	0x00

This frame allows reading **8** blocks from block **4** using key A then B. If no card is present after 5 seconds, "STATUS_CARD_NO_PRESENT" is returned.

I	L	Function ID	B	N	C	
0x5F	0x04	0x00	0x1B	0x00	0x00	0x00

This frame returns the serial number of the card present in the field or returns immediately.

Reply

Identifier value	CMD_CLESS_CARD_FUNCTION [0x5F]	1 byte
Length value	3+L	2 bytes
Value (Parameters)	Function ID	1 byte
	Contactless Status	1 byte
	Function Data	L bytes

Contactless Status. Returned only if Request Status is ILV_OK.

STATUS_CARD_PRESENT [0x00]	The function succeeded.
STATUS_AUTHENT_ERROR [0x0C]	A card was detected but the authentication failed.
STATUS_CARD_NO_PRESENT [0x0B]	Timeout occurs while waiting for a card
STATUS_ERROR [0x0F]	A card was detected but an error occurred.

Function ID

ID of the performed function.

- AC_READ_SERIAL_NUMBER [0x1B]
- AC_READ_DATA_CARD [0x1C]
- AC_READ_SERIAL_NUMBER_TIMEOUT [0x2B]
- AC_READ_DATA_CARD_TIMEOUT [0x2C]

Function Data. Returned only if Request Status is ILV_OK.

If function ID is AC_READ_SERIAL_NUMBER [0x1B]: function data is a L bytes ASCII string containing the card serial number.

If function ID is AC_READ_DATA_CARD [0x1C]: function data is a L bytes binary data read on the card.

Note

See also [Set Contactless Keys](#) function to change the contactless keys.

The [Contactless card mapping](#) section describes the card structure.

Compatibility note

It can also be combined with an authentication to proceed to a contactless authentication.

PROXY MODE FUNCTIONS DESCRIPTION

OVERVIEW

This mode allows controlling the MorphoAccess™ remotely using a set of biometric and database management function interface access commands.

In this mode the terminal is entirely commanded by a client device.

All ILV commands are available, but command presented in this section are reserved to *proxy mode*. When a MorphoAccess™ terminal is not in *proxy mode*, these commands are invalid. Therefore its reply is CMD_INVALID_REQUEST (ID = 0x50).

ACCESS AUTHORIZATION [PROXY MODE ONLY]

Description

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

Command

Request

Identifier value	CMD_ACCESS_AUTHORIZATION [0x56]	1 byte
Length value	1	2 bytes
Value (Parameters)	Access Authorization	1 byte

Access Authorization

0x00, access refused: the led is red; a long “beep” is emitted.

0x01, access granted: the led is green, a short “beep” is emitted and the relay switches. The duration is defined in the relay parameters.

Reply

Identifier value	CMD_ACCESS_AUTHORIZATION [0x56]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request Status	1 byte

Request status

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

Compatibility note

This command controls the led and the buzzer. It was not the case before.

DATA SEND [PROXY MODE ONLY]

Description

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

Command

Request

Identifier value	CMD_DATA_SEND [0x51]	1 byte
Length value	L	2 bytes
Value (Parameters)	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 [SAGEM ONLY].

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

Reply

Identifier value	CMD_WIEGAND_DATACLK_SEND [0x51]	1 byte
Length value	1	2 bytes
Value (Parameters)	Request Status	1 byte

Request Status

ILV_OK [0x00]

The execution succeeded.

ILVERR_ERROR [0xFF]

Execution failed.

ILVERR_BAD_PARAMETER [0xFE]

Invalid parameters.

BIOMETRIC DATA DESCRIPTION

PK_COMP V2

Description

This is the MorphoAccess™ native template format.

This template format is compatible with the other SAGEM biometric terminal and system.

The maximum template size is 256 bytes but it can be compressed to 170 bytes without loss of information.

SAGEM recommends using this template format.

Data structure

Identifier value	PK_COMP V2 [0X02]	1 byte
Length value	L	2 bytes
Value (Parameters)	Minutiae	L bytes

Note

ID_PK_COMP_NORM [0x55] can be used instead of ID_PK_COMP. This is reserved for compatibility with existing systems or specific usage. For more information please contact SAGEM.

The PK_COMP size depends on the number of minutiae. It is usually less than 100 bytes.

See also [Verify](#), [Enroll](#), [Add Base Record](#).

PK_MAT

Description

This is an oldest template format that is still used for compatibility with existing systems.

This template format is only compatible with MorphoKit™ or AFIS that have encoded the PK_MAT in little endian. (PK_MAT generated by MorphoTouch™ are encoded in big endian)

The template size is equal to 512 bytes.

Data structure

Identifier value	PK_MAT [0X03]	1 byte
Length value	512	2 bytes
Value (Parameters)	Minutiae	512 bytes

Note

ID_PK_MAT_NORM [0x35] can be used instead of ID_PK_MAT. This is reserved for compatibility with existing systems or specific usage. For more information please contact SAGEM Défense Sécurité.

