MorphoSmart™ Host System Interface Specification

Version 2.22, Sept 2008

Produced by ***Sagem Sécurité***
Headquarters :
27, rue Leblanc
75512 PARIS CEDEX 12
FRANCE

Customer Service :
Sagem Sécurité
SAV terminaux biométriques
Boulevard Lénine - BP428
76805 Saint Etienne du Rouvray
FRANCE
Tel : +33 2 35 64 55 05

Hotline :
Sagem Sécurité
Support terminaux biométriques
24, Av du gros chêne
95610 ERAGNY – FRANCE
hotline.biometrics@sagem.com
Tel : + 33 1 58 11 39 19
www.biometric-terminals.com

Copyright notices

REVISION HISTORY

Revision	Date	Author	Document revision history
1.0	January17th, 2003	SAGEM SA	First official revision
1.1	May 6th, 2003	SAGEM SA	Enrolment interface evolution + minor modifications
1.2	July21st, 2003	SAGEM SA	Security interface draft (MSO S) + minor modifications
1.3	August 26th, 2003	SAGEM SA	VERIFY, VERIFY MATCH, IDENTIFY and IDENTIFY MATCH can return the matching score.
1.4	September 30th, 2003	SAGEM SA	Security (MSO S): Biometric token
2.0	October 31st, 2003	SAGEM SA	The MSO accepts the PK_MAT and PK_MATNorm template format.
2.1	November 14th, 2003	SAGEM SA	The Enroll function supports the PK_MAT and PK_MATNorm template format. Verify and VerifyMatch accepts up to 10 reference minutiae.
2.2	January 19th, 2004	SAGEM SA	The Enroll function can activate the fingerprint latent detection. New asynchronous message management: MORPHO_FINGER_OK
2.3	February 24th, 2004	SAGEM SA	Add ILV Data Description chapter
2.4	March 19th, 2004	SAGEM SA	Add coder results on Enroll function
2.5	July 16th, 2004	SAGEM SA	ILV_ADD_BASE_RECORD can optionally store records without verification on templates. ILV_CREATE_DATABASE can create a database that manages up to 2000 records. Add explanations to § 6.6 Communication parameter configuration
2.6	November 18th, 2004	SAGEM SA	Add ILV_GET_MSO_CONFIGURATION and ILV_MODIFY_MSO_CONFIGURATION
2.7	December 14th, 2004	SAGEM SA	Add the CBM
2.8	March 18th, 2005	SAGEM SA	Correction database size and add configuration 2 for sensor position
2.9	March 7th, 2005	SAGEM SA	Update of hotline phone number
2.10	June 29th, 2005	SAGEM SA	Add more information about RS232 protocol Add information about USB protocol
2.11	July 8th, 2005	SAGEM SA	Upgrade ILV_Get_Data description Upgrade DataBase size limits description

2.12	July 19th, 2005	SAGEM DS	Upgrade of VERIFY, ENROLL, IDENTIFY and UPGRADE_PRIVATE_DATA : - new request status : ILVERR_FFD and ILVERR_MOIST_FINGER, - new matching results : ILVSTS_FFD and ILVSTS_MOIST_FINGER, - new constant identifier : ID_FFD_LEVEL
2.13	September 14th, 2005	SAGEM DS	New optional ILV data : ID_EXPORT_NUM_PK. If this ILV is used, VERIFY can return the matching PK number. CREATE DATABASE can create a base whose the finger number per person can be set to 1, 2, 10 or 20. VERIFY MATCH can accept up to 20 references templates. Add ILV_ASYNC_MESS_CODEQUALITY and ILV_ASYNC_MESS_DETECTQUALITY to recover the quality scores from the MSO. Modifications of the ILV Asynchronous message and the ILV asynchronous event. Added the transparent upgrade mode chapter.
2.14	January 6th, 2005	SAGEM DS	MSO accepts new templates format as ANSI 378 or ISO 19794-2. Added information in the USB version supported. Added 'Quality note' information. Corrected a return value on IDENTIFY MATCH. Modified template list max size. Added the coder selection optional ILV description. Added the presence detection selection. Added Unlocking ILVs.
2.15	May 2nd, 2006	SAGEM DS	New feature : management of unique use passwords : One Time Password (OTP). New ILVs functions : OTP_ENROLL_USER, OTP_GENERATE, OTP_GET_STATUS, OTP_SET_PARAMETERS. New ILVs data : ID_OTP_SEQUENCE_NUMBER_MAX, ID_OTP_PASSWORD, ID_OTP_PIN, ID_OTP_PARAM, ID_OTP_ALGO_HASH, ID_OTP_SEQUENCE_NUMBER, ID_OTP_SEED, ID_OTP_USER_DATA Modified the GetDescriptor function description. Modified the maximum database size.
2.16	July 19th, 2006	SAGEM DS	GetDescriptor function description update.

2.17	February 9th, 2007	SAGEM DS	Template V10 management update. Chapter 5 (Application Protocol) upgrade. Updgrade descriptions of database functions
2.18	April 20th, 2007	SAGEM DS	Added the 'Biometric Matching Strategy' optional ILV
2.19	June 11th, 2007	SAGEM Sécurité	Added details about the non orientated configuration. Corrected value length for 'Biometric coder selection' and 'Biometric matching Strategy'.
2.20	Sept 4th, 2007	SAGEM Sécurité	Added HMAC-Based OTP : HOTP algorithm
2.21	December 20th, 2007	SAGEM Sécurité	Added details regarding sensor window position
2.22	Sept. 2008	Sagem Sécurité	Upgrade ISO and ANSI template description. Add Thin Finger option in *ID_Coder_Choice*. Add WSQ compression in *ID_Export_Image* (see also *ENROLL* ILV command)

TABLE OF CONTENTS

List of Figures

List of Tables

1 Conventions

This section presents acronyms, abbreviations, measurement units, and conventions used in this document.

1.1 ACRONYMS AND ABBREVIATION

Acronym/Abbreviation	Definition
RFU	Reserved for Future Usage. RFU parameters must be set to 0.
MOBI	Biometric Modules
ILV	Identifier Length Value
SPILV	Service Protocol ILV
CRC	Cyclic Redundancy Check
STX	Start Text
ETX	End Text
MSB	Most Significant Bit
LSB	Least Significant Bit
PK	Characteristics points of a finger used by biometric process
FFD	False Finger Detection
FAR	False Acceptance Rate: probability that a finger is falsely accepted by the system.
FRR	False Rejection Rate: probability that a correct finger is falsely rejected by the system.
BREAK	The International Consultative Committee for Telephone and Telegraph (CCITT) modem recommendations require a signal to be at least "2m+3" bits long, where "m" is the nominal number of bit times in an asynchronous character, usually 10; this means that the minimum time is 23 bits, with no maximum specified. Usually, much more than the minimum is sent, such as 100 or 200 milliseconds (that is, hundreds of bit times at high data rates).
||	Concatenation
OTP	One Time Password

Table 1: Acronyms and Abbreviations

2 Scope

2.1 IDENTIFICATION

This document defines the interface between a MorphoSmart™ device and a host system (typically a PC). It is composed of 3 chapters:

- The first chapter describes the physical layer characteristics,
- The second chapter describes the low layer protocol,
- The third chapter describes the application protocol.

2.2 SOURCE CODE SAMPLE

A Microsoft Visual C++ ® 6.0 project is provided within this SDK as example. It contains sample source code of:

- SPRS232 low layer protocol,
- A sample implementation of the GetDescriptor, Enroll and Verify ILVs.

2.3 ARCHITECTURE OVERVIEW

2.3.1 Generalities

For more information on MorphoSmart™ product functionalities, fingerprint basic knowledge, or on system architecture, please refer to the MorphoSmart™ Overview.pdf documentation.

2.3.2 System Architecture

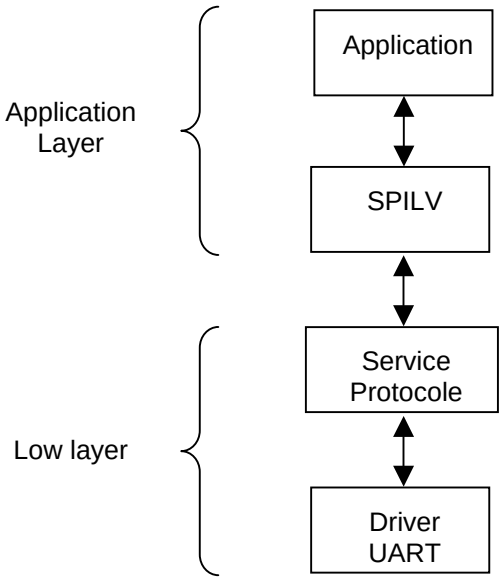


Figure 1: MorphoSmart™ System Architecture

3 Physical layer

The physical layer between the Host and the terminal is RS232.

RS232 Default Configuration is:

Type:	3 wires: TX, RX, GND.
Baudrate:	9600 bps
Data bits:	8 bits
Parity:	none
Stop bits:	1 bit
Flow Control:	Software

After MorphoSmart™ power up, configuration is set to RS232 Default Configuration.

After a BREAK signal, the MorphoSmart™ returns to RS232 Default Configuration, and resets the Request Counter.

RS232 configuration can be modified thanks to high level command CONFIG_UART.

Note: full details are in chapter “Communication parameter negotiation”.

4 Low layer protocol

4.1 RS232 Low Layer Protocol

The RS232 low layer protocol is a symmetric protocol, the rules described in the present section, are integrated in the MorphoSmart™, and have to be implemented also by the calling application in the Host System.

A Microsoft Visual C++ ® 6.0 project (MSO ILV Sample) is provided with MorphoSmart™ SDK contains sample source code of :

- A full SPRS232 low layer protocol
- A sample implementation of the ILV commands Get_Descriptor, Enroll and Verify

4.1.1 Overview

4.1.1.1 Transmission of a Data

When the Transmitter (either the Host or the MorphoSmart™) need to send a Data (either an ILV Command or an ILV Answer) to the Receiver (either the MorphoSmart™ or the Host), the following process is executed:

- The Data is inserted into a Data Packet, and a CRC value is computed, to enable the receiver to check the integrity of the received Data,
- The software handshake characters (XON/XOFF) a “Stuffing” process is executed on the Data Packet before physical transmission to the Receiver.

At the reception of the Data Packet, the Receiver executes the reverse process:

- The Data Packet is “unstuffed” to restore original Data, and the CRC value is checked,
- The Data is extracted from the Data Packet, and provided to the application level.

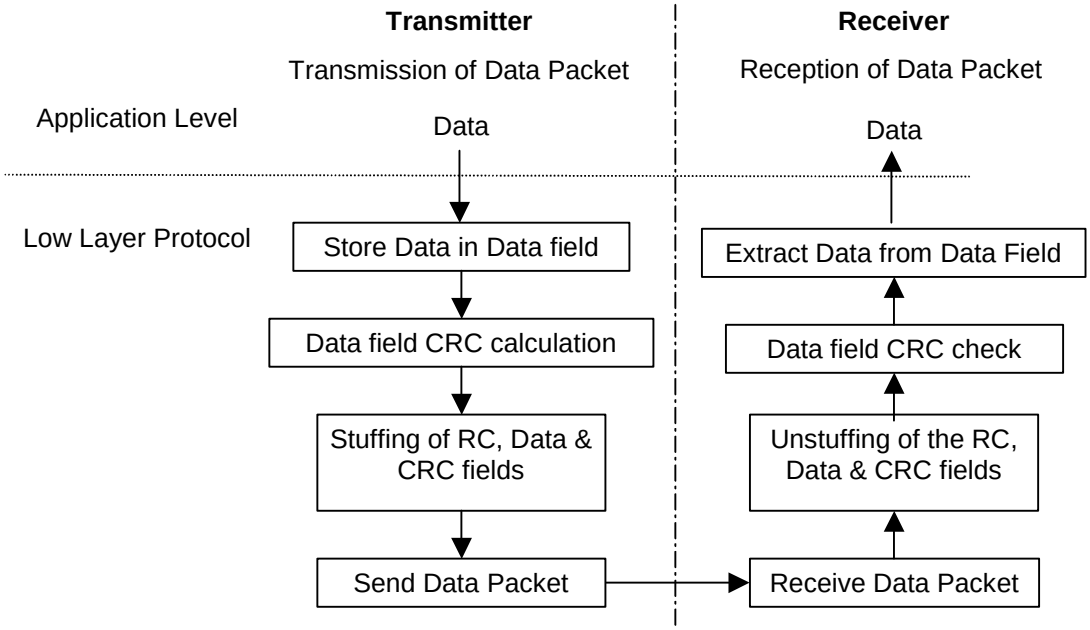


Figure 2 : Data Packet processing overview

4.1.1.2 Packet Acknowledge

In order to secure the transmission, the Receiver acknowledge the reception of each received Data Packet :

- by a ACK Packet if the Data Packet is valid,
- or by a NACK Packet if an error has been detected

The Ack Packet and the Nack Packet return the value of the Request Counter (refer to “Request Counter field” section below), found in the received Data Packet, to confirm to the Transmitter which Data Packet is acknowledged.

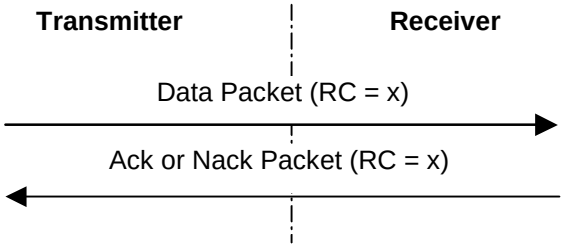


Figure 3 : Data Packet acknowledge overview

The format of the ACK Packet and the NACK packet is described in section below.

The Stuffing process is executed on the ACK/NACK Packet before sending, then at the reception, the ACK/NACK Packet is unstuffed before Request Counter check.

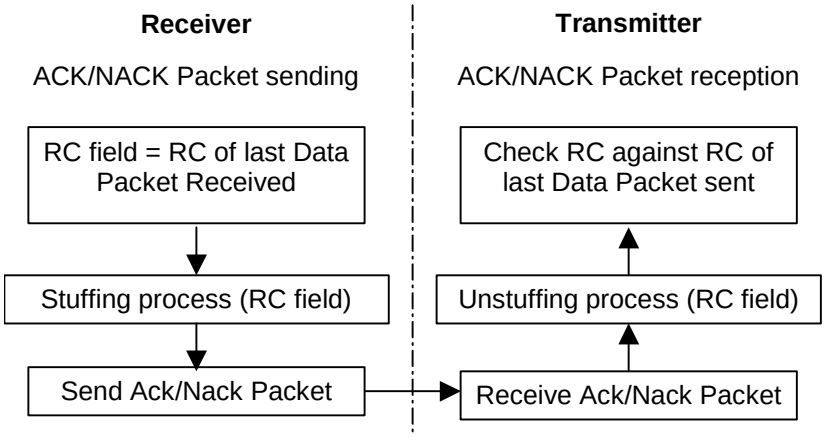


Figure 4 : ACK/NACK Packet processing overview

4.1.2 Packet Definition

4.1.2.1 DATA PACKET STRUCTURE

			Fields to be stuffed			
Start Of Packet					End Of Packet	
STX	ID	RC	**DATA**	CRC	DLE	ETX
1 byte	1 byte	1 byte	1024 bytes max	2 bytes	1 byte	1 byte
Max size when stuffed :		2 bytes	2048 bytes max	4 bytes		

With :

Field name	Definition	Size (bytes)	Value
<STX>	Start Text	1	0x02
<ID>	Packet Identifier	1	Refer to “Packet Identifier” section
<RC>	Request Counter	1	0x00 to 0xFF (Refer to “Request Counter” section)
<DATA>	Data value	1 to 1024	Refer to “Data Field” section
<CRC>	Transmission error control	2	Refer to “CRC” section
<DLE>	Data Link Escape	1	0x1B
<ETX>	End Text	1	0x03

Warning : the maximum size of an unstuffed Data Packet is 1031 bytes :1024 for the DATA field, 1 for STX, ID RC, DLE and ETX fields, and 2 for CRC field. But the stuffing process can double the size of the RC, DATA and CRC fields, then the maximum size of a stuffed Data Packet is 2058 bytes : 2048 for the DATA field, 1 for STX, ID, DLE and ETX fields, 2 bytes for the RC field, and 4 for the CRC field.

For description of stuffing process, please refer to “Stuffing Process” section.

4.1.2.2 ACK / NACK PACKET STRUCTURE

ACK/NACK Packet		
STX	ID	RC
1 byte	1 byte	1 byte
Max size when stuffed :		2 bytes

With the same field definition as the Data Packet (refer to section “Data Packet structure” above).

Warning : the usual size of an unstuffed Ack/Nack Packet is 3 bytes, but the stuffing process can double the size of RC field. In that case, the maximum size of a stuffed Ack/Nack Packet is 4 bytes.

For description of stuffing process, please refer to “Stuffing Process” section.

4.1.2.3 PACKET IDENTIFIER

The identifier is a byte formatted as described below :

Bit	Use	Description
7	IN/OUT flag	Packet direction flag : • This bit must be cleared for each packet sent by the Host to the MorphoSmart™. • This bit is set to 1 for each packet send by the MorphoSmart™ to the Host.
6	F (First)	First Packet flag, this bit is set to “1”, when the Data Packet contains : • the totality of the original Data (original Data size ≤ 1024 bytes) • the first bytes of the original Data (original Data Size > 1024 bytes)
5	L (Last)	Last Packet flag, this bit is set to “1”, when the Data Packet contains : • the totality of the original Data (original Data size ≤ 1024 bytes) • the last bytes of the original Data (original Data Size > 1024 bytes)
4	Reserved	Set to zero.
3 to 0	Packet type	• 0x01 : Data Packet • 0x02 : ACK Packet • 0x04 : NACK Packet

In accordance with description above, the valid Packet Identifier values are :

Value of the Packet Identifier		**Send by**	
Packet type		**Host**	**MorphoSmart™**
Data	Single Data Packet (the whole original Data is stored in the Packet)	0x61	0xE1
	First Data Packet : contains the first bytes of a segmented Data	0x41	0xC1
	Intermediate Data Packet : contains a part of a segmented Data (it is neither the first part, nor the last part of the Data)	0x01	0x81
	Last Data Packet : contains the last bytes of a segmented Data	0x21	0xA1
ACK	Packet	0x62	0xE2
NACK	Packet	0x64	0xE4

For Data segmentation description, please refer to the “Data Field” section below.

4.1.2.4 REQUEST COUNTER FIELD

The Request Counter is a one byte counter handled by the Transmitter, which is used to check the transmission of a Data Packet, by executing the process described below :

- Transmitter : when a Data has to be send, the current value of the Transmitter Request Counter is stored in the RC field of the Data Packet,
- Transmitter : after the physical transmission of the Data Packet, the Transmitter enters in the “Wait for Data Packet acknowledgment” state,
- Receiver : at the reception of a Data Packet, the value of the RC field is extracted and checked against the RC value of the last received Data Packet (Please refer to “Data Packet Reception Process” section),
- Receiver : according to the result of the Data Packet validity check process, an ACK (or a NACK) Packet is returned to the transmitter, with it’s RC field filled with the value of Request Counter extracted from received Data Packet,
- Transmitter : at the reception of a ACK packet, from the Receiver, the Transmitter extracts the value of the RC field, and check it against the Request Counter current value, to verify that the ACK Packet is received for the last Data Packet send,
- Transmitter : If the RC field of the ACK packet is identical to the RC field of the last Data Packet sent, then the Data Packet sending process is completed : the Transmitter Request Counter is incremented (+1), and the Transmitter exits from the “wait for ACK/NACK Packet” status.

Please refer to “Typical transaction workflow” section for samples.

The Transmitter Request Counter value is reset to 0, in case of :

- power up of the Transmitter
- fatal transmission error (no ACK received despite the retry process)
- reception of a BREAK signal

When the current value of the Request Counter reach 0xFF, then the next value is 0x00.

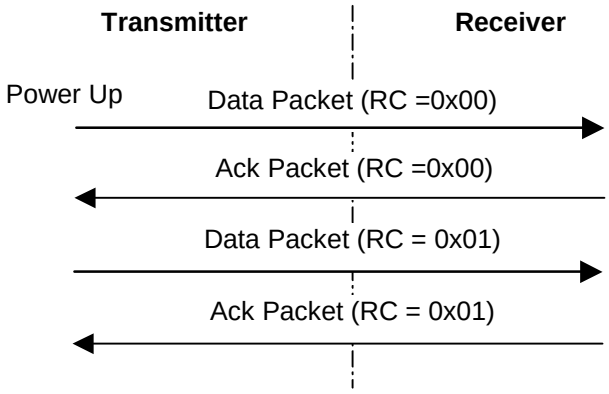


Figure 5 : Influence of Power Up on Request Counter

For the management of the backup of the received RC counter in the Receiver, please refer to “Received Request Counter Backup” section below.

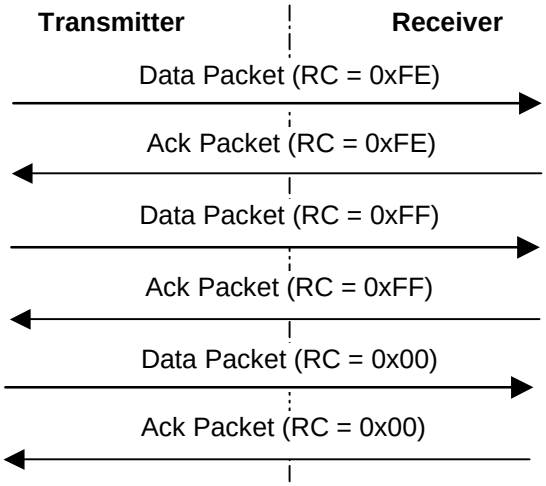


Figure 6 : Request Counter returns to 0x00

4.1.2.5 DATA FIELD

The Data field of a Data Packet contains, either a command (transmission from the Host to the MorphoSmart™), or an answer to a command (Transmission from the MorphoSmart™ to the Host). The command and the answer are formatted according to the ILV protocol, as described in next chapters.

If the size of the Data to transmit, is greater than 1024 bytes, the original Data is segmented into several 1024 bytes segments and one segment with the remaining of the Data. It means that the Transmitter send as much Data Packets as the number of Data segments, for the unique original Data.

When the original Data is segmented, three different Packet Identifiers are used (please refer to “Packet Identifier” section) :

- First Data Packet, for the first bytes of the original Data,
- Intermediate Data Packet, for a part of the segmented Data, which is neither the first part, nor the last part,
- Last Data Packet, for the last bytes of the original Data.

The table below, describes the segmentation of the Data according to it’s size.

Size of original Data	Data Packet to be send (and received)
L ≤ 1024 bytes	Only one Data Packet with Single Data Packet identifier (0x61 if sent by Host, 0xE1 if sent by MorphoSmart™), and the whole original data stored in the Data field.
1024 < L ≤ 2 x 1024 bytes	• A 1st Data Packet with the “First Data Packet” Identifier (0x41 if sent bye Host, 0xC1 if sent by MorphoSmart™), and the first 1024 bytes of the original Data, • A 2nd Data Packet with the “Last Data Packet” identifier (0x21 if sent by Host, 0xA1 if sent by MorphoSmart™), and the remaining bytes of the original Data (L – 1024).
N x 1024 < L ≤ (N+1) x 1024	• A 1st Data Packet, with the “First Data Packet” Identifier (0x41 if sent bye Host, 0xC1 if sent by MorphoSmart™), and the first 1024 bytes of the original Data, • If N – 1 > 0, N - 1 Data Packets with the “Intermediate Data Packet” Identifier (0x01 if sent by Host, 0x81 if sent by MorphoSmart™, and the next 1024 bytes of the original data (none if N = 1, one if N = 2, two if N = 3, and so on), • A (N + 1)th Data Packet with the “Last Data Packet” identifier (0x21 if sent by Host, 0xA1 if sent by MorphoSmart™), and the remaining bytes of the original Data (L – (N x 1024)).

4.1.2.6 CRC CALCULATION

The CRC field is computed by the Transmitter, from the content of the DATA field only, before stuffing process.

The CRC value of the Data field has to be checked by the Receiver after unstuffing process.

The CRC value is computed as a CRC16, in compliance with V.41 ITU-T recommendation : the polynomial used for generation is $x^{16} + x^{12} + x^5 + 1$, and the initial value is 0x0000.

A sample of CRC compute and check functions are provided in the “MSO ILV Sample” project included with MorphoSmart ™ SDK (please refer to file crc.c)

4.1.2.7 END OF DATA PACKET

Due to the unpredictable content of the RC, DATA and CRC fields, the <DLE> <ETX> sequence (0x1B, 0x03) is not enough to determine the end of a packet. In order to provide a reliable condition for “End of Packet” “the stuffing process modifies the content of the RC, DATA and CRC field.

On a stuffed Data Packet, an <ETX> (0x03) character immediately preceded by an odd number of <DLE> (0x1B) characters (1, 3, 5, and so on), indicates the “End of the Data Packet”.

4.1.2.8 STUFFING PROCESS

The stuffing process is perform to avoid confusion between data characters and software handshake characters (XON/XOFF), and to provide a reliable way to find the End of the Packet (please refer to “End of Data Packet” section above).

This process scan every packet to be send (Data, Ack and Nack Packet) in order to find specific characters, in the RC, the DATA and the CRC fields :

- The XON character (0x11) is replaced by a couple of one DLE character (0x1B) and one <XON + 1> character (0x12),
- The XOFF character (0x13) is replaced by a couple of one DLE character (0x1B) and one <XOFF + 1> character (0x14),
- An additional DLE character (0x1B) is inserted before each DLE character found.

The stuffing process, as described above, must be performed before sending any Packet.

Here is a stuffing process sample :

	RC, DATA and CRC fields
Before stuffing	… 0x05 0x54 **0x11** 0x65 0x85 **0x1B** 0x23 0x35 **0x13** 0x03 0x22 0x25 …
After stuffing	… 0x05 0x54 **0x1B 0x12** 0x65 0x85 **0x1B 0x1B** 0x23 0x35 **0x1B 0x14** 0x03 0x22 0x25 …

The unstuffing process (restoration of original RC, Data and CRC values, must be performed after receiving the packet, and before any process of the content of this fields (RC and Data extraction, CRC check).

The unstuffing process generates an error if a DLE character (0x1B) is not followed by one of the expected character : DLE character (0x1B), <XON + 1> (0x12), <XOFF + 1> (0x14), or ETX (0x03)

4.1.2.9 BYTE ORDER

The Packet byte order is Little Endian : multibyte data are sent least significant byte first (LSB). It means that 0x1234 is stored as 0x34, 0x12.

