SALTO
inspiredaccess

INDUSTRY STANDARD PMS PROTOCOL:
TECHNICAL REFERENCE

Version 1.16

Versions review

Date	Version	Description
17/7/02	1.0	First version.
3/12/02	1.1	One shot key: 1) only valid for one hour. 2) No authorisations can be included in the key. 'LT' command: in the previous document version 1.0, the content of the frame sent by the PC interface after key reading is not correctly explained. Moreover, the retention mode parameter (field #2) must not be included in the answer.
20/11/03	1.2	'CN' and 'CC' commands extended: these two commands can optionally include a new field (number #14) the content of which is written in track #3. New 'CP' command: a new command ('CP') for cancelling a guest lost key has been implemented. Card retention mode ('E', 'R', 'T'): for non-motorised key encoders, the retention mode indicates whether the PC interface should wait for the key to be removed from the encoder after the command is processed. This is necessary for encoding processes in which more than one command for the same key are implied ('CN' and 'P1' or 'L1' and 'L2', for example).
8/11/04	1.3	'WF', 'WN' and 'WR' commands implemented: the commands for collecting audit trail incidences are now available.
23/1/08	1.4	The 'CN' and 'CC' commands include a new input parameter (field number #15): if set to 1, then the PC interface will return the serial number of the recently issued card.
13/11/09	1.5	New commands 'CNB' and 'CCB': these commands are the same as 'CN' and 'CC' except that the binary content of the requested card is returned rather than having the PC interface issue the card in a proprietary SALTO encoder.
11/1/2012	1.6	The commands for collecting audit trail events ('WF', 'WN' and 'WR') now include new types for the copy number parameter ('S0', 'S2' & 'S3').
28/6/2012	1.7	Commands CNB and CCB now support the 'TagIt' card technology. Some minor wording mistakes fixed.
23/8/2012	1.8	Commands for collecting audit trail events ('WF', 'WN' & 'WR'): updated the list of possible values for field #7, i.e., copy number ('S0' removed). Some wording mistakes have been fixed.
13/5/2013	1.9	Command for getting card binary image (i.e., 'CNB' and 'CCB') now supports 7-byte Mifare UID.
24/3/2014	1.10	The number of authorisations within the CN & CC commands has been augmented from 20 to 62.
3/12/2014	1.11	New check-in commands for mobile phone apps: 'CNM' & 'CCM'.
7/4/2015	1.12	New commands 'CNMB' & 'CCMB': these commands are the same as 'CNM' & 'CCM', respectively, except that they return the binary data to be used by mobile apps.
17/9/2015	1.13	New commands 'CNMBX' and 'CCMBX': they are an extended version of the 'CNMB' and 'CCMB' commands, respectively. They just include a new parameter for supporting different types of card binary images.
16/12/2015	1.14	The protocol now offers an alternative to the 'CNMBX' command (used for obtaining the card binary image for mobile apps) as requested by a specific hotel chain.

		The maximum length of the 'Display text' (field #14) in the command 'CNM' has increased from 128 to 256 characters.
25/01/2016	1.15	Upon request, the "CN" and "CC" commands (encoding of guest cards) may return a list of card serial numbers.
15/06/2016	1.16	New command ('MC') for modifying guest check-in data, such as "room move" and extended stay.

Table of content

1. Message format

The following table shows the standard ASCII control characters used in the exchange of data between the PMS and the PC interface.

ASCII	HEX	Description
STX	02	Start of text, indicates the start of a message.
ETX	03	End of text, indicates the end of a message
ENQ	05	Enquiry about the PC interface being ready to receive a new message.
ACK	06	Positive acknowledgement to a PMS message or enquiry (ENQ).
NAK	15	Negative acknowledgement to a PMS message or enquiry (ENQ).

Table 1: ASCII control characters used in the protocol.

STX and *ETX* control characters indicate, respectively, the start and end of a message, as shown below. A message is comprised of readable ASCII characters.

STX	Message	ETX	LRC

In order to ensure the integrity of a message, an LRC (Longitudinal Redundancy Check) value must be specified just after the *ETX* character. LRC is calculated by performing the exclusive OR operation (XOR) on all characters after *STX* (*ETX* included) starting with 00H as seed. The PMS can avoid LRC calculation by sending a 0DH value (return character) instead of actual LRC. In this case, the PC interface will not check the integrity of the received message.

The ENQ control character is sent by the PMS to enquire the PC interface about its availability to process new requests.

Finally, the ACK and NAK control characters are sent by the PC interface to indicate, respectively, positive and negative acknowledgement of a PMS request.

2. Communication between PMS and PC interface

At the time of this writing, two communication transport layers are supported: RS-232 and tcp/ip. Communication parameters are fully configurable from the PC interface software.

When the PMS sends a message to the PC interface, the PMS must wait for an ACK or NAK before sending a new message. If an ACK or NAK has not been received within 2 seconds, then the PMS can assume that the server cannot receive the message and appropriate action must be taken.

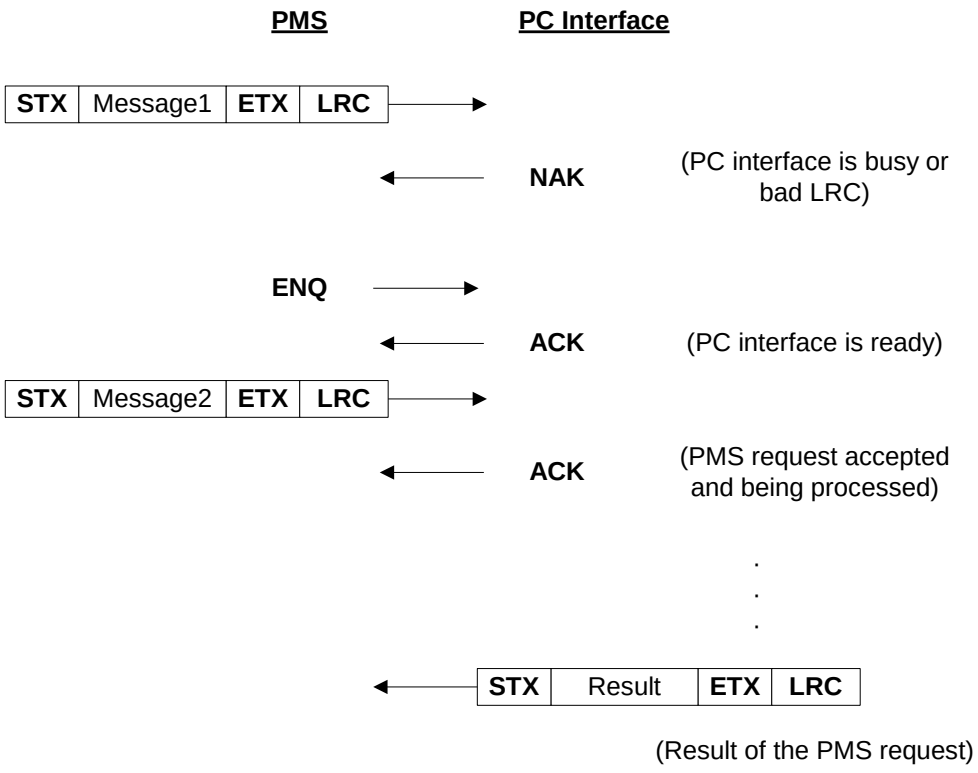
An ACK from the PC interface means that the requested PMS command has been accepted and is being processed. The PC interface will send the result of the request as soon as it has been processed.

When the PMS receives a NAK character, it means that either the PC interface is not prepared to process the requested command or the message sent by the PMS is not correct (bad LRC).

The PMS can send the ENQ character at any time to know if the PC interface is prepared to receive a new message.

In the following lines, an example of communication between the PMS and the PC interface is shown.

Example 1:



3. Commands

Basically, a message is a concatenation of fields as shown below. Each field is delimited by a separator character, the ASCII code of which is **hexadecimal B3H or decimal 179** (represented by '|'). Note that standard PS2 keyboards do not contain such a key: you must press the ALT key and type 179 in the numeric keypad.

The first field (field #0) contains a command to be executed by the PC interface and subsequent fields contain the parameters of the command. Commands are comprised of two readable ASCII characters. Table 2 shows a summary of the available commands.

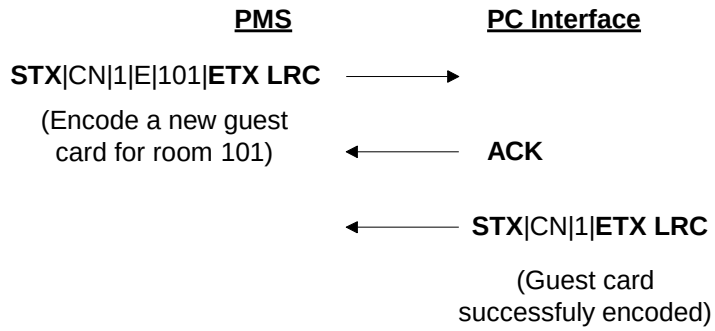
Command	Description
'CN'[x]	Encode a new guest card. Optionally, this command can be followed by a number 'x', which indicates the amount of cards to be encoded: the first issued card will be the original and the rest (x-1) will be copies of the first one. If 'x' is not specified, the default value is 1. This command is used for hotel check-in.
'CC'[x]	Encode a copy of a given guest card (for people who share the same room). Optionally, a number 'x' of copies to be encoded can follow this command. If 'x' is not specified, the default value is 1.
'CO'	Perform a guest checkout.
'CNB'	Same as 'CN' except that the binary content of the card is returned rather than having the Salto server issue the card in a proprietary SALTO encoder.
'CCB'	Same as 'CC' except that a the binary content of the card is returned rather than having the Salto server issue the card in a proprietary SALTO encoder.
'CNM'	Same as 'CN' except that the access permissions are "written" (over the air) on a mobile app rather than on a physical card.
'CCM'	Same as 'CC' except that the access permissions are "written" (over the air) on a mobile app rather than on a physical card.
'CNMB'	Same as 'CNM' except that actual binary data is returned rather than being sent (over the air) to the target mobile phone.
'CCMB'	Same as 'CCM' except that actual binary data is returned rather than being sent (over the air) to the target mobile phone.
'CNMBX'	An extended version of 'CNMB'.
'CCMBX'	An extended version of 'CCMB'.
'CA'	Encode a single opening card. This kind of cards can only be used once on room locks.
'EX'	Abort an operation being executed on the specified encoder.
'LT'	Read the content of a card.
'P1'	Write information on track #1.
'P2'	Write information on track #2.
'P3'	Write information on track #3.

'L1'	Read information from track #1.
'L2'	Read information from track #2.
'L3'	Read information from track #3.
'CP'	Cancel guest lost key.
'WF'	Obtain the oldest opening (or rejection) incidence of a given door/peripheral.
'WN'	Obtain the next opening (or rejection) incidence of a given door/peripheral.
'WR'	Resend the last opening (or rejection) incidence of a given door/peripheral.
'MC'	Modify check-in data for a existing room guest, such as moving to another room or extending stay (without the need of reencoding guest cards).

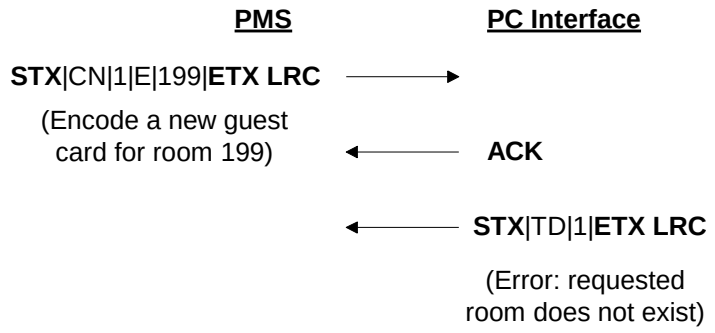
Table 2: available PMS commands.

The next field after the command (field #1) must contain the encoder number in which the command is to be executed. If the PC interface successfully executes the PMS request, the answer will include at least the executed command plus the editor in which the operation was performed. Otherwise, an error message is sent back (see examples below).

Example 2: In this example, the PMS is asking for a new guest card (for room 101) to be issued in encoder #1. After positive acknowledgement, the PC interface returns a message saying that the requested card has been successfully encoded.



Example 3: In this example, the PMS is asking for a new guest card (for room 199) to be issued in encoder #1. After positive acknowledgement, the PC interface returns an error message saying that the specified room does not exist.



The following sections describe the characteristics of each command.

3.1 Command 'CN': encode a new guest card.

Table 3 shows the parameters for this command.