See also [Verify](#), [Enroll](#), [Add Base Record](#).

DATABASE DATA DESCRIPTION

USER INDEX

Description

ILV formatted data that contains the unique internal database index for a record. It is managed by the MorphoSmart database. The value is returned by ENROLL or ADD_BASE_RECORD.

Data structure

Identifier value	ID_USER_INDEX [0X36]	1 byte
Length value	4	2 bytes
Value (Parameters)	Index	4 bytes

Note

See also: [Remove Record](#), [Get Data](#), Update Public data.

USER ID

Description

ILV formatted data that contains 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 enrollment, 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

Identifier value	ID_USER_ID [0x04]	1 byte
Length value	L	2 bytes
Value (Parameters)	ID User	L bytes

Note

See also: [Enroll](#), [Identify](#), [Add Base Record](#), [Remove Record](#).

PKBASE

Description

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

Data structure

Identifier value	PKBASE [0x3A]	1 byte
Length value	1+ L _{UID}	2 bytes
Value (Parameters)	Database identifier	1 byte
	ILV User ID or User Index	L _{UID}

Database identifier

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Note

See also [Verify](#).

BASE CONFIGURATION

Description

This ILV contains the database configuration.

Data structure

Identifier value	ID_BASE_CONF [0x3A]	1 byte
Length value	19	2 bytes
Value (Parameters)	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

Current release of MorphoAccess™ 100 series does not support management of multiple databases. Set this parameter to 0.

Base Valid

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

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

ADDITIONAL DATA FIELD DESCRIPTION

FIELD CONTENT

Description

ILV formatted data packet that contains field content.

Data structure

Identifier value	ID_FIELD_CONTENT [0x32]	1 byte
Length value	4 + L	2 bytes
Value (Parameters)	Data Index	4 bytes
	Data length	4 bytes
	Data	L bytes

Data Index

Field index which 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

Identifier value	ID_PUBLIC_FIELD or ID_FIELD [0x0F]	1 byte
Length value	2 + L	2 bytes
Value (Parameters)	Field size	2 bytes
	Field name	L bytes

Field Size

Define 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 must not 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

Identifier value	ID_PRIVATE_FIELD [0x31]	1 byte
Length value	2 + L	2 bytes
Value (Parameters)	Field size	2 bytes
	Field name	L bytes

Field Size

Define 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 private field must not exceed 32 bytes when the number of database records is superior to 100.

See also: [Create Database](#), [Get Base Config](#).

ADDITIONAL USER DATA

Description

One 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 content of the buffer must be defined by the user. All the additional user data fields defined in the [Create Database](#) command have to be set in the [Enroll](#) command. Size and order of additional data fields must be consistent with database structure.

Data structure

Identifier value	ID_PUC_DATA [0x14]	1 byte
Length value	L	2 bytes
Value (Parameters)	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 128 bytes.

If a field is empty, its length is 0.

Note

All the '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: [Enroll](#), [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

Identifier value	ID_NO_CHECK_ON_TEMPLATE [0x60]	1 byte
Length value	1	2 bytes
Value (Parameters)	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](#).

WIEGAND DATACLOCK 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

Identifier value	ID_WIEGAND_DATA [0x11]	1 byte
Length value	12	2 bytes
Value (Parameters)	Site	4 bytes
	ID	4 bytes
	Custom	4 bytes

Note

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

DATALOCK DATA

Description

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

Data structure

Identifier value	ID_DATALOCK_DATA [0x21]	1 byte
Length value	N	2 bytes
Value (Parameters)	ID	n bytes

Note

ID is an null terminated ASCII string. Null character is not sent on Dataclock output.

CONTACTLESS DATA DESCRIPTION

ADDRESS BIOMETRIC DATA

Description

This parameters give addresses containing the reference minutiae of fingerprint. The reference minutiae are compressed with algorithm PKCOMP (with MorphoKit for example.)

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

Data structure

Identifier value	ID_ADDRPK [0x18]	1 byte
Length value	3	2 bytes
Value (Parameters)	Address	3 bytes

Address

An address of card parameter has this structure:

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

: 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 a absolute way, 1 for block 0 sector 0, then 3 blocks for each sectors.

CONFIGURATION DATA DESCRIPTION

DATE AND TIME CONFIGURATION

Description

This packet configures the time and the date of the system.

Data structure

Identifier value	ID_DATE [0x13]	1 byte
Length value	13	2 bytes
Value (Parameters)	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 sting is NULL terminated.

MISCELLANEOUS DATA DESCRIPTION

MATCHING SCORE

Description

This ILV is optional. It can be used in a request to force the MorphoAccess™ to return the matching score. In a reply, it contains the matching score value.

Data structure

Identifier value	ID_MATCHING_SCORE [0x56]	1 byte
Length value	1 or 4	2 bytes
Value (Parameters)	Value or Score	1 or 4 bytes

Value or Score

Value: Set to a value different from 0 to force the MorphoSmart™ to send the resulting matching score.

