GEMCLUB COMMAND FORMAT

Overview

This chapter discusses the **Application Protocol Data Unit** (**APDU**) message structure used by GemClub.

GemClub cards accept commands and responses in compliance with the Application Data Protocol (APDU) format defined by the ISO 7816-4 standard. This enables GemClub commands to be compatible with the Gemplus GCR Interface Driver Library. You should bear in mind, however that the GemClub transport layer protocol complies with the ISO 7816-3 T=0 standard, which converts APDUs into Transport Protocol Data Units (TPDU)s. The reader sends TPDU commands to the card, and the card returns TPDU responses to the reader.

GemClub cards accept commands in any of the following cases:

Case 1-No command or response data. This is sent as a T=0 ISO IN TPDU with length = 0.

Case 2-Short format: no command data, but with between 1 and 256 bytes response data. This is sent as a T=0 ISO OUT TPDU.

Case 3-Short format: command data between 1 and 255 bytes and no response data. This is sent as a T=0 ISO IN TPDU.

Case 4-Short format: command data between 1 and 255 bytes, response data between 1 and 256 bytes. The command is sent as a T=0 ISO IN TPDU. It must be followed by a **Get Response** command sent as a T=0 ISO OUT TPDU. The **Get Response** mechanism is compliant with the ISO 7816-4 standard.

Command Format

In their default configuration (T=0), GemClub cards accept commands in the following format:

Header	**Body**		
CLA INS P1 P2	Lc	Parameters/data	Le

The fields are described below:

Header Fields

Header fields are mandatory and are used as follows:

Field	**Length**	**What it is**
CLA	**1**	Class of Command—type of command
INS	**1**	Command Code—code that has been specified for the command
P1	**1**	Command Parameter
P2	**1**	Command Parameter

Body Fields

The command body is optional. It includes the following fields:

Field	Length	What it is
Lc	**1**	Length of the Subsequent Data Field—the length of the data to follow
Data		Command Parameters or Data
Le	**1**	Expected Length of Data to be returned

Response Format

GemClub cards return responses to commands in the following format:

Body	Trailer
Data	SW1, SW2

The Body field is optional and holds the data returned by the card.

The Trailer field includes the following two mandatory bytes:

SW1: Status byte 1 that returns the command processing status

SW2: Status byte 2 that returns the command processing qualification

Commands Quick Reference Guide

The following commands are used by GemClub to ensure that it operates correctly at your site:

Application Commands

GemClub **Application commands** are as follows:

Command	CLA	INS	P1	P2	Lc	Le	Page
Append Record		E2h	00h	SFI	var		46
Without Secure Messaging	00h						
With Secure Messaging	04h					0Ah	
Award	80h	4Eh	01h	SFI	16h	15h	49
Get Response	00h	C0h	00h	00h	—	var	58
Read Parameter	80h	BEh	elm	SFI	08h	var	60
Read Record	00h	B2h	rec	SFI	—	var	63
Redeem	80h	4Eh	02h	SFI	16h	15h	58
Select AF by Name	00h	A4h	04h	00h	var	var	68
Update Parameter	80h	DEh	elm	SFI	var	0Ah	73
Update Record—		DCh	rec	SFI	var		76
Without Secure Messaging	00h					—	
With Secure Messaging	04h					0Ah	
Use Rule	80h	4Eh	03h	SFI	16h	14h	79
Verify Secret Code	00h	20h	00h	SFI		—	82
Compare Secret Code					08h		
Read further allowed entries							

GemClub Administrative Commands

GemClub **Administrative commands** are as follows:

Command	CLA	INS	P1	P2	Lc	Le	Page
Create Object	80h	EEh	type	SFI	var	0Ah	52
Delete Object	80h	CEh	type	SFI	10h	0Ah	56
Select Communication Speed	80h	14h	speed	cegt	—	—	71

Note: *All commands where CLA=00h are compatible with ISO 7816-4 standards.*

Command Verification

GemClub automatically carries out a number of verification actions prior to running a command. These verifications are as follows:

Syntax Verification

Prior to running a command, the GemClub operating system carries out the following procedures:

Check	Return Code if command fails at this stage
CLA is allowed.	6E00h
INS belongs to the set of commands	6D00h
INS matches with **CLA**	6D00h
Syntax of Parameters **P1**—**P2**	6A00h
Incoming data length—**Lc**	6700h

Note: CLA = FFh is reserved for Protocol Type Selection which is not supported by GemClub and will thus be rejected. This is the only check performed on the CLA field.

Access Condition Checking

GemClub performs the following access condition checks for Gemplus Proprietary Commands:

Check	Return Code if command fails at this stage
Access Condition is not '**Locked**'—For '**Read**' operations, '**Secure Messaging**' is interpreted as being '**Locked**.'	6982h
Presence and validity of Relevant **Secret Code** or **Secret Key**.	6400h
Relevant **Secret Code** or **Secret Key** is not blocked.	6983h
Decrease in the number of allowed false presentations—EEPROM.	6581h
Comparison between the **Secret Code** or the **MAC**.	6982h
Ratification counter is **Reset**.	6581h

Append Record

This command is used to create a new record in an ISO 7816-4 record file and then initialize the content of the new record. For more information, see '*Record Files*'.

For information on the MACs exchanged in the **Append Record** command, see '*Cryptography and Commands in GemClub*'.

Note: *If secure messaging is not used, the record length can be in the range of 1 255.*

For secure messaging, all of the data field shall be received in the card buffer at the same time during MAC verification. Accordingly when using secure messaging only records which are no more than 24-bytes in size can be appended.