4.1.3 Data Packet Transmission Process (Transmitter)

4.1.3.1 TRANSMISSION OF A DATA PACKET

When the Transmitter needs to send a Data to the Receiver, the Data is included in a Data Packet, as described in the previous sections, and then physically transmitted to the Receiver.

4.1.3.2 WAIT FOR DATA PACKET ACKNOWLEDGE

After the physical transmission of a Data Packet, the transmitter enters in the “Wait for the related ACK/NACK Packet” state, in which the transmission of another Data Packet is not allowed.

4.1.3.3 RECEPTION OF AN INVALID ACK/NACK PACKET

The Transmitter waits for an ACK/NACK Packet with the same Request Counter value as the last Data Packet sent. Then an ACK or a NACK Packet is received with a different RC value is ignored :

- the Transmitter Request Counter is not incremented,
- the Transmitter stays in “Wait for Data packet acknowledgement” state,
- the “Wait for Ack/Nack Packet” timer is not reset.

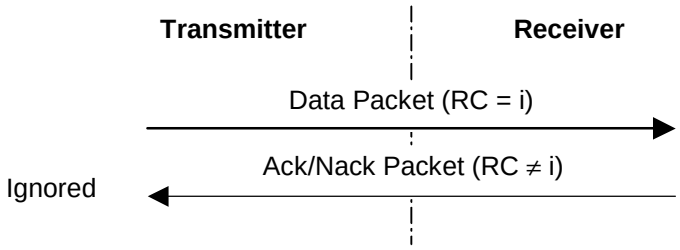


Figure 7 : ACK/NACK Packet with unexpected RC value is ignored

4.1.3.4 RECEPTION OF A NACK PACKET

If a NACK Packet is received, it means that the Receiver has detected a reception error, which usually indicates a physical transmission error; the Transmitter has to send back the same Data Packet.

The Receiver checks the reception of the Data Packet, and if it is still invalid, returns another NACK Packet to the Transmitter. This process is repeated until an ACK Packet is received or until the retry counter is exhausted.

The transmitter is expected to try up to 5 times to send the same Data Packet, if the 5th answer of the receiver is still a NACK Packet, then the Transmitter returns a “Transmission error” to the application level.

In case of transmission error, the MorphoSmart™ reset the communication parameters and data (refer to “communication reset” section).

If a ACK Packet is received at the 2nd, 3rd, 4th or 5th try, the transmission of the Data Packet is completed : the Transmitter Request counter is incremented (+1), and the Transmitter exits from the “Wait for Data Packet acknowledgement” state.

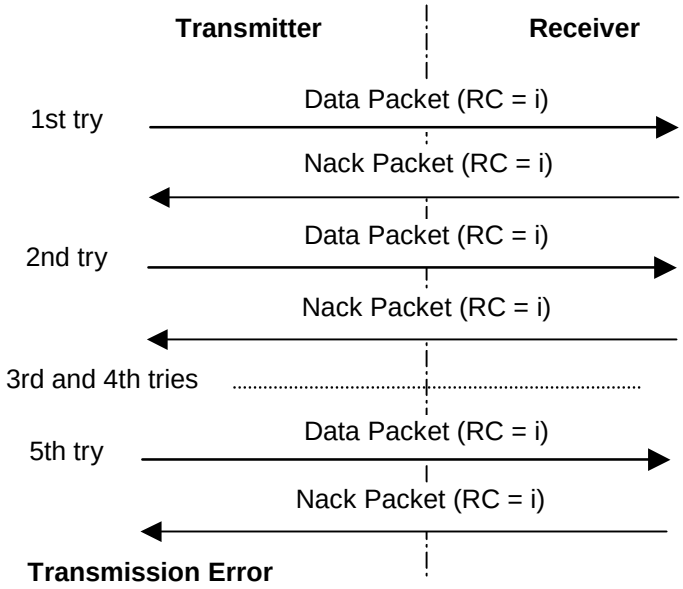


Figure 8 : Retry on NACK Packet reception

4.1.3.5 TIMEOUT WAITING FOR ACKNOWLEDGEMENT

The Transmitter waits for an ACK/NACK Packet during a specified time :

- At the first try, the transmitter waits, after the transmission of the Packet Data, for the reception of the related ACK/NACK Packet,
- If the expected ACK/NACK Packet is not received, the transmitter send back the same Data Packet, with the same timeout value,
- If the expected ACK/NACK Packet is still not received, the transmitter send back the same Data Packet, and with the same timeout : it's the last try,
- If the expected ACK/NACK Packet is still not received, the transmitter doesn't try anymore to send this Data Packet, and return a "Transmission Timeout Error" to the application level.

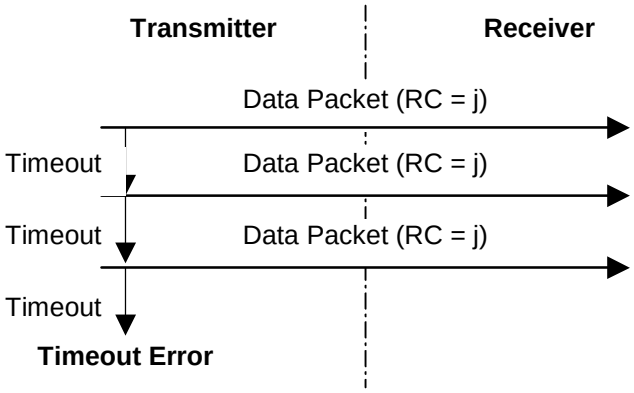


Figure 9 : Timeout waiting for acknowledge

The MorphoSmart™ waits 500 ms for ACK/NACK packet, but 1000 ms is a typical value for the Host.

In case of Timeout transmission error, the MorphoSmart™ reset the communication parameters and data (refer to "communication reset" section).

4.1.4 Data Packet Reception Process (Receiver side)

4.1.4.1 WAIT FOR DATA PACKET STATE

The normal state of the Receiver is to wait for a Data Packet :

- without any time limit, when the Receiver is the MorphoSmart™
- with a timeout value adapted to the ILV command sent, when the Receiver is the Host.

4.1.4.2 WAIT FOR DATA PACKET PROCESS IN MORPHOSMART™

The process is as described below for the three first fields :

- The MorphoSmart™ waits for a STX character : if the received character is not a STX, the character is ignored.
- When a STX character is received, the MorphoSmart™ waits for the next character, which should be a Data Packet Identifier : the character is stored but not checked.
- the MorphoSmart™ waits for the next character which should be a the RC value or a DLE stuffing character.
- If the 3rd character is a DLE character, then the RC byte is stuffed, and the MorphoSmart™ waits for the next character for unstuffing process on the RC field.

Then the Packet Identifier is checked, if it is not a valid Packet Identifier, the received characters are ignored, and the MorphoSmart™ restart at “wait for STX character” step.

If the Packet Identifier is valid, the RC value is checked against the previously received RC value : if it is the same value, the received characters are ignored and the MorphoSmart™ restart at “wait for STX character” step.

If the received RC value is different from the previously received RC, then the MorphoSmart™ expects the remaining of the Data Packet : the received characters are stored until a DLE character followed by a ETX character (End of Packet). The unstuffing process is executed during character reception : the Data Packet is stored unstuffed.

At the end of Packet Data reception, the CRC value is checked : if valid, a ACK Packet is retuned to the Host, else it is a NACK Packet which is returned to the Host.

Finally, the received Data Packet Identifier is checked :

- Single Data Packet : the Data field content is provided to the application level for the process of the ILV Request
- First Data Packet : the Data field content is stored, and the MorphoSmart™ waits for next part of the original Data
- Intermediate Data Packet : the Data field content is added to previously received Data segment, and the MorphoSmart™ waits for next part of the original Data.
- Last Data Packet : the Data field content is added to previously received Data segment, and the whole Data is provided to the application level for the process of the ILV Request

4.1.4.3 WAIT FOR DATA PACKET PROCESS IN HOST

The following process should be implemented in the calling application.

When a Data Packet is received, the Host checks if :

- the received Request Counter value is different than the one received in the previous Data Packet,
- and if the Data Packet is valid (see below),

If both conditions are verified, the Host :

- Send back an ACK Packet, which contains the received Request Counter value, to the MorphoSmart™,
- Provides the content of the Data field, after unstuffing process, to the application level.

The Host returns a NACK Packet if one of the error listed below is detected :

- Timeout between the reception of two bytes of the same packet. the time measure starts after the reception of STX character, and the timeout value is 100 ms,
- Unstuffing error : a <DLE> character (0x1B) is followed by an unexpected character (refer to “Stuffing Process” section above),
- CRC check failure : the received CRC value is not identical to the CRC computed from Data field (after unstuffing).

4.1.4.4 REQUEST COUNTER ERROR IN MORPHOSMART™

If a Data Packet is received with the same RC value than the previously received Data Packet, then the MorphoSmart™ ignore the Data Packet :

- The Data Packet is not acknowledged,
- The Data field content is not provided to the application level.

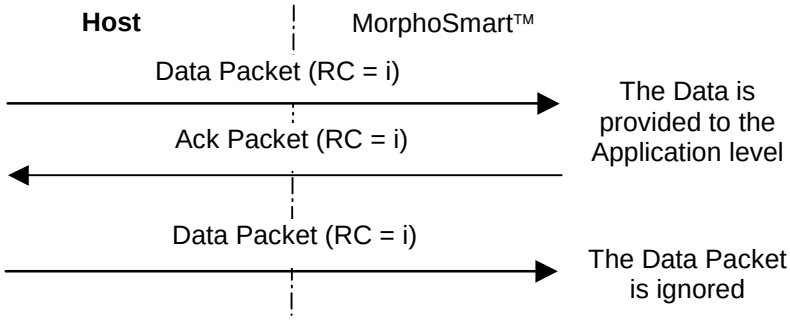


Figure 10 : Request Counter error (MorphoSmart™)

4.1.4.5 REQUEST COUNTER ERROR IN HOST

The following process should be implemented in the calling application :

- When the received RC value is identical to the previously received Data Packet, it means that the ACK/NACK Packet returned for the previous received Data packet has not been received by the MorphoSmart™ (refer to “timeout waiting for acknowledgement” section above).
- Then according to the transmission error process, the MorphoSmart™ send again the same Data Packet. The Host process it as a new Data Packet : the validity is checked and an ACK/NACK Packet is returned.
- If the Data Packet is valid, the content of the Data Field, is not provided to the application level. The Host assumes that the Data has already been provided to the Application Level during the process of the previously received Data Packet with this RC value.

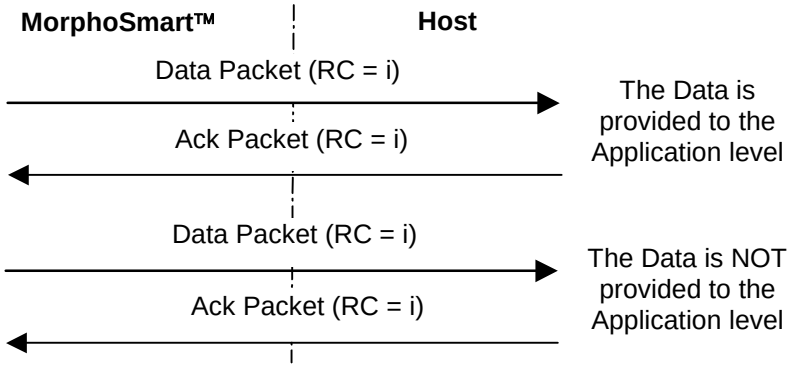


Figure 11 : Request Counter error (Host)

4.1.4.6 RECEIVED REQUEST COUNTER BACKUP (RECEIVER)

As the Receiver has to check the RC of the received Data packet, against the previously received Data packet, it means that the Receiver must store the RC value of the last received packet.

This backup value is erased when one of the events listed below occurs :

- Power-up,
- Reception of a BREAK signal on the RS232 port (MorphoSmart™ only)
- Fatal transmission error (no ACK received despite the retry process)

When the backup of the RC is erased, the RC check of the next Data Packet to be received will be successful. Then the received RC value will be stored, and used for the RC check of the next Data packet.

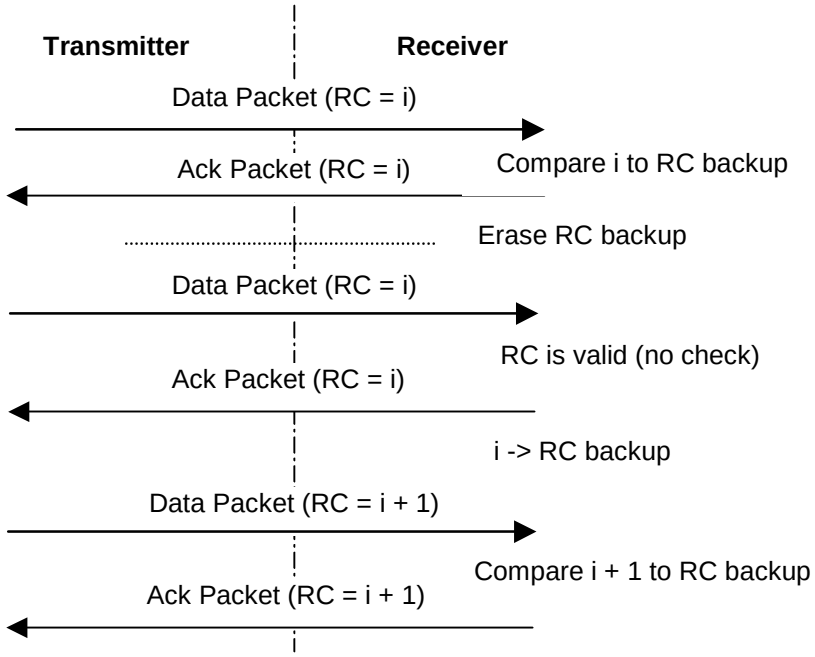


Figure 12 : Influence of Request Counter Backup deleting in Receiver

4.1.5 Typical transaction workflow

4.1.5.1 DATA SIZE ≤1024 BYTES

When the size of the Data to be transmitted is less than or equal 1024 bytes, only one Data Packet is required (the data is not segmented in several Data Packets).

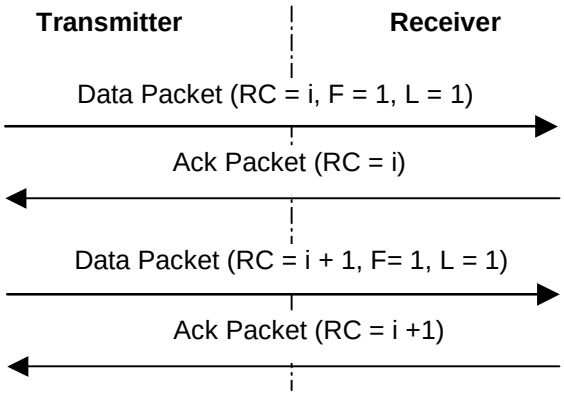


Figure 13 : Transmission of data (size <= 1024 bytes)

The Request counter is incremented after each completed Data Packet transmission.

The Data Packet Identifier has both bit 6 (First Packet flag) and 5 (Last Packet flag) set to “1” value.

The ACK packet returned by the receiver contains the Request Counter value of the received data packet.

Data Packet	Data Packet Identifier	ACK Packet Identifier
Send by Host to MorphoSmart ™	0x61	0xE2
Send by MorphoSmart ™ to Host	0xE1	0x62

4.1.5.2 DATA SIZE > 1024 AND ≤2048 BYTES

When the size of data to be transmitted is more than 1024 bytes, but less or equal than 2048 bytes, the data is segmented into two Data Packets : the first one with 1024 of effective data, and the second one the remaining of the data.

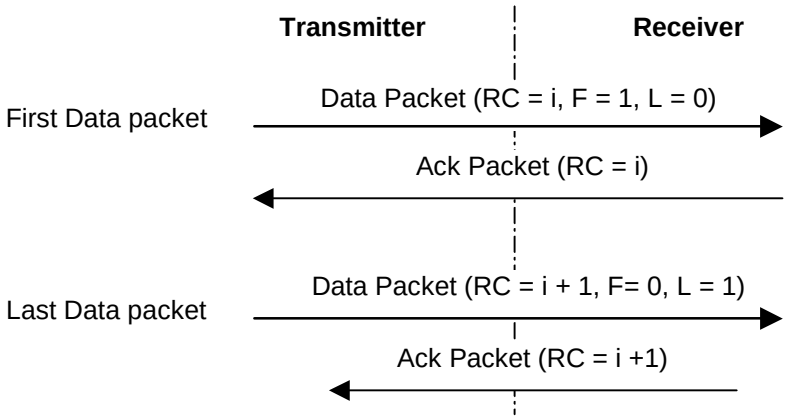


Figure 14 : Transmission of Data (1024 < size ≤ 2048 bytes)

The Request counter is incremented after each completed Data Packet transmission.

The First Data Packet Identifier has only bit 6 (First Packet flag) set to “1” value.

The Last Data Packet Identifier has only bit 5 (Last Packet flag) set to “1” value.

The ACK packet returned by the receiver contains the Request Counter value of the received data packet.

Data Packet	Data Packet Identifier	ACK Packet Identifier
First Packet send by Host to MorphoSmart ™	0x41	0xE2
Last Packet send by Host to MorphoSmart ™	0x21	0xE2
First Packet send by MorphoSmart ™ to Host	0xC1	0x62
Last Packet send by MorphoSmart ™ to Host	0xA1	0x62

4.1.5.3 DATA SIZE > 2048 BYTES

When the size of data to be transmitted is more than 2048 bytes, the data is segmented into more than two data packets : the first one with 1024 of effective data, then the required number of data packet for each other segment of 1024 bytes of effective data, and the last one with the remaining of the data.

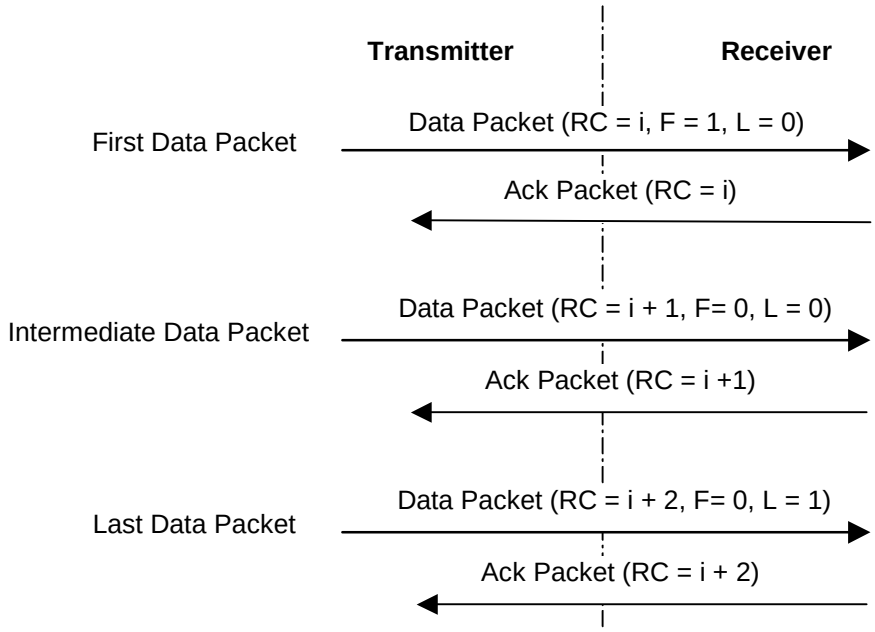


Figure 15 : Transmission of Data (2048 < size <= 3076 bytes)

The Request counter is incremented after each completed Data Packet transmission.

The First Data Packet Identifier has only bit 6 (First Packet flag) set to “1” value.

The intermediate Data Packet Identifier has both bit6 (Last Packet flag) and Bit5 cleared (zero value).

The Last Data Packet Identifier has only bit 5 (Last Packet flag) set to “1” value.

The ACK packet returned by the receiver contains the Request Counter value of the received data packet.

Data Packet	Data Packet Identifier	ACK Packet Identifier
First Packet send by Host to MorphoSmart ™	0x41	0xE2
Intermediate Packet send by Host to MorphoSmart ™	0x01	0xE2
Last Packet send by Host to MorphoSmart ™	0x21	0xE2
First Packet send by MorphoSmart ™ to Host	0xC1	0x62
Intermediate Packet send by MorphoSmart ™ to Host	0x81	0x62
Last Packet send by MorphoSmart ™ to Host	0xA1	0x62

4.1.6 Typical communication between Host and MorphoSmart™

The Host send a command (ILV format) to the MorphoSmart™, which returns a reception acknowledge. When the command is processed, the MorphoSmart™ send the result (ILV format) to the Host, which returns also a reception acknowledge.

When the Data to be sent is a Command, the transmitter is the Host, and the Receiver is the MorphoSmart™. But when the Data to be sent is an Answer, the MorphoSmart™ is the Transmitter, and the Host the Receiver.

Warning : As the Host and the MorphoSmart™ are Transmitters, there is two different and independent Transmitter Request Counters : one for each Transmitter.

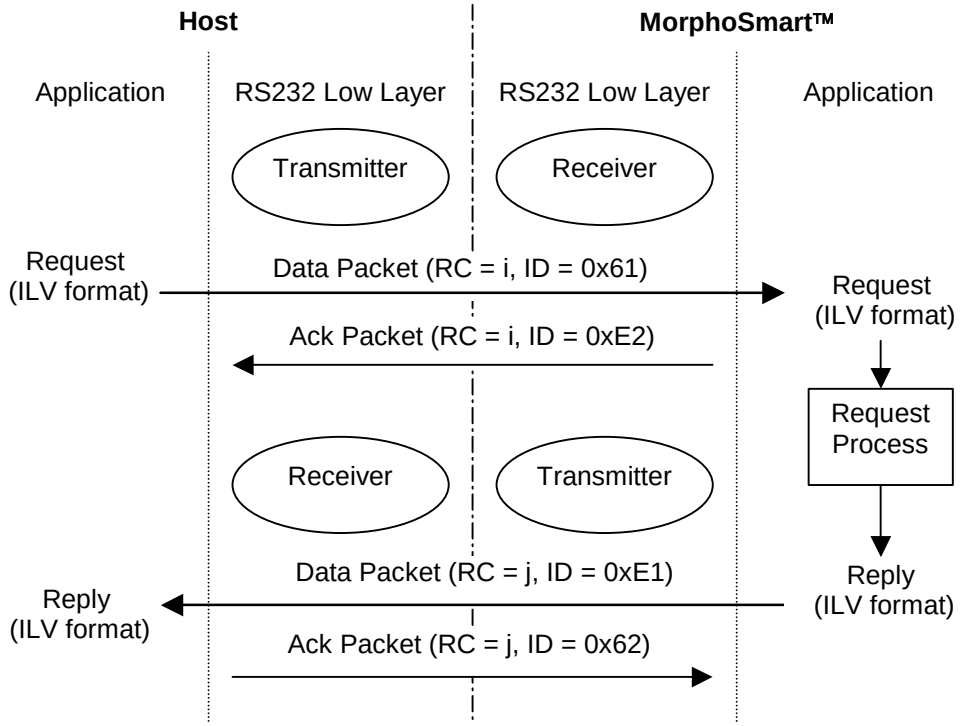


Figure 16 : Typical Command/Answer cycle between Host and MorphoSmart™

According to the figure above, the next values of the Request Counters are :

- "i+1" for the transmission of the next Request by the Host
- "j+1" for the transmission of the next Reply by the MorphoSmart™

4.1.7 Communication Reset

When specific events occurs, the MorphoSmart™ reset communication parameters :

- the RS232 port parameters are reset to default value : 9600 Bd, 8 bits, no parity, 1 bit stop signal, software flow control,
- the Transmitter Request counter is reset to zero (0x00),
- the last received RC value is erased : the RC control is disabled for the next received Data Packet.

The communication parameters are reset when :

- the MorphoSmart™ is powered up.
- or when a BREAK signal is received from the host,
- or when a transmission timeout error occurs : neither ACK nor NACK packet received for the last Data Packet sent (even after the retry process),
- or when a transmission error occurs : no ACK Packet received for the last Data Packet sent. (even after the retry process).

4.2 USB LOW LAYER PROTOCOL

4.2.1 Overview

The integrated USB driver of the MorphoSmart™ terminal emulates a RS232 serial port.

The MorphoSmart™ terminal is processed as a Communication Device Class (CDC), in accordance with the "USB Device Class Specifications, version 1.1" available from the *www.usb.org* web site, in "developers, documents" section. The device may also be used with a USB2.0 controller, but will still use the USB 1.1 transfer rate.

4.2.2 Data Transmission

The Data to be transmitted is included in a frame, which is send by writing in a RS232 virtual port.

A frame is received by reading in the same RS232 virtual port.

When a frame is received, the Data is extracted from the frame, and provided to the Application level.

The Data to be transmitted is either a Command (from Host to MorphoSmart™) or a Reply to a Command (MorphoSmart™ to Host), build in accordance with the ILV protocol described in next chapters.

4.2.3 Frame Format

The data to be transmitted is included in a frame according to the format described below :

Part	**Description**
Start of frame	A set of 4 ASCII characters : ‘S’Y’N’C
Length of the frame	4 bytes (unsigned long big endian)
1 complement of the length of the frame,	4 octets (unsigned long big endian)
Data	Command or answer (ILV format as described in next chapters)
End of Frame	A set of 2 ASCII characters : ‘E’N

Sample : to send the Get descriptor command, write the following sequence :

0x53 0x59 0x4E 0x43 0x04 0x00 0x00 0x00 0xFB 0xFF 0xFF 0xFF 0x05 0x01 0x00 0x2F 0x45 0x4E

5 Application Level : ILV Protocol

5.1 SCOPE

At application level, the terminal expects a command which complies to the ILV format, and returns to the Host, a reply which also complies to the same format.

5.2 ILV FORMAT

The application data structure exchanged between the terminal and the host, has three fields :

	Field name	Description	Size
I	**I**dentifier	Identifier of the command. The reply returned by the terminal starts with the same identifier value as the one found in the received command (if valid).	1 byte
L	**L**ength	Length of the ***Value*** field (in byte). This field is in little endian format. The number of bytes of the Value field is directly related to the command and it's parameters. The size of this field is either 2 bytes or 6 bytes, depending on the Value field size : either lower than 0xFFFF or not (see below).	2 or 6 bytes
V	**V**alue	Data and/or parameters. This field may contains one or several ILV structures. This field is in little endian format.	L bytes

For a Value field size less than 65535 (0xFFFF) bytes, the ILV structure is formatted as described below :

Identifier value	I	1 byte
Length value	Number of bytes of Value field (L)	2 bytes
Value (Parameters)	Data and/or parameters.	L bytes

For a Value field size higher or equal to 65535 (0xFFFF) bytes, the ILV structure is formatted as described below :

Identifier value	I	1 bytes
Length value	0xFFFF	2 bytes
	Number of bytes of Value field (L)	4 bytes
Value (Parameters)	Data and/or parameters.	L bytes

5.3 REPLY FORMAT IN CASE OF ERROR

Any command received from the Host, is analyzed and then executed by the terminal. If an error occurs, then the terminal returns an ILV formatted reply to the Host.