Field	Description
0	'CN'[x]
1	Encoder number in which the requested card will be encoded.
2	For motorised encoders (not available in Salto), this field indicates whether encoder card must be retained or ejected from the encoder. Otherwise, whether the PC interface waits for the card to be removed from the encoder: 'E' → Ejection. The PC interface waits for the key to be removed from the encoder. 'R' → Retention. The PC interface does not wait for the key to be removed. 'T' → Ejection by the rear side (not used). Same effect as 'E'.
3	First room to be opened by the card (main room). Max. 24 characters.
4	Second room to be opened by the card. Max. 24 characters.
5	Third room to be opened by the card. Max. 24 characters.
6	Fourth room to be opened by the card. Max. 24 characters.
7	Authorisations granted to guest.
8	Authorisations denied to guest.
9	Starting date and time of the card.
10	Expiring date and time of the card.
11	Data of the operator who makes the request. Max. 24 characters.
12	Information to be written on track #1.
13	Information to be written on track #2.
14	Information to be written on track #3.
15	Whether the PC interface is to return the card's serial number or not.

Table 3: input parameters for command 'CN'.

The first three input parameters after the command (encoder number, retention mode and room name) are obligatory; the remainder can be left empty. Blank fields located between the last non-empty field and *ETX* can be eliminated from the message if desired.

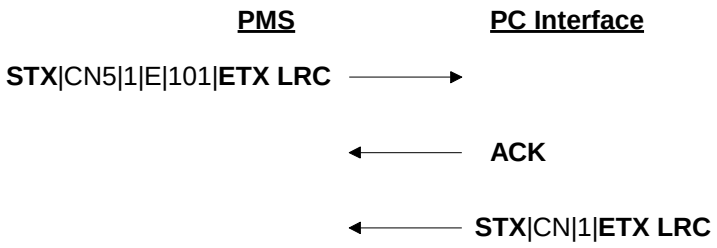
Example 4:

> **STX**|CN|1|E|101|||||||||**|ETX LRC**
>
> is the same as
>
> **STX**|CN|1|E|101**|ETX LRC**
>
> The PC interface will issue a new guest card for room number 101 in encoder #1.

Optionally, a number 'x' indicating the amount of cards to be encoded can follow the 'CN' command. In this case, the first issued card is considered to be the original one, whereas the rest (x-1) are copies of the first card. If no value is specified, the PC interface will encode one card only.

If the request is processed correctly, the PC interface will return the 'CN' command plus the encoder name, as shown in the example below. Additionally, the PC interface may return the serial number of the recently issued card, depending on the content of the input field #15 (more on this later).

Example 5: In this example, the PC interface will issue 1 new guest card and 4 copies.



3.1.1 Special processing of encoder '0' (zero)

As stated in Table 3, field #1 contains the name of the Salto proprietary encoder on which the operator should place the card in order to write the requested access permissions. Any alphanumeric value is allowed for this field, including just '0'.

However, there exists a special setting within the Salto software (designed for a specific hotel chain) that, when enabled, will make the "CN" command be processed differently depending on the content of this field: if the name of the encoder equals to '0', then the Salto software will return a binary image of the requested guest key suitable for mobile apps developed by a specific hotel chain (see section 3.11 for further details). Otherwise (that is, encoder name different from '0'), the "CN" command will behave as usual.

Note that this special setting within the Salto software has been designed for a specific hotel chain. For the rest of the hotel chains and integrators, this setting should be regarded as being disabled and treat encoder '0' as if it were any other encoder name.

3.1.2 Retention mode (field #2)

For non-motorised key encoders, the retention mode indicates whether the PC interface should wait for the key to be removed from the encoder before sending the result of the encoding back to the PMS.

If a 'R' value is specified in this field, the PC interface will send the result to the PMS as soon as the key is encoded, without waiting for the key to be removed from the encoder. On the contrary, the 'E' or 'T' values indicate that the PC interface will wait for the key to be removed and then send the corresponding response.

The retention mode is necessary for encoding processes in which more than one command for the same key (or room) are implied (for example 'CN' and 'P1').

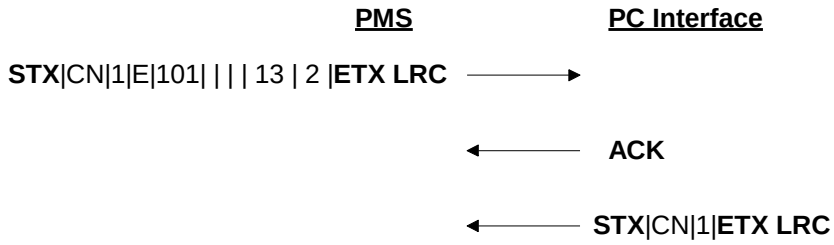
3.1.3 Authorisations (fields #7 and #8)

Cardholders can access doors other than rooms by means of authorisations. An authorisation is an access permission to one or more doors. This mapping between authorisations and doors must be specified in the locking plan (Salto DB).

The maximum number of authorisations is 62. Each authorisation is represented by an ANSI character. The Appendix B section shows a list of valid ANSI characters and their corresponding authorisation number within the Salto DB. For example, character '1' represents authorisation number 1 whereas character '&' represents authorisation number 40.

Example 6: if we want to grant to guest 101 the use of sauna plus pool and gym (authorisations '3' and '1') and deny the use of parking (authorisation '2'), the message should look like as follows:

Authorisation symbol		Door
'1'	→	Pool and Gym
'2'	→	Parking
'3'	→	Sauna
…		



Note that non-specified authorisations will take default values from locking plan.

3.1.4 Starting and expiration date (fields #9 and #10)

Fields number 9 and 10 contain card starting and expiration date, respectively. The format for these fields is as follows:

'hh[mm]DDMMYY'

where **hh**: hour (00 to 23).

mm: minutes (00 to 59).

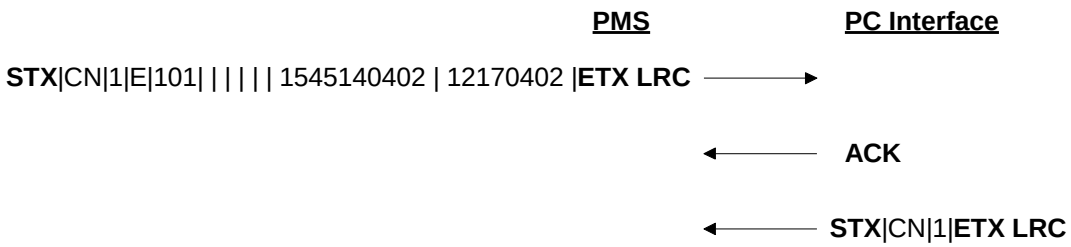
DD: date (01 to 31).

MM: month (01 to 12).

YY: year (00 to 99).

Note that minute information is optional. If no minute information is specified, the default value will be 0.

Example 7: In this example, the PC interface will issue a guest card for room 101 valid from 15:45 April 14th 2002 to 12:00 April 17th 2002.

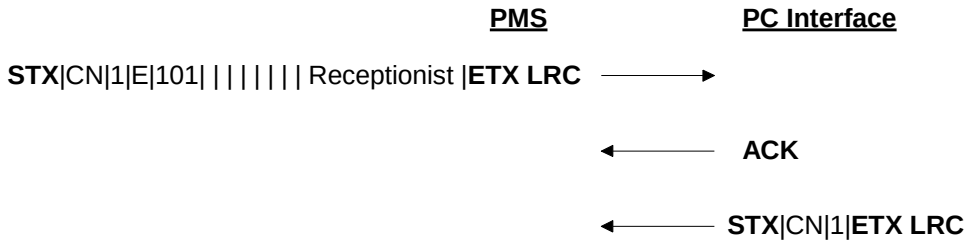


If field #9 corresponding to starting date is left empty, the issued card will be operative immediately. On the other hand, if no expiration is specified (field #10 empty), the issued card will be valid for one day only (this default value can be modified in the locking plan).

3.1.5 Operator data (field #11)[1]

In general, the PC interface audits every issued card requested from the PMS. Sometimes, it might be desirable to include in the audit record the name of the operator who made the request. The PMS can provide with this information by filling field #11.

Example 8: In this example, the PC interface will issue a guest card for room 101 and it will generate an audit record that includes 'Receptionist' as the request applicant.



Note that if no value is given to this field, the default value is 'PMS'.

3.1.6 Messages on track 1 and track 2 (fields #12, #13 and #14)

When issuing a card, the PC interface writes encrypted information that is meaningful to electronic locks only. However, it might be desirable to write other kind of information that can make sense to other devices in the property, such as 'point-of-sale' terminals, etc. This protocol allows non-encrypted information to be written on previously allocated areas in the card (named as track 1, 2 and 3).

Fields #12, #13 and #14 contain the information to be written on track 1, track 2 and track3, respectively. This information must be comprised of readable ASCII characters and should not include the field separator character, '|'.

[1] As of this writing, the content of field #11 is not processed by the Salto software.

Example 9: Let's suppose we need to issue a card having the following features:

- Access to room 1.
- Additional access to room 2.
- Parking granted (authorisation number 2).
- No starting date.
- Expiration at 1 PM on 4th July 2002.
- Name of the guest on track 1.
- Main room number on track 2.

The resulting message to be sent by the PMS will look like:

STX|CN|1|E|101|102| | | 2 | | |13040702| |MR J. BROWN|101|**ETX LRC**

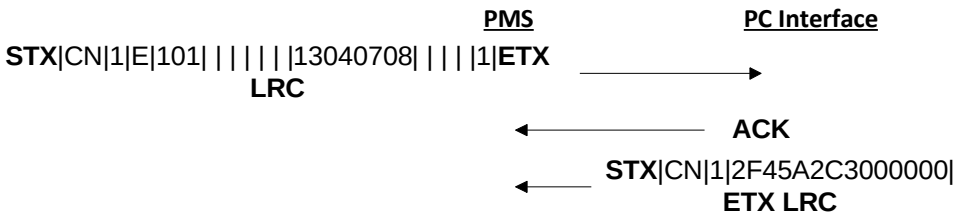
3.1.7 Return card serial number (field #15)

If desired, the PMS client may ask the Salto server to return the the serial number of the issued card. This can be asked with field #15, which may contain one of the following values:

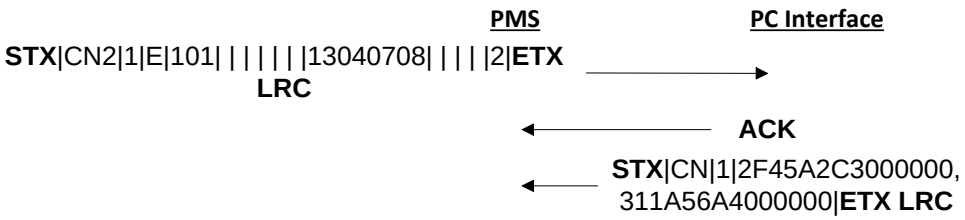
- 0 (or blank): no serial number is returned.
- 1: one serial number is returned. If several cards are requested ("CNx"), then the serial number of the last issued card is returned.
- 2: a list of serial numbers is returned, one per issued card. Each value is separated by comma.

The resulting serial number will be written in the output field #2 within the response frame.

Example 10: In this example, the PC interface will issue a guest card for room 101 and will return the serial number of the card in field #2.



Example 11: this is the same example as the one above except for the following differences: (1) the PMS client asks 2 cards to be issued ("CN2"); (2) the PMS client asks that a list of serial numbers should be returned ("2" in field #15); (3) finally, the Salto server returns a list of serial numbers (within field #2).


PMS
PC Interface
STX|CN2|1|E|101| | | | | | |13040708| | | | |2|ETX LRC
ACK
STX|CN|1|2F45A2C3000000, 311A56A4000000|ETX LRC

3.2 Command 'CC': copy guest card

This command is used to encode copies of guest cards. Note that the PC interface cannot perform this command if the specified rooms are checked out or not occupied. The parameters for this command are exactly the same as for 'CN'.

3.3 Command 'CNB': get binary data for a new guest card

In certain scenarios, it is desirable to get the actual content of the requested card in binary format rather than having the PC interface issue the card in a proprietary SALTO encoder. This allows the PMS to use a 3rd party (non-SALTO) encoder to write the raw data on the card. This feature is especially useful when the card issuing process involves several steps and only one card encoder is allowed: for example, in a badging process, first a picture is printed on the card and then the SALTO access control permissions are written by the same device.

Important warning #1: the 'CNB' command (like any command returning card binary image) requires card memory access to be protected by your own security keys (rather than Salto's security keys). This implies that your card security keys must be shared with all the Salto devices in the installation so that it can be possible to read from or write to your cards. This process of sharing card security keys is achieved by means of a special card called SAM. Thus, make sure you ask Salto for a customised SAM card (containing your security keys) before working with the "CNB" and "CCB" commands.