Command Format

The **Command** is structured as follows:

Field	Description Value
CLA	00h—Without secure messaging. 04h—With secure messaging.
INS	E2h.
P1	00h.
P2	SFI—Coded over bits b_3-b_7 as follows: xxxxx000b (1 ≤ SFI ≤ 30). Other values are RFU.
Lc field	**Record length**—Without Secure Messaging. **Eight bytes plus Record length** With Secure Messaging.
Record Field	**Data to be updated**—Record length. **Incoming MAC**—Eight bytes, where secure messaging has been specified.
Le field	**0Ah** Only where secure messaging is specified.

Response

The **Response** is structured as follows:

Field	Description
Data	**Card Transaction Counter** (only where secure messaging is specified) Two bytes. **Outgoing MAC** (only where secure messaging is specified) Eight Bytes.
SW1 SW2	Status bytes.

Command Processing

Before running an **Append Record** command, the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	**Return Code if Command fails at this stage**
Verify the syntax of command is correct. See '*Syntax Verification*'.	—
Verify the length of the incoming piece of data to ensure that it conforms with secure messaging requirements—8<'Lc' ≤ 32.	6700h
Verify that system file is present and valid.	6400h
Verify CTC<FFFFh	6983h
Increase CTC—EEPROM by one unit.	6581h
Verify that referenced file is present.	6A82h
Verify referenced file integrity.	6400h
Verify that access condition for **Append Record**. See '*Access Condition Checking*'.	—
Verify that length of the incoming **Lc** data is in accordance with the record length.	6A84h
Create the record (**EEPROM**)	6581h
Normal ending of the command—Where secure messaging has been specified.	61h'La'
Normal ending of the command—Where secure messaging has not been specified.	9000h

Context Modification after Execution

During Context Modification after running the command, the global security status is not affected.

Status Codes

The **Status Codes** which can be returned by this command are:

SW1	SW2	Description
61h	'La'	'La' bytes available for '**Get Response**.'
64h	00h	No precise diagnostic.
65h	81h	Memory failure.
67h	00h	Wrong length.
69h	82h	Security status not satisfied.
69h	83h	Authentication method blocked.
69h	85h	Conditions of use not satisfied.
6Ah	82h	File not found.
6Ah	86h	Incorrect parameters P1-P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if:

- One of the data elements needed to perform the **Award** command is absent or invalid.
- The access condition for **Award** is not fulfilled.

The command is aborted if:

- The current date does not verify the following condition:

  Award start date ≤ current date ≤ Award end date.

- One of the following equations is not verified (overflow):

  Balance + Transaction Amount ≤ 16,777,215

  Cumulative balance + Amount ≤ 16,777,215

  Visit Counter+1 ≤ 65535

Note: If the validity dates for ***Award*** *are not present in the counter file, then the date verification will be by-passed.*

If either the cumulative balance or the visit counter is not present in the counter file, then the relevant verification will be bypassed.

A ***Get Response*** *command is required to validate the transaction inside the card. Thus if a command other than* ***Get Response*** *is received after an* ***Award****, EEPROM data will be restored. See* GemClub Backup Procedure**.**

If no ***Transaction Proof*** *key file is referenced in the counter file, then a data package consisting of zeroes will be returned instead.*

Create Object

This command is used to create a new file in the GemClub card application memory. During this procedure memory space is allocated and a new file is created in EEPROM.

For information on the MACs exchanged in the **Create Object** command, see '*Cryptography and Commands in GemClub*'.

Command Format

The **Command** is structured as follows:

Field	**Description Value**
CLA	80h
INS	EEh
P1	Data object type: 01h—System File. 02h—ISO 7816-4 Record File. 03h—Counter File. 04h—Rule File. 05h—Secret Code File. 06h—Secret Key File.
P2	SFI—(01–1Eh).
Lc field	14h
Data field	**Terminal Data**—Eight bytes. **Data Object Attributes**—Four bytes. **Incoming MAC** or **Secret code** Eight Bytes.
Le field	0Ah

Data Object Attributes

Data objects have the following attributes in GemClub:

Data Object Type	**Attributes**
System File	Four bytes reserved for future use.
Record File	**File descriptor byte**—one byte—coded in accordance with ISO 7816-4 standards. It indicates the structure of the file. 04h and 05h are the values supported by GemClub. Other values are reserved for future use. **File Size**—two bytes—unsigned value indicating the size occupied by the entire data file. The file size includes eight bytes of o/s internal information. It also includes one byte per record reserved for o/s internal information. The file size also takes account of the fact that the address of a record always starts on a multiple of four. Two bytes are reserved for future use.

SW1	SW2	Description
6Ah	80h	Incorrect parameters in the data field.
6Ah	86h	Incorrect parameters P1-P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if one of the data elements needed to perform the '**Create Object**' command is absent or invalid. The **System file** must be created in advance of any other files. Accordingly, this condition does not apply to **System file** creation.

The command is rejected if file attributes do not confirm to the following rules:

- For a **Rule file**, the number of macro instructions must fall within the range of 1–4.
- For a **Record file**, the size must be ≥ 12 bytes—Record files must consist of at least one record which is one byte in size (see *Length of a Record File*).
- For a **Record file**, FDB must be in the range of 4–5. For further details, see ISO 7816-4 documentation.

After system file creation, a command is rejected if the '**Create**' access condition—which is referenced in the **System file**—is not fulfilled.

The command is aborted if the file to be created already exists.

The command cannot be run if the relevant memory space cannot be allocated.