5.3.1 ILV format error

If the format of the received command is invalid, then the terminal returns the ILV formatted reply below :

Identifier value	**ILV_INVALID (0x50)**	1 byte
Length value	0x0000	2 bytes
Value (Parameters)	None	0 byte

Usually it means that a mismatch between the L value and the length of the V area, has been found in the received command.

5.3.2 Unknown request

If the value of the request code (I), found in the received command, doesn't correspond to a function described in the following chapters, then the terminal returns the same ILV formatted reply :

Identifier value	**ILV_INVALID (0x50)**	1 byte
Length value	0x0000	2 bytes
Value (Parameters)	None	0 byte

5.3.3 Error during request process

If the request cannot be successfully executed, then the terminal returns the ILV formatted reply below :

Identifier value	Received I value	1 byte
Length value	0x0005	2 bytes
Value (Parameters)	ILV_Status	1 byte
	Internal error code	4 bytes

With :

ILV_Status : different from ILV_OK (0x00), the possible values are listed for each command in next chapters. Please note that new ILV_Status codes can be added in future release of the firmware.

Internal error code : reserved for Sagem Sécurité internal use (mainly for development purpose).

5.3.4 Device locked

Since embedded version 06.03, the MSO can be locked to a user or to an application. A secret Key is inserted into the MSO at production time and given to the authorized user. Then using a unlocking process, the user is able to execute all the functions of the device.

Please refer to the Chapter 6.8 *Unlocking functions description* for more information on this feature.

If the device is locked, then the terminal returns the usual ILV formatted error reply (please refer to previous description), with the ILV status equal to **ILVERR_APPLI_LOCKED**.

5.4 COMPATIBILITY AND RECOMMENDATIONS

In order to remain compatible with future MorphoSmart™ firmware releases, we recommend not being too restrictive on ILV reply analysis process :

- New data can be added at the end of a reply. It means that the length (L value) can be higher than current value. Then applications should be able to process the required data, and ignore extra data without triggering any error.
- New error codes can be added, then applications should be able to deal with unexpected error codes.

6 ILV Functions Description

6.1 TABLE OF MORPHOSMART ™ ILV FUNCTIONS

Initialization functions		
0x05	*GET_DESCRIPTOR*	Retrieves information from biometric module
Biometric functions		
0x20	*VERIFY*	Capture and verify against a reference template
0x21	*ENROLL*	Capture and add to database and/or export templates
0x22	*IDENTIFY*	Capture and identify against the local database
0x23	*VERIFY MATCH*	Verify a list of reference templates against a search template
0x24	*IDENTIFY MATCH*	Identify a search template against the local database
Database management functions		
0x30	*CREATE BASE*	Create a local database
0x32	*ERASE BASE*	Erase the local database (destroy all records, not database structure)
0x34	*ERASE ALL BASE*	Erase all local database (destroy all records, not database structure)
0x3B	*DESTROY BASE*	Destroy the local database
0x33	*DESTROY ALL BASE*	Destroy all local databases
0x35	*ADD RECORD*	Add a record to the local database
0x36	*REMOVE RECORD*	Remove a record from the local database
0x38	*FIND USER BASE*	Search record matching a given pattern
0x3C	*UPDATE PUBLIC DATA*	Update a database public field
0x3D	*UPDATE PRIVATE DATA*	Update a database field (public or private)
0x3E	*GET PUBLIC FIELDS*	Get database public field list
0x3F	*GET DATA*	Get a database public field
0x07	*GET BASE CONFIG*	Get database configuration
Security management functions		
0x80	*SECU GET CONFIG*	Get MorphoSmart™ security configuration
0x81	*SECU READ CERTIFICATE ID*	Get X509 certificate from MSO certification path
0x82	*SECU STORE CERTIFICATE*	Load a host X509 certificate
0x83	*SECU STORE PKCS12*	Load a PKCS#12 token
0x84	*SECU MUTUAL AUTH INIT 1*	First step to establish a secure tunnel
0x85	*SECU MUTUAL AUTH INIT 2*	Second step to establish a secure tunnel
0x86	*SECU PROTOCOLE*	Security envelope
OTP management functions		
0xB0	*OTP ENROLL USER*	Enroll the user in the OTP token in the OTP database

0xB1	*OTP GENERATE*	Generate the OTP
0xB2	*OTP GET STATUS*	Get the status of the OTP token
0xB3	*OTP SET PARAMETERS*	Set the parameters of the OTP token
Configuration		
0x90	*GET_MSO_CONFIG*	Retrieve the value of one configuration parameter
0x91	*MODIFY_MSO_CONFIG*	Modify the value of one configuration parameter
Unlocking		
0x8B	*GET_UNLOCK_SEED*	Retreives a seed to initialize the unlock process
0x8C	*UNLOCK*	Unlocks the functions using the seed
Miscellaneous		
0xEE	*CONFIG UART*	Change UART configuration
0x71	*ASYNC MESSAGE*	Asynchronous message
0x70	*CANCEL*	Cancel a live finger acquisition
Invalid ILV		
0x50	ILV_INVALID	Invalid ILV

Table 2: List of functions

6.2 INITIALIZATION FUNCTIONS DESCRIPTION

6.2.1 GET_DESCRIPTOR ID = 0x05

This function gets MorphoSmart™ descriptors.

6.2.1.1 REQUEST

Identifier value	0x05	
Length value	0x0001	
Value(Parameters)	Format	1 byte

Format:

Specifies the required descriptor(s) among the values listed below :

ID_FORMAT_TEXT: All main descriptors formatted in separate character strings (see below).

ID_FORMAT_BIN_VERSION: software version descriptor.

ID_FORMAT_BIN_MAX_USER: maximum number of records that can be stored into the internal database, according to hardware and software configuration of the MorphoSmart™ device.

6.2.1.2 REPLY (ID_FORMAT_TEXT)

Warning: these descriptors are subject to change and in human readable form only, for user information. Your software should not rely on these answers for any automatic information retrieval, since these descriptors change from one release to the other.

Identifier value	0x05	
Length value	0x0001 + <3 + L_1> + <3 + L_2> + <3 + L_3>	
Value(Parameters)	Request Status	1 byte
	Product Information	3 + L_1
	Sensor Information	3 + L_2
	Software Information	3 + L_3

Request Status:

ILV_OK	The function succeeded.
ILVERR_BADPARAMETER	Format is not valid.
ILVERR_ERROR	An internal error occurred.

Product Information: Product information data structure definition.

Identifier value	ID_DESC_PRODUCT	
Length value	L_1	
Value	String	L_1

Sensor Information: Sensor information data structure definition.

Identifier value	ID_DESC_SENSOR	
Length value	L_2	
Value	String	L_2

Software Information: Software information data structure definition.

Identifier value	ID_DESC_SOFTWARE	
Length value	L_3	
Value	String	L_3

6.2.1.3 REPLY (ID_FORMAT_BIN_VERSION)

Identifier value	0x05	
Length value	0x000B	
Value(Parameters)	Request Status	1 byte
	Software Version	10 bytes

Request Status:

ILV_OK	The function succeeded.
ILVERR_BADPARAMETER	Format is not valid.
ILVERR_ERROR	An internal error occurred.

Software Version : ILV formatted data with software release identification.

Identifier value	ID_FORMAT_BIN_VERSION	
Length value	0x0007	
Value	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

Sample : for software release 07.02.h, the terminal returns the value 0x30, 0x37, 0x2E, 0x30, 0x32, 0x2E, 0x68

6.2.1.4 REPLY (ID_FORMAT_BIN_MAX_USER)

Identifier value	0x05	
Length value	0x0006	
Value(Parameters)	Request Status	1 byte
	Maximum User number allowed	5 bytes

Request Status:

ILV_OK	The function succeeded.
ILVERR_BADPARAMETER	Format is not valid.
ILVERR_ERROR	An internal error occurred.

Maximum User number allowed : ILV formatted data with the maximum number of records of the data base, allowed by the MorphoSmart™ device configuration (hardware and software). Please refer to the chapter 7.3.4.5. for more information about database size.

Identifier value	ID_FORMAT_BIN_MAX_USER	
Length value	0x0002	
Value	Maximum number of records allowed by the configuration	2 ASCII digits

Sample : if the terminal configuration allows to create a database with a maximum of 500 records, then the value returned is : 0xF4, 0x01

6.3 BIOMETRIC FUNCTIONS DESCRIPTION

6.3.1 VERIFY ID = 0x20

This function captures a fingerprint 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 20.

6.3.1.1 REQUEST

Identifier value	0x20	
Length value	0x0005 + < L_1> (+ < L_2> +... +< L_n> + 7 + 4 + 7 + 4 + 7 + 7 + 7 + 7)	
Value (Parameters)	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 ≤ 20)	L_n bytes
	Asynchronous event (optional)	7 bytes
	Matching Score (optional)	4 bytes
	FFD Security Level (optional)	7 bytes
	Matching PK number (optional)	4 bytes
	Biometric coder selection (optional)	7 bytes
	Biometric presence detection mode (optional)	7 bytes
	Biometric Matching strategy (optional)	7 bytes
	Alive asynchronous message (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 (Sagem Sécurité recommends 5). This parameter specifies how tight the matching threshold is. For some more information, please refer to the "MorphoSmartOverview.pdf" documentation.

Acquisition Quality Threshold: Not used. Set to 0.

Reference template i: This is a list of ILV formatted data that contains the referenced templates. The maximum number of referenced templates is twenty. The biometric data can be one of the following ILV:

PK_COMP V2, or *PK_MAT*, *ISO_PK*, *PKV10*
PKBase: Verify the captured finger against the templates of a database record (depending on the database format),
It is not possible to process verification against more than one database record.
TKB: Check the templates integrity then verify the captured finger against the templates contains in the X9.84 token. The certificate should have been previously sent to the MorphoSmart™ in order to process the verification of the token signature.

***Asynchronous Event*:** This ILV is optional. If it is not present, no asynchronous messages will be sent.

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

***FFD Security Level*:** This ILV is optional. If it is not present, the false finger security level is set to 0 (lowest level – default security level).

***Matching PK number*:** This ILV is optional. If it is not present, the matching PK number will not be returned.

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

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

Biometric matching strategy : This ILV is optional. If it is not present, the default biometric matching strategy will be used.

Alive asynchronous message : ILV formatted optional parameter used to allow redundant finger position asynchronous message, for a “terminal is still alive” function during fingerprint wait. If not present, the default process is executed (no redundant finger position message).

6.3.1.2 REPLY

Identifier value	0x20	
Length value	0x0002 (+ 7 + 1)	
Value (Parameters)	Request status	1 byte
	Matching result	1 byte
	Matching score (optional)	7 bytes
	Matching PK number (optional)	1 byte

Request status:

ILV_OK	The function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	The matching threshold value is out of range, or timeout value is out of range, or there is no input biometric data, or first finger is ID_PKBASE and second finger is present, or the matching is performed with more than the maximum number of templates, or you select index 0 with an X9.84 template.
ILVERR_INVALID_MINUTIAE	The reference ILV minutiae are not valid: bad identifier, corrupted minutiae.
ILVERR_TIMEOUT	The finger detection timeout has expired.
ILVERR_CMDE_ABORTED	Command has been cancelled.
ILVERR_BIO_IN_PROGRESS	Command received during biometric processing
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_BASE_NOT_FOUND	The specified base does not exist
ILVERR_INVALID_USER_ID	The record identifier does not exist in the database
ILVERR_USER_NOT_FOUND	The searched user is not found.
ILVERR_BAD_SIGNATURE	Invalid digital signature
ILVERR_SECU_CERTIF_NOT_EXIST	The required certificate does not exist.
ILVERR_SECU_ASN1	Error while decoding an ASN1 object or bad X984 template index
ILVERR_SECU	Cryptographic error
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

Matching result:

ILVSTS_HIT	The comparison succeeded.
ILVSTS_NO_HIT	It is not the same finger.
ILVSTS_DB_EMPTY	The database is empty.
ILVSTS_FFD	False finger detected.
ILVSTS_MOIST_FINGER	The finger can be too moist or the scanner is wet.

***Matching Score*:** This ILV is optional. It is returned on request.

***Matching PK number*:** This ILV is optional. It is returned on request.

6.3.2 ENROLL ID = 0x21

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 getting 3 images for enrollment purpose, and 1 image for verification purpose. An enrollment based on 3 images per finger will increase the system accuracy.

The number of fingers to be enrolled is either one or two.

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.

6.3.2.1 REQUEST

Identifier value	0x21	
Length value	0x0008 + (<L_{UID}> + < L_{Data1}> + … <L_{Datai}>+ 7 + 4 + 9 + <L_{TKB}> + 4 + 4 + 7 + 7 + 7 + 7 + 7)	
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
	User ID (optional)	L_{UID} bytes
	Additional user data field 1 (optional)	L_{Data1} bytes
	…	
	Additional user data field i (optional)	L_{Datai} bytes
	Asynchronous event (optional)	7 bytes
	Biometric Algorithm Parameter (optional)	4 bytes
	Export Image (optional)	9 bytes
	Export biometric token (optional)	L_{TKB} bytes
	Fingerprint latent detection (optional)	4 bytes
	Export Coder Results (optional)	4 bytes
	FFD Security Level (optional)	7 bytes
	Biometric coder selection (optional)	7 bytes
	Biometric presence detection mode (opt)	7 bytes
	Alive asynchronous message (optional)	7 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Timeout: Finger detection timeout in second. 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 per finger. Allowed values are 0, 1 and 3.
We strongly recommend setting 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: in this case, 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 to store calculated minutiae into the local database. Otherwise set it to FALSE.
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: this field is used to specified if the template build from captured fingerprint has to be included in the reply or not. It also allows to specified the maximum size of a template when PK_Comp format is selected. When the **Number of finger** field value is 2, the reply includes two templates in acquisition order (first template is related to first captured fingerprint).

- Set this field to 0x00 to exclude the template (fingerprint minutiae) from the reply.
- Set this field to 0x01 to include the template (fingerprint minutiae) in the reply. See the ***Algo Param Field*** parameter (described below) to choose the template format.
- For PK_COMP only, this field can be set to a value from 170 (0xAA) to 255 (0xFF) to limit the size of the template. If the template size is higher than the required value, it is compressed to match the requirement before being include in the reply, otherwise the template is included without modification. It means that the template size is less or equal to the specified value.
- For PK ANSI 378, PK ISO FMC CS, PK ISO FMC NS, PK ISO FMR, this field can be set to a value from [2-255] to limit the number of PK in a template.

User ID: Optional ILV formatted data which contains the unique user identifier. This ILV data is mandatory when the *Save Record* value is set to TRUE. The value of the User ID is retrieved by the IDENTIFY function under Hit condition (the captured fingerprint match with one of the templates stored in the database).

***Additional User Data j*:** optional ILV formatted data which contains the data to store in the additional user data field # i. When the *Save Record* value is set to TRUE, the terminal expects one Additional user Data ILV for each of the additional user data fields defined with the CREATE DATABASE command. The order of the additional user data must be the same as the order of the additional user data fields. As the content of the data is not checked by the terminal (except excessive size), the data can be empty (0 bytes)

***Asynchronous Event*:** This ILV is optional. If it is not present, no asynchronous messages will be sent.

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

***Export Image*:** ILV formatted optional parameter, used to require to include the fingerprint image in the reply.

Export Biometric Token (*TKB Settings*): This ILV is optional. Use this ILV to export the template in a X9.84 biometric format.

***Fingerprint latent detection*:** This ILV is optional. Use this ILV 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 *Enrollment type* is set to 1).

***Export Coder Results*:** This ILV is optional. Use this ILV to export the coder results for each captured finger. This is reserved for advanced users.

***FFD Security Level*:** This ILV is optional. If it is not present, the false finger security level is set to 0 (lowest level – default security level).

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

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

Alive asynchronous message : ILV formatted optional parameter used to allow redundant finger position asynchronous message, for a “terminal is still alive” function during fingerprint wait. If not present, the default process is executed (no redundant finger position message).

6.3.2.2 REPLY

Identifier value	0x21	
Length value	0x0006 + (< L_{BIO1}> + < L_{BIO2}> + < L_{IMG1}> + < L_{IMG2}> + < L_{CR1}> + < L_{CR2}>))	
Value (Parameters)	Request status	1 byte
	Enroll status	1 byte
	User database index	4 bytes
	Biometric data 1	L_{BIO1} bytes
	Biometric data 2 (optional)	L_{BIO2} bytes
	Image 1 (optional)	L_{IMG1} bytes
	Image 2 (optional)	L_{IMG2} bytes
	Coder Results finger 1 (optional)	L_{CR1} bytes
	Coder Results finger 2 (optional)	L_{CR2} bytes

Request status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	One or more input parameters are out of range.
ILVERR_INVALID_USER_DATA	The input ILV user data is not valid: bad identifier or wrong size.
ILVERR_TIMEOUT	The finger detection timeout has expired.
ILVERR_BASE_NOT_FOUND	The specified database does not exist.
ILVERR_CMDE_ABORTED	Command has been cancelled.
ILVERR_SAME_FINGER	User used the same finger twice.
ILVERR_FIELD_NOT_FOUND	Invalid field number.
ILVERR_INVALID_USER_ID	The specified User ID is already used in the database.
ILVERR_ALREADY_ENROLLED	Enrolled fingerprint templates already exist in the database.
ILVERR_BIO_IN_PROGRESS	Command received during biometric processing
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_SECU	Cryptographic error
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

Enroll status: If the *request status* is not ILV_OK, the *enroll status* is not returned.

ILVSTS_OK	The enrollment succeeded.
ILVSTS_DB_FULL	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.
ILVERR_BADPARAMETER	One or more input parameters are out of range.
ILVSTS_FFD	False finger detected.
ILVSTS_MOIST_FINGER	The finger can be too moist or the scanner is wet.

User database index: value of the index of the record in the database : 0 means 1st entry of the data base, 1 means 2nd entry, and so on.
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: The resulting templates returned by the MorphoSmart™ if the *Export Minutiae Size* is different than 0. The following templates format can be returned:

PK_COMP_V2: this is the default template format, used if *Biometric Algorithm Parameter* is not specified
PK_MAT: this is the returned template format when specified in the *Algo Param Field* request parameter. This format is still used for compatibility issue.
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
X984 biometric token**:** this is the returned template format when specified in the *Export Biometric Token* request parameter. This format is used for security issue.

Note: ANSI 378 and ISO FMR templates formats support more than one fingerprint per template. However, MSO returns only one fingerprint per template.

Image i **:** This ILV formatted data, which contains a fingerprint image, is included in the reply when required by the *Export Image* parameter. The number of images is equal to the number of fingers captured (usually one or two). When the required *Enrollment type* is set to 3 images per finger, the terminal returns the last captured image (i.e. the 3rd image).

Coder Results**:** This ILV is sent if the *Export Coder Results* request parameter has been set. For each captured finger, it contains advanced coder results information (reserved for experimented users).For more information please contact Sagem Sécurité.

6.3.3 IDENTIFY ID = 0x22

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

6.3.3.1 REQUEST

Identifier value	0x22	
Length value	0x0006 (+ 7 + 4 + 7+ 7+ 7+ 7+ 7)	
Value(Parameters)	Database identifier	1 byte
	Timeout	2 bytes
	Matching threshold	2 bytes
	RFU	1 byte
	Asynchronous event (optional)	7 bytes
	Matching score (optional)	4 bytes
	FFD Security Level (optional)	7 bytes
	Biometric coder selection (optional)	7 bytes
	Biometric presence mode detection (opt)	7 bytes
	Biometric matching strategy (optional)	7 bytes
	Alive asynchronous message (optional)	7 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Timeout: Finger detection timeout in seconds. Zero value corresponds to an infinite timeout.

Matching Threshold: This parameter can be set to values from 0 to 10 (Sagem Sécurité recommends 5). This parameter specifies how tight the matching threshold is. For some more information, please refer to the "MorphoSmart Overview.pdf" documentation.

RFU: Not used, set to 0.

***Asynchronous Event*:** This ILV is optional. If it is not present, no asynchronous messages will be sent.

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

***FFD Security Level*:** This ILV is optional. If it is not present, the false finger security level is set to 0 (lowest level – default security level).

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

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

Biometric matching strategy : This ILV is optional. If it is not present, the default biometric matching strategy will be used.

Alive asynchronous message : ILV formatted optional parameter used to allow redundant finger position asynchronous message, for a "terminal is still alive" function during fingerprint wait. If not present, the default process is executed (no redundant finger position message).

6.3.3.2 REPLY

Identifier value	0x22	
Length value	0x0002 + 4 + < L_{UID}> + <L_1> + … + <L_n> (+ 0x0007)	
Value(Parameters)	Request Status	1 byte
	Matching Result	1 byte
	User Database Index (if applicable)	4 bytes
	User ID (if applicable)	L_{UID} bytes
	Additional user data field 1 (if applicable)	L_1 bytes
	…	
	Additional user data field n (if applicable)	L_n bytes
	Matching score (optional)	7 bytes

Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	The matching threshold value or timeout value are not in range.
ILVERR_TIMEOUT	The finger detection timeout has expired
ILVERR_BASE_NOT_FOUND	The specified database doesn't exist
ILVERR_CMDE_ABORTED	Command has been cancelled.
ILVERR_BIO_IN_PROGRESS	Command received during biometric processing
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

Matching Result: This is the result of the identification of the coded fingerprint in the database. If the request status is not ILV_OK, the matching result is not returned.

ILVSTS_HIT	The comparison succeeded
ILVSTS_NO_HIT	It is not the same finger
ILVSTS_DB_EMPTY	The database is empty.
ILVSTS_FFD	False finger detected.
ILVSTS_MOIST_FINGER	The finger can be too moist or the scanner is wet.

User Database index: This is the index of the record (person) in the database : 0 means 1st entry of the database, 1 the 2nd entry, and so on. This field is returned only if the request status is ILV_OK, and if the matching status is ILVSTS_HIT.

User ID: ILV formatted data which contains the unique user identifier of the record that match with the fingerprint**.** This data is returned only if the request status is ILV_OK, and if the matching status is ILVSTS_HIT.

Additional User Data j**:** ILV formatted data which contains the value of the additional user data fields j, as it was provided by the *ENROLL* command, or by the *Add_BASE_Record* commands. The content of all the public fields and of all the private fields are returned in the same order as at the creation of the database. The command returns also empty fields. This data is returned only if the request status is ILV_OK, and if the matching status is ILVSTS_HIT**.**

Matching Score: This is an optional data, returned only if requested in the command. This field is returned only if the request status is ILV_OK, and if the matching status is not ILVSTS_DB_EMPTY.

6.3.4 VERIFY MATCH ID = 0x23

This function compares a searched template with several reference templates, and searches for a match. All templates (search and reference) are provided as parameters of the function.

6.3.4.1 REQUEST

Identifier value	0x23	
Length value	0x02 + <Ls> + <Lr_1> (+ <Lr_n> + 0x0004)	
Value (Parameters)	Matching threshold	2 bytes
	Search Template	Ls
	Reference template 1	Lr_1
	Reference template 2 (optional)	Lr_2
	…	
	Reference template n (optional) (n ≤ 20)	Lr_n
	Matching score (optional)	4 bytes

Matching Threshold: This parameter can be set to values from 0 to 10 (Sagem Sécurité recommends 5). This parameter specifies how tight the matching threshold is. For some more information, please refer to the "MorphoSmart Overview.pdf" documentation.

Search Template: Only one ILV formatted data packet containing the searched minutiae of one finger. The following templates format can be used: *PK_COMP V2*, *PK_MAT*, *ISO_PK* , *PKV10* or *X984 biometric token*.

Reference Templates i: One ILV formatted data packets containing a reference minutia. The following templates format can be used: *PK_COMP V2*, *PK_MAT*, *ISO_PK* , *PKV10* or *X984 biometric token*. The maximum number of reference templates is 20.

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

6.3.4.2 REPLY

Identifier value	0x23	
Length value	0x0003 (+ 0x0007)	
Value (Parameters)	Request status	1 byte
	Matching result	1 byte
	Index	1 byte
	Matching score (optional)	7 bytes

Request status:

ILV_OK	The function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	The matching threshold value, or timeout value are not in range, or there is no input biometric data, or the matching is performed with more than the maximum number of templates, or you selected index 0 with an X9.84 template.
ILVERR_INVALID_MINUTIAE	The reference ILV minutiae is not valid: bad identifier, corrupted minutiae.
ILVERR_BAD_SIGNATURE	Invalid digital signature
ILVERR_SECU_CERTIF_NOT_EXIST	The required certificate does not exist.
ILVERR_SECU_ASN1	Error while decoding an ASN1 object or bad X984 template index
ILVERR_SECU	Cryptographic error
ILVERR_CMD_INPROGRESS	Command received while another command is running

Matching result:

ILVSTS_HIT	The comparison succeeded.
ILVSTS_NO_HIT	It is not the same finger.

Index: Index in the “Biometric Data Ref i” list that matches with the “Biometric Data Search”. If a false or a too moist finger is detected, or if the finger is not authenticated, the returned index value is 0xFF. If the comparison succeeds with the reference template i (i>=1), the Index value is i-1. For example, if the comparison succeeds with the reference template 2, the Index value is 1.

***Matching Score*:** This ILV is optional. It is returned on request.

6.3.5 IDENTIFY MATCH ID = 0x24

This function identifies one biometric data sent to the terminal against the local database.
The maximum number of search template is 1.
If the database is empty, the function will return immediately

6.3.5.1 REQUEST

Identifier value	0x24	
Length value	0x0003 + <Ls> (+ 0x0004)	
Value(Parameters)	Database identifier	1 byte
	Matching threshold	2 bytes
	Search template	Ls bytes
	Matching score (optional)	4 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Matching Threshold: This parameter can be set to values from 0 to 10 (Sagem Sécurité recommends 5). This parameter specifies how tight the matching threshold is. For some more information, please refer to the "MorphoSmart Overview.pdf" documentation.

Search Template: one ILV formatted data packet containing the searched minutiae of one finger. The following templates format can be used: *PK_COMP V2*, *PK_MAT*, *ISO_PK* , *PKV10* or *X984 biometric token*.