Important warning #2: the 'CNB' command (like any command returning card binary image) requires a special hardware, called dongle, without which it would not work.

As shown in the table below, the 'CNB' command is almost the same as the 'CN' one (see section 3.1 in page 10), except for the following differences:

- No number of cards (immediately after the command) can be specified.
- Field #1 now contains the serial number of the target card.
- Field #2 now contains information about the structure of the target card.
- As for the rest of the fields, they preserve the same meaning and considerations as in the CN command.

The table below shows the input parameters for this command:

Field	Description
0	'CNB'
1	Serial number of the target card.
2	Information about the structure of the target card on which the returned binary data is to be written.
3	First room to be opened by the card (main room). Max. 24 characters.
4	Second room to be opened by the card. Max. 24 characters.
5	Third room to be opened by the card. Max. 24 characters.
6	Fourth room to be opened by the card. Max. 24 characters.
7	Authorisations granted to guest.
8	Authorisations denied to guest.
9	Starting date and time of the card.
10	Expiring date and time of the card.
11	Data of the operator who makes the request. Max. 24 characters.
12	Information to be written on track #1.
13	Information to be written on track #2.
14	Information to be written on track #3.

Table 4: input parameters for the command 'CNB'.

The first three parameters after the command (i.e., target card's serial number, structure data and room name) are obligatory; the remainder may be left empty, in which case the default values come into place (note that blank fields located between the last non-empty field and *ETX* can be eliminated from the message if desired).

On processed, the PC interface will return the content of the card in binary format as shown in the table below.

Field	Description
0	'CNB'
1	Serial number of the target card.
2	Binary content of the card.

Table 5: output parameters for the command 'CNB'.

In the following sections, only input fields #1 and #2 will be explained. As for the rest of the fields, the reader should refer to the section concerning the 'CN' command since the same considerations apply.

3.3.1 Card's serial number

Both the input field #1 in the request frame and the output field #1 in the response frame contain the target card's serial number, which is a unique identified provided by the card manufacturer. Depending on the actual card's technology, the length of this field varies (for example, in case of a Mifare card, the serial number is either 4 or 7 bytes long whereas for a HIDiCLASS card, the serial number takes 8 bytes long). Table 6 shows the size of serial number for each of the supported card technology.

Serial numbers are represented as a string of hexadecimal ASCII characters (that is, '0' to '9', 'a' to 'f' and 'A' to 'F'). It is obvious to say that every to two characters within the string correspond to one byte in the card serial number.

Note that the most significant byte comes first in the string, that is, the first two hexadecimal characters represent the most significant byte of the card serial number, whereas the last two characters represent the least significant byte.

3.3.2 Card's structure information

The input field #2 must contain information about the structure of the target card on which the binary data is eventually to be written. The actual content depends on the target card's technology (e.g., Mifare, Desfire, etc.) and the memory allocated to SALTO.

In general, this field is comprised of a sequence of values separated by comma. Unless otherwise stated, the first value within the sequence will indicate the target card's technology and the rest of the values will depend on the specified technology.

In the table below, the currently supported technologies and their corresponding code are shown:

Code	Card technology	Size (in bytes) of card serial number
1	Mifare.	4 or 7
2	HIDiCLASS	8
3	Desfire	7
4	TagIt	8

Table 6: supported card technologies and the size of their serial number.

The following sections describe, for each of the supported technology, the card structure format to be stored in the input field #2.

3.3.3 Mifare cards

For Mifare cards, the first value within field #2 is set to '1', indicating that what follows represents the structure of a Mifare card. More specifically, the values after the Mifare code represent a list of Mifare sector IDs allocated to SALTO within the target card. The table below shows the actual content of the input field #2 for Mifare cards:

Position	Type	Description
0	Integer	Mifare type code (=1).
1	Integer	ID of the 1st Mifare sector allocated to SALTO.
2	Integer	ID of the 2nd Mifare sector allocated to SALTO.
...		
N	Integer	ID of the Nth Mifare sector allocated to SALTO.

Table 7: representation of a card structure for Mifare within input field #2.

Example 12: Let's suppose that for a given Mifare card, sectors #11, #12, #13, #14 and #15 are allocated to SALTO. The content of field #2 should be as follows:

```
Content of input field #2 = |1, 11, 12, 13, 14, 15|
```

Base on the provided structure data, the PC interface will construct the corresponding binary card data and place it on the output field #2. Actually, the card binary data for Mifare is comprised of a set of binary blocks, each of which is, in turn, comprised of 16 bytes and is defined by the following three parameters:

- Sector ID (0 to 39).
- Block ID within the specified sector (0 to 15).
- A series of bytes (16 bytes) representing binary content of the specified block.

Note that the first bytes (from left to right) in the binary string correspond to the lowest locations within the block (i.e., address 0) and the last bytes correspond to the highest locations (address 15). For example:

```
Binary block
Address:      00              08            15
Binary data: 1F643BC67DD3A8AC625AF98F6AF8C380
```

The table below describes the actual content of the output field #2. Except for the first value, which indicates the type of technology (in this case, Mifare), every consecutive three values (corresponding to sector ID, block ID and binary data, respectively) represents a binary block.

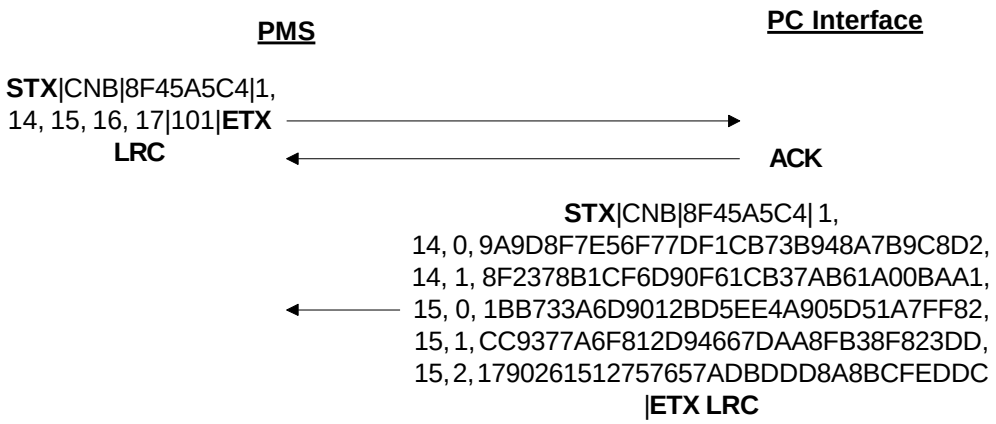
Position	Type	Description
0	Integer	Mifare type code (=1).
1	Integer	Mifare sector ID of the 1st block image.
2	Integer	Block ID (within the above specified sector) of the 1st block image.
3	Alphanumeric(32)	Hexadecimal representation of the 1st block image.
4	Integer	Mifare sector ID of the 2nd block image.
5	Integer	Block ID (within the specified sector) of the 2nd block image.
6	Alphanumeric(32)	Hexadecimal representation of the 2nd block image.
...		
N-2	Integer	Mifare sector ID of the N/3-th block image.
N-1	Integer	Block ID (within the specified sector) of the N/3-th block image.
N	Alphanumeric(32)	Hexadecimal representation of the N/3-th block image.

Table 8: parameters within output field #2 representing the requested binary data for a Mifare card.

Example: The following lines represent 5 binary blocks of a Mifare card (two blocks in sector #14 and three in sector #15):

```
Sector Block ID     Data
14     0            9A9D8F7E56F77DF1CB73B948A7B9C8D2
14     1            8F2378B1CF6D90F61CB37AB61A00BAA1
15     0            1BB733A6D9012BD5EE4A905D51A7FF82
15     1            CC9377A6F812D94667DAA8FB38F823DD
15     2            1790261512757657ADBDDD8A8BCFEDDC

Content of field #2:
|1, 14, 0, 9A9D8F7E56F77DF1CB73B948A7B9C8D2,
    14, 1, 8F2378B1CF6D90F61CB37AB61A00BAA1,
    15, 0, 1BB733A6D9012BD5EE4A905D51A7FF82,
    15, 1, CC9377A6F812D94667DAA8FB38F823DD,
    15, 2, 1790261512757657ADBDDD8A8BCFEDDC|
```



In this example, the PMS is requesting for the binary data of a new guest card accessing room 101. Both the serial number and the structure of a Mifare card are given in fields #1 and #2, respectively. The PC interface returns a set of 5 binary blocks.

3.3.4 HIDiCLASS cards

The first value within field #2 must be 2, indicating that what follows corresponds to the structure data of a HIDiCLASS card. More specifically, the structure data is represented by a list of HIDiCLASS pages allocated to SALTO in the target card. Each of the allocated pages is defined by the following couple of parameters:

- Page ID (from 0 to 15).
- Size of the page (either 2Kbits or 16 Kbits).

The table below shows the actual content of the input field #2 for representing the allocated structure within a HIDiCLASS card:

Position	Type	Description
0	Integer	HIDiCLASS type code (=2).
1	Integer	ID of the 1st HIDiCLASS page allocated to SALTO.
2	Boolean	Whether the 1st page is 16Kbits long (1) or 2Kbits (0).
3	Integer	ID of the 2nd HIDiCLASS page allocated to SALTO.
4	Boolean	Whether the 2nd page is 16Kbits long (1) or 2Kbits (0).
...		
N-1	Integer	ID of the N/2-th HIDiCLASS page allocated to SALTO.
N	Boolean	Whether the N/2-th page is 16Kbits long (1) or 2Kbits (0).

Table 9: representation of a card structure for HIDiCLASS within input field #2.

Example 13: Let's assume that, for a given HIDiCLASS card, pages #5, #6 and #8 are allocated to SALTO, all of which are small pages (i.e., 2 Kbits) except page #8, whose size is (16 Kbits). The content of field #2 for this example will be as follows:

```
Content of input field #2 = |2, 5, 0, 6, 0, 8, 1|
```

The resulting binary data within the output field #2 is actually comprised of a set of binary blocks (8 bytes long), each of which is defined by the following three parameters:

- Page ID (from 0 to 15).
- Block ID within the page (from 6 to 255).
- Binary image data (8 bytes) within the block.

The table below describes the actual content of the output field #2. The first value indicates the type of technology (in this case, HIDiCLASS); as for the rest, every consecutive three values (corresponding to page ID, block ID and binary data, respectively) represent one binary block.

Position	Type	Description
0	Integer	HIDiCLASS type code (=2).
1	Integer	HIDiCLASS page ID of the 1st block image.
2	Integer	Block ID (within the above specified page) of the 1st block image.
3	Alphanumeric(16)	Hexadecimal representation of the 1st block image.
4	Integer	HIDiCLASS page ID of the 2nd block image.
5	Integer	Block ID (within the specified page) of the 2nd block image.
6	Alphanumeric(16)	Hexadecimal representation of the 2nd block image.
...		
N-2	Integer	HIDiCLASS page ID of the N/3-th block image.
N-1	Integer	Block ID (within the specified page) of the N/3-th block image.
N	Alphanumeric(16)	Hexadecimal representation of the N/3-th block image.

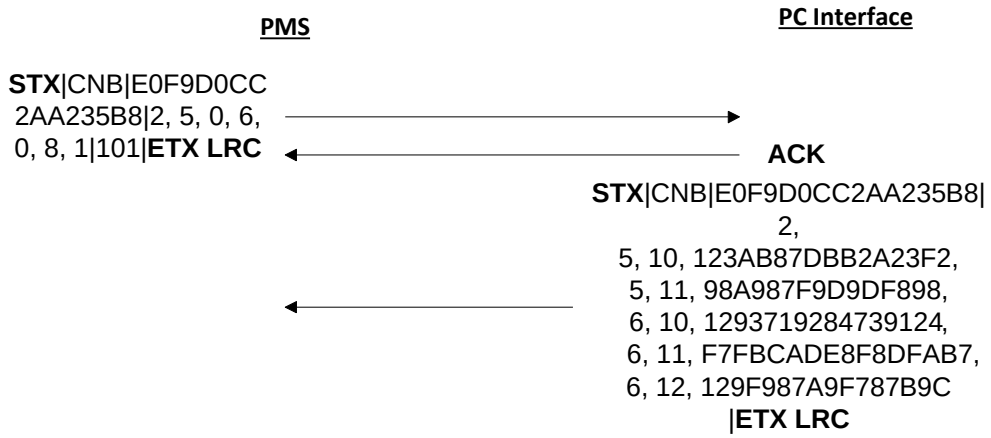
Table 10: parameters within output field #2 representing the requested binary data for a HIDiCLASS card.