*Note: A **Get Response** command is needed to validate the **Create Object** command within the card. Accordingly, if a command other than **Get Response** is received after **Create Object,** EEPROM data will be restored using the GemClub Backup Procedure. For more information, see GemClub Backup Procedure.*

Delete Object

This command is used to delete a file in the GemClub card application memory. During this procedure the referenced file is deleted, and the EEPROM taken up by the deleted file is freed for other uses.

Note: If you delete the system file that has been specified for the card, you delete all files on the memory.

*If secure messaging is used for **Delete System File**, then the Card Transaction Counter and the outgoing MAC will be replaced by zeroes.*

*If secure messaging is used for **Delete Secret Key File** when the same key is used for MAC calculation, then zeroes will be returned at the end of the command instead of the outgoing MAC.*

For information on the MACs exchanged in the **Delete Object** command, see '*Cryptography and Commands in GemClub*'.

Command Format

The **Command** is structured as follows:

Field	**Description Value**
CLA	80h.
INS	CEh.
P1	Data object type: 01h—System File. 02h—ISO 7816-4 Record File. 03h—Counter File. 04h—Rule File. 05h—Secret Code File. 06h—Secret Key File.
P2	SFI—(01–1Eh).
Lc field	10h.
Data field	**Terminal Data**—Eight bytes. **Incoming MAC** or **Secret code** Eight Bytes.
Le field	0Ah.

Response

The **Response** is structured as follows:

Field	**Description**
Data	**Card Transaction Counter**—Two bytes. **Outgoing MAC**—Eight bytes, zeroes if **Secure Messaging** is not specified.
SW1 SW2	Status bytes.

Get Response

This command is used to transmit response APDU(s) from the GemClub card to the terminal.

For information on provisions for **Case 4** commands when the **T=O** protocol is in use, see '*GemClub Command Format*'.

At the end of a case 4 command, the card returns the messages **SW1=61h** and **SW2='La**,' where 'La' represents the number of bytes available in the card for response. In this situation, the terminal, immediately issues a '**Get Response**' command to retrieve the APDU response.

Note: A ***Get Response*** *command is mandatory for the transaction validation at the end of the following commands:*

Award

Redeem

Use Rule

Update Parameter

Create Object

Delete Object

If the ***Get Response*** *command is not immediately issued after the relevant command, EEPROM data will be restored. See* GemClub Backup Procedure'.

Command Format

The **Command** is structured as follows:

Field	Description Value
CLA	00h—Not tested by the Card.
INS	C0h.
P1	00h.
P2	00h.
Le field	Number of bytes available in the card—This depends on the status byte **SW2** of the last command that has been run.

Response

The **Response** is structured as follows:

Field	Description
Data	Data available in the card buffer from the last command that has been run. (Le bytes).
SW1 SW2	Status bytes.

Read Parameter

This command is used to read a data element or card information (card serial number and the issuer reference) stored in a file.

Note: The card serial number is a unique number assigned by Gemplus during the card manufacturing process. It is coded over eight bytes.

*The **Read Parameter** command with specific option P1-P2=0000h is used to read the card serial number and the issuer reference.*

Command Format

The **Command** to read a data element is structured as follows:

Field	**Description Value**
CLA	80h.
INS	BEh.
P1	Data element tag—For example, the ID of the data element that is to be read. See '*Data Element Tags*'.
P2	SFI—(01–1Eh).
Lc field	08h.
Data field	**Secret code**—Eight bytes.
Le field	Data element length.

The **Command** to read card information is structured as follows:

Field	**Description Value**
CLA	80h.
INS	BEh.
P1	00h.
P2	00h.
Le field	1Ch.

Lc

Response

The **Response** from reading a data element is structured as follows:

Field	**Description**
Data	The data element that is to be read.
SW1 SW2	Status bytes.

The **Response** from reading card information is structured as follows:

Field	**Description**
Data	Card traceability information, including card serial number and issuer reference—28 bytes. For more information see the Card Information graphic on the following page.
SW1 SW2	Status bytes.

Command Processing

Before running a **Read Parameter** command the operating system must carry out certain verification procedures.

The OS verifies the following:

To **read a data element:**

Action	Return Code if Command fails at this stage
Verify that syntax of command is correct. See '*Syntax Verification*'.	—
Verify that system file is present.	6400h
Verify that referenced file is present.	6A82h
Verify reference file integrity.	6400h
Verify the access Condition for '**Read**.' See '*Access Condition Checking*'.	—
Verify the existence of referenced data elements—Optional data elements in **Counter** and **Rule** files.	6A88h
Normal ending of the command.	61h 'La'

To **Read Card Information**

Action	Return Code if Command fails at this stage
Verify the syntax of command is correct. See '*Syntax Verification*'.	—
Verify the outgoing data length 'Le.'	6Ch'xx'
Verify the normal ending of the command.	9000h

Context Modification after Execution

During Context Modification after running the command, the global security status is unchanged.

Conditions of Use and Security

The command is rejected if:

- One of the data elements needed to run the '**Read Record**' command is absent or invalid.
- A short file identifier is not transmitted in the command (SFI=0).
- Access condition for the **Read** command is not fulfilled.

The operating system allows part of the record to be read. Accordingly a length error '6C xx' will be returned if

- 'Le' field =256 **or**
- 'Le' field is greater than the record length.

Command Processing

Before running a **Read Record** command, the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	Return Code if Command fails at this stage
Verify the syntax of command is correct. See '*Syntax Verification*'.	—
Verify that system file is present and valid.	6400h
Verify that referenced file is present.	6A82h
Verify referenced file integrity.	6400h
Verify that referenced record exists.	6A83h
Verify the access condition for **Read**. See '*Access Condition Checking*'.	—
Verify the outgoing data length 'Le' according to the record length.	6Ch'xx
Normal ending of the command.	9000h

Context Modification after Execution

During Context Modification after running the command, the global security status is not affected.