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

6.3.5.2 REPLY

Identifier value	0x24	
Length value	0x0006 + <L_{UID}> (+ 0x0007)	
Value(Parameters)	Request Status	1 byte
	Matching Result	1 byte
	User Database Index	4 bytes
	User ID	L_{UID} bytes
	Matching score (optional)	7 bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	The matching threshold value, or timeout value are not in range, or the matching is performed with more than the maximum number of templates, or you selected index 0 with an X9.84 template.
ILVERR_TIMEOUT	The finger detection timeout has expired
ILVERR_BASE_NOT_FOUND	The specified database doesn't exist
ILVERR_BAD_SIGNATURE	Invalid digital signature
ILVERR_SECU_CERTIF_NOT_EXIST	The required certificate does not exist.
ILVERR_SECU_ASN1	Error while decoding an ASN1 object or bad X984 template index
ILVERR_SECU	Cryptographic error
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

Matching Result: This is the result of the identification of the coded fingerprint in the database. If the request status is not ILV_OK, the matching result is not returned.

ILVSTS_HIT	The comparison succeeded
ILVSTS_NO_HIT	It is not the same finger
ILVSTS_DB_EMPTY	The database is empty

User Database index: This is the index of the record in the database : 0 means 1st entry of the data base, 1 means 2nd entry, and so on. 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 contains the unique user identifier of the record that match with the fingerprint. This data is returned only if the request status is ILV_OK, and if the matching status is ILVSTS_HIT.

***Matching Score*:** This ILV is optional. It is returned on request.

6.4 DATABASE FUNCTIONS DESCRIPTION

6.4.1 CREATE DATABASE ID = 0x30

This function creates a biometric database in flash memory.

6.4.1.1 REQUEST

Identifier value	0x30	
Length value	0x0005 +(i x 0x000B) (+ 0x0004)	
Value(Parameters)	Database identifier	1 byte
	RFU	1 byte
	Maximum number of records	2 bytes
	Maximum number of fingerprints per record	1 byte
	Description of data Field 1 (optional)	11 bytes
	Description of data Field 2 (optional)	11 bytes
	…	…
	Description of data Field I (optional)	11 bytes
	Biometric Algorithm Param (optional)	4 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

RFU: Must be set to 0.

Maximum number of records : maximum number of records which can be stored in the database. Must not exceed 500 without a specific license (Please refer to the 7.3.4.5 chapter for further information). Usually, there is one record per user.

Maximum number of fingerprints per record: maximum number of fingerprints which can be stored in each record of the database (1, 2, 10 or 20, please refer to the 7.3.4.5 chapter for further information).

At record generation, the terminal allows the host to save less fingerprints per user than specified in the Create Database command (i.e. 1 in case of 2 fingerprints maximum per record).

Description of User Data field i : A list of optional *Public* and/or *private* fields dedicated to store data related to the user, such as the first name, the last name, and other useful data. These fields are provided only for Host convenience : except size check, the terminal doesn't verify the contents of these fields.

The maximum number of additional fields available depends on MorphoSmart™ type and configuration (see "**DataBase size limits**" section in ***Additional User Data field i*** description)

Biometric Algorithm Parameter: optional ILV formatted data, used to specify the type of templates to be stored in the database : either regular templates or normalized templates (such as PK_Comp_Norm and PK_MAT_Norm). When omitted, the default value (0x00) applies automatically, which specifies a database for regular template formats such as PK_Comp, PK_MAT, ISO and ANSI formatted templates

For more information about normalized templates please contact Sagem Sécurité.

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 doesn't 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, is 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.

6.4.1.2 REPLY

Identifier value	0x30	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BASE_ALREADY_EXISTS	The Database identifier is wrong, or this database already exist
ILVERR_NO_SPACE_LEFT	The Database can not be created because there is not enough memory
ILVERR_BADPARAMETER	Wrong number of finger, ILV Format incorrect or invalid database configuration
ILVERR_OUT_OF_FIELD	The number of additional fields is superior to the maximum allowed (***Please refer to the 7.3.4.5 chapter for further information)***
ILVERR_FIELD_INVALID	Additional field name length is more than 6
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.2 ERASE BASE ID = 0x32

This function erases all records in the biometric module local database.

6.4.2.1 REQUEST

Identifier value	0x32	
Length value	0x0001	
Value(Parameters)	Database Identifier	1 byte

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

6.4.2.2 REPLY

Identifier value	0x32	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BASE_NOT_FOUND	The specified database doesn't exist
ILVERR_BADPARAMETER	ILV Format incorrect
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.3 ERASE ALL BASE ID = 0x34

This function erases all records from all local databases (since current release of MorphoSmart™ does not support management of multiple databases, there is only one database available for now).

6.4.3.1 REQUEST

Identifier value	0x34
Length value	0x0000

6.4.3.2 REPLY

Identifier value	0x34	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BADPARAMETER	ILV Format incorrect
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.4 DESTROY BASE ID = 0x3B

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

6.4.4.1 REQUEST

Identifier value	0x3B	
Length value	0x0001	
Value(Parameters)	Database Identifier	1 byte

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.

6.4.4.2 REPLY

Identifier value	0x32	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BASE_NOT_FOUND	The specified database doesn't exist
ILVERR_BADPARAMETER	ILV Format incorrect
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.5 DESTROY ALL BASE ID = 0x33

This function deletes all the local databases in flash memory (when only one, then this command is identical to *DESTROY_BASE* command).

6.4.5.1 REQUEST

Identifier value	0x33	
Length value	0x0001	
Value(parameters)	RFU	1 byte

RFU: Must be set to 0.

6.4.5.2 REPLY

Identifier value	0x33	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Wrong number of finger or ILV Format incorrect.
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.6 ADD BASE RECORD ID = 0x35

This function adds a record for a user to the specified local database.

6.4.6.1 REQUEST

Identifier value	0x35	
Length value	0x0001 + <L_1> (+ <L_2>) + < L_{UID}>+ (<L_{Data1} >+ … + < L_{Datai}>)	
Value(Parameters)	Database identifier	1 byte
	Reference template 1	L_1 bytes
	Reference template 2 (optional)	L_2 bytes
	…	
	Reference template 20 (optional)	L_{20} bytes
	User ID	L_{UID} bytes
	Additional User Data 1 (optional)	L_{Data1}bytes
	…	
	…	
	Additional User Data i (optional)	L_{Datai} bytes
	No check on template (optional)	4 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Reference Template i : ILV formatted data which contains a fingerprint minutiae, according to one of the following format : *PK_COMP V2*, *PK_MAT*, *ISO_PK*, *PKV10* or *X984 biometric token*. The terminal checks that the template doesn't match with any of the already stored templates, otherwise the command is rejected with a specific error code The number of reference templates must be less or equal to the maximum number of fingerprints per record (i.e. 1, 2, 10 or 20), as specified at the creation of the DataBase by the CREATE DATABASE command

***User ID*:** ILV formatted data which contains 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.

***Additional User Data i* :** optional ILV formatted data which contains the data to store in the additional user data field # i. The terminal expects one Additional user Data ILV for each of the additional user data fields defined with the CREATE DATABASE command. The order of the additional user data must be the same as the order of the additional user data fields. As the content of the data is not checked by the terminal (except excessive size), the data can be empty (0 bytes).

***No check on template* :** optional ILV formatted data used to disable reference templates checks.

6.4.6.2 REPLY

Identifier value	0x35	
Length value	0x0006	
Value(Parameters)	Request Status	1 byte
	Base Status	1 byte
	User Database Index	4 bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BADPARAMETER	At least one biometric data parameter is required
ILVERR_INVALID_USER_DATA	The input ILV user data is not valid: bad identifier, wrong size
ILVERR_INVALID_MINUTIAE	The reference ILV minutiae is not valid: bad identifier, corrupted minutiae

ILVERR_ALREADY_ENROLLED	The person is already in this database
ILVERR_BASE_NOT_FOUND	The specified database doesn't exist
ILVERR_INVALID_USER_ID	The record identifier already exists in the database
ILVERR_SAME_FINGER	User used the same finger twice
ILVERR_FIELD_NOT_FOUND	Invalid field number
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

Base Status:

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

User Database Index: index of the record in the database : 0 means 1st entry of the data base, 1 means 2nd entry, and so on. If the request status is not ILV_OK, the User Database Index is not returned.

6.4.7 REMOVE BASE RECORD ID = 0x36

This function removes a record from the specified local database.

6.4.7.1 REQUEST

Identifier value	0x36	
Length value	0x0001 +< L_1>	
Value(Parameters)	Database identifier	1 byte
	User ID or User Index	L_1

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

***User ID*:** ILV formatted data, which contains the unique user identifier of the record to remove, as it was provided with by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation.

***User Index*:** ILV formatted data, which contains the index of the record in the database, returned by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation

6.4.7.2 REPLY

Identifier value	0x36	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BADPARAMETER	The input ILV Identifier Data is not valid: bad ID, bad length
ILVERR_INVALID_USER_ID	The User ID does not exist in the database
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.8 FIND USER BASE ID = 0x38

This function searches for the 1st record which has a User ID, or an additional user data public field) equal to a specified value (same length, same content).

6.4.8.1 REQUEST

Identifier value	0x38	
Length value	0x0009 + < L>	
Value(Parameters)	Database identifier	1 byte
	Field index	4 bytes
	Record offset	4 bytes
	Data to find	L bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Field index: Field index on which the search is performed. 0 specifies User ID field, 1 the 1st additional user data field (if any), 2 the 2nd additional data field (if any), and so on. The specified additional user data field must be public.

Record offset: Value of the index of the entry of the database to start searching**.** 0 specifies to start with 1st entry of the database. This parameter allows the host to provided a "search for next match" function.

Data to find: An ILV formatted data packet that contains the data to find.

Identifier value	ID_PUC_DATA	
Length value	<L>	
Value (Parameters)	Value of the Data to find	L bytes

Value of the Data to find : binary value to search for.

6.4.8.2 REPLY

Identifier value	0x38	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte
	User DataBase index	4 bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BADPARAMETER	The input ILV Identifier Data is not valid: bad Id, bad length
ILVERR_BASE_NOT_FOUND	The specified database does not exist
ILVERR_USER_NOT_FOUND	No record contains the searched data
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

User Database index: index of the matching record in the database : 0 means 1st entry of the data base, 1 means 2nd entry, and so on. If the request status is not ILV_OK, the User Database Index is not returned.

6.4.9 GET DATA ID = 0x3F

This function return the content of one of the additional user data public field, for a specified record.

6.4.9.1 REQUEST

Identifier value	0x3F	
Length value	0x0005 +< L_1 >	
Value(Parameters)	Database identifier	1 byte
	Field index	4 bytes
	User ID or User Index	L_1

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Field index: index of the additional user data field to read. 0 the 1st additional user data field (if any), 1 the 2nd additional data field (if any), and so on. The specified field must be public.

***User ID*:** ILV formatted data, which contains the unique user identifier of the record, as it was provided with by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation.

***User Index*:** ILV formatted data, which contains the index of the record in the database, returned by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation.

6.4.9.2 REPLY

Identifier value	0x3F	
Length value	0x0001 + <L>	
Value(Parameters)	Request Status	1 byte
	User Data	L bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BADPARAMETER	The input ILV Identifier Data is not valid: bad Id, bad length
ILVERR_BASE_NOT_FOUND	The specified database does not exist
ILVERR_USER_NOT_FOUND	The required User ID has not been found in the database
ILVERR_FIELD_NOT_FOUND	The required field doesn't exist in the database
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

User Data : content of the specified ***Additional User Data field***, a string of character without end of string byte (0x00). Returned only if the specified record has been found (Request Status is ILV_OK).

6.4.10 GET_PUBLIC_FIELDS ID = 0x3E

This function return the content of a specified public Additional User Data Field, for all records stored in the DataBase.

6.4.10.1 REQUEST

Identifier value	0x3E	
Length value	0x0005	
Value(Parameters)	Database identifier	1 byte
	Field index	4 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Field index: value witch indicates which User Data Field has to be read. Value 0 means User ID data field and 1 the first Additional User Data Field. The Additional User Data Field must be public. Please refer to ***Additional User Data field i*** description.

6.4.10.2 REPLY

Identifier value	0x3E	
Length value	0x0005 + < N_1> + … < N_k>	
Value(Parameters)	Request Status	1 byte
	User number	4 bytes
	Field 1	N_1 bytes
	Field 2	N_2 bytes
	…	
	Field k	N_k bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BADPARAMETER	Index identifies an index that is not valid: non-existent, or private field
ILVERR_BASE_NOT_FOUND	The specified database does not exist
ILVERR_FIELD_NOT_FOUND	The required field doesn't exist in the database
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

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

6.4.11 GET BASE CONFIG ID = 0x07

This function return the description of the specified DataBase.

6.4.11.1 REQUEST

Identifier value	0x07	
Length value	0x0001	
Value(Parameters)	Database Identifier	1 byte

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

6.4.11.2 REPLY

Identifier value	0x07	
Length value	0x0012 + L_{ts} + (i x 0x000B) + 4	
Value(Parameters)	Request Status	1 byte
	Number fingers/record	1 byte
	Max record number	4 bytes
	Current Record Number	4 bytes
	Free Record Number	4 bytes
	Number of Fields	4 bytes
	ILV Timestamp (not used)	L_{ts} bytes
	Description of data Field 1	11 bytes
	Description of data Field 2	11 bytes
	…	
	Description of data Field i	11 bytes
	Biometric Algorithm Param	4 bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BASE_NOT_FOUND	There is no Database corresponding to the Identifier specified in the Request
ILVERR_BADPARAMETER	The input ILV Identifier Data is not valid: bad ID, bad length
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

Number fingers/record: Maximum number of fingerprints per record.

Max record number: Maximum number of records which can be stored in the database.

Current Record Number: Number of records currently saved in database.

Free Record Number: Number of entries currently available in database.

Number of Fields: Number of Additional User Data Field of the database.

ILV Timestamp: Field only present for compatibility with MorphoAccess™ product.

Identifier value	ID_TIMESTAMP	
Length value	L_{ts}	
Value	Value	L_{ts} bytes (usually 12 bytes)

Description of Data field i : ILV formatted data which contains the description of the Additional User Data Field i. There is one ILV data per additional user data field (*Public* or *private*) defined in the structure of the Database .

Biometric Algorithm Param **:** ILV formatted data which contains the type of database : either standard (0x00), or normalized (0x01). For more information about normalization please contact Sagem Sécurité.

6.4.12 UPDATE PUBLIC DATA ID = 0x3C

This function modifies the content of one or more Additional User Data Public Fields of a specified record. The record to be updated is specified either by it's User ID field or by it's index into the DataBase.

6.4.12.1 REQUEST

Identifier value	0x3C	
Length value	0x0002 + L_1 + L_2 + … + L_n	
Value(Parameters)	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: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Field number: Indicate number of fields to update in the record. The number of public fields to be updated it at least one, and at maximum equal to the number of public fields defined at the creation of the DataBase (please refer to *CREATE_BASE*).

***User ID*:** ILV formatted data, which contains the unique user identifier of the record, as it was provided with by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation.

***User Index*:** ILV formatted data, which contains the index of the record in the database, returned by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation.

***Field*:** An ILV formatted data packet that identify the public field to be updated, and the new value of this field.

6.4.12.2 REPLY

Identifier value	0x3C	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BASE_NOT_FOUND	There is no database corresponding to the Identifier specified in the Request
ILVERR_BADPARAMETER	The input ILV Identifier Data is not valid: bad Id, bad length
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.4.13 UPDATE PRIVATE DATA ID = 0x3D

This function modifies the content of one or more Additional User Data Fields (Public or Private) of a specified record. The record to be updated is specified either by it's User ID or by it's index into the DataBase.

In order to protect the content if Private Fields, this function performs first a user identity verification (like Verify function), and then, if successful, the contents of all specified data fields are updated.

6.4.13.1 REQUEST

Identifier value	0x3D	
Length value	0x0006 + L_1 + L_2 + … + L_n + <7> + <7> + <4>	
Value(Parameters)	Database Identifier	1 byte
	Field number = n+1	1 byte
	Timeout	2 bytes
	Matching threshold	2 bytes
	User ID or User Index	L_1
	Field 0	L_2
	Field 1	L_3
	….	…
	Field n	L_n
	Asynchronous event (optional)	7 bytes
	FFD Security Level (optional)	7 bytes
	Biometric coder selection (optional)	7 bytes
	Biometric presence mode detection (opt)	7 bytes
	Biometric matching strategy (optional)	7 bytes

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

Field number: Indicates number of fields to update. The number of fields to be updated it at least one, and at maximum equal to the number of fields (public and private) defined at the creation of the DataBase (please refer to *CREATE_BASE*).

***User ID*:** ILV formatted data, which contains the unique user identifier of the record, as it was provided with by the *ENROLL* command, or by the *ADD BASE RECORD* command, at record creation

***User Index*:** index of the record, into the database, returned by *ENROLL* or *ADD BASE RECORD* commands.

***Field*:** An ILV formatted data packet that identify the field (either public or private) to be updated, and the new value of this field.

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 Sécurité recommends 5). This parameter specifies how tight the matching threshold is. For some more information, please refer to the "MorphoSmart Overview.pdf" documentation.

***Asynchronous Event*:** This ILV is optional. If it is not present, no asynchronous messages will be sent.

***FFD Security Level*:** This ILV is optional. If it is not present, the false finger security level is set to 0 (lowest level – default security level).

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

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

Biometric matching strategy : This ILV is optional. If it is not present, the default biometric matching strategy will be used.

6.4.13.2 REPLY

Identifier value	0x3D	
Length value	0x01	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function
ILVERR_BASE_NOT_FOUND	There is no database corresponding to the Identifier specified in the Request
ILVERR_BADPARAMETER	The input ILV Identifier data is not valid: bad ID, bad length
ILVERR_NO_HIT	Finger does not match record template
ILVERR_CMD_INPROGRESS	Command received while another command is running
ILVERR_FFD	False finger detected
ILVERR_MOIST_FINGER	The finger can be too moist or the scanner is wet
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported

6.5 SECURITY FUNCTIONS DESCRIPTION

IMPORTANT:

Those commands are only available with secured MorphoSmart™ (MorphoSmart™ S).

Some more information can be found in the document "MorphoSmartKeyManagementSystem.pdf" or "MorphoSmartOverview.pdf". Reading this document is important for a good understanding of those commands.

6.5.1 Implementation scheme

6.5.2 Offered security protocol

With "offered security" security option, the MorphoSmart™ is authenticated, the Host is not authenticated. This mode accepts secured frames as well as non secured frames. It is the host responsibility to decide which ILV frames shall or shall not be secured.

This security option insures:

- MorphoSmart™ authentication
- Frame integrity (on host request only) thanks to an asymmetric digital signature
- Anti-replay

Advantages of this mode are:

- This mode is less complex to implement than the "Tunneling" protocol (no Host private key to manage).

In order to implement offered security implementation, the host shall perform the following operations:

INITIALISATION

- Check MorphoSmart™ configuration (SECU_GET_CONFIG)
- Get MorphoSmart™ certificate (MSO_SECU_ReadCertificate)
- Retrieve MorphoSmart™ certification path (access to a LDAP directory, MSO_SECU_ReadCertificate, …) and verify MorphoSmart™ certification path.

FOR EACH SECURED FRAME

- Send a request: Add the security envelope (SECU_PROTOCOLE)
- Get the response:
    - Extract the security envelop
    - Verify the signature
    - Verify the hash of the sent ILV frame

Request (command issued by the Host)

SECU_PROTOCOLE	Length	DATA	RAND

DATA = regular ILV request.

RAND (4 bytes) = Anti-replay random number.

Response (command issued by the MSO):

SECU_PROTOCOLE	Length	ILV status	DATA	RAND	HASH	SIGN

ILV status: refer to chapter "Error codes"

DATA = regular ILV response.

RAND (4 bytes) = Anti-replay random number

HASH (20 bytes) = SHA-1(DATA $_{request\_frame}$) = hash computed on DATA field of the request frame

SIGN (x bytes) = [DATA || RAND || HASH] $_{PRIV\_MSO}$ = the MorphoSmart™ signs the concatenation of DATA, RAND, HASH.

Note: possible signature algorithms are:

- DSA
- RSA 1024, PKCS#1 v1.5 padding
- RSA 2048, PKCS#1 v1.5 padding

6.5.3 Tunneling implementation

With “tunneling protocol”, both the MorphoSmart™ and the Host are authenticated. This mode only accepts secured frames.

This security mode insures:

- Mutual authentication between the host and the MorphoSmart™. Mutual authentication is performed according to Needham-Schroeder Protocol for Public Keys.
- Anti-replay
- Frame integrity (signature)
- Confidentiality (encryption)

Advantages of this mode are:

- A mutual authentication is performed.
- This mode is more efficient: after key negotiation, this mode only uses symmetric keys that have a more efficient computation time.
- Frames are encrypted

In order to implement tunneling security implementation, the host shall perform the following operations:

INITIALISATION – MUTUAL AUTHENTICATION

- Check MorphoSmart™ configuration (SECU_GET_CONFIG)

- The host performs the first part of the mutual authentication (SECU_MUTUAL_AUTH_INIT1)
  - If the MSO returns the error ILVERR_BAD_SIGNATURE
    - The HOST shall give the MorphoSmart™ its whole certification path (SECU_STO_CERTIFICATE).
    - The host retries the first part of the mutual authentication (SECU_MUTUAL_AUTH_INIT1)
  - the host decrypts the returned cryptogram using its private RSA key (DSA is not available with this protocol).

    [rand1 || MSO identifier] $_{HOST\_PUB}$

    rand1 is a 48 bytes random number generated by the HOST.

    MSO identifier (20 bytes) is the hash (SHA-1) of MorphoSmart™ distinguish name.

    HOST_PUB is the host RSA public key.
- The host performs the second part of the mutual authentication (SECU_MUTUAL_AUTH_INIT2)
  - The HOST computes:

    [HOST identifier || rand2 || rand1] $_{MSO\_PUB}$

    rand2 is a 48 bytes random number generated by the MSO.

    HOST identifier (20 bytes) is the hash (SHA-1) of the HOST distinguish name.
  - the HOST verifies:

    [rand2] $_{PUB\_HOST}$
- The host and the MorphoSmart™ computes:
  - the signature key **KS = rand1[1..24] xor rand2[1..24]**
  - the encryption key **KC = rand1[25..48] xor rand2[25..48]**

Note: key exchange possible algorithms are:

- RSA 1024, PKCS#1 v1.5 padding
- RSA 2048, PKCS#1 v1.5 padding

FOR EACH FRAME

- Send the request:
  - Increase the anti-replay counter
  - Add the security envelope (*SECU_PROTOCOLE*)
- Get the response:
  - Extract the security envelop
  - Decrypt the frame
  - Verify the signature
  - Verify the anti-replay counter

The security encapsulation is performed as:

Request (command issued by the Host)

SECU_PROTOCOLE	Length	DATA (regular ILV)	CNT	SIGN

Response (command issued by the MSO):

SECU_PROTOCOLE	Length	ILV Status	DATA (regular ILV)	CNT	SIGN

ILV status: refer to chapter "Error codes"

DATA = regular ILV to secure

CNT (4 bytes) = Anti-replay counter.

This counter must be incremented by the host each time a new request is sent. The anti-replay counter of frame N must be strictly higher than anti-replay counter of frame N-1.

Frames (N, N-1) anti-replay counter do not need to be consecutive numbers. This characteristic allows a frame to be lost without having a synchronization problem. Even when the HOST does not receives any answer from the MSO; the HOST shall increase this random number.

Once anti-replay counter is equal to 0xFFFFFFFF, there is no higher CNT, another secure context must be established thanks to another mutual authentication (but such a case shall never occurs if the host application regularly renews its security context).

SIGN (8 bytes) = [DATA || CNT || 0…0] $_{KS}$ = the MorphoSmart™ signs the concatenation of DATA, CNT, and from 0 till 7 padding characters (0x00) to sign a buffer multiple of 8 bytes. Signature algorithm is triple DES EDE3 CBC MAC. Initialization vector is 0x0000000000000000. Encryption key is KS.

Encryption: [DATA || CNT || X...X] $_{KC}$ = the MorphoSmart™ encrypts the concatenation of DATA, CNT, and from 0 till 7 padding characters from the signature in order to have a buffer multiple of 8 bytes. Encryption algorithm is triple DES EDE3 CBC. Initialization vector is 0x0000000000000000. Encryption key is KC.

6.5.4 SECU_GET_CONFIG ID = 0x80

This function returns MorphoSmart™ security configuration.

6.5.4.1 REQUEST

Identifier value	0x80	
Length value	0x0000	
Value(Parameters)	None	0 byte

6.5.4.2 REPLY

Identifier value	0x80	
Length value	0x001C	
Value(Parameters)	Request Status	1 byte
	Serial Number	24 bytes
	Security options	1 byte
	FAR limit	2 bytes

Serial Number: MorphoSmart™ serial number

Security options: Binary mask as follow:

b8	b7	b6	b5	b4	b3	b2	b1	Biometric Subtype
0	0	0	0	0	0	0	0	No security options.
							1	The MorphoSmart™ uses the “tunneling” protocol.
						1		The MorphoSmart™ uses the “offered security” protocol.
					1			The MorphoSmart™ uses only templates with an X9.84 envelop and a signature.
				1				The MorphoSmart™ only accept signed firmware packages during retrofit. With a secure MorphoSmart™ (series S) this option is always set and cannot be canceled.
			1					The MorphoSmart™ never export its matching score.
X	x	x	x					RFU

Note: if b1 or b2 are set, and if b4 is not set, the MorphoSmart™ accepts templates with or without the X9.84 envelope.

FAR limit: All matching commands with a FAR less restrictive than FAR limit are rejected. It is still possible to require a more restrictive FAR value. Range of this value is 0 to 10, see “Matching Threshold value” chapter for some more details.

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Request.
ILVERR_CMD_INPROGRESS	Command received while another command is running

6.5.5 SECU_READCERTIFICATE ID = 0x81

This function reads an X509 certificate from MSO certification path.

6.5.5.1 REQUEST

Identifier value	0x81	
Length value	0x0001	
Value(Parameters)	Certificate index	1 byte

Certificate index: Certificate index within the certification path. This parameter can have the following values:

0: the returned certificate is the MSO certificate.

1: the returned certificate is the certificate that has signed MSO certificate.

…

N: the certificate that will be returned is the certificate that signed 'N-1' certificate.

6.5.5.2 REPLY

Identifier value	0x81	
Length value	0x0001 + <L>	
Value(Parameters)	Request Status	1 byte
	X509 certificate	L bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Request.
ILVERR_CERTIF_NOT_EXIST	The required signature does not exist
ILVERR_CMD_INPROGRESS	Command received while another command is running

***X509 certificate*:** The requested certificate returned in a DER format.

6.5.6 SECU_STO_CERTIFICATE ID = 0x82

This function stores a host X509 certificate within the MorphoSmart™ circular area. The MorphoSmart™ can store 30 certificates in its certificate area, including host certificates, and its own certificate certification path (the deeper the MorphoSmart™ certification path is, the less place it remains for host certificates). Once there is no more room, the oldest host certificate is replaced.