Example 14: The following lines represent 5 binary blocks of a HIDiCLASS card (two blocks in page #5 and three in page #6):

```
Page   Block  Binary data
5      10     123AB87DBB2A23F2
5      11     98A987F9D9DF898F
6      10     1293719284739124
6      11     F7FBCADE8F8DFAB7
6      12     129F987A9F787B9C


Content of the output field #2:
|2, 5, 10, 123AB87DBB2A23F2, 5, 11, 98A987F9D9DF898, 6, 10, 1293719284739124,
    6, 11, F7FBCADE8F8DFAB7, 6, 12, 129F987A9F787B9C|
```

In this example, the PMS is requesting for the binary data of a new guest card accessing room 101. Both the serial number and the structure of a HIDiCLASS card are given in fields #1 and #2, respectively. The PC interface returns a set of 5 binary blocks.



3.3.5 Desfire cards

In case of Desfire, the first value within field #2 must be 3, indicating that what follows corresponds to the structure data of a Desfire card. More specifically, the PMS must provide information about the files allocated to SALTO within the target card. Each of the allocated files is defined by the following two parameters:

- File ID.
- File size.

The table below shows the how the structure information for a Desfire card is represented within the input field #2:

Position	Type	Description
0	Integer	Desfire type code (=3).
1	Integer	ID of the 1st Desfire file allocated to SALTO.
2	Integer	Size of the specified file.
3	Integer	ID of the 2nd Desfire file allocated to SALTO.
4	Integer	Size of the specified file.
...		
N-1	Integer	ID of the N/2-th Desfire file allocated to SALTO.
N	Integer	Size of the specified file.

Table 11: representation of a card structure for Desfire within input field #2.

Example 15: Let's assume that for a given card, files #1 and #2 (whose size is, respectively, 1024 and 560 bytes) are allocated to SALTO. The content of field #2 is as follows:

```
Content of input field #2= |3, 1, 1024, 2, 560|
```

As for the resulting binary data, it is actually comprised of a set of binary block, each of which is defined by the following three parameters:

- File ID (from 0 to 15).
- Starting address within the file.
- Binary data within the file starting from the specified address.

The table below describes the actual content of the output field #2. The first value indicates the type of technology (in this case, Desfire), as for the rest, every consecutive three values (corresponding to file ID, starting address and binary data, respectively) represent a binary block.

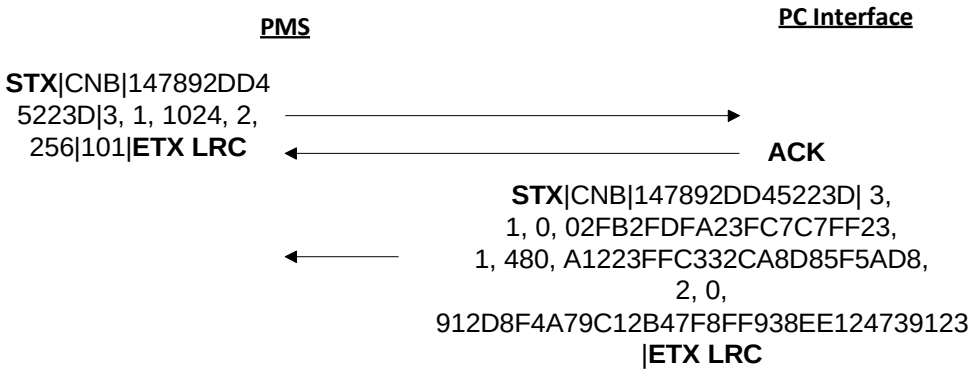
Position	Type	Description
0	Integer	Desfire type code (=3).
1	Integer	Desfire file ID of the 1st binary block.
2	Integer	Starting address of the binary data within the specified file.
3	Alphanumeric	Hexadecimal representation of the 1st binary block.
4	Integer	Desfire file ID of the 2nd binary block.
5	Integer	Starting address of the image data within the specified file.
6	Alphanumeric	Hexadecimal representation of the 2nd binary block.
...		
N-2	Integer	Desfire file ID of the N/3-th binary block.
N-1	Integer	Starting address of the image data within the specified file.
N	Alphanumeric	Hexadecimal representation of the N/3-th binary block.

Table 12: parameters within output field #2 representing the requested binary data for a Desfire card.

Example 16: The following lines three binary blocks for a Desfire card. Note that the first two binary blocks are contained within the same file (i.e., file #1) but they are located at different addresses (0 and 480, respectively). The third binary block is contained within file #2 at address 0:

```
[File] [Start address]     [Binary block]
#1            0            02FB2FDFA23FC7C7FF23
#1            480          A1223FFC332CA8D85F5AD8
#2            0            912D8F4A79C12B47F8FF938EE124739123
Content of the output field #2:
|3, 1, 0, 02FB2FDFA23FC7C7FF23,
    1, 480, A1223FFC332CA8D85F5AD8,
    2, 0, 912D8F4A79C12B47F8FF938EE124739123|
```

In this example, the PMS is requesting for the binary data of a new guest card accessing room 101. Both the serial number and the structure of a Desfire card are given in fields #1 and #2, respectively. The PC interface returns a set of 3 binary blocks.



3.3.6 TagIt cards

In case of TagIt cards, the technology type code (first value within field #2) must be '4', indicating that what follows corresponds to a TagIt configuration. This configuration, as shown in the table below, is just comprised of one value, namely, the memory size reserved to Salto.

Position	Type	Description
0	Integer	TagIt type code (=4).
1	Integer	Memory size (in bytes) of the Salto application.

Table 13: representation of a card structure for the TagIt technology within input field #2.

Example 17: Let's assume that a given TagIt card has 128 bytes allocated to Salto. The content of field #2 would be as follows:

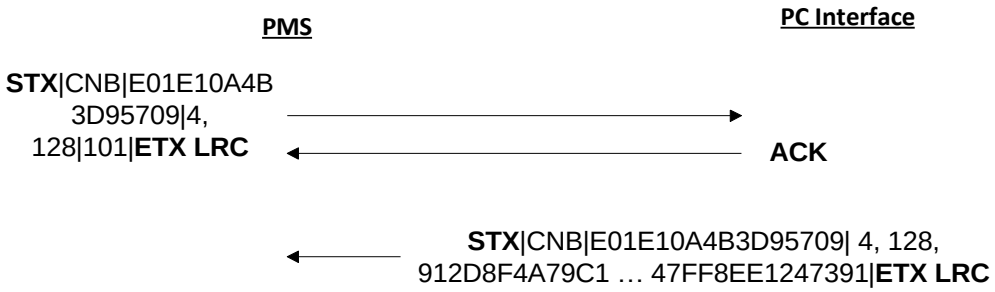
```
Content of input field #2= |4, 128|
```

As for the Salto response, the table below describes the actual content of the output field #2. The first value indicates the type of card technology (in this case, '4'). The second value just represents the whole content of the allocated card memory (in hexadecimal format).

Position	Type	Description
0	Integer	TagIt type code (=4).
1	Integer	Starting memory address (in decimal) from which to write the returned binary data.
2	Alphanumeric	Memory image in hexadecimal format.

Table 14: parameters within output field #2 representing the requested binary data for a TagIt card.

Example 18: Let's assume that PMS is requesting for the binary data of a new guest card accessing room 101. Both the serial number and the memory size of a TagIt card are given in fields #1 and #2, respectively. The PC interface returns the corresponding binary image for the Salto application.


PMS
PC Interface
STX|CNB|E01E10A4B 3D95709|4, 128|101|ETX LRC
ACK
STX|CNB|E01E10A4B3D95709| 4, 128, 912D8F4A79C1 … 47FF8EE1247391|ETX LRC

3.4 Command 'CCB': get binary data for a copy guest card

This command is the binary counterpart of the 'CC' command (copy guest card), as explained in section 3.2, page 16. Thus, it is used to get the binary content of a copy guest card. The parameters for this command are exactly the same as for 'CNB' (see section 3.3 in page 17).

As with the 'CC' commnad, an error will be produced if the specified rooms are checked out or not occupied.

3.5 Command 'CNM': check-in for mobile apps

Salto locks are designed to support not only card-based contactless technologies (such as Mifare) but also phone-based ones (such as BLE or NFC). The advantage of BLE, for instance, is that you may upload (over the air) a given smart-phone with the appropriate access permissions data and use this device to open doors as if it were a conventional proximity card.

The protocol command that makes this possible is the 'CNM' command: its syntax is very similar to the card-based counterpart, i.e., the check-in command 'CN' (see section 3.1 in page 10) except for the following differences:

- No number of cards is accepted (in field #0).
- The telephone number of the target smart phone must be provided (in field #1).
- Text message may be specified for showing up in the phone's screen (field #14).
- As for the rest of the fields, they preserve the same meaning and considerations as in the CN command (though their position is shifted by one).

The table below shows the input parameters for this command:

Field	Description
0	'CNM'
1	Telephone number of the target smart phone.[2]
2	First room to be opened by the card (main room). Max. 24 characters.
3	Second room to be opened by the card. Max. 24 characters.
4	Third room to be opened by the card. Max. 24 characters.
5	Fourth room to be opened by the card. Max. 24 characters.
6	Authorisations granted to guest.
7	Authorisations denied to guest.
8	Starting date and time of the card.
9	Expiring date and time of the card.
10	Data of the operator who makes the request. Max. 24 characters.
11	Information to be written on track #1.[3]
12	Information to be written on track #2.
13	Information to be written on track #3.
14	Text message to be shown on the phone's display. Max. 256 characters.

Table 15: input parameters for the command 'CNM'.

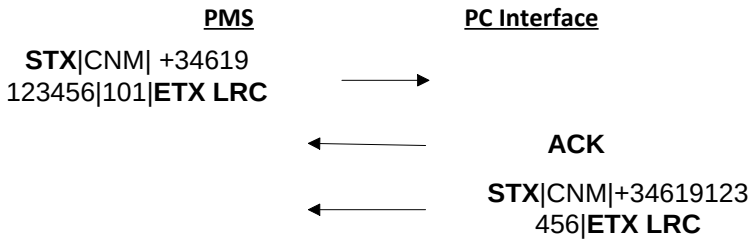
For a more detailed explanation about the meaning of each field, see the equivalent command 'CN', located in section 3.1 in page 10.

[2] The telephone number to which to send access permissions must contain the international prefix. That is, it must start with the '+' symbol and be followed by at least 6 digits. Otherwise, an error will be returned.

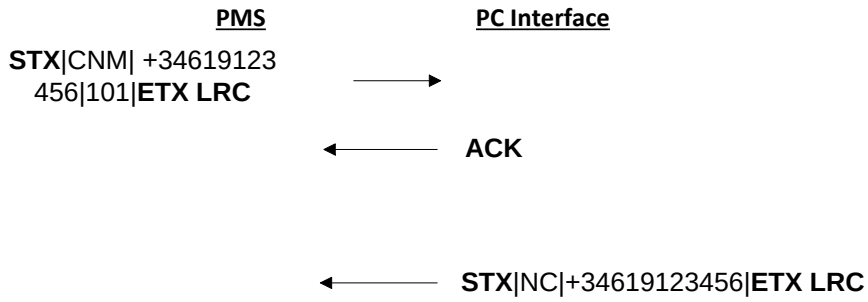
[3] As of this writing, track data (i.e., fields #11, #12 y #13) is not supported yet by the Salto software.

If the request is processed correctly, Salto will return the same 'CNM' message plus the specified telephone number to which card data has been sent, as shown in the example below.

Example 19: In this example, the PMS is asking Salto to send card data to a new guest of room 101 who can be reached in the telephone number +34 619 123 456. Salto responds that the request has been correctly processed (i.e., card data has been correctly sent over the air).

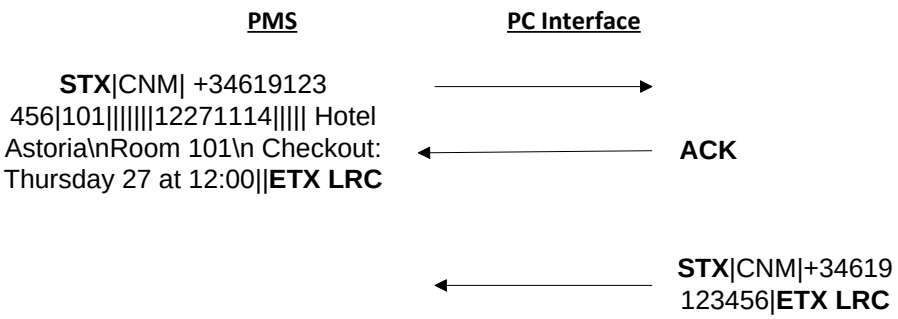


Example 20: This is the same example as above except that Salto responds that an error has occurred (e.g., card data cannot be sent over the air due to connection problems). Note that the error code ("NC", no communication, in the example bellow) is followed be the phone number requested phone number:



As for the display text parameter (field #14) used to show up customised messages on the client's phone screen, bear in mind that the maximum length for the text is 128 characters. Note also that for inserting line breaks (carry return), you should use the '\n' sequence (the escaped character '\' can be printed as follows: '\\').