Status Codes

The **Status Codes** which can be returned by this command are:

SW1	SW2	Description
61h	'La'	'La' bytes available for '**Get Response**.'
64h	00h	No precise diagnostic.
65h	81h	Memory failure.
67h	00h	Wrong length.
69h	82h	Security status not satisfied.
69h	83h	Authentication method blocked.
69h	85h	Conditions of use not satisfied.
6Ah	82h	File not found.
6Ah	86h	Incorrect parameters P1-P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if:

- One of the data elements needed to perform the **Redeem** command is absent or invalid.
- The access condition for **Redeem** is not fulfilled.

The command is aborted if:

- The current date does not verify the following condition:

  Redeem start date ≤ current date ≤ Redeem end date.

- The balance is less than the amount to be redeemed.

Note: If the validity dates for ***Redeem*** *are not present in the counter file, then the date verification will be by-passed.*

A ***Get Response*** *command is required to validate the transaction inside the card. Thus if a command other than* ***Get Response*** *is received after an* ***Redeem****, EEPROM data will be restored. See* GemClub Backup Procedure'.

If no ***Transaction Proof*** *key file is referenced in the counter file, then a data package consisting of zeroes will be returned instead.*

Select Application File by Name

This command is used to emulate a subset of the ISO 7816-4 '**Select File**' command in order to comply with EMV application selection features.

In cases which take in EMV-PSE issues, the **Application file** name (possibly right-truncated) is compared to the following string '**1PAY.SYS.DDF01**.' See '*Referencing by Name EMV Simulation*'.

In all other cases, the **Record file** is located in the EMV-DIR file—if any. When the **Record file** is located, a pre-defined EMV-compatible response is returned to the terminal.

Note: This command can only be run on a GemClub-EMV card.

Command Format

The **Command** is structured as follows:

Field	**Description Value**
CLA	00h—Not tested by the Card.
INS	A4h.
P1	04h.
P2	00h.
Lc field	01h–10h.
Data field	Application file name—possibly right truncated.
Le field	09h + relevant record file name length—01h–10h.

Response

The **Response** is structured as follows:

Field	**Description**
Data	File control information.
SW1 SW2	Status bytes.

File Control Information for EMV-PSE

File Control Information for EMV-DIR files is structured as follows:

Field	**Value**	**Length in Bytes**
Template—File control information (FCI).	6Fh	1
Length—07h + relevant Application file name length.	15h	1
Tag— Application file name.	84h	1
Length—Actual application file name length.	0Eh	1
Value— Application file name.	**'1PAY.SYS.DDF01'**	14
Template—Proprietary data.	A5h	1
Length.	03h	1
Tag—SFI of the EMV-DIR file.	88h	1
Length.	01h	1
Value—SFI of the EMV-DIR file. This is coded in the system file.	SFI	1

Command Processing

Before running a **Select Application file by Name** command the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	Return Code if Command fails at this stage
Verify that syntax of command is correct. See '*Syntax Verification*'.	—
Verify that system file is present and valid.	6400h
Verify that **EMV-DIR** file is present.	6A82h
Verify **EMV-DIR** file integrity.	6400h
Browse for **Application file** name string in the EMV-DIR.	6A82h
Verify relevant EMV-DIR application length according to **Application file** name length.	6A82h
Normal ending of the command—'La' bytes still available for '**Get Response**.'	61h'La'

Context Modification after Execution

During Context Modification after running the command, the global security status is not affected.

Status Codes

The **Status Codes** which can be returned by this command are:

SW1	SW2	Description
90h	00h	Normal ending of the command.
67h	00h	Wrong length.
6Ah	86h	Incorrect parameters P1-P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

If the command '**Select Communication Speed**' is accepted by the card, then the baud rate is switched after transmission of the status bytes. Accordingly, any newly specified communication speed will apply to the next command until a **Reset** or another successfully implemented '**Select Communication Speed**' command has been specified.

Note: Depending on the selected baud rate and especially for the 115,200 baud rate ***the terminal will insert a minimum delay between receipt of the status byte SW2 of a command and receipt of the next command sent to the card.*** *This applies for the* ***Select Communication Speed*** *command and any other command exchanged using the higher baud rate. Extra-guardtime has been specified for the terminal to ensure that the card is ready to receive the next command.*

Guardtime to be inserted is defined as the number of ***etus*** *between the start bit of* ***SW2*** *and the start bit of the first byte of the next command (CLA). Accordingly, the following values are defined:*

For ***TA1=95h*** *(highest baud rate): terminal extra-guardtime =* ***24 etus*** *(32 clock cycles per etu).*

Command Processing

Before running a '**Select Communication Speed**' command, the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	Return Code if Command fails at this stage
Check that syntax of command is correct. See '*Syntax Verification*'.	—
Normal ending of the command.	9000h

Context Modification after Execution

During Context Modification after running the command, the global security status is not affected.

SW1	SW2	Description
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if:

- One of the data elements needed to perform the '**Update Parameter**' transaction is absent or invalid.
- The access condition for '**Update Parameter**' is not fulfilled.

Note: Certain data elements may have implicit access conditions specified for them.

For the secret key value update, where secret key file protection has been specified, the secret half-key is encrypted.

*A **Get Response** command is needed to validate the **Update Parameter** command within the card. Accordingly, if a command other than **Get Response** is received after **Update Parameter,** EEPROM data will be restored using the* GemClub Backup Procedure. *For more information, see* GemClub Backup Procedure'.