Score: Resulting 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™ to return the matching score. In a reply, it contains the matching score value.

Data structure

Identifier value	ID_LATENT_SETTING [0x39]	1 byte
Length value	1	2 bytes
Value (Parameters)	Value	1 byte

Value

0x00: the fingerprint latent detection is disabled

0x01: the fingerprint latent detection is enabled.

Note

See also: [Enroll](#).

CONSTANT VALUES

IDENTIFIER - STATUS

Remote management

DISTANT_SESSION_OPENED	1
DISTANT_SESSION_CLOSED	2
DISTANT_SESSION_MEMORY_ERROR	0xE0
DISTANT_SESSION_GENERIC_ERROR	0xFF

ERROR CODES VALUE

ILV_OK	0	0x00
ILVERR_ERROR	-1	0xFF
ILVERR_BADPARAMETER	-2	0xFE
ILVERR_INVALID_MINUTIAE	-3	0xFD
ILVERR_INVALID_USER_ID	-4	0xFC
ILVERR_INVALID_USER_DATA	-5	0xFB
ILVERR_TIMEOUT	-6	0xFA
ILVERR_INVALID_ID_PROTOCOL	-7	0xF9
ILVERR_ALREADY_ENROLLED	-8	0xF8
ILVERR_BASE_NOT_FOUND	-9	0xF7
ILVERR_BASE_ALREADY_EXISTS	-10	0xF6
ILVERR_BIO_ENCOURS	-11	0xF5
ILVERR_CMD_ENCOURS	-12	0xF4
ILVERR_FLASH_INVALID	-13	0xF3
ILVERR_NO_SPACE_LEFT	-14	0xF2
ILVERR_ADMIN_ENCOURS	-15	0xF1

ILVERR_BAD_SIGNATURE	-16	0xF0
ILVERR_CARD_PROBLEM	-17	0xEF
ILVERR_ERROR_SL	-18	0xEE
ILVERR_ERROR_WIEGAND	-19	0xED
ILVERR_ERROR_TCPIP	-20	0xEC
ILVERR_OUTOFFIELD	-21	0xEB
ILVERR_INVALID_ACCESS	-22	0xEA
ILVERR_FIELD_NOT_FOUND	-23	0xE9
ILVERR_FIELD_INVALID	-24	0xE8
ILVERR_SECURITY	-25	0xE7
ILVERR_USER_NOT_FOUND	-26	0xE6
ILVERR_CMDE_ABORTED	-27	0xE5
ILVERR_SAME_FINGER	-28	0xE4
ILVERR_FAKE_FINGER_DETECTION_DISABLE	-29	0xE3
ILVERR_KEY_ALREADY_EXIST	-30	0xE2
ILVERR_NO_SUCH_KEY	-31	0xE1
ILVERR_BAD_CRC	-32	0xE0
ILVERR_BAD_SESSION_STS	-33	0xDF
ILVERR_DIFFERENT_SN	-34	0xDE
ILVERR_BIO_FAILED	-35	0xDD

CONFIGURATION FILES

APPLICATION PARAMETERS

File name: app.cfg

[bio ctrl]

bypass authentication=0

authent timeout=10

authent card mode=0

authent PK contactless=0

authent ID contactless=0

identification=1

[info]

release=3

minor=2

major=1

type=120

[relay]

aperture time in 10 ms=300

enabled=1

[send ID UDP]

host address or name=10.10.161.72

host port=11020

enabled=1

[log file]

enabled=0

[buzzer]

volume=5

[contactless]

C=1

B=4

[send ID wiegand]

valid format=1

custom format=0.0

ID format=9.16

site format=1.8

stop format=3.12

start format=2.12

frame length=26

site code=7

enabled=0

[send ID dataclock]

data inverted=0

clock inverted=0

enabled=0

[failure ID]

generic error ID=65535

timeout ID=65535

not in DB ID=65535

not recognized ID=65535

enabled=0

[send ID RS485]

terminal identifier=145

parity=0

stopbits=1

databits=8

speed=115200

enabled=0

[led IN]

controller ack timeout=300

enabled=0

REMOTE MANAGEMENT PARAMETERS

File name: adm.cfg

[remote management TCP]

inactivity timeout=0

port=11010

[terminal]

group=255

NETWORK PARAMETERS

File name: net.cfg

[boot proto]

DHCP activated=0

[parameters]

network mask=255.255.240.0

default gateway=10.10.160.1

network address=10.10.161.40

host name=MA1210000012

SENSOR PARAMETERS PARAMETERS

File name: bio.cfg

[bio ctrl]

matching th=5

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 THE 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.

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



Sagem Défense Sécurité
SAFRAN Group

Siège social : Le Ponant de Paris

27, rue Leblanc - 75512 PARIS CEDEX 15 - FRANCE