Example 21: In this example a check-in for room 101 is being performed (expiration is Nov. 27th 2014 at 12:00).



The display on the phone will show the following message:

Hotel Astoria
Room 101
Checkout: Thursday 27 at 12:00

3.6 Command 'CCM': copy guest card for mobile apps

This command is used for sending to a mobile phone (over the air) a copy of permissions for a specified room. The parameters for this command are exactly the same as for 'CNM' (see section 3.3 in page 17).

Note that this command will not work if the specified rooms are checked out or not occupied.

As with the 'CC' or 'CCB' commnads, an error will be produced if the specified rooms are checked out or not occupied.

3.7 Command 'CNMB': get binary data for a new guest card for mobile apps

This command is the binary counterpart of the 'CNM' command (check-in for mobile apps, as explained in section 3.5 in page 28). More specifically, it returns the binary data for a check-in operation to be stored in and used by third-party mobile apps[4].

As shown in the table below, the 'CNMB' command is almost the same as the 'CNM' one (see **Table 15** in page 28), except for the following differences:

- Telephone number (in field #1) is optional. If provided, the Salto software system will produce more accurate and complete reports.
- The customised display messages are not considered in this command (as it was in the 'CNM' command, in field #14). It makes no sense since third-party mobile apps are free to display whatever message is desired.
- As for the rest of the fields, they preserve the same meaning and considerations as in the CNM command.

The table below shows the input parameters for this command:

Field	Description
0	'CNMB'
1	Telephone number of the target smart phone (optional).[5]
2	First room to be opened by the card (main room). Max. 24 characters.
3	Second room to be opened by the card. Max. 24 characters.
4	Third room to be opened by the card. Max. 24 characters.
5	Fourth room to be opened by the card. Max. 24 characters.
6	Authorisations granted to guest.
7	Authorisations denied to guest.
8	Starting date and time of the card.
9	Expiring date and time of the card.
10	Data of the operator who makes the request. Max. 24 characters.
11	Information to be written on track #1.
12	Information to be written on track #2.
13	Information to be written on track #3.

Table 16: input parameters for the command 'CNMB'.

Note that the name of the main room (field #2) is the only parameter that must be provided. The rest may be left blank, in which case default values will be considered.

On processed, the PC interface will return the content of the card in binary format as shown in the table below.

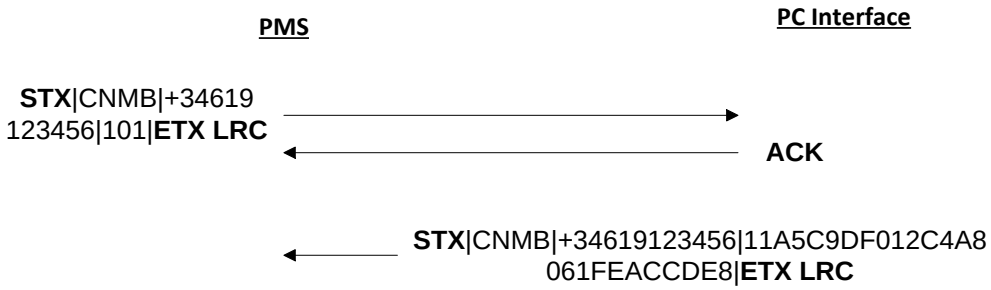
[4]Third-party mobile apps must work in conjuction with an API SDK provided by SALTO.

[5]The telephone number is optional, it is intended for reporting purposes only. Note that, if provided, the it must contain the international prefix. That is, it must start with the '+' symbol and be followed by at least 6 digits. Otherwise, an error will be returned.

Field	Description
0	'CNMB'
1	Telephone number of the target smart phone.
2	Binary content of the card.

Table 17: output parameters for the command 'CNMB'.

Example 22: In the example below the PMS is requesting binary data of a new guest card accessing room 101. The requested data is intended to be used by an app within a mobile phone equipped with BLE.



Note also that the 'CNMB' command is equivalent to 'CNMBX' (see section 3.9 in page 34) when the type of binary data is set to 0. The PMS integrator is free to choose either of them.

3.8 Command 'CCMB': get binary data for a copy guest card for mobile apps

This command is the binary counterpart of the 'CCM' command (copy guest card for mobile apps, as explained in section 3.6 in page 30). More specifically, it returns the binary data of a guest card copy to be stored in and used by third-party mobile apps[6].

The parameters for this command are exactly the same as for 'CNMB' (see section 3.7 in page 31).

As with the 'CC', 'CCB' or 'CCM' commands, an error will be produced if the specified rooms are checked out or not occupied.

Note also that the 'CCMB' command is equivalent to 'CCMBX' (see section 3.10 in page 36) when the type of binary data is set to 0. The PMS integrator is free to choose either of them.

[6]Third-party mobile apps must work in conjuction with an API SDK provided by SALTO.

3.9 Command 'CNMBX': an extended version of 'CNMB'

This command is the same as the 'CNMB' command (used for obtaining card binary data for mobile apps, as explained in section 3.7 in page 31) except that it includes an extra input parameter (field #1) necessary for supporting different types of card binary image.

The table below enumerates the input parameters for this command:

Field	Description
0	'CNMBX'
1	Type of binary image.
2	Telephone number of the target smart phone (optional[7]).
3	First room to be opened by the card (main room). Max. 24 characters.
4	Second room to be opened by the card. Max. 24 characters.
5	Third room to be opened by the card. Max. 24 characters.
6	Fourth room to be opened by the card. Max. 24 characters.
7	Authorisations granted to guest.
8	Authorisations denied to guest.
9	Starting date and time of the card.
10	Expiring date and time of the card.
11	Data of the operator who makes the request. Max. 24 characters.
12	Information to be written on track #1.
13	Information to be written on track #2.
14	Information to be written on track #3.

Table 18: input parameters for the command 'CNMBX'.

Note that only fields #1 (type of card binary image to return) and #3 (name of the main room) are obligatory. The rest may be left blank, in which case default values will be considered.

As of this writing, the following types of card binary image are supported for field #1:

Image type	Description	Salto's app SDK required?
0	Salto format.	Mobile apps should use API SDK provided by Salto.
1	HOTEL1 format. Only valid for a very special type of room lock (used by a specific hotel chain codenamed HOTEL1).	No API SDK from Salto required. (Only valid for apps from a specific hotel chain).

Table 19: supported types of card binary image.

Generally, PMS integrators must use the Salto format (type 0). As for the "HOTEL1" format (type 1), it is only valid for a special type of room lock (used by specific hotel chain).

[7] The telephone number is optional, it is intended for reporting purposes only. Note that, if provided, the telephone number must contain the international prefix. That is, it must start with the '+' symbol and be followed by at least 6 digits. Otherwise, an error will be returned.

Note that the 'CNMB' command also produces a binary image of type 0 (Salto format). That is, both 'CNMB' and 'CNMBX'commands are equivalent when the type field in 'CNMBX' is 0.

Field #2 contains the phone number the binary image is to intended for. It can be safely omitted since it is only intended for reporting purposes.

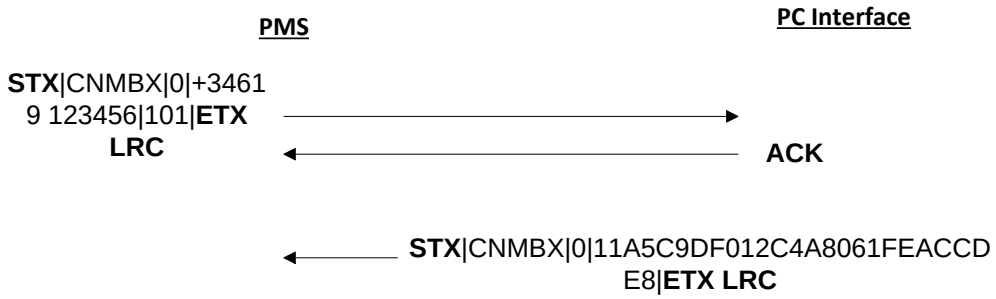
For the meaning of the rest of the fields (from field #3 onward), the reader is referred to section 3.1 in page 10, where a similar command 'CN' is explained.

On processed, the Salto interface will return the binary content of the card in the specified format. The table below enumerates the structure of the response.

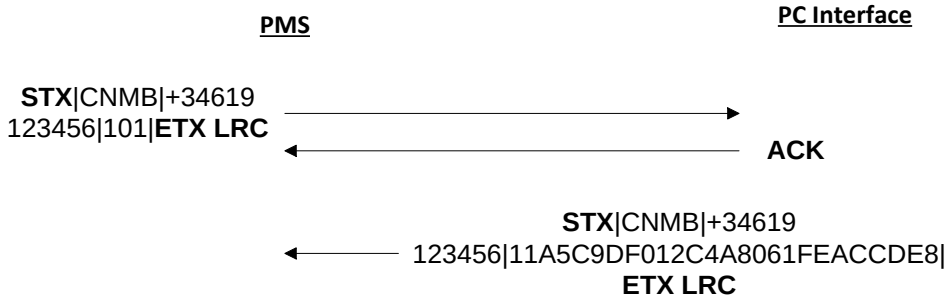
Field	Description
0	'CNMBX'
1	Type of binary image (as requested).
2	Binary content of the card. If the image in Salto format (type=0), then mobile apps must use an API SDK provided by SALTO.

Table 20: output parameters for the command 'CNMBX'.

Example 23: in the example below the PMS is requesting binary data (in Salto format) of a new guest card accessing room 101. The requested data is intended to be used by an app within a mobile phone equipped with BLE.



The same result as above can be obtained through the 'CNMB' command:



3.10 Command 'CCMBX': an extended version of 'CCMB'

The 'CCMBX' command is similar to 'CNMBX' except that the returned card binary image corresponds to a guest card copy (rather than to a new guest card). See previous section for further information about the input and output parameters.

Note that an error will be produced if the specified rooms are checked out or not occupied.

3.11 Special use case: using 'CN'/'CC' commands and encoder '0' for getting binary data for mobile apps

As stated in section 3.1, the 'CN' and 'CC' commands are generally used to encode guest cards with the requested access permissions. However, there exists a special scenario in which this normal behaviour might be altered. This scenario occurs when two conditions are met: a special setting within the Salto software is enabled and the specified encoder name within the 'CN' command is '0' (zero): in this case the 'CN' response will instead contains a binary image of the requested guest card (to be used by mobile apps). The same considerations apply to the 'CC' command.

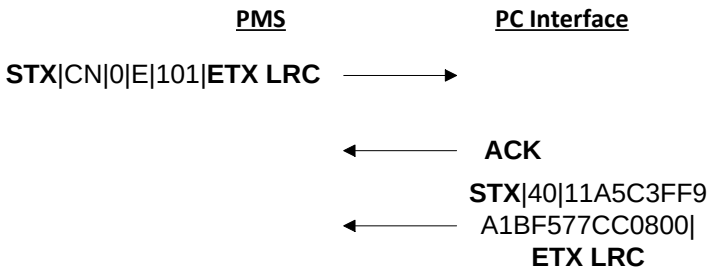
Important: as of this writing, the format of this binary image is only meaningful for a specific hotel chain. Thus, all the PMS integrators (except for this specific hotel chain) should not use the 'CN' for obtaining binary images for mobile apps. They should instead use the CNMB or CNMBX commands (see sections 3.7 or 3.9).

Interistingly, the format of the response in this special escenario is as follows:

SXT|40|*payload of binary image*|**ETX LRC**

Note that the content of the first field (after STX) is not 'CN' (or 'CC') but '40'.

Example 24: in this example, the PMS is asking for the binary image of a new guest card accessing room 101. It is assumed that a special setting within the Salto software is enabled. After positive acknowledgement, the Salto software returns the requested binary image, which is then expected to be sent to the gest's phone by the PMS.