Command Processing

Before running an **Update Parameter** command the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	Return Code if Command fails at this stage
Verify the syntax of command is correct. See '*Syntax Verification*'.	—
Verify that system file is present and valid.	6400h
Verify the referenced file is present.	6A82h
Verify the reference file integrity.	6400h
Verify CTC<FFFFh.	6983h
Increase CTC—EEPROM by one unit.	6581h
Verify the access condition for 'Update.' See '*Access Condition Checking*'.	—
Verify the existence of referenced file—Optional data elements in **Counter** and **Rule** file.	6A88h
Update data element—EEPROM.	6581h

Status Codes

The **Status Codes** which can be returned by this command are:

SW1	SW2	Description
90h	00h	Normal ending of the command.
61h	'La'	'La' bytes available for '**Get Response**'—only where secure messaging is specified.
64h	00h	No precise diagnostic.
65h	81h	Memory failure.
67h	00h	Wrong length.
69h	82h	Security status not satisfied.
6Ah	82h	File not found.
6Ah	83h	Record not found.
6Ah	84h	Not enough memory space.
6Ah	86h	Incorrect Parameters P1–P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if:

- One of the data elements needed to run the '**Update Record**' command is absent or invalid.
- A short file identifier is not transmitted in the command—SFI=0.
- Access condition for the **Update** command is not fulfilled.
- The **Lc** field is not the same length as the record length.
- If the **Lc** field does not contain at least eight bytes of MAC and one record byte—In cases involving secure messaging.
- If the Lc field is higher than the card reception buffer size (32 bytes)—In cases involving secure messaging.

Command Processing

Before running an **Update Record** command, the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	**Return Code if Command fails at this stage**
Verify the syntax of command is correct. See '*Syntax Verification*'.	—
The incoming piece of data—8<'Lc'≤32.	6700h
Verify that system file is present and valid.	6400h
Verify that referenced file is present.	6A82h
Verify referenced file integrity.	6400h
Verify CTC<FFFFh	6983h
Increase CTC—EEPROM by one unit.	6581h
Verify access condition for **Update**. See '*Access Condition Checking*'.	—
Verify that referenced record exists.	6A83h
Verify that length of the incoming **Lc** data is in accordance with the record length.	6A84h
Update the record—**EEPROM**.	6581h
Normal ending of the command—Where secure messaging has been specified.	61h'La'
Normal ending of the command—Where secure messaging has not been specified.	9000h

Context Modification after Execution

During Context Modification after running the command, the global security status is not affected.

Status Codes

The **Status Codes** which can be returned by this command are:

SW1	SW2	Description
61h	‘La’	‘La’ bytes available for ‘**Get Response**.’
64h	00h	No precise diagnostic.
65h	81h	Memory failure.
67h	00h	Wrong length.
69h	82h	Security status not satisfied.
69h	83h	Authentication method blocked.
69h	85h	Conditions of use not satisfied.
6Ah	82h	File not found.
6Ah	86h	Incorrect parameters P1-P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if:

- One of the data elements needed to perform the **Use Rule** command is absent or invalid.
- The access condition for **Use Rule** is not fulfilled.

*Note: A **Get Response** command is required to validate the transaction inside the card. Thus if a command other than **Get Response** is received after an **Use Rule,** EEPROM data will be restored. See* GemClub Backup Procedure’.

*If no **Transaction Proof** key file is referenced in the counter file, then a data package consisting of zeroes will be returned instead.*

Command Processing

Before **Command processing** of a **Use Rule** command the operating system must carry out certain verification procedures.

The OS verifies the following:

Action	Return Code if Command fails at this stage
Verify that syntax of command is correct. See ‘*Syntax Verification*’.	—
Verify that system file is present and valid.	6400h
Verify that referenced rule is present.	6A82h
Verify referenced rule integrity.	6400h
Verify that last rule ran correctly.	6400h
Check the existence and validity of **Transaction Proof** key—if any.	6400h

Stays in force until warm reset of Card.

Verify Secret Code

This command is used to compare the secret code stored in the GemClub card with the secret code sent by the terminal.

This command updates the global security status and can also be used to find out the number of attempts remaining to present a secret code.

Command Format

The **Command** is structured as follows:

Field	Description Value
CLA	00h.
INS	20h.
P1	00h.
P2	Secret code references: **0** for the **PIN** code and **1-30** for a secret code SFI. Other values are RFU.
Lc field	**08h** Verify secret code. **00h**—Return the number of attempts remaining.
Record Field	**Secret code**—Optional, must be eight bytes in length.

Response

The **Response** is structured as follows:

Field	Description
SW1 SW2	Status bytes.

Status Codes

The **Status Codes** which can be returned by this command are:

SW1	SW2	Description
90h	00h	Normal ending of the command.
63h	C'x'h	Verification failed—'x' shows the number or attempts remaining.
64h	00h	No precise diagnosis.
65h	81h	Memory failure.
67h	00h	Wrong length.
69h	83h	Authentication method blocked.
6Ah	82h	File not found.
6Ah	86h	Incorrect parameters P1-P2.
6Dh	00h	Instruction code not supported or incompatible with class.
6Eh	00h	Class not supported.

Conditions of Use and Security

The command is rejected if one of the data elements needed to run the '**Verify Secret Code**' command is absent or invalid.

Data Element Tags

The following is a table of all tags (IDs) of all data elements used in GemClub.