6.5.6.1 REQUEST

Identifier value	0x82	
Length value	<L>	
Value(Parameters)	X509 certificate	L bytes

***X509 certificate*:** A X509 certificate, encoded in DER format, to saved into the MorphoSmart™.

6.5.6.2 REPLY

Identifier value	0x82	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Request.
ILVERR_SECU_BAD_STATE	Request is performed within a bad security state.
ILVERR_BAD_SIGNATURE	Error while verifying certificate signature: wrong signature unknown certificate issuer
ILVERR_SECU_ASN1	Error while decoding the X509 certificate.
ILVERR_CMD_INPROGRESS	Command received while another command is running

6.5.7 SECU_STO_PKCS12 ID = 0x83

This function loads a PKCS#12 token in the MorphoSmart™. The PKCS12 token contains MorphoSmart™ whole certification path and the MorphoSmart™ private key. This token is protected thanks to an anti-replay counter and thanks a specific MorphoSmart™ internal key (distinct for each MorphoSmart™).

6.5.7.1 REQUEST

Identifier value	0x83	
Length value	0x0004 +<L>	
Value(Parameters)	Anti-replay counter	4 bytes
	PKCS#12 token	L bytes

Anti-replay counter: Counter that participates to protect the PKCS#12 token.

PKCS#12 token: An ILV formatted data that contains the PKCS#12 token, encoded in DER format.

Identifier value	ID_PKCS12_TOKEN	
Length value	L	
Value	Protected PKCS#12 token	L bytes

6.5.7.2 REPLY

Identifier value	0x83	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Request.
ILVERR_SECU_BAD_STATE	Request is performed within a bad security state.
ILVERR_SECU_ANTIREPLAY	The PKCS#12 token is older or equal than the already embedded one.
ILVERR_BAD_SIGNATURE	The integrity MAC of the PKCS#12 token is wrong.
ILVERR_SECU_ASN1	Error while decoding the PKCS#12 token.
ILVERR_CMD_INPROGRESS	Command received while another command is running

6.5.8 SECU_MUTUAL_AUTH_INIT1 ID = 0x84

This command is used to establish a secure tunnel between the MorphoSmart™ and the host (phase 1).

6.5.8.1 REQUEST

Identifier value	0x84	
Length value	<L>	
Value(Parameters)	Host X509 certificate	L bytes

***Host X509 certificate*:** The host certificate encoded in DER format.

6.5.8.2 REPLY

Identifier value	0x83	
Length value	0x0001 + <L1> + <L2>	
Value(Parameters)	Request Status	1 byte
	MSO X509 certificate	L1 bytes
	Cryptogram	L2 bytes

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Request.
ILVERR_SECU_BAD_STATE	Request is performed within a bad security state.
ILVERR_BAD_SIGNATURE	Error while verifying certificate signature: wrong signature unknown certificate issuer
ILVERR_SECU_ASN1	Error while decoding the X509 certificate.
ILVERR_CRYPTO_ERROR	Error while performing a cryptographic operation.
ILVERR_CMD_INPROGRESS	Command received while another command is running

***MSO X509 certificate*:** The MSO certificate encoded in DER format.

***Cryptogram*:** Contains data encrypted with Host public key. The plaintext data contain:
S1 (24 bytes) | C1 (24 bytes) | MSO ident (20 bytes)
S1 and C1 are random numbers generated by the MorphoSmart™.

6.5.9 SECU_MUTUAL_AUTH_INIT2 ID = 0x85

This command is used to establish a secure tunnel between the MorphoSmart™ and the host (phase 2).

6.5.9.1 REQUEST

Identifier value	0x85	
Length value	<L>	
Value(Parameters)	Cryptogram	L bytes

***Cryptogram*:** Contains data encrypted with MSO public key. The plaintext data contain:
MSO ident (20 bytes) | S2 (24 bytes) | C2 (24 bytes) | S1 (24 bytes) | C1 (24 bytes)
S2 and C2 are random numbers generated by the host.
The symmetric signature key is S = S1 xor S2.
The symmetric encryption key is C = C1 xor C2.

6.5.9.2 REPLY

Identifier value	0x85	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte
	Cryptogram	L bytes

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Request.
ILVERR_SECU_BAD_STATE	Request is performed within a bad security state.
ILVERR_CRYPTO_ERROR	Error while performing a cryptographic operation.
ILVERR_CMD_INPROGRESS	Command received while another command is running

***Cryptogram*:** Contains data encrypted with Host public key. The plaintext data contain:
S2 (24 bytes) | C2 (24 bytes)

6.5.10 SECU_PROTOCOLE ID = 0x86

This command is used to encapsulates regular ILV with a secured command. Full details are given in the "*implementation scheme*" paragraph.

The implementation depends on the kind of secure protocol used.

6.6 OTP FUNCTIONS DESCRIPTION

The OTP MorphoSmart™ products have a specific database and do not use the standard database. There is only one user enrolled in this specific database : the OTP user.

The OTP user record contains the items listed below :

- 2 reference templates (i.e. 2 fingerprints)
- one User ID : OTP user identifier (optional)
- 2 additional data fields (optional)
- One OTP user data field (optional)

6.6.1 OTP_ENROLL_USER ID = 0xB0

This command enrolls the OTP user in the terminal.

When there is no OTP user recorded in the terminal, the process is :

- The valid enrolment allowance pin code must be provided in the command : the terminal checks it and then the user is requested to place the 1st finger to record on the sensor (three times).
- Then the user is required to place the 2nd finger to record on the sensor (three times).
- Finally, the terminal generates a record with the two fingerprints, the User ID (if provided), and the two additional user data fields (if provided).

When there is already one OTP user recorded, and if the replacement of the current OTP user is allowed, the process :

- starts with the current OTP user identification, if no pin code is provided. It means that the current OTP user is required to place a recorded finger on the sensor, then the process is identical to the 1st OTP user enrolment process.
- Is identical to 1st OTP user enrolment process if the valid enrolment allowance pin code is provided.

6.6.1.1 REQUEST

Identifier value	0xB0	
Length value	1+ 5 + <L_{UID}>+< L_{Data1}>+< L_{Data2}>+< L_{Data3}>+<0x0004>+<L_{TKB}>+0x0007)	
Value (Parameters)	Export Minutiae Size	1 byte
	Pin Code (optional)	5 bytes
	User ID (optional)	L_{UID} bytes
	Additional user data field 1 (optional)	L_{Data1} bytes
	Additional user data field 2 (optional)	L_{Data2} bytes
	Biometric Algorithm Parameter (optional)	4 bytes
	Export biometric token (optional)	L_{TKB} bytes
	Asynchronous event (optional)	7 bytes

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. See the *Algo Param Field* parameter to choose the template format.
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.

Pin Code: optional ILV formatted data, which contains the enrollment allowance pin code, which is mandatory for the first OTP user enrollment. The enrolment allowance check process is different, if the Pin Code ILV is included or not to the command :

- If the OTP pin ILV data is not included, the terminal allows the enrolment process, after identification of the current user : the user is required to place one of the recorded finger on the terminal sensor. If there is none current user, the command is rejected.
- If the OTP pin is included in the OTP ENROLL USER command, the terminal allows the enrolment process, if this ILV contains the valid enrolment allowance pin code (a new pin code is computed after each successful enrolment.

***User ID*:** optional ILV formatted data, which contains the unique user identifier. This identifier can be retrieved by a successful OTP GENERATE command, and by the OTP GET STATUS command.

***Additional User Data field j*:** optional ILV formatted data, which contains additional user data (such as first name, and last name). The terminal allows the host to provide one or two additional user data fields. These data are retrieved by a successful OTP GENERATE command and by the OTP GET STATUS command.

Identifier value	ID_PUC_DATA	0x14
Length value	L_{UID}	
Value	Additional user data field j	L_{Dataj} bytes, up to 15 bytes

***Biometric Algorithm Parameter*:** This ILV is optional. If not present, the computed templates will be exported in the PK_COMP V2 format.

Export Biometric Token (*TKB Settings*): This ILV is optional. Use this ILV to export the template in a X9.84 biometric format.

***Asynchronous Event*:** This ILV is optional. If it is not present, no asynchronous messages will be sent.

6.6.1.2 REPLY

Identifier value	0xB0	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte
	Biometric data	L_{BIO1} bytes

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Parameter
ILVERR_OTP_PIN_NEEDED	Pin needed for first enrolment
ILVERR_OTP_REENROLL_NOT_ALLOWED	Re-enrolment not allowed
ILVERR_OTP_ENROLL_FAILED	Enrolment failed (status != ILV_OK)
ILVERR_OTP_IDENT_FAILED	Identification failed (status != ILV_OK except DB errors)
ILVERR_OTP_LOCK_ENROLL	The OPT_USER_ENROLL command is locked.

Biometric data: The resulting templates returned by the MorphoSmart™ if the *Export Minutiae Size* is different than 0. The following templates format can be returned:

PK_COMP V2: this is the default template format,
PK_MAT: this is the returned template format when specified in the *Algo Param Field* request parameter. This format is still used for compatibility issue.
X984 biometric token**:** this is the returned template format when specified in the *Export Biometric Token* request parameter. This format is used for security issue.

The OTP ENROLL USER command is locked after 5 successive failed enrolments (either invalid pin code provided or failed identification of the current user).

To unlock it, the host must send a *OTP SET PARAMETERS* command with a valid enrolment allowance PIN code. This action clears all the items recorded by the *OTP ENROLL USER* command : the two fingerprints, the user ID, and the two additional data fields. Then the current user, or a new one, must be enrolled (using the *OTP ENROLL USER* command).

6.6.2 OTP_GENERATE ID = 0xB1

This command requires the terminal to generate a OTP (One Time Password), after a successful identification of the OTP user (recorded by the *OTP ENROLL USER* command). To be identified, the user is required to place a finger on the sensor, then the terminal captures the fingerprint, and compares it with the two fingerprints saved during the user enrolment process.

6.6.2.1 REQUEST

Identifier value	0xB1	
Length value	(0x0004+ < $L_{SequenceNumber}$ > + < L_{Seed} > + 0x0007)	
Value (Parameters)	Hash algorithm (optional)	4 bytes
	Sequence number (optional)	$L_{SequenceNumber}$ = 5 up to 11 bytes
	Seed (optional)	L_{Seed} = 4 up to 19 bytes
	Asynchronous event (optional)	7 bytes

Hash algorithm: optional ILV formatted data, which enables to the select the hash algorithm to be used for OTP generation. When this data is not present, the terminal selects the default hash algorithm (SHA1).

Sequence number: optional ILV formatted data, which enables to specify the required OTP value : either the next value or the last generated value. When this data is not present, the terminal generates the next OTP value (default). $\mathbf{L_{SequenceNumber}}$ = 5 bytes for SHA1 Hash algorithm and $\mathbf{L_{SequenceNumber}}$ = 11 for HOTP algorithm.

Seed: optional ILV formatted data which contains the value of the secondary seed to be used to generate the OTP. It is a binary value, to be defined by the host. When the data is not present, a OTP value is generated using primary seed only (Seed is only needed for SHA1 Hash algorithm).

***Asynchronous Event*:** optional ILV formatted data, which specifies the asynchronous message required by the host, during identification process. If it is not present, no asynchronous messages will be sent to the host.

6.6.2.2 REPLY

Identifier value	0xB1	
Length value	0x0001 + L_1 (+ <L_{UID}> + <L_2> + <L_3>+< L_4>)	
Value (Parameters)	Request Status	1 byte
	OTP (One Time Password)	L_1 = 16 bytes
	User ID (optional)	L_{UID} bytes
	Additional user data field 1 (optional)	L_2 bytes
	Additional user data field 2 (optional)	L_3 bytes
	OTP User Data (optionnal)	L_4 bytes

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Parameter
ILVERR_NO_MORE_OTP	No more OTP (sequence number = 0)
ILVERR_OTP_NO_HIT	No Hit (or latent or FFD)
ILVERR_OTP_ENROLL_FAILED	Enrolment failed (status != ILV_OK)
ILVERR_OTP_IDENT_FAILED	Identification failed (status != ILV_OK except DB errors)
ILVERR_OTP_LOCK_GEN_OTP	The OTP_GENERATE command is locked.

***OTP (One Time Password)* :**

- SHA1 hash algorithm: a string of 16 hexadecimal ASCII characters, without end of string byte (0x00).
- HOTP hash algorithm: a string of 6, 7 or 8 decimal ASCII characters padded with 0x00 up to a 16 bytes buffer.

***User ID*:** ILV formatted data which contains the user identifier, as specified during OTP user enrolment process (using the OTP ENROLL USER command).

***Additional User Data field j*:** ILV formatted data which contains the additional user data, as specified during OTP user enrolment process (using the OTP ENROLL USER command).

***OTP user Data*:** ILV formatted data which contains the binary data defined by the host, saved by the terminal (using the OTP SET PARAMETERS command), and returned to the host without modification.

The **User ID**, the **Additional User Data fields** and the **OTP User Data**, can also be retrieved by the OTP GET STATUS command.

The OTP GENERATE command is locked after 10 successive failed identifications.

To unlock it, the host must sent an *OTP SET PARAMETERS* command with a valid enrolment allowance PIN code. This action clears all the items recorded by the *OTP ENROLL USER* command : the two fingerprints, the user ID, and the two additional data fields. Then the current user, or a new one, must be enrolled (using the *OTP ENROLL USER* command).

6.6.3 OTP_GET_STATUS ID = 0xB2

This command returns the status of the OTP function, such as the initialization status (OTP parameter state and value), and the user enrolment status. This command returns also the optional user data : the User ID, the content of Additional user data fields, and the content of the OTP User Data.

6.6.3.1 REQUEST

Identifier value	0xB3
Length value	0x0000
Value (Parameters)	None

6.6.3.2 REPLY

Identifier value	0xB3	
Length value	0x0001 + 0x0001 + 0x0002 + 0x0001(+ <L_{UID}> + <L_1> + <L_2>+ <L_3>)	
Value (Parameters)	Request Status	1 byte
	Param State	1 byte
	OTP param value	2 bytes
	User Enrolled	1 byte
	User ID (optional)	L_{UID} bytes
	Additional user data field 1 (optional)	L_1 bytes
	Additional user data field 2 (optional)	L_2 bytes
	OTP User Data (optional)	L_3 bytes

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_OTP_NOT_INITIALIZED	All OTP parameters are not initialized

Param state : binary mask which indicates the status of the OTP generator, in accordance with the information loaded by the OTP SET PARAMETERS command.

The binary mask is composed with the flags listed below :

OTP_MASK_SEQUENCE_NUMBER	0x01	When this bit is set, the sequence number N is initialized
OTP_MASK_PASSWORD	0x02	When this bit is set, the password is initialized
OTP_MASK_PIN	0x04	When this bit is set, the code pin is initialized
OTP_MASK_PARAM	0x08	When this bit is set, the OTP parameters are initialized
OTP_MASK_SEQUENCE_NUMBER_INVALID	0x10	When this bit is set, N is invalid (no more OTP available). The OTP sequence number must be reset by the OTP Set Parameters command.
OTP_MASK_USER_DATA	0x20	When this bit is set, the OTP User data is not empty

OTP param field **:** two bytes which contain the value of the OTP configuration parameters.

User enrolled**:** This byte provides the user enrollment status. If the user is enrolled, the byte value is 0x01, else this value is 0x00.

User ID**:** ILV formatted data which contains the user identifier, as specified during OTP user enrollment process (using the OTP ENROLL USER command).

***Additional User Data field j*:** ILV formatted data which contains the additional user data, as specified during OTP user enrollment process (using the OTP ENROLL USER command).

***OTP user Data*:** ILV formatted data which contains the binary data defined by the host, saved by the terminal (using the OTP SET PARAMETERS command), and returned to the host without modification.

The **User ID**, the **Additional User Data fields** and the **OTP User Data**, can also be retrieved by a successful OTP GENERATE command.

6.6.4 OTP_SET_PARAMETERS ID = 0xB3

This command allows to specify the value of the OTP parameters.

6.6.4.1 REQUEST

Identifier value	0xB4	
Length value	(0x0005 + < $L_{SequenceNumber}$ > + <L_1> + 0x0008 +< L_2>)	
Value (Parameters)	OTP Parameters (optional)	5 bytes
	Initial sequence number N (optional)	$L_{SequenceNumber}$ = 5 to 11 bytes
	Password (optional)	L_1 = 13 to 66 bytes
	Pin Code (optional)	8 bytes
	OTP User Data (optional)	L_2 = 3 to 1027 bytes

OTP Parameters: optional ILV formatted data, which contains the value of the OTP configuration parameters.

Initial sequence number N: optional ILV formatted data, which contains the initial value of the OTP generation sequence number. For SHA1 hash algorithm, this value is equal to the number of OTP that the terminal is allowed to generate. $\mathbf{L_{SequenceNumber}}$ = 5 bytes for SHA1 Hash algorithm and $\mathbf{L_{SequenceNumber}}$ = 11 for HOTP algorithm.

Password: optional ILV formatted data, which contains the primary seed used for OTP generation. This password must be specified at least once, in order to allow the terminal to be able to generate OTP values.

Pin Code: optional ILV formatted data, which contains the enrollment allowance pin code. A valid pin code must be stored in the terminal to allow the recording of the OTP user, using the OTP ENROLL USER command.

***OTP user Data*:** optional ILV formatted data, which contains binary data, defined by the host, saved by the terminal, and returned without modification by a successful OTP GENERATE command, and by the OTP GET STATUS command.

6.6.4.2 REPLY

Identifier value	0xB4	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Parameter
ILVERR_OTP_LOCK_SET_PARAM	The OTP_SET_PARAMETERS command is locked.

The *OTP SET PARAMETERS* command is locked after 5 successive attempts to store an invalid enrolment allowance pin code. When the command is locked, the OTP configuration is cleared (all parameters are reset to zero). To return to an operational state, the OTP configuration must be initialized : it means that the host has to send an *OTP SET PARAMETERS* command with an initial enrolment allowance PIN code, the initial OTP sequence number, and an OTP password. Then, the current user, or a new one, must be enrolled (using the *OTP ENROLL USER* command).

6.7 CONFIGURATION FUNCTIONS DESCRIPTION

6.7.1 GET_MSO_CONFIG ID = 0x90

This function retrieves the value of one parameter of the system configuration.

6.7.1.1 REQUEST

Identifier value	0x90	
Length value	0x0002	
Value(Parameters)	Parameter identifier	2 bytes

Parameter identifier: It corresponds to the MSO parameter to retrieve. This parameter can be:
- ***Sensor_Window_Position****: orientation of the sensor (only available for MorphoSmart™ CBM)*
- ***Sleep_TimeOut****: TimeOut for the System to Sleep (in ms) (only available for a serial Interface)*

6.7.1.2 REPLY

Identifier value	0x90	
Length value	0x0001+0x0002+<L>	
Value(Parameters)	Request Status	1 byte
	Parameter identifier	2 bytes
	Parameter Value	L bytes

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Parameter
ILVERR_CMD_INPROGRESS	Command received while another command is running

Parameter identifier: The parameter identifier to retrieve. This parameter can be:
- ***Sensor_Window_Position***
- ***Sleep_TimeOut***

Parameter Value: The value of the requested parameter identifier.

6.7.2 MODIFY_MSO_CONFIG ID = 0x91

This function modifies the value of one parameter of the system configuration. This new value will be taken in consideration only if the MSO has reboot.

6.7.2.1 REQUEST

Identifier value	0x91	
Length value	0x0003	
Value(Parameters)	Parameter identifier	2 bytes
	Parameter Value	L bytes

Parameter identifier: This parameter identifier corresponds to the MSO parameter to modify. This parameter can be:

- ***Sensor_Window_Position****: orientation of the sensor (only available for MorphoSmart™ CBM)*
- ***Sleep_TimeOut****: TimeOut for the System to sleep (in ms) (only available for a serial Interface)*

Parameter Value: This parameter value corresponds to the value that could take the parameter identifier.

6.7.2.2 REPLY

Identifier value	0x91	
Length value	0x0001	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded.
ILVERR_ERROR	An error occurred during the execution of the function.
ILVERR_BADPARAMETER	Invalid Parameter
ILVERR_CMD_INPROGRESS	Command received while another command is running

6.8 UNLOCKING FUNCTIONS DESCRIPTION

The MSO/CBM terminal can be delivered with an automatic lock feature, which is activated in the factory plant (on customer request).

At start-up (i.e. power up), the terminal is locked to prevent unauthorized applications to access its functions.

In order to unlock the terminal, the host must execute an unlock process, using a secret key. Then the terminal remains unlocked until next start-up: it means that the unlock process must be executed by the host after each power up of the terminal.

When a locked MSO / CBM is plugged, you must first unlock it to access the desired functions.

Terminal unlocking is performed in two steps:

- **Get an unlock seed** : start unlock process or get back “terminal not locked” status (in that case, the unlock process is useless : the terminal is already unlocked)
- **Cipher this seed** using the secret key and transmit the result to the MSO / CBM.

The Cipher algorithm used is DES.

If the result of the second step is successful, the terminal functions are available as usual.

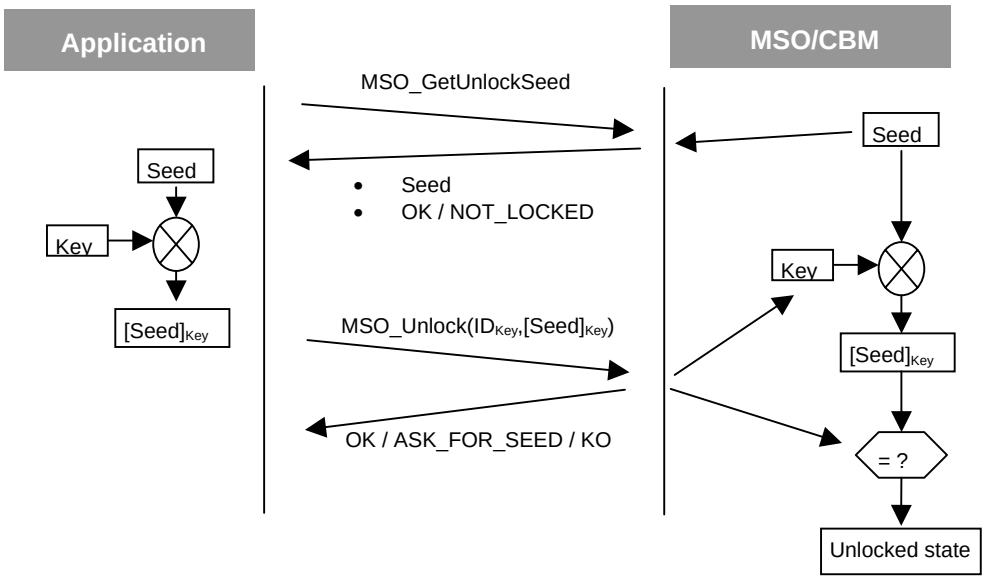


Figure 17: MSO / CBM unlocking process

6.8.1 ILV_GET_UNLOCK_SEED ID = 0x8B

This function requests the terminal to start the unlock process, and to return the value of the seed to be used for the next step of the unlock process.

6.8.1.1 REQUEST

Identifier value	0x8B
Length value	0x0000 Bytes
Value(Parameters)	None

6.8.1.2 ERROR REPLY

Identifier value	0x8B	
Length value	0x0005	
Value(Parameters)	Request Status	1 byte
	Internal Error	4 Bytes

Request Status:

ILVERR_BADPARAMETER	Format is not valid.
ILVERR_ERROR	An internal error occurred.
ILVERR_DEVICE_NOT_LOCK	The device is not in a locked state.

6.8.1.3 REPLY

Identifier value	0x8B	
Length value	0x0009	
Value(Parameters)	Request Status	1 byte
	Seed	8 Bytes

Seed : a random hexadecimal value to be returned encrypted by the host using ILV_Unlock command.

Request Status:

ILV_OK	The function succeeded.

6.8.2 ILV_UNLOCK ID = 0x8C

This function asks to unlock the MSO.

6.8.2.1 REQUEST

Identifier value	0x8C	
Length value	0x0010 Bytes	
Value(Parameters)	Secret ID	8 Bytes
	Seed ciphered by the secret	8 Bytes

Secret ID: a hexadecimal value which specifies the secret key used by the host to encrypt the seed. The secret keys are loaded in the terminal by factory plant, which provides the corresponding secret ID.
The secret key is thus not sent by the host to the terminal: only the identification of the key is sent in order to allow the terminal to find the corresponding key.

Seed: a random hexadecimal value included in the reply to the ILV_Get_Unlock_Seed command.

6.8.2.2 ERROR REPLY

Identifier value	0x0B	
Length value	0x0005	
Value(Parameters)	Request Status	1 byte
	Internal Error	4 Bytes

Request Status:

ILVERR_BADPARAMETER	Format is not valid.
ILVERR_ERROR	An internal error occurred.

6.8.2.3 REPLY

Identifier value	0x8B	
Length value	0x0009	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The function succeeded.