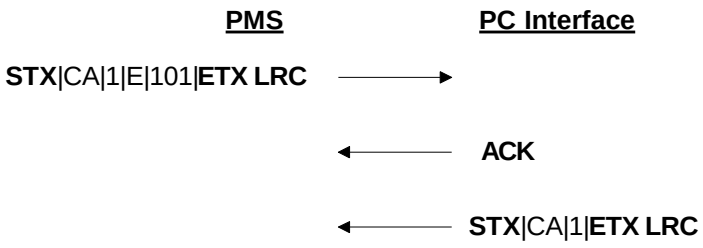
3.12 Command 'CA': single opening card

This command is used to encode a single opening card. The purpose of this kind of cards is to open a specific room just once. After the first opening, the card is not valid any longer. Bear in mind that one shot keys are only valid for one hour.

The parameters for this command are exactly the same as for 'CN', the difference being that the following parameters are not taken into account:

- Starting and expiration date/time (fields #9 and #10): when issuing a one shot key, the starting and expiration date/time are replaced, respectively, by the current time of edition and this same value plus one hour.
- Granted and denied authorizations (fields #7 and #8): these values are not considered.
- Track #1, #2 and #3 (fields #12, #13 and #14, respectively): one shot keys has no track data.

Example 25: The messages below show that the PMS is requesting a single opening card for room 101 to be issued in encoder #1. The PC interface sends an ACK character first and then an answer indicating that the card was correctly encoded. The issued key will be valid for just one hour.



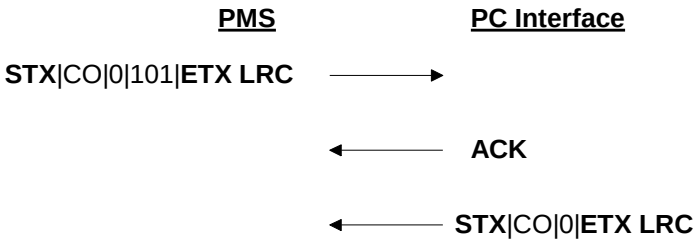
3.13 Command 'CO': checkout

This command performs the checkout of the specified room. There are two parameters involved in this command, though the first one (encoder number) has no effect.

Field	Description
0	'CO'
1	Encoder number. No effect.
2	Room to check-out

Table 21: parameters for command 'CO'.

Example 26: Let's assume that the PMS needs to check out room 101. The message to send should be as follows:



3.14 Command 'MC': modify check-in data

This command is used to modify check-in data of active guests without having to reencode their cards. More specifically, the following types of modification are supported:

- re-assign a new room (a.k.a, re-rooming);
- extend stay, that is, change expiration date/time.

Note that for these changes to take effect, online doors (or, at least, online hotspots) are required.

What follows are the three parameters that may be provided to this command:

Field	Description
0	'MC'
1	Room.
2	New room (optional).
3	New expiry date (optional)

Table 22: parameters for command 'MC'.

The 'room' parameter (field #1) indicates the name of the original room whose check-in data is to be modified. Note that the specified room must be currently occupied, otherwise an error will be produced.

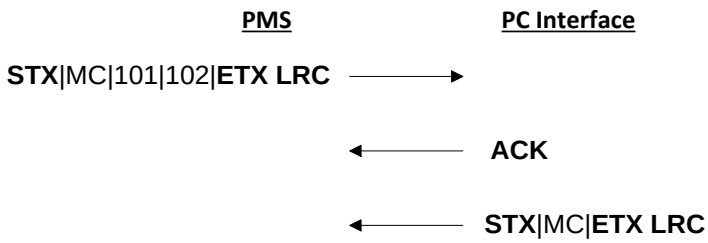
The 'new room' parameter (field #2) indicates the new room to which the guest should be moved. This parameter may be left blank, meaning the no room change is desired.

Finally, the 'new expiry date' parameter (field #3) indicates the new expiration date/time for the specified room guest in "hh[mm]DDMMYY" format where:

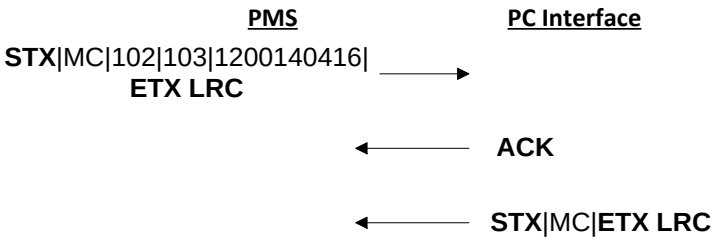
- hh: hour (00 to 23).
- mm: minutes (00 to 59).
- DD: date (01 to 31).
- MM: month (01 to 12).
- YY: year (00 to 99).

If the command is correctly processed, the response frame returned by the Salto server will just contain the name of the command, i.e., 'MC'. Otherwise an error code will be returned.

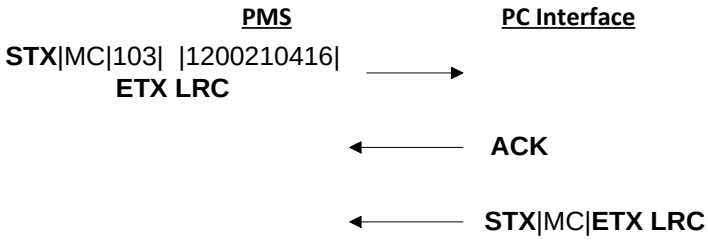
Example 27: in this example, the PMS asks the Salto server that room guest 101 should be moved to 102:



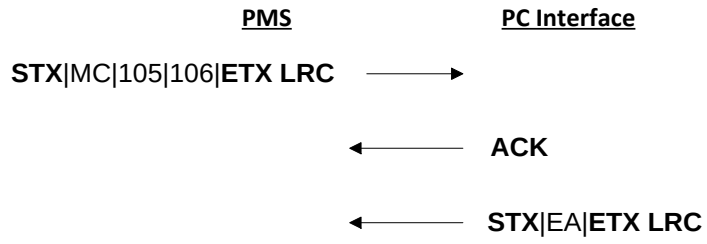
Example 28: in this example, the PMS wants the Salto server (1) to conduct a room-move from 102 to 103 and, at the same time, (2) to extend her stay at the hotel until April 14th 2016 (at 12:00):



Example 29: in this example, the PMS just requests to extend the stay of room guest 103 until April 21th 2016:



Example 30: let's assume that room 105 is not occupied. If the PMS requests to move guest from 105 to 106 an error code ('EA') will be returned:



Important note: in case the lock of the target room is online wireless, this command will take very long time to be processed (in the worst case up to 2 minutes) due to the high latency of the RF channel. Therefore, once the request is sent to the Salto server, the PMS client must be patient and must allow a response time of up to 2 minutes.

3.15 Command 'LT': read the content of a card

This command is used to read the content of a card. Two parameters are required, as shown in the following table:

Field	Description
0	'LT'
1	Encoder/reader number.
2	For motorised encoders, this field indicates whether encoder card must be retained or ejected from the encoder. Otherwise, whether the PC interface waits for the card to be removed from the encoder: 'E' → Ejection. The PC interface waits for the key to be removed from the encoder. 'R' → Retention. The PC interface does not wait for the key to be removed. 'T' → Ejection by the rear side. Same as 'E'.

Table 23: parameters for command 'LT'.

The PC interface answer will depend on the card being read. Moreover, there are four kinds of response depending on the card:

- The card belongs to a guest: the answer will include information about rooms being accessed, starting and expiration dates, authorisations granted, etc. (See Table 24).
- The card is not a guest card but a special one, such as spare card, staff card, etc. In this case the PC interface answer will only include the type of the card read. (See Table 25).
- The card belongs to another property: the PC interface cannot interpret the content.
- Error occurs while reading the card: badly inserted, damaged card, etc. See Table 30 for further possible errors.

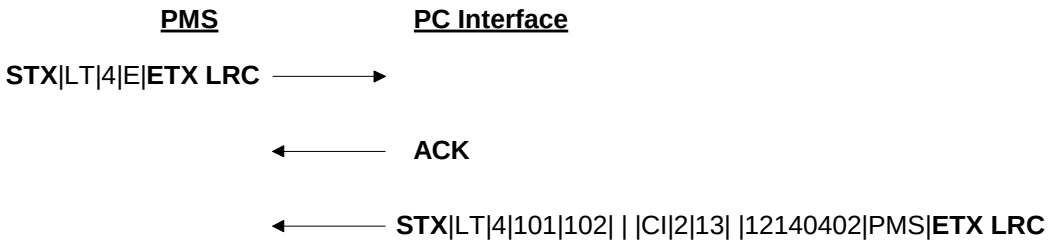
Table 24 shows the content of the message answered by the PC interface after reading a guest card.

Field	Description
0	'LT'
1	Encoder/reader number.
2	Room number 1 (main room) opened by the card. This field should never be empty.
3	Room number 2 opened by the card.
4	Room number 3 opened by the card.
5	Room number 4 opened by the card.
6	Two possible values in this field: 'CI' → card is still valid for main room. 'CO' → card is not valid for main room since it is checked-out.
7	Card's copy number. Possible values in this fields: '0' → original card. '1' → first copy. '2' → second copy. 'I' → undefined copy (third and successive). 'A' → one-shot key.
8	Granted authorisations. This field can be empty.
9	Starting date of the card. This field can be empty.

10	Expiration date of the card. This field can be empty.
11	Name of the operator who issued the card. This field can be empty.

Table 24: answer message when reading a guest card using command 'LT'.

Example 31: In this example, the PMS asks the PC interface to read a card in encoder #4. The answer of the PC interface includes the content of the card as follows:



The length of the answer given by the PC interface clearly indicates that the card belongs to a guest. The actual fields contain …

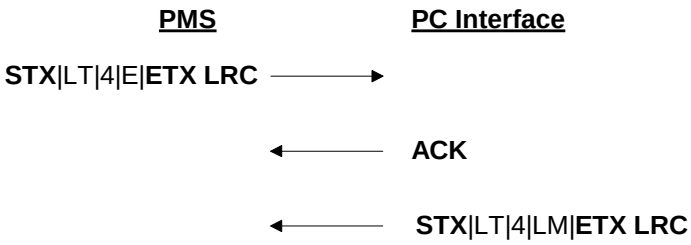
LT:	card correctly read.
4:	card read in encoder #4.
101:	guest card belongs to room 101 (main room).
102:	card can also open room 102
-	Empty field.
-	Empty field.
CI:	card is still valid for room 101 (not checked out).
2:	this card is the second copy issued for main room (101).
13:	Authorisations number 1 and 3 granted to guest.
-	Empty field. No starting date.
12140402:	Expiration date (12:00, 14th of April 2002).
PMS:	Issued by the PMS.

On the other hand, Table 25 shows the answer given by the PC interface when a non-guest card (such as staff card) is inserted.

Field	Description
0	'LT'
1	Encoder/reader number.
2	Card type. Possible values in this field are: 'LM' → staff card or special card (such as programming card, cancelling card, etc.) 'LR' → spare card for guests. 'LC' → guest card not valid. 'LD' → unidentified card (it belongs to another system or erased from the locking plan).

Table 25: answer message when reading a non-guest card using command 'LT'.

Example 32: Let's assume that the PMS requests to read a card and a staff card is inserted in the specified encoder. The messages look like as follows:



The answered message means:

LT:	card read without errors.
4:	card read in encoder #4.
LM:	Staff card or special card.

3.16 Command 'P1': write information on track #1

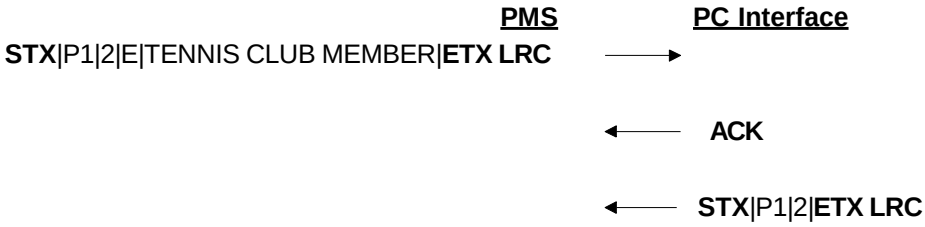
When issuing a card, the PMS can ask the PC interface to write additional information on track 1. Other devices can use this information appropriately. The information must be comprised of readable ASCII characters (excluding the field separator character, '|').

In this command, 3 parameters are involved:

Field	Description
0	'P1'
1	Encoder number.
2	For motorised encoders, this field indicates whether encoder card must be retained or ejected from the encoder. Otherwise, whether the PC interface waits for the card to be removed from the encoder: 'E' → Ejection. The PC interface waits for the key to be removed from the encoder. 'R' → Retention. The PC interface does not wait for the key to be removed. 'T' → Ejection by the rear side. Same as 'E'.
3	ASCII message to be encoded.

Table 26: parameters for command 'P1', 'P2' and 'P3'.