Tag	Data element	
20h	**System file**	**Personalization Data** Group of data elements which can be used to read or update file details. It includes tags **21h**, **22h**, **23h**, **24h** and **26h**.
21h	**System file**	AC for **Update/Delete** Specifies right to **Update** or **Delete** a file.
22h	**System file**	AC for **Read** Specifies right to **Read** a file.
23h	**System file**	AC for **Create** Specifies right to **Create** a file.
24h	**System file**	**PIN Code File Reference information** Secret code used as PIN for cardholder identification.
26h	**System file**	**EMV-DIR File Reference information** Identifier of the file used for EMV-DIR simulation.
27h	**System file**	**Card Transaction Counter** Used for authentication operations. The CTC is used for secure messaging computation.
40h	**Record File**	**Personalization Data** Group of data elements which can be used to read or update file details. It includes tags **41h** and **42h**.
41h	**Record File**	AC for **Update/Delete Access** Specifies right to **Update** or **Delete** records in the record files.
42h	**Record File**	AC for **Read Access** Specifies right to **Read** records in the record files.
60h	**Counter file**	**Personalization Data** Group of data elements which can be used to read or update file details. It includes tags **61h**, **62h**, **63h**, **64h** and **65h**.

Tag	Data element	
75h	**Counter file**	**Rules allowed for this counter** Specifies rules that can be run on this counter. If this field is absent, the '**Use Rule**' operation cannot modify the details on the counter. Otherwise, each bit corresponds to a rule identifier. For more information see: '*Use Rule Command*'.
76h	**Counter file**	**Label** Displays an alphanumeric value which can be used to identify the name and the version of the counter. This can be re-used during the life of the card.
80h	**Rule file**	**Personalization Data** Group of data elements which can be used to read or update file details. It includes tags **81h**, **82h**, **83h**, **84h** and **85h**.
81h	**Rule file**	AC for **Update/Delete** Specifies right to **Update** or **Delete** the Rule file.
82h	**Rule file**	AC for **Read** Specifies right to **Read** the file.
83h	**Rule file**	Key reference for **Transaction Proof** Reference key that is used for transaction proof computation. It ensures that a transaction proof is requested.
84h	**Rule file**	AC for **Use Rule** Specifies right to use the rule by performing a **Use Rule** command.
85h	**Rule file**	**Version** A byte used to show the version of the rule.
90h	**Rule file**	**Macro Instruction 1** Specifies, and describes, the action that is to be carried out on the specified counter.
91h	**Rule file**	**Macro Instruction 2** Specifies, and describes, the rule(s) to be used on this counter.
92h	**Rule file**	**Macro Instruction 3** Specifies, and describes, the rule(s) to be used on this counter.

APPENDIX A: GEMCLUB EXCHANGE STRUCTURE

Overview

This chapter describes the exchange features that have been created for GemClub.

Communication Features

GemClub cards support a clock frequency rate of 1 MHz—5 MHz for a power supply of 5v (frequency rate = 1-2 MHz for a power supply of 3v). The clock signal is derived from a command generated from the terminal. The duty cycle for asynchronous operation is 40%—60% of the period during stable operation.

The card defaults to a low energy consumption level after each command has been run. It is reactivated by either a **Reset Interrupt** command, or upon receipt of the first bit of the first character of any new command that is sent to the terminal.

Physical Layer

Characters sent from or received by GemClub cards should be structured as follows:

- One start bit
- Eight data bits
- One parity bit even
- One or two stop bits—depending on the extra guardtime parameter that is specified for the character

The communications convention supported by the card is the **Direct Convention.** This means that:

- The least significant bit (b0) is sent first
- The logical level '1' is 'mark'
- The first byte of the **Answer-to-Reset** is 3Bh

The parity level is specified as logical level '0' if the number of the logical level '1' bits in the sequence from b7-b0 is even.

The **Extra Guardtime** parameter N used to send characters from the GemClub reader to the card equals 0. This means that the card requires a 12 etu delay between two consecutive leading edges—Represented by two stop bits in the next frame.

No **Extra Guardtime** is needed to send characters from the GemClub card to the GemClub reader.

GemClub Communication Protocol

GemClub supports the **T=0** protocol in accordance with **ISO 7816-3** specifications. For more information on the **T=0** protocol, please consult the relevant ISO documentation.

There are a number of specific parameters which must be taken into account prior to applying the **T=0** protocol. These are as follows:

- The size of the reception buffer is set at 32 bytes. If more than 32 bytes are transmitted, the card supports *slave mode* in reception.
- The work waiting time (wwt) is the maximum delay between the start leading edge of any character sent to the card and the start leading edge of the previous character (sent either by the card or by the reader).

  $$wwt = 960 \times W \times F \times \frac{1}{f}$$

  Where wwt = work waiting time.

  f = frequency clock currently delivered in the card—in Hz.

  The parameter W is coded in byte TC2 of the ATR.

  The default value for W is 10—if f = 3.5712 MHz and F = 372, then wwt is 1 second.

  To reset the work waiting time, the card can send a null byte (60h).
- Each command that is received in GemClub must begin with five header bytes—**CLA**, **INS**, **P1**, **P2** and **P3**

Header Byte	What it means
CLA	Instruction Class
INS	Instruction Code
P1	The first parameter.
P2	The second parameter.
P3	The length parameter.