6.9 COMMUNICATION PARAMETER NEGOTIATION

After power-up, MorphoSmart ™ always starts communication with default parameters:

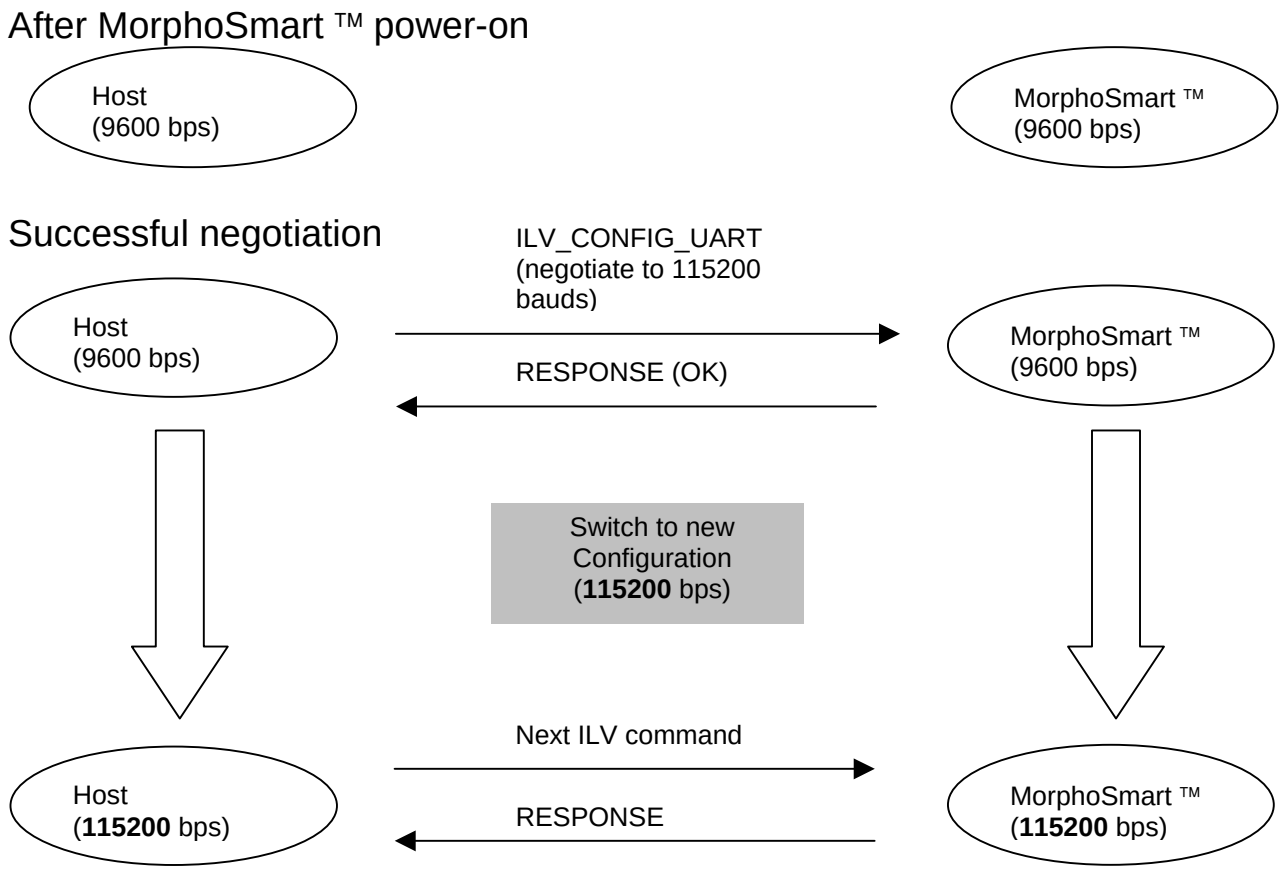
Baudrate:	9600 bps
Data bits:	8 bits
Parity:	None
Stop bits:	1 bit
Flow Control:	Software

Negotiated communication parameters are discarded with next power-up. This feature allows a MorphoSmart™ with unknown parameters (for example, if the host did not save the configuration).

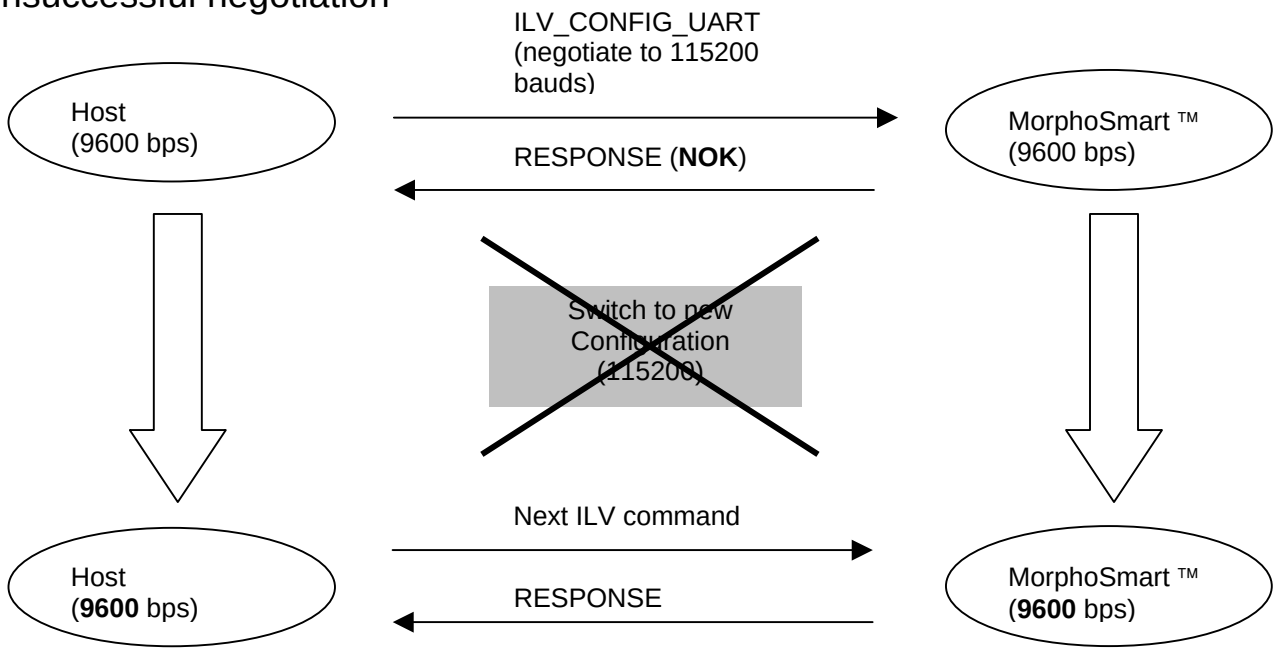
Default parameters can always be retrieved after a BREAK (all the bits are logic 0, see definition in “1.1 Acronyms and abbreviation”).

6.9.1 Scheme

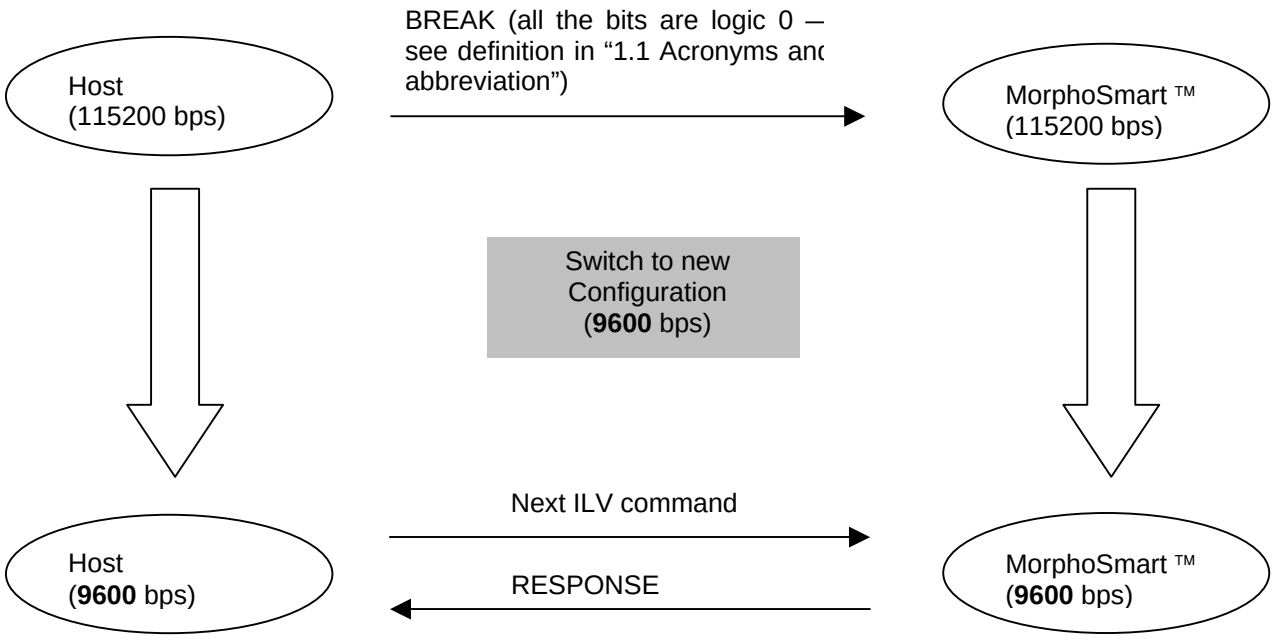
Here are the different scenarios that may happen during communication parameter negotiation.



Unsuccessful negotiation



Return to default communication parameters



6.9.2 CONFIG_UART ID = 0xEE

This function configures RS232 communication parameters.

Note: To avoid synchronization problem (host does not know MorphoSmart ™ parameters because of a crash for example) we recommend that the host perform a BREAK and return to default communication configuration before issuing this command.

6.9.2.1 REQUEST

Identifier value	0xEE	
Length value	0x0D	
Value(Parameters)	Communication parameter	0x0D byte

Communication parameter: An ILV formatted data packet that contains RS-232 communication parameters.

Identifier value	ID_COM1	
Length value	0x0000A	
Value	Bps	4 bytes
	Byte size	1 byte
	Number of stop bits	1 byte
	Parity	1 byte
	Flow control	1 byte
	RFU	2 bytes

Warning: we recommend you use only tested parameters. Other values are not guaranteed.

Bps: Bit Per Second rate (little Endian). Minimum bps is 1200. Maximum bps is 115200. The step is 100.

Example: *11100 bps is possible.*
11150 bps is forbidden.

Tested values are 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 76800, 115200 bps.

Example: *9600 bps is 0x80 0x25 0x00 0x00.*
115200 bps is 0x00 0xC2 0x01 0x00.

Flow control:

0	No flow control (not tested).
2	Software control (XON/XOFF).

Byte size:

8	8-bit data.

Parity:

0	No parity.
1	Odd parity (not tested).
2	Even parity (not tested).

Number of stop bits:

1	1 stop bit.
2	2 stop bit (not tested).

RFU: Not used, set to 0.

6.9.2.2 REPLY

Identifier value	0xEE	
Length value	0x01	
Value(Parameters)	Request Status	1 byte

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred during the execution of the function.

ILVERR_CMD_INPROGRESS	Command received while another command is running

6.10 ASYNCHRONOUS MESSAGES

In order to build friendly user interfaces, the MorphoSmart™ manages asynchronous messages that indicate current status of a current live acquisition.

Asynchronous messages are managed for all live-finger acquisition functions: Enroll, Verify, and Identify. Reception of those messages is fully customizable. Asynchronous information statuses are:

- user directions (press harder, move left, remove finger…)
- the finger number and acquisition number during the enrollment process
- a low-resolution image of the finger is received

Live finger acquisition ends when one of the following occurs:

- timeout expiration (timeout could be infinite if required),
- a finger is detected,
- Cancel command is executed.

6.10.1 Example of Cancel command

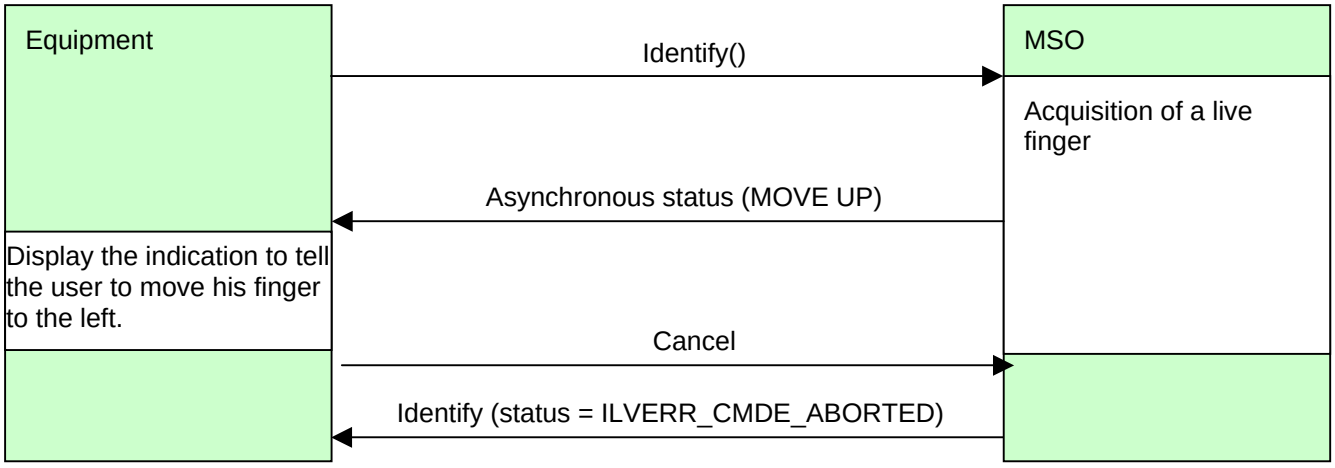


Figure 18: Live acquisition management

6.10.2 CANCEL ID = 0x70

This function cancels any biometric function with a live finger acquisition (it means: ENROLL, IDENTIFY, VERIFY and UDPATE_PRIVATE_FIELD). After reception of this command, MorphoSmart™ stops the function and sends a reply to the biometric function, with the ILVERR_CMDE_ABORTED status.

6.10.2.1 REQUEST

Identifier value	0x70
Length value	0x0000
Value(Parameters)	None

6.10.2.2 REPLY

None: there is no reply for this command (the reply is the one from the live acquisition: verify, identify, enroll or update private field.

7 ILV Data Description

7.1 TABLE OF MORPHOSMART ™ ILV DATA

Biometric Data		
0x02	*ID_PK_COMP*	Identifies a PK_COMP ILV (minutiae)
0x03	*ID_PK_MAT*	Identifies a PK_MAT little endian ILV (minutiae).
0x37	*ID_PK_COMP_NORM*	Identifies a PK_COMP_NORM ILV (minutiae).
0x35	*ID_PK_MAT_NORM*	Identifies a PK_MAT_NORM ILV (minutiae).
0x55	*ID_X984_DATA*	The X984 biometric token
0x54	*ID_X984_PARAM*	The X984 biometric token parameter
0x57	*ID_TKB*	The biometric token
0x53	*ID_TKB_SETTINGS*	Choose and set the biometric token envelop
0x38	*ID_BIO_ALGO_PARAM*	Used to define specific biometric parameters
Database Data		
0x36	*ID_USER_INDEX*	Identifies a user index ILV.
0x04	*ID_USER_ID*	Identifies a user data ILV
0x3A	*ID_PKBASE*	Identifies a database record template.
0x32	*ID_FIELD_CONTENT*	Identifies user public field content
0x0F	*ID_PUBLIC_FIELD*	Identifies the name of a public field in database
0x31	*ID_PRIVATE_FIELD*	Identifies the name of a private field in database
0x14	*ID_PUC_DATA*	Identifies additional user data in database
Asynchronous message data		
0x34	*ID_ASYNCHRONOUS_EVENT*	Reception of asynchronous events during live finger acquisition.
0x71	*ID_ASYNCHRONOUS_MESSAGE*	Signals an asynchronous message
0x01	*MESSAGE_COMMAND_CMD*	Signals an action that the user has to perform
0x02	*MESSAGE_IMAGE_CMD*	Image for graphical user interface
0x04	*MESSAGE_ENROLLMENT_CMD*	Signals enrollment status
0x08	*MESSAGE_IMAGE_FULL_RES_CMD*	Acquired fingerprint image
0x40	*MESSAGE_CODE_QUALITY_CMD*	Quality computed by the image coder
0x80	*MESSAGE_DETECT_QUALITY_CMD*	Quality computed by finger detection process
Image data		
0x3D	*Export Image*	Used to define export image parameters or that contains the image
Security data		
0x50	*X509 certificate*	X509 certificate
0x52	*Cryptogram*	Cryptogram used for mutual authentication
OTP data		
0x64	*ID_OTP_SEQUENCE_NUMBER_MAX*	Identifies the initial value of the sequence number (N)
0x65	*ID_OTP_PASSWORD*	Identifies the password
0x66	*ID_OTP_PIN*	Identifies the re-enrollment code pin
0x67	*ID_OTP_PARAM*	Identifies the parameters which administer the behavior of the MorphoSmart™ product.
0x68	*ID_OTP_ALGO_HASH*	Identifies the hash algorithm
0x69	*ID_OTP_SEQUENCE_NUMBER*	Identifies the current value of the sequence number (N)
0x6A	*ID_OTP_SEED*	Identifies the seed

0x71	*ID_OTP_USER_DATA*	The data attached to the user (max 1024 bytes)
Miscellaneous data		
0x56	*Matching Score*	Used to export matching score
0x39	*Latent detection*	Enable the fingerprint latent detection
0x42	*False Finger Detection*	Set the security level of the false finger detection
0x43	*Biometric coder selection*	Selects the biometric coder to use
0x63	*Matching PK Number*	Used to export the matching PK number
Coder Results data		
0x58	*Export Coder Results*	Used obtain coder results for a captured finger (matrix of directions, matrix of qualities, quality and frequency).
0x59	*ID_CODER_MATDIR*	Matrix of directions.
0x5A	*ID_CODER_MATQUAL*	Matrix of qualities.

Table 3: List of ILV data

7.2 BIOMETRIC DATA

7.2.1 PK_COMP V2 ID = 0x02

This is the MorphoSmart™ native template format, which is compatible with the other Sagem Sécurité biometric terminals and systems.

Sagem Sécurité recommends using this template format.

Identifier value	ID_PK_COMP	
Length value	L	
Value (Parameters)	PK_Comp template	L bytes (L is less or equal to 256)

PK_Comp template : biometric template which complies with PK_Comp V2 SAGEM private format.

Template size (L) : the maximum size of a PK_Comp template is 256 bytes.

The template output by the terminal have usually a size less than 256 bytes, depending directly on the number of the minutiae detected. However the terminal accepts a template padded to 256 bytes with any value (0x00, 0xFF,..).

Note : for storage purpose on an external device (such a smart card), a PK_Comp template can be limited to a size of 170 bytes without loss of biometric performances. It means that a template with a size greater than 170 bytes need to be compressed with the relevant fonction. The terminal is able to generate PK_Comp templates with a limited size (please refer to T*ENROLL*T command).

Note:

ID_PK_COMP_NORM (ID = 0x37) 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 Sécurité.

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

See also T*VERIFY*, *ENROLL*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*.

7.2.2 PK_MAT ID = 0x03

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.

Identifier value	ID_PK_MAT	
Length value	0x0200	
Value (Parameters)	Minutiae	512 bytes

Note:

ID_PK_MAT_NORM (ID = 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 Sécurité.

See also *VERIFY*, *ENROLL*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*.

7.2.3 PKV10 ID = 0x78

This is a new template format that is a combination of two existing template formats (V6 and V9). This type of template format is generated (and used) by applications based on MorphoKit™ product. This template format allow to have better performance (better FRR rate for instance).

The template size is equal to 512 bytes.

Identifier value	ID_PKV10	
Length value	0x0200	
Value (Parameters)	Minutiae	512 bytes

Note:

See also *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*.

7.2.4 ISO and ANSI template formats

7.2.4.1 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 template with only one fingerprint inside, but is able to manage ISO/ANSI templates with several fingerprints inside, generated by other devices.

The size a template is computed with the formula below :

Template Size = [FingerMinutiaeHeaderSize] + NbFingerprintViews x ([SingleFingerHeaderSize] + NbMinutiae x [MinutiaSize] + ExtendedDataSize)

Number of Minutiae per fingerprint :

- Usually the enrolment process initially detects 100 minutiaes (average value), and in a second step, invalid minutiaes are removed (about 60% of initially detected minutiaes). Then only 40 minutiaes (average value) are stored in the template.
- The minimum number of minutiaes, required for a normal matching process, is between 15 and 20.
- The maximum number of minutiaes for a fingerprint is not limited, but it can reach 70 or more.

7.2.4.2 ISO_PK ID = 0x3F

This ILV formatted data is 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).

Identifier value	ID_ISO_PK	
Length value	5 + L	
Value (Parameters)	ILV *ISO_PK_PARAM*	5 bytes
	ILV *ISO_PK_DATA_x*	L bytes

ISO_PK_PARAM : ILV formatted data which select the fingerprint to be used (refer to *ISO_PK_PARAM* ILV description)

ISO_PK_DATA_x : ILV formatted data which contains 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*, or *ISO_PK_DATA_ISO_FMC_CS*.

See also *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*.

7.2.4.3 ISO_PK_PARAM ID = 0x40

This ILV formatted data is used, inside an *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 included several fingerprints.

Identifier value	ID_ISO_PK_PARAM	
Length value	0x0002	
Value	Finger Selection	1 byte
	All templates	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. The value 0 selects first fingerprint, 1 selects 2nd fingerprint, and so on…. The value in this field is ignored when the “All template” field is set to 0x01.

All template: 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.

7.2.4.4 ISO_PK_DATA_ANSI 378 ID = 0x41

Identifier value	ID_ISO_PK_DATA_ANSI_378	
Length value	L	
Value (Parameters)	ANSI 378 template	L bytes

ANSI 378 template : biometric template which 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.

Template size (L) :

- FingerMinutiaeHeaderSize : 32 bytes
- NbFingerprintViews : 255 maximum
- SingleFingerHeaderSize : 4 bytes
- MinutiaSize : 6 bytes
- ExtendedDataSize : 65535 bytes maximum

See also *ENROLL*, *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*

7.2.4.5 ISO_PK_DATA_MINEX_A ID = 0x6F

Identifier value	ID_ISO_PK_DATA_MINEX_A	
Length value	L	
Value (Parameters)	MINEX A template	L bytes

MINEX A template : biometric template which 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.

Template size :

- FingerMinutiaeHeaderSize : 32 bytes
- NbFingerprintViews : 1
- SingleFingerHeaderSize : 4 bytes
- MinutiaSize : 6 bytes
- ExtendedDataSize : 65535 bytes maximum

See also *ENROLL*, *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*

7.2.4.6 ISO_PK_DATA_ISO_FMR ID = 0x6E

Identifier value	ID_ISO_PK_DATA_ISO_FMR	
Length value	L	
Value (Parameters)	ISO FMR template	L bytes

ISO FMR template : biometric template which 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.

Template size (L) :

- FingerMinutiaeHeaderSize : 24 bytes
- NbFingerprintViews : 255 maximum
- SingleFingerHeaderSize : 4 bytes
- MinutiaSize : 6 bytes
- ExtendedDataSize : 65535 bytes maximum

See also *ENROLL*, *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*

7.2.4.7 ISO_PK_DATA_ISO_FMC_NS ID = 0x6D

Identifier value	ID_ISO_PK_DATA_FMC_NS	
Length value	L	
Value (Parameters)	ISO FMC NS template	L bytes

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

Template Size (L) :

- FingerMinutiaeHeaderSize : none
- NbFingerprintViews : 1
- SingleFingerHeaderSize : 0
- MinutiaSize : 5 bytes
- ExtendedDataSize : none (0)

See also *ENROLL*, *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*

7.2.4.8 ISO_PK_DATA_ISO_FMC_CS ID = 0x6C

Identifier value	ID_ISO_PK_DATA_FMC_CS	
Length value	L	
Value (Parameters)	ISO FMC CS template	L bytes

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

Template Size (L) :

- FingerMinutiaeHeaderSize : none
- NbFingerprintViews : 1
- SingleFingerHeaderSize : 0
- MinutiaSize : 3 bytes
- ExtendedDataSize : none (0)

See also *ENROLL*, *VERIFY*, *VERIFY MATCH*, *IDENTIFY MATCH*, *ADD BASE RECORD*

7.2.5 Biometric Token

This is a secured biometric template encapsulated within a X9.84 format: a secure MorphoSmart™ can export such a template or verify its integrity by verifying its signature.
The X9.84 biometric token is only supported by the MorphoSmart™ S.

7.2.5.1 X9.84 DATA ID = 0x55

ILV formatted data that contains the X984 token.
The X9.84 format is ASN.1 DER encoded. The syntax is described in the MorphoSmartOverview document.
A X9.84 data can contain up to ten PK_COMP V2, or PK_MAT or normalized templates.

Identifier value	ID_X984_DATA	
Length value	$L_{x9.84}$	
Value	X984 biometric token	$L_{x9.84}$ bytes

7.2.5.2 X9.84 PARAM ID = 0x54

ILV formatted data that contains the X984 template index used for verification.

Identifier value	ID_X984_PARAM	
Length value	1	
Value	X984 template index	1 byte

X984 template index: This parameter is used to select the templates contained in the X984 structure. Set 0 to use all templates, set 1 to use the first template only, 9 to use the last template only.

7.2.5.3 TKB ID = 0x57

ILV formatted data that contains the X984 token and the X984 template index used for verification.

Identifier value	ID_TKB	
Length value	$L_1 + L_2$	
Value (Parameters)	ILV *X9.84 param*	L_1 bytes
	ILV *X9.84 data*	L_2 bytes

See also *VERIFY*, *VERIFYMATCH*,

7.2.5.4 TKB SETTINGS ID= 0x53

ILV formatted data that defines the type of biometric template envelop and the data value to be included in the token. It is used with ILV ENROLL to generate and export the biometric in the specified envelope.

Identifier value	ID_TKB_SETTINGS	
Length value	<0x0005 + N>	
Value	Envelope type	1 byte
	Size of application data	4 bytes
	Application data	N bytes

Envelope type:
MORPHO_X984_SIGNED_TEMPLATE: the X9.84 envelope is constructed with the template, the application data, and signature.

Size of application data: This value is set to N.

Application data: The application data that can be inserted in the biometric token. This parameter is used only by your application.

See also: *ENROLL*

7.2.6 Biometric Algorithm Parameter ID = 0x38

When used in *CREATE DATABASE* ILV command, this optional ILV formatted data that specifies the format of the biometric templates to be stored in the database either regular template or normalized templates. When this ILV is omitted, the default value applies : only regular template format are allowed to be stored in the database. Regular template formats means PK_Comp, PK_MAT, and ISO or ANSI formatted templates.

Identifier value	ID_BIO_ALGO_PARAM		
Length value	0x0001		
Value	0:	regular template format	1 byte
	1:	normalized template format	

Except if it is mandatory to be able to stored in the database, only normalized templates (such as PK_Comp_Norm and PK_MAT_Norm), the parameter value must be set to default.
Normalization is reserved for compatibility with existing systems or specific usage. For more information please contact Sagem Sécurité.

See also: *CREATEBASE*, *GETBASECONFIG*

When used in *ENROLL* command this optional ILV formatted data that specifies the format of the biometric templates to be output : default template format is PK_COMP V2.

Identifier value	ID_BIO_ALGO_PARAM		
Length value	0x0001		
Value	0:	PK_COMP templates	1 byte
	1:	PK_COMP_NORM templates	
	2:	PK_MAT templates	
	3:	PK_MAT_NORM templates	
	65:	ANSI 378 templates	
	108:	ISO FMC CS templates	
	109:	ISO FMC NS templates	
	110:	ISO FMR templates	
	111:	MINEX A templates	

See also: *ENROLL*,

7.3 DATABASE DATA

7.3.1 User Index ID = 0x36

ILV formatted data that contains the index of the record in the specified internal database.