Example 33: In the following communication frames, the PMS is asking the PC interface to encode message '*TENNIS CLUB MEMBER'* on track 1:



3.17 Command 'P2': write information on track #2

This command is the same as 'P1' the difference being that information is written on track 2 rather than track 1.

3.18 Command 'P3': write information on track #3

This command is the same as 'P1' the difference being that information is written on track 3 rather than track 1.

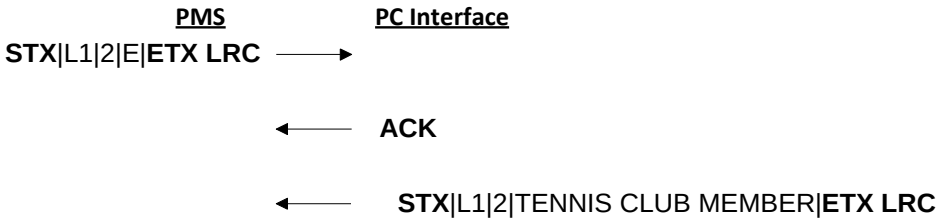
3.19 Command 'L1': read information from track #1

This is the inverse command for 'P1', that is, the PC interface reads information from track 1. Two parameters are needed in this command:

Field	Description
0	'L1'
1	Encoder number.
2	For motorised encoders, this field indicates whether encoder card must be retained or ejected from the encoder. Otherwise, whether the PC interface waits for the card to be removed from the encoder: 'E' → Ejection. The PC interface waits for the key to be removed from the encoder. 'R' → Retention. The PC interface does not wait for the key to be removed. 'T' → Ejection by the rear side. Same as 'E'.

Table 27: parameters for the commands 'L1', 'L2' and 'L3'.

Example 34: In this example, the PMS is asking the PC interface to read track 1 information from a card inserted in encoder number 2. After the acknowledgement answer, the PC interface sends back the content of track 1.



3.20 Command 'L2': read information from track #2

This command is the same as 'L1' the difference being that information is read from track 2 rather than track 1.

3.21 Command 'L3': read information from track #3

This command is the same as 'L1' the difference being that information is read from track 3 rather than track 1.

3.22 Command 'CP': cancel lost key

The 'CP' command can be used for cancelling a guest key, making the key be invalid in all the doors in the property. You should ONLY use this command when all the following conditions are met:

- The guest has lost his key or has left the hotel several dates before the expected leaving date.
- The expiration of the lost key is to occur within several dates.
- The lost key opens other doors besides its own room.

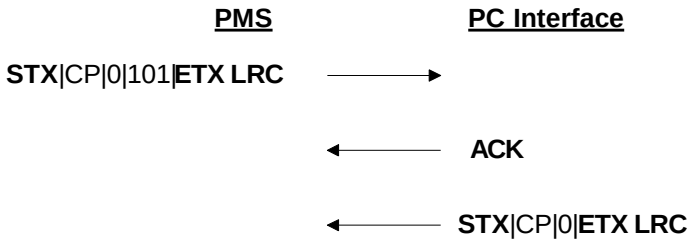
Note that you should not use this command as a substitute of the checkout ('CO') command.

The parameters for this command are as follows:

Field	Description
0	'CP'
1	Encoder number. No effect.
2	Number of the room to which the lost key belonged.

Table 28: parameters for command 'CP'.

Example 35: Let's assume that a guest using room 101 has lost his key. The message to send should be as follows:



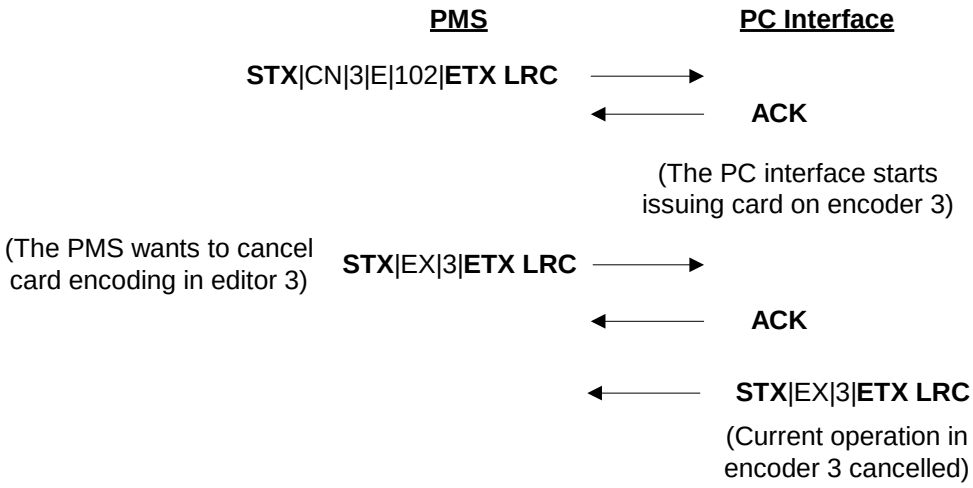
3.23 Command 'EX': abort task

This command aborts the current operation being executed in the specified encoder.

Field	Description
0	'EX'
1	Encoder number.

Table 29: parameters for command 'EX'.

Example 36: Let's assume that the PMS asks for a new guest card to be issued in encoder 3. The PC interface will answer with an ACK and start executing the request on encoder 3. However, for some reason (room number mistakenly assigned, etc.), the PMS wants to immediately cancel this request. The following example illustrates this situation:



4. Error messages

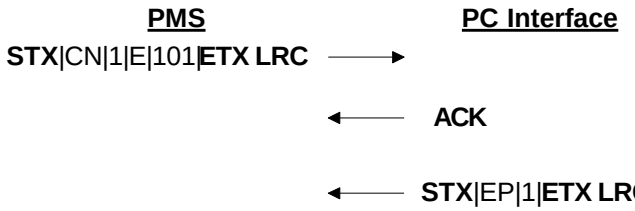
Sometimes, the operation required by the PMS cannot be carried out for different reasons: card badly inserted, encoder is occupied, etc. In this case, the PC interface will respond with a message containing, at least, an error command and the encoder number in which the operation failed.

The table below summarises the possible error messages generated by the PC interface.

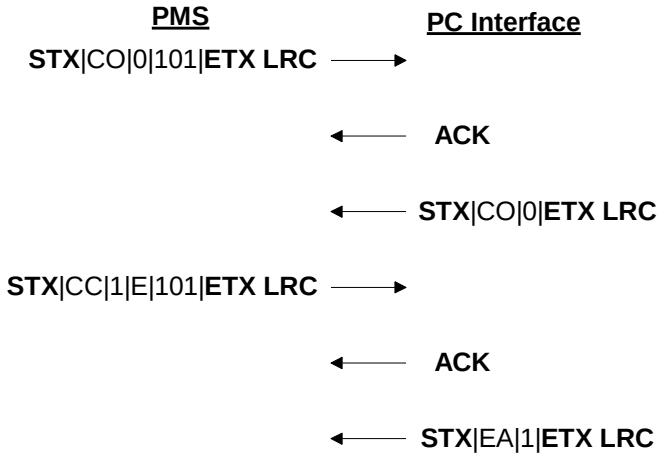
Error	Description
'ES'	Syntax error. The received message from the PMS is not correct (unknown command, nonsense parameters, prohibited characters, etc.)
'NC'	No communication. The specified encoder does not answer (encoder is switched off, disconnected from the PC interface, etc.)
'NF'	No files. Database file in the PC interface is damaged, corrupted or not found.
'OV'	Overflow. The encoder is still busy executing a previous task and cannot accept a new one.
'EP'	Card error. Card not found or wrongly inserted in the encoder.
'EF'	Format error. The card has been encoded by another system or may be damaged.
'TD'	Unknown room. This error occurs when trying to encode a card for a non-existing room.
'ED'	Timeout error. The encoder has been waiting too long for a card to be inserted. The operation is cancelled.
'EA'	This error occurs when the PC interface cannot execute the 'CC' command (encode copies of a guest card) because the room is checked out.
'OS'	This error occurs when the requested room is out of service.
'EO'	The requested guest card is being encoded by another station.
'EV'	Card validity error. This error occurs when the inserted card for a 'CN', 'CC' or 'CA' command belongs to a valid staff user.
'ER'	General error. When the resulting error is none of the above described, the PC interface returns an 'ER' followed by an encoder number, an error code and an error description.

Table 30: possible error messages.

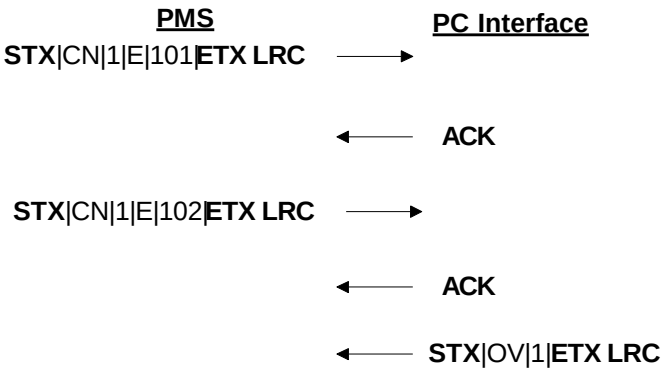
Example 37: In this example, the PMS asks for a guest card to be issued in encoder #1. The operator wrongly inserts the card and, therefore, the PC interface returns a card error.



Example 38: In this example, the PMS makes two requests: the first one is a room checkout, which is successfully carried out by the PC interface. However, as for the second request (encode a guest copy card), an error occurs for it is not allowed to issue a guest copy card for a room with no active guest (checked-out). The PMS must first send a new guest card command ('CN') before using 'CC'.



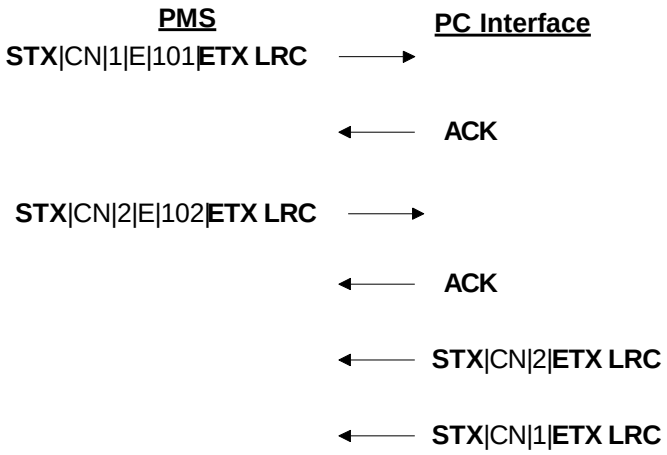
Example 39: The PMS asks the PC interface to issue a guest card for room 101 in encoder #1. However, while the requested card is being encoded, the PMS asks for another card to be issued in the same encoder. The PC interface returns an overflow error indicating that the requested second card cannot be issued because encoder #1 is still occupied.



5. Tasks simultaneity in different encoders

It is possible for the PMS to request a new operation without waiting for the previous one to finish, provided that the new operation is to be performed in a different encoder. The following example illustrates this situation:

Example 40: In this example, the PMS asks the PC interface to issue two guest cards at the same time: one card for room 101 using encoder #1 and another card for room 102 using encoder #2. Note that the operation in encoder #2 finished sooner than the one in encoder #1.



This example also shows the fact that the order in which operations are terminated does not necessarily correspond to the order in which requests were made. In other words, the PC interface will transmit the operation result as soon as it is finished.

Obviously, if the PMS wants two different cards to be issued in the same encoder, it must wait for the first operation to be finished before sending the second request. Otherwise, an overflow error is raised, as shown in example 26.

As stated before, the PMS can cancel an operation being performed in a specific encoder by means of the 'EX' command (see example 23).

6. Retrieving peripheral and door audit trail

6.1 Commands for audit trail requests: 'WF', 'WN' and 'WR'

The DB of the interface application stores the audit trail (openings and rejections) of all the doors and peripherals (wall readers, etc.) installed in the property. By means of the following commands, it is possible for the PMS to collect the audit trail events of a particular door or peripheral:

Command	Description
'WF'	The PMS requests the interface to send the *oldest* audit trail event (opening or rejection) of a particular peripheral or door.
'WN'	The PMS requests the interface to send the next audit trail event (with respect to the previous requested event) of a particular peripheral/door.
'WR'	The PMS requests the interface to resend the last requested audit trail event.

Table 31: PMS commands for retrieving audit trail incidences.

All these commands require just one parameter, namely, the name of the door or peripheral from which the requested audit trail event is obtained, as shown in the table below:

Field	Description
0	'WF', 'WN' or 'WR'
1	Door or peripheral identification.