Notes**: The length parameter* ***P3 *is used to specify the number of data bytes which are to be transmitted during the command. You must also specify the direction in which these pieces of data are to move.*

In an outgoing command, ***P3=0*** *results in transfer of a 256-byte piece of data from the card.*

In an incoming data command, ***P3=0*** *results in no pieces of data being transferred.*

After receipt of this 5-byte header, the GemClub reader waits for receipt of a ***Procedure byte****.*

APPENDIX B: ANSWER-TO-RESET

Answer-To-Reset

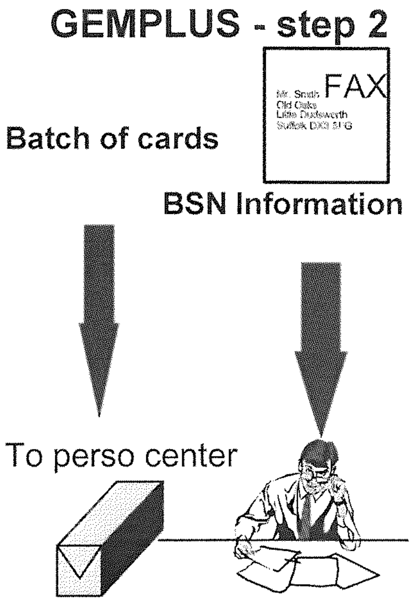
The **Answer-To-Reset** command is structured in the GemClub card as follows:

Character	**Hex. value**	**Meaning**
TS	3Bh	Direct convention
T0	6xh	TB_1 and TC_1 present, x historical characters (default: x=8)
TB_1	00h	Vpp not required
TC_1	00h	No extra guardtime required
T1-T8		Historical bytes compatible 7816-4 and status information like a TLV compact object.
T1	80h	Status information is contained in an optional TLV object.
T2	66h	Tag: 6 (Pre-issuing data), length: 6 bytes
T3	xx	FMN : OS family name
T4	xx	PRN : Product name
T5	xx	OSV : OS version
T6	xx	PRV : Program version
T7	xx	CIV : Chip reference
T8	0Eh	Cardlife status byte

Historical bytes T3 T7

Product definition information

FMN	**PRN**	**OSV**	**PRV**	**CIV**
A2h	06h	02h	01h	32h



Items Required

Under the customer card shipment process, four different items are sent to the card issuer for the first order of the year (unless a customer card from the previous year has not yet expired see *Customer Card Description* below):

1. GemClub cards,
2. *Customer card* (1 card is supplied by default. Additional customer cards can be supplied upon request when placing the order).
3. *Customer card information sheet* (containing the customer card protection code and the key verification code).
4. *BSN information sheet* (containing the batch seed parts AA and BB and the key verification code).

These items are sent to you by different means to ensure optimal security during the shipment to your premises.

Customer Card Description

Personalization cannot be started until all four items are available.
Customer cards are used to ensure that the customer key (necessary to derive the mother batch key) is transported to your personalization center under secure conditions. Once received, the customer card can be used to forward the customer key under secure conditions to another site.

The customer card is a dedicated MPCOS-EMV card, meaning that it accepts the same commands as all other MPCOS-EMV cards.

Please refer to the *MPCOS-EMV Reference Manual* for more information. If you are not familiar with this card, you will find all the commands (in APDU format) you need to send to the customer card to retrieve the customer key in *Recovering the Customer Key* later in this appendix.

The customer card and two information sheets are valid for one year exactly, from the date that the customer card is issued (so if it was issued in September, it would still be valid for an order in April the following year).

Customer Card and BSN Forms

The customer card contains the customer key which is used in conjunction with the batch seed to compute the mother batch key.

The customer card is protected by a secret code (the customer card protection code). This code and a key verification code are sent to the customer by fax. See the following page for a copy of the *Customer Card Information Form.*

The batch seed number (parts AA and BB) and another key verification code are sent by a separate fax. See the page following the *Customer Card Information Form* for a copy of the *BSN Information Form.*

BSN INFORMATION

Customer Name

CUSTOMER: BANK Y

INTERNAL	**REF.:**	**9734001**
ORDER	**REF.:**	**12345678**
BATCH	**REF.:**	**9734042**

Dear Customer,

Please find hereafter the information required to start personalizing your cards.

- Batch seed part **AA:**

8002 E1A7 7ACF BC1E (D2)

Inverted Checksum (last byte)

- Batch seed part **BB:**

EA79 4DE6 6692 6EC5 (3E)

- Key Verification Code (KVC):

1AB4CB

Key Verification Code
= 3 MSB of 3DES (0,MKbatch)

In the case of any problem, please contact the G+ Hot Line (33 442 36 50 50). The internal reference above must be provided to your Gemplus contact.

Yours Faithfully.

Recovering the Mother Batch Key

	Deriving the mother batch key from the BSN form
1.	Compute $MKbatch_a = 3DES^{-1}$ (Batch seed AA, Kcust) Compute $MKbatch_b = 3DES^{-1}$ (Batch seed BB, Kcust)
2.	Compute $MKbatch = MKbatch_a \| MKbatch_b$
3.	Use KVC (from BSN Information form) to verify that the mother batch key is correct. If it is correct then: KVC = 3MSB 3DES(0,MKbatch).

Card-by-Card Personalization

Before an individual card can be personalized, its card key, 01, must be recovered.

	Recovering the Key of an Individual Card
1.	Recover the Card Serial Number using the **Read Parameter**() command (see 'Read Parameter').
2.	Use the following function to recover the Card Key: **Card Key = 3DES_16 (Card Serial Number, Mother Batch Key).**
3.	Use Secure Messaging using the GemClub commands and the recovered Card Key.