Identifier value	ID_USER_INDEX	
Length value	0x0004	
Value	User Database Index	4 bytes

See also: *REMOVERECORD*, *GETDATA*, *UPDATEPUBLICDATA*, *UPDATEPRIVATEDATA*

User database index: index of the record in the database : 0 means 1st entry of the data base, 1 means 2nd entry, and so on. Its' value is returned by the ***ENROLL*** command, and by the ***ADD_BASE_RECORD*** commands at record generation.

7.3.2 User ID ID = 0x04

ILV formatted data that contains the unique identifier of the user in the specified internal database. (The terminal insure that each user id is unique in the database).

Identifier value	ID_USER_ID	
Length value	L	
Value	User ID	L bytes

See also: *ENROLL*, *IDENTIFY*, *IDENTIFYMATCH*, *ADDRECORD*, *REMOVERECORD*, *GETDATA*, *UPDATEPUBLICDATA*, *UPDATEPRIVATEDATA*

User ID : unique identifier of a user (a record). This field is managed as a byte array. If the host requires the User Id to be a character string, do not forget to manage the ending '\0'. The maximum size (L) of the User ID is 24 bytes.

The content of the 'User ID' field is returned by the ***Identify*** function when a Hit (match) occurs.

At record generation (either by the *ENROLL* command, or by the *ADD_RECORD* commands), the User Id field content is automatically generated by the terminal when the size of the field I(L) is equal to zero. In that case, the user ID generated is an ASCII character string (without ending NULL character), equal to the value of the index of the record in the database : '0' (0x30) for index 0, '10' (0x31 0x30) for index 10, and so on.

7.3.3 PKBase ID = 0x3A

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

Identifier value	ID_PKBASE	
Length value	0x0001 + L_{UID}	
Value (Parameters)	Database identifier	1 byte
	ILV *User ID* or *User Index*	L_{UID}

Database identifier: Current release of MorphoSmart™ does not support management of multiple databases. Set this parameter to 0.

See also *VERIFY*,

7.3.4 Additional User Data Field

7.3.4.1 FIELD CONTENT ID = 0x32

ILV formatted data packet that contains field content.

Identifier value	ID_FIELD_CONTENT	
Length value	0x00004+ 0x00004 + <L>	
Value	Field Index	4 bytes
	Data length	4 bytes
	Data	L bytes

Field Index: index of the Additional User Data Field. 1 for 1st field, 2 for 2nd field, and so on.

Data length: Field data size (= L).

Data: Value of the field data.

See also: *GET_PUBLIC_FIELD*, *UPDATE_PUBLIC_DATA*, *UPDATE_PRIVATE_DATA*

7.3.4.2 PUBLIC FIELD ID = 0x0F

ILV formatted data packet that the public field structure definition

Identifier value	ID_PUBLIC_FIELD	
Length value	0x0008	
Value	Field size	2 bytes
	Field name	6 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.

The maximum number of additional fields available depends on MorphoSmart™ type and configuration (see "**DataBase size limits**" section below)

See also: *CREATEBASE*, *GETBASECONFIG*

7.3.4.3 PRIVATE FIELD ID = 0x31

ILV formatted data packet that the private field structure definition

Identifier value	ID_PRIVATE_FIELD	
Length value	0x0008	
Value	Field size	2 bytes
	Field name	6 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.

The maximum number of additional fields available depends on MorphoSmart™ type and configuration (see "**DataBase size limits**" section below)

See also: *CREATEBASE*, *GETBASECONFIG*

7.3.4.4 ADDITIONAL USER DATA ID = 0X14

One ILV formatted data, containing personal user data included in the record saved into the database (i.e. First Name or Last Name of the User).

Identifier value	ID_PUC_DATA	
Length value	<L>	
Value	User Data	L bytes

User Data : Buffer containing the data. The format of the User Data, which is defined by the calling application, is not checked by the terminal. The size of the User Data must be less or equal to the size defined at database creation (by ***CREATE_BASE*** command).

The User Data can be empty, then the length of the ILV is equal to 0 (L = 0x00). This is the only way, for the host, to clear the content of a Additional User Data Field (using either the *UPDATE_PUBLIC_DATA*, or the *UPDATE_PRIVATE_DATA* command).

See also: *CREATE_BASE*, *GET_BASE_CONFIG* *ENROLL*, *IDENTIFY*, *ADD_RECORD*, *GET_DATA*, *UPDATE_PUBLIC_DATA*, *UPDATE_PRIVATE_DATA*.

7.3.4.5 DATABASE SIZE LIMITS

The table below indicates the limits for the size of the Database, which can be created with the ***CREATE_BASE*** command, according to the type of MorphoSmart™ device flash size and if an IdentLite or IdentPlus license is installed or not. The size of a database is defined with :

- The maximum number of records (users) which can be stored in the Database
- The maximum number of fingerprints per record
- The maximum number and the maximum size of additional user's fields per record

		Number of users specified at the creation of the Database					
Terminal flash size	**Licence**	**1 to 100**	**101 to 250**	**251 to 500**	**501 to 2000**	**2001 to 3000**	**3000 to 5200**
1Mbyte	None	16 fields of 128 bytes each (total size = 2048 bytes)	2 fields (max total size = 64 bytes)		Not Available		
2Mbytes	None	20 fields of 128 bytes each (total size = 2560 bytes)		2 fields (total size = 128 bytes)			
	IdentLite	20 fields of 128 bytes each (total size = 2560 bytes)		2 fields (total size = 128 bytes)			
4Mbytes	None	20 fields of 128 bytes each (total size = 2560 bytes)		2 fields (total size = 128 bytes)			
	IdentLite	20 fields of 128 bytes each (total size = 2560 bytes)		2 fields (total size = 128 bytes)	2 fields (total size = 128 bytes)	2 fields (total size = 128 bytes)	
	IdentPlus	20 fields of 128 bytes each (total size = 2560 bytes)		2 fields (total size = 128 bytes)	2 fields (total size = 128 bytes)	2 fields (total size = 128 bytes)	2 fields (total size = 128 bytes)

Table 4 : DataBase Size Limits for records with one or two fingers

Note : to know the maximum number of records you can store into your MorphoSmart™ use the *GetDescriptor* function.

In addition to regular DataBase for records with one or two fingerprints, the MorphoSmart™ supports two formats of DataBase for a higher number of fingerprints per record.

- When the maximum number of fingerprints per record is set to 10, then the maximum number of records of the database must be set to 100.
- When the maximum number of fingerprints per record is set to 20, then the maximum number of records of the database must be set to 50.
- In these two previous combinations (finger number per person / person number in database : 10/100 and 20/50), it is not possible to create more than 2 additional user data fields of a maximum 32 bytes long each, whatever the terminal type and configuration.

Number of fingerprints per record	Number of records (mandatory)	Additional User Data fields
10	100	A maximum of 2 fields of 32 bytes maximum each (total size = 64 bytes)
20	50	

Table 5 : DataBase Size limits for records with 10 or 20 fingers

These two formats are supported by all configuration of MorphoSmart™ with or without licence.

Warning : at DataBase creation (with Create_DataBase command), the value of the number of records and the number of fingerprints per record is checked, and if it doesn't match with the table above, the DataBase creation is rejected (i.e. a DataBase of 99 records with 10 fingerprints each is not allowed).

7.3.5 No_Check_On_Template ID = 0x60

Optional ILV formatted data, used with the *ADD_BASE_RECORD* command, to disable checks on reference templates. In order to insure that each fingerprint is unique in the database, before storing a new record the terminal checks that :

- all the fingerprints provided in the command must be different,
- and none of the provided fingerprints is already stored in the database.

These checks are performed with biometric matching functions.

It is not recommended to disable reference templates check for single *ADD_BASE_RECORD* commands.

But when a empty large base as to filled with a lot of *ADD_BASE_RECORD* command, disabling reference template checks will reduce the global duration of the process. Warning : in that case the database coherence must be checked by the host before sending the Add_Record commands.

Identifier value	ID_NO_CHECK_ON_TEMPLATE	
Length value	0x0001	
Value	Value	1 byte

Value:

0x00 : Normal process : the terminal checks the reference templates

0x01: the reference templates check is disabled.

Other values are rejected with a "bad parameter" error.

See also: *ADD_BASE_RECORD* command.

7.4 ASYNCHRONOUS MESSAGES

7.4.1 Asynchronous Event ID = 0x34

This optional parameter (ILV format) is use to request the MorphoSmart™ to sent asynchronous messages during biometric process, as described in the *Asynchronous messages* section.

Identifier value	ID_ASYNCHRONOUS_EVENT	
Length value	0x0004	
Value (Parameters)	Binary mask (little endian format)	4 bytes

Binary Mask: The binary mask contains a bit for each kind of asynchronous message required. The allowed flags are listed below, all other bits of the binary mask must be set to 0.

0x01 **Finger position messages** : no finger detected, move left, move right, move up, move down, remove finger, press harder, fingerprint acquired, …

Please refer to *ASYNC_MESSAGE (COMMAND)*

0x02 **Live low-resolution images, and acquired fingerprint image (normal resolution)**

Please refer to *ASYNC_MESSAGE (IMAGE)*

0x04 **Enrolment step messages** (*ENROLL* command only)

Please refer to *ASYNC_MESSAGE (ENROLMENT)*

0x08 **Image of the acquired fingerprint only**.

When this flag is set up the, regular images (flag 0x02) messages are automatically disabled.

Please refer to *ASYNC_MESSAGE (IMAGE_FULL_RES).*

0x40 **Quality of the acquired fingerprint.**

Please refer to *ASYNC_MESSAGE (CODE_QUALITY)*

0x80 **Live quality of the detected fingerprint**.

Please refer to *ASYNC_MESSAGE (DETECT_QUALITY)*.

See also: *VERIFY*, *ENROLL*, *IDENTIFY*, *UPDATEPRIVATEDATA*

Example : To require the MorphoSmart™ to send the asynchronous message *ASYNC_MESSAGE (COMMAND)* and *ASYNC_MESSAGE (IMAGE)*, in an enroll command, the ILV is (hexadecimal values):

```
21 0F 00 00 14 00 00 00 01 00 00 34 04 00 03 00 00 00
```

7.4.2 Asynchronous Message ID = 0x71

Each asynchronous messages sent by the MorphoSmart™ is ILV formatted, and has the same Identifier value.

Identifier value	ID_ASYNCHRONOUS_MESSAGE	
Length value	1 + <L1> + … + <Ln>	
Value(Parameters)	Request Status	1 byte
	ILV Asynchronous message 1	L1 bytes
	ILV Asynchronous message 2	L2 bytes
	…	…
	ILV Asynchronous message n	Ln bytes

Request Status:

ILV_OK	The execution of the function succeeded
ILVERR_ERROR	An error occurred.

ILV Asynchronous message: One or more formatted ILV data packets that correspond to asynchronous messages (current version is limited to one asynchronous message). The possible types of asynchronous messages are:

- *MESSAGE_COMMAND_CMD*
- *MESSAGE_IMAGE_CMD*
- *MESSAGE_ENROLLMENT_CMD*
- *MESSAGE IMAGE_FULL_RES_CMD*
- *MESSAGE_CODE_QUALITY_CMD*
- *MESSAGE_DETECT_QUALITY_CMD*

Example : The following asynchronous messages is sent by the MorphoSmart to signal *MESSAGE_COMMAND_CMD* with the MORPHO_PRESS_FINGER_HARDER action (hexadecimal values):

71 08 00 00 01 04 00 05 00 00 00

7.4.2.1 MESSAGE_COMMAND_CMD ID=0x01

This type of asynchronous message sent by the MorphoSmart™, is used to :

- indicate that there is a wrongly placed finger on the sensor, and then the action that the user has to perform
- provide information about the state of the fingerprint process (fingerprint wait started, fingerprint acquisition completed).

This ILV data is encapsulated in the 'Value' field of an asynchronous ILV.

Identifier value	MESSAGE_COMMAND_CMD	
Length value	0x0004	
Value	Command	4 bytes

Command:

0	MORPHO_MOVE_NO_FINGER	No finger is detected. This message is send once when the sensor is activated, and no finger is detected. This message is also send when no finger is detected after a wrongly placed finger (it means that the finger has been removed).
1	MORPHO_MOVE_FINGER_UP	User must move his finger up
2	MORPHO_MOVE_FINGER_DOWN	User must move his finger down
3	MORPHO_MOVE_FINGER_LEFT	User must move his finger to the left.
4	MORPHO_MOVE_FINGER_RIGHT	User must move his finger to the right.
5	MORPHO_PRESS_FINGER_HARDER	User must press his finger harder. It means also that the fingerprint size is too small.
6	MORPHO_LATENT	Possible latent fingerprint. The fingerprint detected on the sensor is at the same place as a previous fingerprint acquisition. The MORPHO_LATENT message is always followed by a MORPHO_REMOVE_FINGER message.
7	MORPHO_REMOVE_FINGER	User must remove his finger. This message is sent after a MORPHO_LATENT message, or after each fingerprint acquisition (except the last one) of a enrolment process.
8	MORPHO_FINGER_OK	The fingerprint acquisition is completed with success. For Identify and Verify command, it means also that the matching process starts.

7.4.2.2 MESSAGE_IMAGE_CMD ID=0x02

This type of asynchronous message, sent by the MorphoSmart™, is used to transmit images from the sensor, for the host GUI.

Two different images are send by the terminal, with the same identifier value, but with a different size (specified in the image header) :

- live image : it is an image captured by the sensor whatever there is a fingerprint or not. The terminal sent live images until the end of the fingerprint acquisition process, with a reduced size to provide a convenient image throughput.
- end image : last captured image. When the quality of the fingerprint is enough, the fingerprint acquisition stops, and the last captured image is send without size reduction (except on RS232 terminal).

This ILV data is encapsulated in the 'Value' field of an asynchronous ILV.

Identifier value	MESSAGE_IMAGE_CMD	
Length value	0x000C + <L>	
Value	Image header	0x0C bytes
	Raw image	L bytes

***Image Header*:** Specify the image format.

Raw image: Image data.

7.4.2.3 MESSAGE_ENROLLMENT_CMD ID=0x04

This type of asynchronous message sent by the MorphoSmart™, is used to indicate the enrollment process step.

For example, during the enrolment of one finger with three images, the terminal send three messages (enrolment of finger 1 of 1, acquisition 1, 2 and 3).

This ILV data is encapsulated in the 'Value' field of an asynchronous ILV.

Identifier value	MESSAGE_ENROLLMENT_CMD	
Length value	0x0004	
Value	Finger number	1 byte
	Finger Total	1 byte
	Capture number	1 byte
	Capture Total	1 byte

Finger number: Current number of enrolled fingers (start from 1).

Finger Total: Total number of fingers to enroll.

Capture number: Current number of acquisitions of the current enrolled finger (start from 1).

Capture Total: Total number of acquisitions one finger.

7.4.2.4 MESSAGE_IMAGE_FULL_RES_CMD ID=0x08

This type of asynchronous message, sent by the MorphoSmart™, is used to transmit only one image : the image of the acquired fingerprint.

The size of this image is not reduced, then the transmit time can be long on RS232 terminals.

When this kind of asynchronous message is requested in the biometric command, the regular images (see *MESSAGE_IMAGE_CMD*), if also requested are automatically disabled.

This ILV data is encapsulated in the 'Value' field of an asynchronous ILV.

Identifier value	MESSAGE_IMAGE_FULL_RES_CMD	
Length value	0x000C + <L>	
Value	Image header	0x0C bytes
	Raw image	L bytes

***Image Header*:** Specify the image format.

Raw image: Image data.

7.4.2.5 MESSAGE_CODE_QUALITY_CMD ID=0x40

This type of asynchronous message, sent by the MorphoSmart™, is used to provide the quality value of the acquired fingerprint image. This quality value is provided by the biometric coder of the terminal.

Identifier value	MESSAGE_CODE_QUALITY_CMD	
Length value	0x0001	
Value	Code quality (Value between 0 and 255)	1 byte

Code quality: quality note score of the image detained to be coded. The value of 255 corresponds to the best quality note score. The code quality depends a lot on the type of finger. It is admitted that a correct finger starts at 40, and a really good finger is over 120.

This ILV is sent when the quality score of the finger's picture is greater than the minimum quality score required for storage or comparison.

7.4.2.6 MESSAGE_DETECT_QUALITY_CMD ID=0x80

This type of asynchronous message, sent by the MorphoSmart™, is used to provide the quality note of the live fingerprint image, which is calculated by the "presence detection" function.

This ILV data is encapsulated in the 'Value' field of an asynchronous ILV.

Identifier value	MESSAGE_DETECT_QUALITY_CMD	
Length value	0x0001	
Value	Detect quality (Value between 0 and 255)	1 byte

Detect quality: quality note score calculated by the "presence detection" function. The value of 255 corresponds to the best quality note score. The detect quality depends a lot on the type of finger. It is admitted that a correct finger starts at 40, and a really good finger is over 120.

To avoid a system overload, this ILV is sent only if the communication is free, as the messages relative to the finger actions (finger move left, remove finger…) that the user has to perform. The Host system receives this ILV by means of the callbacks system of the asynchronous ILVs (encapsulated in the Asynchronous Message ILV 0x71).

7.4.3 Alive Asynchronous Message ID = 0x99

This optional ILV command is useful when the host system has to wait indefinitely for a fingerprint on the sensor, but want to know if the terminal is still alive, without requesting live images from the terminal (please refer to *MESSAGE_IMAGE_CMD*).

To use this function, the host system must require the finger position messages (flag *ASYNC_MESSAGE (COMMAND)* in the binary mask of the optional ILV *ID_ASYNCHRONOUS_EVENT*.

When a biometric command, which requires a fingerprint acquisition, is send to the terminal, the first asynchronous message returned by the terminal is a **MORPHO_MOVE_NO_FINGER**. Then until a fingerprint is detected on sensor and then removed, this message is no more send to the host system. It mean that as long as there is no fingerprint, there is no message transmitted with the host system. Then the "alive" function enables the terminal to send periodically the **MORPHO_MOVE_NO_FINGER** message, when no fingerprint is detected.

When a badly placed fingerprint is detected, the terminal send one of relevant message only once, until the finger is moved. For example if the fingerprint is to close to the right side of the sensor, the terminal send only one **MORPHO_MOVE_FINGER_LEFT** message, and if the "alive" function is activated, the terminal will send periodically the **MORPHO_MOVE_FINGER_LEFT** until the finger is moved or removed.

To activate the function, the system specifies a "maximum idle time" allowed (in seconds), which is used by the terminal during fingerprint acquisition process. Then after the sending of each asynchronous message the terminal start time monitoring. If the measured time reach the maximum value specified, then the terminal will transmit again the last message sent (which is usually a **MORPHO_MOVE_NO_FINGER** message). As for "pertinent message" the terminal restarts the time measure process to be able to send again a "still alive" message.

Please note that the function is automatically disabled after a *ASYNC_MESSAGE (COMMAND* with **MORPHO_FINGER_OK**. This message indicates the successful end of the fingerprint acquisition process, and that the terminal starts the "matching" process. This means that the final answer to the biometric command will be received soon (usually less than a few seconds, depending of the size of the biometric database).

One typical use of this function is access control system which is usually waiting for a finger on the sensor to start the access control process itself (usually : search for the fingerprint in a database, and check of the access rights assigned to the fingerprint when found).

Identifier value	ID_ALIVE_MESSAGE_TIME	
Length value	0x0004	
Value (Parameters)	Maximum idle time (0 or 10 to 3600)	4 bytes

Maximum idle time: value, in second, of the duration during which the terminal is allowed to disable the sending of redundant finger position messages. Allowed value are :

- **0** when the function is not required (same effect as if the ID_ALIVE_MESSAGE_TIME optional ILV is not included in the biometric command).
- higher or equal to **10** s, and lower or equal to **3600** s. Please note that due to some random internal process, it is possible that the terminal send a message a few second after the required time. Then for a better operation, **it is highly recommended to choose a value for the Host System timeout, much larger than the value of the Maximum Idle Time**. For example, if the maximum idle time is set to 30 s, it is recommended that the Host System wait at least 20 second more before considering that the terminal is in an unknown state.

A biometric command including a ID_ALIVE_MESSAGE_TIME optional ILV will be rejected, with a ILVERR_BADPAMETER error code value, if :

- the **Maximum idle time** value is higher than 0 and the lower than the minimum value, or is higher than the maximum value
- or if the *ID_ASYNCHRONOUS_EVENT* optional ILV is missing
- or if the binary mask of the *ID_ASYNCHRONOUS_EVENT* optional ILV doesn't include the **ASYNC_MESSAGE (COMMAND)** flag or the **ASYNC_MESSAGE (IMAGE)** flag.

The ID_ALIVE_MESSAGE_TIME optional ILV is ignored when :

- the **Maximum idle time** value is equal to 0
- or when the *ASYNC_MESSAGE (IMAGE)* flag is set in the *ID_ASYNCHRONOUS_EVENT* optional ILV

See also: *ASYNC_MESSAGE (COMMAND)*, *VERIFY*, *ENROLL*, *IDENTIFY*, *UPDATEPRIVATEDATA*.

Example : To enable MorphoSmart™ alive asynchronous message in an enroll command, the ILV is (hexadecimal values):

```
21 16 00 00 14 00 00 00 01 00 00 34 04 00 01 00 00 00 99 04 00 0A 00 00 00
```

7.5 IMAGE

7.5.1 Export Image ID = 0x3D

ILV formatted data used within ILV ENROLL to require the terminal to return in the reply, the image that have been captured to extract the minutiae.

Identifier value	ID_EXPORT_IMAGE	
Length value	0x0006	
Value	Image type	1 byte
	Compression	5 bytes

Image type:

Defines the required type of image : only one value is allowed for this parameter.

ID_DEFAULT_IMAGE The resolution of the returned image is 500 x 500 dpi.

The size of the returned image is :

- 416 x 416 pixels for the MorphoSmart™
- 400 x 256 pixels for the MorphoSmart™ CBM.

Compression:

ILV formatted data that specifies the type of compression required.

Identifier value	ID_COMPRESSION	
Length value	0x0002	
Value	Compression type	1 byte
	Compression parameter	1 byte

Compression type :

This parameter defines the type of compression to apply on the captured image.

ID_COMPRESSION_NULL	The image is returned without any compression (which means an size of 173056 bytes for a MorphoSmart ™, and 102400 bytes for a MorphoSmart ™ CBM.
ID_COMPRESSION_V1	The image is returned compressed, using a fingerprint image compression algorithm (Sagem private format). The compression ratio depends of the image to be compress, but it is usually close to 4. To be displayed the compressed image must be uncompressed using the ImageCompress.dll library provided with MorphoSmart ™ SDK. This compression format is recommended when the terminal is connected by a RS232 port, and internal database management is required.
ID_COMPRESSION_WSQ	The image is returned compressed, using WSQ compression algorithm (Wavelets scalar quantization). This feature is available since 08.07 software revision only, and requires a specific firmware (MSO_WSQ) which disables internal database management. This compression format is recommended when the terminal is connected by a RS232 port, and internal database management is not required. Please note that WSQ compression format is also used by several police services, to store fingerprint images.

Compression parameter:

This parameter defines the value of compression parameter.

ID_COMPRESSION_NULL	The value of this parameter must be set to 0x00.
ID_COMPRESSION_V1	The value of this parameter must be set to 0x00.
ID_COMPRESSION_WSQ	Required compression ratio [2 to 255], recommended value is "15"

See also: *ENROLL*,

7.5.2 Image reply ID = 0x3D

ILV formatted data containing a MSO image (included in a reply to ENROLL ILV command when required by the ID_EXPORT_IMAGE optional ILV parameter).

Identifier value	ID_EXPORT_IMAGE	
Length value	0x000C + <L>	
Value	*Image Header*	0x0C bytes
	Image binary data	L bytes

7.5.2.1 IMAGE HEADER

Formatted data containing the image parameters information.

Header Revision	Version number. Current version number is 0	
Header Size	10. (0x0A)	1 byte
Header	Row number	2 bytes
	Column number	2 bytes
	Vertical resolution	2 bytes
	Horizontal resolution	2 bytes
	Compression	1 byte
	Compression Parameter	1 byte

Row number: number of row of the image (in pixels).

Column number: number of column of the image (in pixels).

Vertical resolution : number of dot per inch in vertical direction (usually 500 dpi)

Horizontal resolution X: number of dot per inch in vertical direction (usually 500 dpi)

Compression: type of compression performed on the fingerprint image

ID_COMPRESSION_NULL	The image is returned without any compression (which means an size of 173056 bytes for a MorphoSmart ™, and 102400 bytes for a MorphoSmart ™ CBM.
ID_COMPRESSION_V1	The image is returned compressed, using a fingerprint image compression algorithm (Sagem private format). The compression ratio depends of the image to be compress, but it is usually close to 4. To be displayed the compressed image must be uncompressed using the ImageCompress.dll library provided with MorphoSmart ™ SDK.
ID_COMPRESSION_WSQ	The image is returned compressed, using WSQ compression algorithm (Wavelets scalar quantization). This feature is available since 08.07 software revision only, and requires a specific firmware (MSO_WSQ) which disables internal database management.

Compression parameter: depends of the kind of compression

ID_COMPRESSION_NULL	The compression parameter value is the number of bit per pixel (for example 8 means that we have 8 bits per pixel).
ID_COMPRESSION_V1	The compression parameter value is required by the uncompress library ImageCompress.dll to be able to uncompress the image. Warning : images compressed with ID_COMPRESSION_V1 can only be uncompressed with ImageCompress.dll library.
ID_COMPRESSION_WSQ	Compression ratio [2 to 255], recommended value is "15". Same value as the one specified in the ID_EXPORT_IMAGE optional parameter of the ENROLL ILV command.

Image binary data : depends of the kind of compression

ID_COMPRESSION_NULL	L bytes : usually one byte per pixel (for example 416 x 416 means 173056 bytes). The size of the image depends on the sensor size and on the image type.
ID_COMPRESSION_V1	L bytes of binary data. The size of the binary data depends on the original size of the image (uncompressed image size) and on the fingerprint image. Usually the average compressed ratio is equal to 4 (it means that the compressed size is close to the original image size divided by 4).
ID_COMPRESSION_WSQ	8 bytes of WSQ image header + Lwsq bytes of WSQ format compressed image

WSQ Image Header: formatted data containing the WSQ image parameters information**.**

Header Revision	Version number. Current version number is 0	2 bytes
Header Size	0x08	2 bytes
Header	Compression ratio x 1000, reached for output image	4 bytes
	Lwsq: WSQ Image size in bytes	4 bytes

See also: *ENROLL*,

7.6 OTP DATA

7.6.1 OTP sequence number max ID = 0x64

ILV formatted data that contains the initial value of the OTP sequence number.