Table 32: parameters for requesting audit trail events (commands 'WF', 'WN' or 'WR').

The syntax of the response from the interface will be as follows:

Field	Description
0	'WF', 'WN' or 'WR'
1	Door or peripheral identification.
2	Date of the opening (or rejection). 5 characters. Format: 'day/month' or 'month/day', depending on the configuration of the interface.
3	Time of the opening (or rejection). 5 characters. Format: 'hours:minutes', from 00:00 to 23:59.
4	Type of incidence (1 character): it contains a code number indicating whether the incidence is either an opening or a rejection and its cause. Type = 0: the incidence corresponds to an opening. Type = 2: the incidence corresponds to a rejection, the cause being that the card is invalid (expired, checked out, cancelled, etc.) Type = 3: the incidence corresponds to a rejection, the cause being that the card has not been granted the permission of accessing the specified door or peripheral. Type = 4: the incidence corresponds to a rejection, the cause being that the card is out of time. Type = 5: the incidence corresponds to a rejection due to antipassback.
5	The direction of the opening/rejection (1 character): -'I': input or entrance reader. -'O': output or exit reader.
6	Card identification (8 characters minimum).

	The content of this field depends on the type of the card owner: -Hotel guest: if the card corresponds to a hotel guest, this field will contain the name of the room to which the guest belongs. -Staff: if the card corresponds to a staff user, this field will contain the word 'STAFF ' (8 characters) and field #8 will contain the name of the user (see below). -Special users: for other kind of users (such as spare card) this field is left empty (8 blank characters).
7	Copy number of the card (2 characters): -'#0': original card. -'#1': first copy. -'#2': second copy. -'#D': indefinite copy (third or successive). -'@1': single opening card number 1 (one-shot key). -'S1': spare card. -'S2': opening caused by means of a switch, button, keypad, etc. -'S3': opening caused online from the computer.
8	Name of a staff user (20 characters minimum).

Table 33: syntax of the frame returning audit trail information.

6.2 Card types

Note that the interface distinguishes 3 kinds of cards, depending on the holder: hotel guests, staff users and special cards (spare cards). The answer from the interface will vary according to the card type:

- Hotel guests: the field #6 (card identification) will contain the name of the room the current guest belongs to. The field #8 (name of the user) will be empty (20 blank characters).
- Staff user (such as master cards, maintenance, cleaners and, in general, non-hotel guests): the field #6 (card identification) will contain the word 'STAFF'. Additionally, field #8 (name of the user) will include the actual name of the user (if the length of the name is shorter than 20 characters, blank characters are added).
- Special cards: both fields #6 and #8 are left empty (filled with blank characters). If the incidence is caused by a spare card, the field #7 (copy number) will contain 'S1'.

6.3 Procedure for collecting audit trail

Given a peripheral or door, the PMS should consider its audit trail as being a *pool of incidences* (openings and rejections) sorted by date/time. There is one pool per door and per peripheral. This pool is date-limited, that is, only the most recent incidences are kept in the pool whereas the old ones are removed. By default, the valid day range in the pool is 7 days (this is a default range value that can be shortened or enlarged as desired in the interface configuration, see appendix). In other words, the oldest incidence in the pool will have occurred 7 days ago. This means that, as days go by, some of the incidences will become too old and will be 'dropped' from the pool: in order to not 'loose' any incidence, the PMS should collect the audit trail of a given peripheral on a regular basis (once a week at least).

From the PMS side, the procedure for obtaining the entire audit trail of a given door or peripheral should be as follows:

- If the PMS has not got any previous incidence collected from a specific door (i. e., it is the first time), the 'WF' command must be used to make the interface point to the oldest incidence.
- Subsequent requests should include the 'WN' command, meaning that the PMS is asking for the next incidence with respect to the previously obtained one. The PMS should keep asking for the next incidence ('WN') until it receives an overflow error message ('WO', see below), meaning that there is no more incidences to be sent.
- If, by any chance, a given requested incidence is not properly received by the PMS (noisy channel, temporary disconnection, etc.), the 'WR' command can be used for the interface to repeat the last incidence.

6.4 Error messages when collecting audit trail

There are two possible error messages when requesting incidences from the PMS, as shown in the following table:

Command	Description
'WE'	General error: it is not possible to send the requested incidence. Possible causes could be: - specified door or peripheral does not exist. - database not accessible. - etc.
'WO'	Overflow error: there is no more incidence to be sent. Depending on the command, the meaning of this error is as follows: -'WF': the audit trail of the specified door or peripheral is empty, i. e., no opening or rejection has been produced in the last days. -'WN': the last incidence in the audit trail has already be sent to the PMS, so no more incidence is available. -'WR': this case only occurs when the 'WR' command has been sent before any actual incidence collection request ('WF' or 'WN'): the interface cannot repeat anything since not previous request has been made.

Table 34: possible error messages for the commands 'WF', 'WN' or 'WR'.

6.5 Examples of audit trail collection

The following two examples illustrate the frames originated between both the PMS and interface when retrieving incidences.

Example 41: For an illustrative example, let's make the following assumptions:

1) Today is 8th of October 2001. Besides, the day range parameter has been set to 7 days in the interface configuration. This means that any incidence occurring before 1/10/2001 will not be considered.

2) Door named 'Entrance' has got the following four incidences (note that the oldest incidence is out of day range):

Date/Time	Door	User	Type	Direction
15/9/2001, 12:30:50	Entrance	Cleaner	Rejection (no permission granted)	Input
1/10/2001, 14:05:10	Entrance	101	Opening	Input
1/10/2001, 15:47:23	Entrance	Maintenance	Opening	Output
2/10/2001, 16:32:09	Entrance	Cleaner	Rejection (no permission granted)	Input
2/10/2001, 19:01:47	Entrance	One shot key	Opening	Input

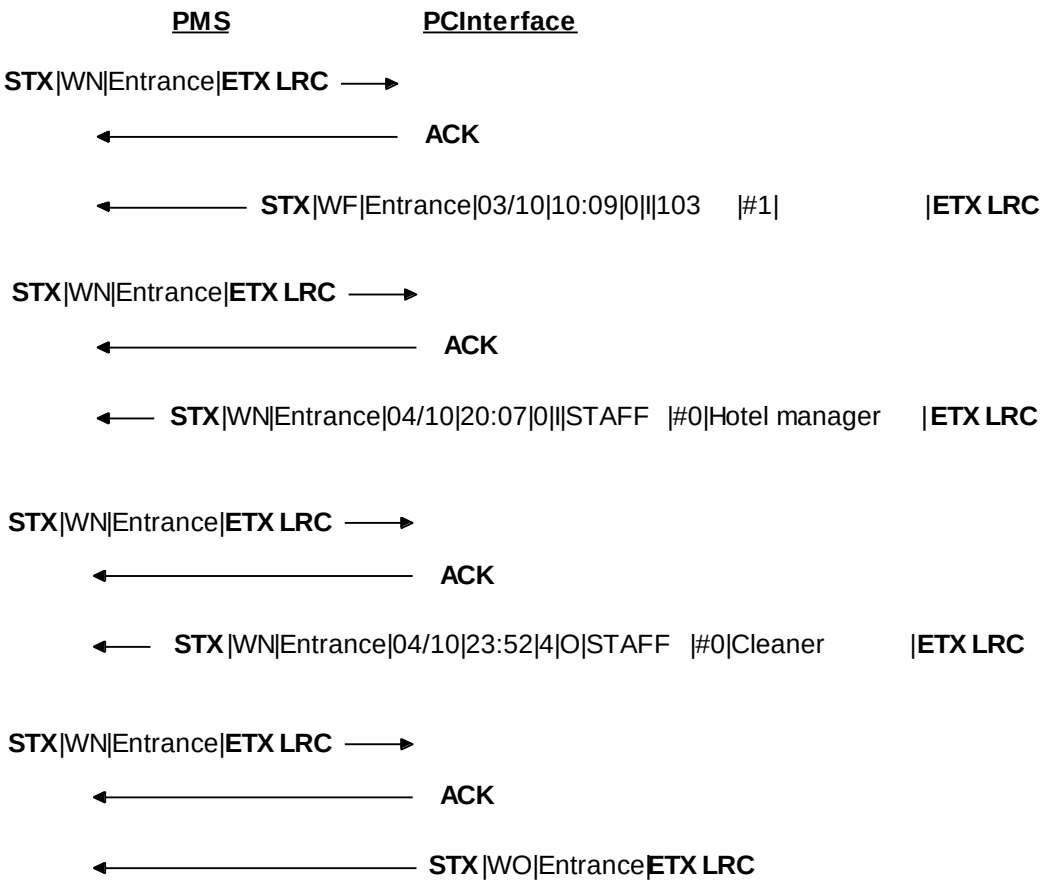
3) Finally, the PMS has got no audit trail information at all about the door 'Entrance' (because it is the first time or previous information has been lost).

The frame sequence between the PMS and the interface will look like as follows:

Example 42: Let's suppose now that, few days later, there are new incidences from the same door to be collected by the PMS:

Date/Time	Door	User	Type	Direction
3/10/2001, 10:09:04	Entrance	103 (first card copy)	Opening	Input
4/10/2001, 20:07:53	Entrance	Hotel manager	Opening	Input
4/10/2001, 23:52:49	Entrance	Cleaner	Rejection (out of time)	Output

The frame sequence between the PMS and the interface will look like as follows:


PMS
PCInterface
STX|WN|Entrance|ETX LRC
ACK
STX|WF|Entrance|03/10|10:09|0|I|103 |#1| |ETX LRC
STX|WN|Entrance|ETX LRC
ACK
STX|WN|Entrance|04/10|20:07|0|I|STAFF |#0|Hotel manager |ETX LRC
STX|WN|Entrance|ETX LRC
ACK
STX|WN|Entrance|04/10|23:52|4|O|STAFF |#0|Cleaner |ETX LRC
STX|WN|Entrance|ETX LRC
ACK
STX|WO|Entrance|ETX LRC

Appendix A. SALTO software configuration

There are several configurable parameters in the SALTO software that affect the protocol performance:

Parameter	Description
PMS_MONTH_DAY_FORMAT	This parameter indicates the date format to be used in the audit trail collection commands ('WF', 'WN' and 'WR'): '1': USA format, i.e., Month/Day. '0': standard format, i.e., Day/Month. Default value: '0'.
PMS_AUDIT_EVENTS_PERIOD	This parameter indicates the size (in terms of days) of the audit trail for each door or peripheral. Incidences older that the specified value will be removed from the audit trail. Default value: 7 days

Table 35: configurable parameters in the SALTO software.

Appendix B. ANSI characters for representing authorisations

The following table shows the supported ANSI characters for representing authorisation numbers within the check-in commands **CN** and **CC**.

Dec	Hex	ANSI Character	Authorisation Number (Salto DB)
49	31	1	1
50	32	2	2
51	33	3	3
52	34	4	4
53	35	5	5
54	36	6	6
55	37	7	7
56	38	8	8
57	39	9	9
65	41	A	10
66	42	B	11
67	43	C	12
68	44	D	13
69	45	E	14
70	46	F	15
71	47	G	16
72	48	H	17
73	49	I	18
74	4A	J	19
75	4B	K	20
76	4C	L	21
77	4D	M	22
78	4E	N	23
79	4F	O	24
80	50	P	25
81	51	Q	26

82	52	R	27
83	53	S	28
84	54	T	29
85	55	U	30
86	56	V	31
87	57	W	32
88	58	X	33
89	59	Y	34
90	5A	Z	35
97	61	a	10
98	62	b	11
99	63	c	12
100	64	d	13
101	65	e	14
102	66	f	15
103	67	g	16
104	68	h	17
105	69	i	18
106	6A	j	19
107	6B	k	20
108	6C	l	21
109	6D	m	22
110	6E	n	23
111	6F	o	24
112	70	p	25
113	71	q	26
114	72	r	27
115	73	s	28
116	74	t	29
117	75	u	30
118	76	v	31
119	77	w	32
120	78	x	33

121	79	y	34
122	7A	z	35
33	21	!	36
35	23	#	37
36	24	$	38
37	25	%	39
38	26	&	40
40	28	(	41
41	29	)	42
42	2A	*	43
43	2B	+	44
44	2C	,	45
45	2D	-	46
46	2E	.	47
47	2F	/	48
58	3A	:	49
59	3B	;	50
60	3C	<	51
61	3D	=	52
62	3E	>	53
63	3F	?	54
64	40	@	55
91	5B	[	56
92	5C	\	57
93	5D	]	58
94	5E	^	59
95	5F	_	60
123	7B	{	61
125	7D	}	62