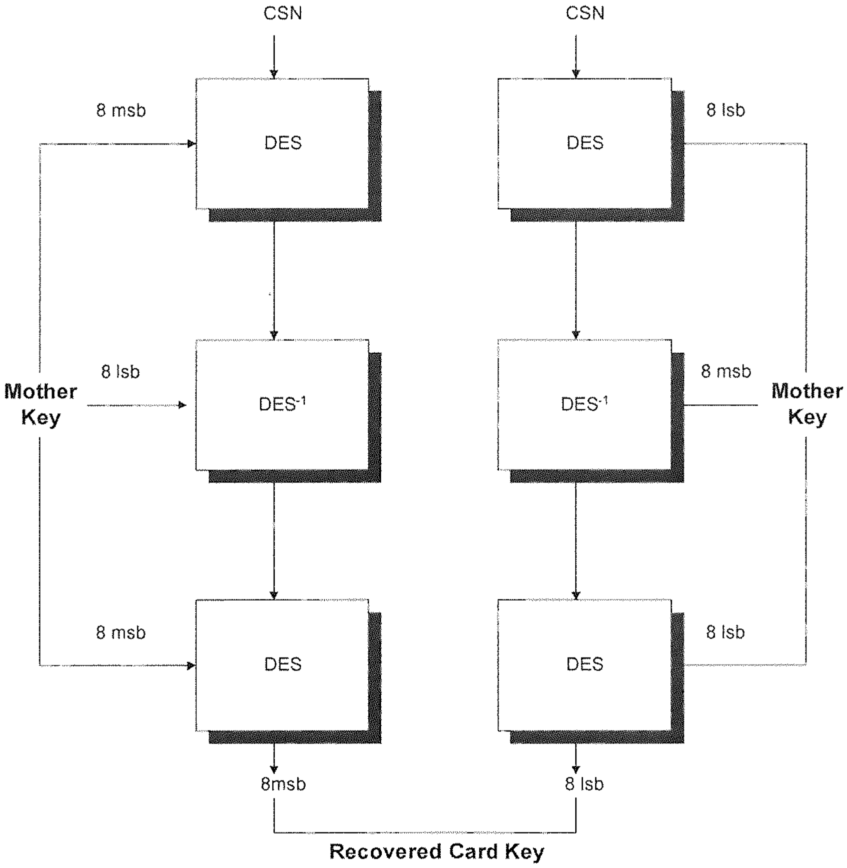


Figure 8 Key 01 recovery

GLOSSARY

For the purpose of this specification, the following definitions apply:

Term	**Definition**
Access conditions	A set of security attributes attached to a file.
Acquirer	An organization which collects and possibly aggregates transactions from several terminals and/or from other acquirers for delivery to one or more Loyalty operators.
APDU	Data exchange protocol between a card and a reader.
Award	Points awarded to a loyal customer in return for the purchase of goods that are linked with a loyalty program.
Balance	The current amount of points stored into the counter.
Batch Seed	Part of the BSN used to compute the mother key for a batch of delivered cards.
Batch Serial Number	Used for key diversification in the customer card shipment process.
Card session	A link between the GemClub card and the terminal starting with the Answer To Reset and ending with a subsequent reset or a deactivation of the card.
Card Transaction Counter (CTC)	Used for authentication operations. The CTC is used for secure messaging computation.
Counter file	A file in the card which allows for point storage, awards and redemptions.
Cumulative balance	The number of points awarded since the creation of the counter file.
Current date	The current date as recorded on the terminal.
Data element	A string of bytes which are processed in one package by the application—either card application layer or terminal application layer.
Data object	A set of data elements—a file.
Diversified key	A key derived from a 'mother key' and a unique identifier—for example, the card serial number—using cryptography features. In this document, the term 'diversified key' refers to any key stored in the card's non-volatile memory.
File	A set of data elements.
Global Security Status	Used to record correct presentation of the PIN code and of the most recently presented secret code. It is stored in a GemClub card's RAM.

Term	Definition
Record	String of bytes which can be handled as a whole by the card and absolutely referenced by a record number or relatively referenced to the current record.
Record file	Used to store data in the GemClub card.
Record number	A number which uniquely identifies the record within its record file. The number is assigned in sequential order.
Redemption	The exchange of points earned on the card for a service, a product or a discount, in accordance with rules specified in the relevant loyalty program.
Rules	Used to link several actions pertaining to awards or redemptions. The rules are included in rules files.
Secret Code	Data which the application may require to be presented to the card by its user before data can be processed.
Secret key	Value with a 16-byte length used in an algorithm to compute 3DES authentication.
Secure Messaging	Used to ensure that communications between card and reader are not corrupted in transit and for mutual authentication between a card and terminal.
Session	Period of time between two card Resets, or between a power up and a power down.
System file	File used for the global security of the card.
Tag	A unique number which is one byte long.
Terminal	A device which supports a given loyalty application and accepts the corresponding loyalty cards at the point of sale.
Transaction Proof	Used to prove that a data transfer has taken place between a card and a terminal.
Triple DES	A variant of the DES algorithm, consisting of a triple encryption.
Visit counter	The number of customer purchases using the card.

Abbreviation/Acronym	**Definition**
Le	Length expected
LRC	Longitudinal Redundancy Check
l.s.	Least Significant
LSB	Least Significant Byte
MAC	Message Authentication Code
m.s.	Most Significant
MSB	Most Significant Byte
obj	Object (by extension: data object).
ofs	Offset
OS	Operating System
OTP	One-Time Programmable
P1/P2	Parameter 1/Parameter 2 of the command header
PIN	Personal Identification Number
rec	Record number
RFU	Reserved for Future Use
ROM	Read Only Memory
SAM	Secure Access Module
SFI	Short File Identifier
SM	Secure messaging
SVC	Stored Value Card
SW1/SW2	Status Word 1/Status Word 2
TLV	Tag length value
TPDU	Transmission Protocol Data Unit
var	Variable
Vcc	Supply Voltage
Vpp	Programming Voltage