Identifier value	ID_OTP_SEQUENCE_NUMBER_MAX	
Length value	0x0002	
Value	Initial sequence value	2 up to 8 bytes

Initial sequence value:

For SHA1 algorithm: a binary value stored on 2 bytes, which is also the number of OTP that the terminal is allowed to generate. As the OTP generation time is proportional to the sequence number, it is recommended to specify a value less or equal to 10000, to avoid excessive duration.

For HOTP algorithm: a binary value stored on 8 bytes, which is the initial sequence value. The number of OTP that the terminal is allowed to generate is equal to (2^{64} -1) - Initial sequence value.

See also: *OTP SET PARAMETERS*, *OTP Sequence number*

7.6.2 OTP password ID = 0x65

ILV formatted data that contains the OTP password, which is the primary seed for OTP generation.
An OTP password must be written in the terminal to initialize the OTP generator.

Identifier value	ID_OTP_PASSWORD	
Length value	L	
Value	OTP password	L = 10 up to 63 bytes

OTP password : a binary value stored on several bytes.

See also: *OTP SET PARAMETERS*

7.6.3 OTP pin ID = 0x66

ILV formatted data that contains the pin code required to allow the enrolment of the OTP user.

When there is no OTP reference template record in the terminal (i.e. before the first OTP user enrollment), a pin code is needed, to allow the enrolment of the user.

Before 1st OTP user enrolment, the enrolment allowance pin code must be loaded in the terminal using the OTP SET PARAMETERS command. Then the OTP ENROLL USER command is allowed, if the same pin code value is specified in the command.

After a successful OTP user enrolment, a next allowance pin code is computed by the terminal (and by the host). It means that the enrolment allowance pin code is valid for only one successful user enrolment. When the OTP user must be replaced, the next valid enrolment allowance pin code must be provided.

When an OTP user is already enrolled, the current OTP user record can be replaced by an other, if allowed by OTP parameters (see *OTP SET PARAMETERS* command).

When the *OTP SET PARAMETERS* command is used to load a pin code in the terminal, all recorded OTP data is deleted, including the OTP current user's fingerprint templates. Then the enrollment of an OTP user is again allowed.

Identifier value	ID_OTP_PIN	
Length value	L	
Value	Enrolment allowance pin code	L = 5 bytes

Enrolment allowance Pin code: a binary value stored on 4 bytes, followed by a CRC value stored in one byte.

See also: *OTP SET PARAMETERS*, *OTP ENROLL USER*

7.6.4 OTP parameters ID = 0x67

ILV formatted data that contains the value of the parameters which define the behavior of the OTP function. The OTP param field value is loaded by the OTP SET PARAMETERS command, and its content is returned by the OTP GET STATUS command.

Identifier value	ID_OTP_PARAM	
Length value	0x0002	
Value (Parameters)	OTP Param field	2 bytes

OTP Param field : 16 bits mask composed with the flags listed below :

BIT0	0x01	When this bit is set, on site re-enrolment is allowed. It means that, when a OTP user is already recorded, the *OTP ENROLL USER* command allow replacing the current user. In that case, the enrolment allowance pin code is not included, and as the first step of the process, the current user is required to place an already recorded finger on the sensor, for successful identification. Then normal enrolment process starts (the new user is required to place 1st finger 3 times, and then the 2nd finger, 3 times). When this bit is cleared, the *OTP ENROLL USER* command must include a valid enrolment allowance pin code, to enable the replacement of the current user. In that case, the biometric identification of the current user is not performed.
BIT1 to BIT15	-	Reserved.

See also: *OTP SET PARAMETERS*, *OTP GET STATUS*

7.6.5 OTP algo hash ID = 0x68

ILV formatted data that specify the hash algorithm to be used, by the terminal, to generate the OTP value.

Identifier value	ID_OTP_ALGO_HASH	
Length value	0x0001	
Value (Parameters)	Hash function	1 byte

Hash function:

- 0x01, which means SHA-1 hash algorithm (default value).
- 0x02, which means HOTP6 hash algorithm.
- 0x03, which means HOTP7 hash algorithm.
- 0x04, which means HOTP8 hash algorithm.

See also: *OTP GENERATE*

7.6.6 OTP sequence number ID = 0x69

ILV formatted data that contains the OTP current sequence number, used to specify which OTP has to be returned :

- the next one : the terminal must generate the next OTP value
- the last one : the terminal must send again the last generated OTP value

Identifier value	ID_OTP_SEQUENCE_NUMBER	
Length value	0x0002	
Value (Parameters)	Sequence Number	2 up to 8 bytes

Sequence Number:

- For SHA1 hash algorithm: 2 bytes numeric value.

  If N is the current value of the internal OTP sequence number, N + 1 value requires to received again last generated OTP, 0 or N requires to received next OTP value (default function). The OTP sequence number, managed by the terminal, is initialized by *OTP SET PARAMETERS* command.

  After each successful OTP generation, the terminal decrements its internal sequence number. When the sequence number reaches zero, the terminal is not allowed to generate an OTP value anymore.

  In order to be able to send the valid N or N+1 value to the terminal, the host must keep track of the number of OTP generated by the terminal.

- For HOTP algorithm: : 8 bytes numeric value.

  If N is the current value of the internal OTP sequence number, N - 1 value requires to received again last generated OTP, 0 or N requires to received next OTP value (default function). The OTP sequence number, managed by the terminal, is initialized by *OTP SET PARAMETERS* command.

  After each successful OTP generation, the terminal increments its internal sequence number. When the sequence number reaches (2^{64} -1), the terminal is not allowed to generate an OTP value anymore.

  In order to be able to send the valid N or N - 1 value to the terminal, the host must keep track of the number of OTP generated by the terminal.

See also: *OTP GENERATE*, *OTP SET PARAMETERS*

7.6.7 OTP seed ID = 0x6A

ILV formatted data that contains the secondary seed used to generate the OTP. This can be used by servers wishing to increase the security level by diversifying the Hash input.

Identifier value	ID_OTP_SEED	
Length value	L	
Value (Parameters)	Seed	L = 1 up to 16 bytes

Seed : numeric binary value, to be provided by the host.

Seed is only needed for SHA1 hash algorithm.

See also: *OTP GENERATE*

7.6.8 OTP User Data ID = 0x71

ILV formatted data that contains the OTP User Data, defined by the host, saved in the terminal by the *OTP SET PARAMETERS* command, and returned to the host, by a successful *OTP GENERATE* command, and by a *OTP GET STATUS* command.

The content of the OTP User Data is not checked by the terminal; it is stored and returned to the host without any modification.

When the host sends a *OTP SET PARAMETERS* command with an empty OTP User Data ILV (L = 0, V not provided), then the OTP User Data is deleted in the terminal.

Identifier value	ID_OTP_USER_DATA	
Length value	L	
Value (Parameters)	User Data	L = 1 up to 1024 bytes

User data : binary data defined by host.

See also: *OTP SET PARAMETERS*, *OTP GET STATUS*, *OTP ENROLL USER*, *OTP GENERATE*

7.7 SECURITY

7.7.1 X509 certificate ID = 0x50

An ILV formatted data that contains the X509 certificate, encoded in DER format.

Identifier value	ID_X509_CERTIFICATE	
Length value	L	
Value	X509 certificate	L bytes

X509 certificate: Certificate encoded in DER format.

See also: *READCERTIFICATE*, *STORECERTIFICATE*, *MUTUALAUTHINIT1*

7.7.2 Cryptogram ID =0x52

An ILV formatted data that contains an encrypted token.

Identifier value	ID_CRYPTOGRAM_MUTUAL_AUTH	
Length value	L	
Value	Cryptogram	L bytes

Cryptogram: Encrypted buffer data.

See also: *MUTUALAUTHINIT1*, *MUTUALAUTHINIT2*

7.8 MISCELLANEOUS

7.8.1 Matching Score ID = 0x56

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

Identifier value	ID_MATCHING_SCORE	
Length value	0x0001 or 0x0004	
Value (Parameters)	Value or Score	1 or 4 bytes

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.

See also: *VERIFY*, *VERIFYMATCH*, *IDENTIFY*, *IDENTIFYMATCH*

7.8.2 Latent detection ID = 0x39

This ILV is optional. It can be used in the *ENROLL* request to enable the latent detection.

Identifier value	ID_LATENT_SETTING	
Length value	0x0001	
Value	Value	1 byte

Value: The possible value is:

0x00: the fingerprint latent detection is disabled

0x01: the fingerprint latent detection is enabled.

See also: *ENROLL*

7.8.3 False Finger Detection ID = 0x42

This ILV is optional. It can be used in the *VERIFY, ENROLL, IDENTIFY and UPDATE_PRIVATE_DATA* requests to set the false finger security level.

Identifier value	ID_FFD_LEVEL	
Length value	0x0004	
Value	Value	4 bytes

Value: The possible value is:

0x00: low false finger security level (default)

0x01: medium false finger security level

0x02: high false finger security level.

See also: *VERIFY*, *ENROLL*, *IDENTIFY*, *UPDATE_PRIVATE_DATA*

7.8.4 Biometric coder selection ID = 0x43

Optional ILV formatted parameter, used to configure the MorphoSmart™ integrated biometric coder. The biometric coder is the software component that convert a fingerprint image into a template. The best configuration depends on the kind of fingerprint to capture, and on the kind of fingerprints to be compared. This optional ILV is available since software release v06.03.

Identifier value	ID_CODER_CHOICE	
Length value	0x0004	
Value (Parameters)	Coder algorithm option	4 bytes

Coder algorithm option : 4 bytes in little endian

- 0x00000000 : default value. Parameter value is ignored. Standard function is automatically selected.
- 0x00000003 : standard function. Best choice for fingerprint with normal to large ridges.
- 0x00000007 : Juvenile function. Mandatory choice when it is required that adult and childhood fingerprints of the same person, match together (see description below).
- 0x00000008 : Thin finger function (available since software release 08.04). Best choice for fingerprint with small to normal ridges (see description below).

Common to Thin finger and Juvenile functions :

This functions provides a solution, with reliable performance, to the automatic processing of fingerprint with small to normal ridges, extending the capability of the terminal to a wider range of population. Please note that, when these options are turned on :

- the encoding time is much longer
- the biometric performances (FAR and FRR), are lower with fingerprints with normal to large ridges.

Juvenile function :

In addition to common behavior, each fingerprint is encoded only into a 'normalized' format such as PK_Comp_Norm or PK_MAT_Norm. This template format allows to find a match between an adult and a childhood fingerprint of the same person.

Please note that as the comparison of a 'normalized' template with a 'regular' template (such as a PK_Comp format template) is not significant, then it is not allowed by the terminal.

Then this function modifies the regular way of using the biometric commands :

- ENROLL command : the exported template is must be in a 'normalized' format (either PK_Comp_Norm or PK_Mat_Norm), other format are forbidden. If a record has to be created, the database must be created with the 'juvenile' option.
- IDENTIFY command : the database must be created with the 'juvenile' option.
- VERIFY command : the reference templates must be in 'normalized' format (either in PK_Comp_Norm or in PK_Mat_Norm).

This function should not be used if the terminal has to process only adult fingerprints.

Thin finger function :

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 ENROLL 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

See also: *VERIFY*, *ENROLL*, *IDENTIFY*, *UPDATE_PRIVATE_DATA*

7.8.5 Biometric presence detection mode ID = 0x44

This ILV is optional. It can be used to select the detection mode suited to your application. This parameter is used to fine-tune the system, especially if you face problem with dry fingers.

Identifier value	ID_MODE_DETECT	
Length value	0x04	
Value (Parameters)	Value	4 bytes

Value: bitmask of the following:

0x01: MORPHO_VERIF_DETECT_MODE this is a more permissive mode than default. In this mode, the MorphoSmart™ detects more easily finger presence, but might issue lower quality templates.

0x02: MORPHO_ENROLL_DETECT_MODE (default).

0x04: MORPHO_WAKEUP_LED_OFF : (only available on MSO xx1) turns off the leds on a MSO xx1 device while waiting for a finger (impedance wakeup).

Note : DETECT_PRES_MODE_VERIF and DETECT_PRES_MODE_ENROLL *cannot* be used together

See also: *VERIFY*, *ENROLL*, *IDENTIFY*, *UPDATE_PRIVATE_DATA*

7.8.6 Biometric matching strategy ID = 0x0B

This ILV is optional. It can be used to select the matching strategy suited to your application. This parameter is used to fine-tune the system, especially if you face problem with dry fingers, or if you need more accurate matchings, even if they take more time.

Identifier value	ID_MATCHING_STRATEGY	
Length value	0x01	
Value (Parameters)	Value	4 bytes

Value:

0x00: MORPHO_STANDARD_MATCHING_STRATEGY (default).

0x01: MORPHO_ADVANCED_MATCHING_STRATEGY : more powerful matching strategy resulting in lower FRR but increased processing time.

See also: *VERIFY*, *IDENTIFY*, *UPDATE_PRIVATE_DATA*

7.8.7 Matching PK Number ID = 0x63

This ILV is optional. It can be used in the *VERIFY* request to force the MorphoSmart™ to return the matching PK number. In a reply, it contains the matching PK number value.

Identifier value	ID_EXPORT_NUM_PK	
Length value	0x0001	
Value (Parameters)	Value or PK number	1 bytes

Value: Set to a value different from 0 to request the MorphoSmart™ to send the resulting matching PK number.

PK Number: Resulting matching PK number on 1 byte.

If a false or a too moist finger is detected, or if the finger is not authenticated, the returned matching PK number value is 0xFF.

If the comparison succeeds with the reference template i (i>=1), the matching PK number value is i-1. For example, if the comparison succeeds with the reference template 2, the matching PK number value is 1.

See also: *VERIFY*

7.9 MORPHOSMART ™ CONFIGURABLE PARAMETER

This table describes the parameters which can be get or modify with the *GET_MSO_CONFIG_VALUE* or *MODIFY_MSO_CONFIG_VALUE*.

ID	Meaning	Value	Length
0x0E10	*Sensor_Window_Position*	0 (default value), 1, 2	1 byte
0x0510	*Sleep_TimeOut*	0 (default value) to =<4000 (ms)	4 bytes

7.9.1 Sensor_Window_Position ID=0x0E10

Only available on MorphoSmart™ CBM.

This parameter configures the position of the sensor window.
The value of Sensor_Window_Position allows to orientate the image according to the orientation of the sensor window (see figure 7)
The sensor window has a default position to calculate the image: if the sensor has turned off 180 degrees then the image will be turned of 180 degrees too.

Parameter Value is 0 (default value, default position)

Parameter Value is 1:

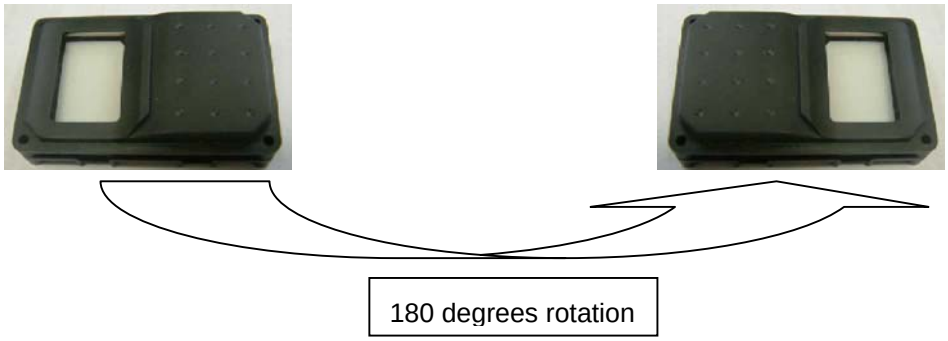


Figure 19: MorphoSmart™ CBM Sensor Window Position Configuration

If the parameter value is 2 then the matching is non oriented. The matching doesn't depend of the sensor position. You can put the sensor into the two previous positions.

Important note : using the non oriented matching, the identification time is increased, due to the internal behavior of the device.

The identification time increase depends also on the adopted 'matching strategy' :

Default : the identification time can double,

Enhanced : the identification time can go up 8 times.

Examples :

To set the Sensor_Window_Position to 180 degre, the ILV should be (hexadecimal values):

91 03 00 10 0E 01

To set the Sensor_Window_Position to non oriented matching, the ILV should be (hexadecimal values):

91 03 00 10 0E 02

To set the Sensor_Window_Position to default position, the ILV should be (hexadecimal values):

91 03 00 10 0E 00

7.9.2 Sleep_TimeOut ID=0x0510

This parameter configures, only for a serial Interface, the Timeout (in ms) when the system goes on LowPower. Its value is:

- 0 (default value): No LowPower
- >= 4000: Timeout (in ms) when the system goes on LowPower

8 Error codes

Error code	Definition	Value	Hex
ILV_OK	Successful result	0	0x00
ILVERR_ERROR	An error occurred	-1	0xFF
ILVERR_BADPARAMETER	Input parameters are not valid	-2	0xFE
ILVERR_INVALID_MINUTIAE	The minutiae is not valid	-3	0xFD
ILVERR_INVALID_USER_ID	The User ID is not valid.	-4	0xFC
ILVERR_INVALID_USER_DATA	The user data is not valid	-5	0xFB
ILVERR_TIMEOUT	No response after defined time.	-6	0xFA
ILVERR_ALREADY_ENROLLED	The person is already in this database	-8	0xF8
ILVERR_BASE_NOT_FOUND	The specified database doesn't exists	-9	0xF7
ILVERR_BASE_ALREADY_EXISTS	The specified database already exists	-10	0xF6
ILVERR_BIO_IN_PROGRESS	Command received during biometric processing	-11	0xF5
ILVERR_CMD_INPROGRESS	Command received while another command is running	-12	0xF4
ILVERR_FLASH_INVALID	Flash type invalid	-13	0xF3
ILVERR_NO_SPACE_LEFT	Not enough memory for the creation of a database	-14	0xF2
ILVERR_BAD_SIGNATURE	Invalid signature	-16	0xF0
ILVERR_OUT_OF_FIELD	Number of the additional field is greater than allowed or an additional field length is greater than allowed.	-21	0xEB
ILVERR_FIELD_NOT_FOUND	The required field does not exist in the database.	-23	0xE9
ILVERR_FIELD_INVALID	Field size or field name is invalid	-24	0xE8
ILVERR_USER_NOT_FOUND	The required User ID has not been found in the database	-26	0xE6
ILVERR_CMDE_ABORTED	Command has been cancelled.	-27	0xE5
ILVERR_SAME_FINGER	There are two templates of the same finger	-28	0xE4
ILVERR_NO_HIT	Presented finger does not match	-29	0xE3
ILVERR_SECU_CERTIF_NOT_EXIST	The required certificate does not exist	-30	0xE2
ILVERR_SECU_BAD_STATE	Invalid security state	-31	0xE1
ILVERR_SECU_ANTIREPLAY	An anti-replay error occurred	-32	0xE0
ILVERR_SECU_ASN1	Error while decoding an ASN1 object	-33	0xDF
ILVERR_SECU	Cryptographic error	-34	0xDE
ILVERR_SECU_AUTHENTICATION	Mutual authentication error	-35	0xDD
ILVERR_FFD	False Finger Detected	-37	0xDB
ILVERR_MOIST_FINGER	The finger can be too moist or the scanner is wet	-38	0xDA
ILVERR_APPLI_LOCKED	The MSO is locked	-68	0xBC
ILVERR_OTP_NOT_INITIALIZED	All OTP parameters are not initialized	-39	0xD9
ILVERR_NO_MORE_OTP	No more OTP (sequence number = 0)	-40	0xD8
ILVERR_OTP_ENROLL_NEEDED	OTP database is empty : enroll is needed	-41	0xD7
ILVERR_OTP_NO_HIT	No Hit (or latent or FFD)	-42	0xD6
ILVERR_OTP_REENROLL_NOT_ALLOWED	Re-enrolment not allowed	-43	0xD5

ILVERR_OTP_ENROLL_FAILED	Enrolment failed (status != ILV_OK)	-44	0xD4
ILVERR_OTP_IDENT_FAILED	Identification failed (status != ILV_OK except DB errors)	-45	0xD3
ILVERR_OTP_PIN_NEEDED	Pin needed for first enrolment	-46	0xD2
ILVERR_OTP_LOCK_SET_PARAM	After 5 false Pin presentation, the OtpSetParameters ILV is locked.	-65	0xBF
ILVERR_OTP_LOCK_ENROLL	After 5 failed enrolment with pin or 5 failed identification, the OTP_Enrol_User ILV is locked.	-66	0xBE
ILVERR_OTP_LOCK_GEN_OTP	After 10 failed identification, the OtpGenerate ILV is locked.	-67	0xBD
ILVERR_OPERATION_NOT_SUPPORTED	Operation not supported. For example, database operation are not supported by a WSQ software.	-57	0xC7
ILV_NOT_IMPLEMENTED	The request is not yet implemented	-99	0x9D

Table 6: Error codes

9 Status codes

Status code	Definition	Value	Hex
ILVSTS_OK	Successful	0	0x00
ILVSTS_HIT	Authentication or Identification succeeded	1	0x01
ILVSTS_NO_HIT	Authentication or Identification failed	2	0x02
ILVSTS_DB_FULL	The database is full.	4	0x04
ILVSTS_DB_EMPTY	The database is empty.	5	0x05
ILVSTS_FFD	False finger detected.	34	0x22
ILVSTS_MOIST_FINGER	The finger can be too moist or the scanner is wet.	35	0x23

Table 7: Status codes

10 Constants

Constant	Definition	Value	Hex
ID_USER_DATA	Identifies a user ID ILV	5	0x05
ID_COM1	Identifies serial communication port #1	6	0x06
ID_FIELD_SIZE	Identifies the size of a field in database	16	0x10
ID_TIME_STAMP	Not used. For compatibility with other MorphoAccess™.	17	0x11
ID_PUC_DATA	Identifies that the value is a string	20	0x14
ID_DESC_PRODUCT	Product descriptor	41	0x29
ID_DESC_SOFTWARE	Software descriptor	42	0x2A
ID_DESC_SENSOR	Sensor descriptor	43	0x2B
ID_COMPRESSION_NULL	No compression is used	44	0x2C
ID_FORMAT_TEXT	Text format for descriptor	47	0x2F
ID_FORMAT_BIN_VERSION	Version as return of the descriptor	116	0x74
ID_FORMAT_BIN_MAX_USER	Max nb of person enrolled in the base fort he descriptor	117	0x75
ID_COMPRESSION	Used to define compression parameters	62	0x3E
ID_PKCS12_TOKEN	PKCS#12 token	81	0x51
ID_NO_CHECK_ON_TEMPLATE	Checks on template during add base record	96	0x60
ID_DEFAULT_IMAGE	Default size and default resolution image	0	0x00
ID_COMPRESSION_V1	Default compression algorithm	60	0x3C
ID_COMPRESSION_WSQ	WSQ compression algorithm	156	0x9C
ID_FFD_LEVEL	Identifies the FFD security level	66	0x42
MORPHO_X984_SIGNED_TEMPLATE	Used to define the X9.84 biometric token envelope	1	0x01

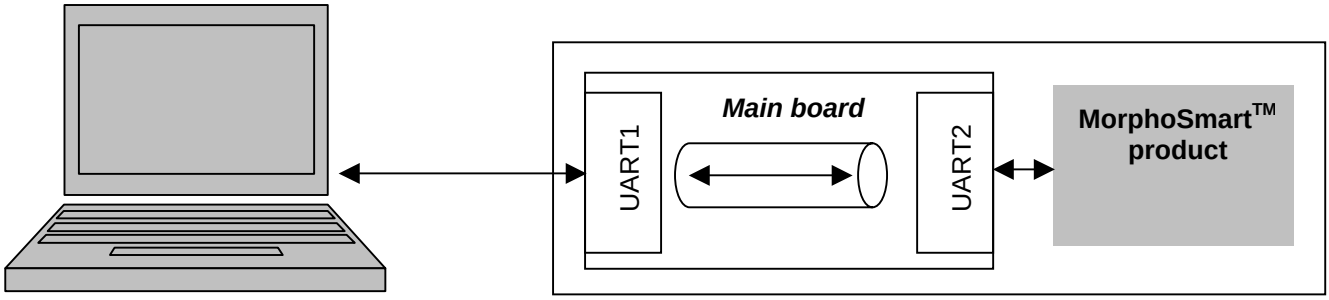
Table 8: Constant

11 Transparent firmware upgrade

When the MorphoSmart™ product is embedded in another equipment, the firmware upgrade has to be still available. To achieve the firmware upgrade, some conditions have to be met to ensure a transparent loading using Sagem Sécurité tools.

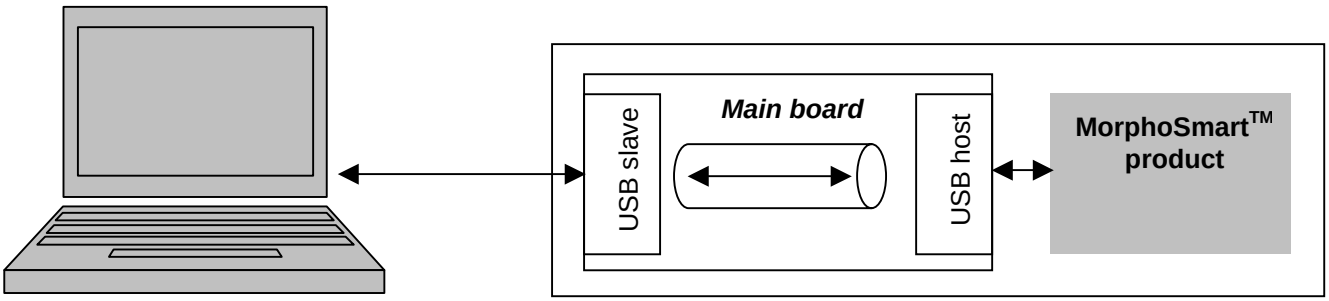
The tool used to upgrade the firmware is the Quickloader. The equipment embedding the MorphoSmart™ product must behave transparently, and the operation has to behave just like if the MorphoSmart™ product was directly connected to the PC.

11.1 SERIAL TRANSPARENT FIRMWARE UPGRADE



To enable correct firmware upgrade, it is necessary to develop in the embedding equipment a specific software being able to transfer frames in transparent mode (pipe) from UART1 to UART2 in both ways.

11.2 USB TRANSPARENT FIRMWARE UPGRADE



In USB, the principle is the same as in serial mode. It is necessary to develop in the embedding equipment a specific software able to transfer frames directly from the PC to MorphoSmart™ transparently in both ways.

A specific USB feature has to be taken into account: the ***Reset phase*** of the MorphoSmart™ should be also reported correctly to the PC in order to enable correct completion of the upgrade.

The QuickLoader tool needs the device to simulate its disconnection, and its enumeration according to the corresponding behaviour of the MorphoSmart™ product. This means the device has to monitor and report the MorphoSmart™ disconnection by its own disconnection of the bus.