# Interoperability Specification for ICCs and Personal Computer Systems

## *Part 3. Requirements for PC-Connected Interface Devices*

*Gemalto*

*HID Global*

*NXP Semiconductors N.V.*

*Oracle America*

*SCM Microsystems*

**Revision 2.01.09**
**June 2007**

**AMENDMENT 1**
**2011-06-03**

**AMENDMENT 1:**

- Clarification of key type identifier for picopass® keys
- Definition and use of Increment and Decrement data objects
- Definition of a vendor specific, generic command

## Scope

The scope of this amendment is:

A) Clarification of key type identifier for picopass® keys.

B) Definition and use of Increment and Decrement data objects.

C) Definition of a vendor specific, generic command.

Amendment 1 to PC/SC part 3, revision 2.01.09 from June 2007 was prepared by the members of PC/SC Workgroup, Tech Meeting, June 3, 2011, on request of HID Global

 

# Interoperability Specification for ICCs and Personal Computer Systems

Part 3:
## Requirements for PC-Connected Interface Devices

AMENDMENT 1: Clarify key type identifier for picopass® keys

Clause 3.2.2.1.6, page 29, term Byte 3 (Key Type):

Replace the existing definition with the following:

**Byte 4 (Key Type):**

The type of the key: E.g. for MIFARE® KEY_A (0x60) or KEY_B (0x61), for picopass® Debit_key (0x00) or Credit_key (0x01)

 

# Interoperability Specification for ICCs and Personal Computer Systems

Part 3:
## Requirements for PC-Connected Interface Devices

AMENDMENT 1: Definition and use of Increment and Decrement data objects

Clause 3.2.2.1 – Storage Card Functionality Support

Add the following chapter:

### 3.2.2.1.10 Increment / Decrement Value

This command increments or decrements the value of a data object if the card supports this functionality.

| Command | CLA | INS | P1 | P2 | Lc | Data | Le |
|---|---|---|---|---|---|---|---|
| INCREMENT / DECREMENT VALUE | 0xFF | 0xC2 | 0x00 | 0x03 | xx | BER-TLV | -- |

P2 = 00, 01and 02 is already reserved and more specifically detailed in pcsc3_v2 02 00_sup2. Value, destination and mode are coded in the BER-TLV data field.

The increment / decrement TLV structure starts with the context specific, constructed TAG:
0xA0 (INCREMENT)
0xA1 (DECREMENT)
followed by context specific, primitive TAGs for destination(s) and value. It is allowed to use more then one destination in one value sequence or to use more then one value choice in one data field.

CHOICE is defined as a context specific, constructed TAG 101x xxxx
SEQUENCE is defined as a context specific, primitive TAG 100x xxxx

```
VALUE CMD ::= CHOICE {
        Increment Value     [0]  INCREMENT,
        Decrement Value     [1]  DECREMENT
}

INCREMENT ::= SEQUENCE {
        Destination         [0]  OCTET SRTING, OPTIONAL     -- See warning 1)
        Value               [1]  OCTET STRING               -- See warning 2)
}

DECREMENT ::= SEQUENCE {
        Destination         [0]  OCTET SRTING, OPTIONAL     -- See warning 1)
        Value               [1]  OCTET STRING               -- See warning 2)
}
```

 

Warning [1]): The length of Destination - OCTET string depends on the card size (number of blocks). PC/SC does not define a bit- and byte-endianness for the destination. The endianness is predefined by the card.

Warning [2]): The length of Value - OCTET string depends on the value size of the card. PC/SC does not define a bit- and byte-endianness for the value. The endianness is predefined by the card.

Note: If more then one destination is coded in an increment or decrement SEQUENCE and the card is a MIFARE® card then the MIFARE® restore command should be used after the first increment or decrement to transfer the values to 2nd or 3rd block.

| Tag | Length | Value |
|---|---|---|
| 0xA0 | xx | INCREMENT SEQUENCE |
| 0xA1 | xx | DECREMENT SEQUENCE |
| 0x80 | xx | Destination block number, optional,<br>Length depends from card type, e.g. 1 for a MIFARE® block |
| 0x81 | xx | Value,<br>Length depends from card type, e.g. 4 for a MIFARE® value block |

For example: Decrement MIFARE® block 5 and restore to block 6 (backup)

```
FF C2 00 03 0E
            A1 0C                       // decrement
               80 01 05                 // block 5 and
               80 01 06                 // restore to block 6
               81 04 01 00 00 00        // value = 1
```

For example: Decrement MIFARE® block 5 (value = 100) and increment block 6 (value = 2)

```
FF C2 00 03 16
            A1 09                       // decrement
               80 01 05                 // block 5
               81 04 64 00 00 00        // value = 100
            A0 09                       // increment
               80 01 06                 // block 6
               81 04 02 00 00 00        // value = 2
```

For example: Decrement picopass® e-purse from selected page (implicit block 2)

```
FF C2 00 03 08
            A1 04                       // decrement
               81 02 01 00              // value = 1
```

**Increment / Decrement Command Error Codes**

| | SW1 | SW2 | Meaning, status word as described below |
|---|---|---|---|
| Success | '90' | '00' | Increment operation successful |
| Errors | '65' | '81' | memory failure (unsuccessful increment) |
| | '69' | '81' | incompatible command |

 

| | | | |
|---|---|---|---|
| | '69' | '82' | security status not satisfied |
| | '69' | '86' | command not allowed |
| | '6A' | '81' | function not supported |
| | '6A' | '82' | invalid block address |

 

# Interoperability Specification for ICCs and Personal Computer Systems

Part 3:
## Requirements for PC-Connected Interface Devices

AMENDMENT 1: Definition of a vendor specific, generic command

Clause 3.2.2.1 – Storage Card Functionality Support

Add the following chapter:

#### 3.2.2.1.11 Vendor specific generic command

IFDs often support features outside the specified commands of PC/SC. To allow applications to control these features a generic command needs to be defined. The definition of such a generic command prevents conflicts of reserved INS values but used by certain IFDs.

Note: *This command shall not be used to implement features already defined in any of the parts of the PC/SC specification. An IFD having implemented such features within this generic command will be considered non-compliant with PC/SC.*

| Command | CLA | INS | P1 | P2 | Lc | Data field | Le |
|---|---|---|---|---|---|---|---|
| VENDOR CMD | 0xFF | 0x70 | VID MSB | VID LSB | xx | variable | xx |

P1 and P2 constitute the world wide unique vendor ID as assigned by the USB organization. The INS byte indicates this command clearly as a vendor specific command. P1 and P2 indicate the individual vendor who owns this particular command. For this reason the data field is fully under control of the individual vendor and does not need to be structured at all.

 