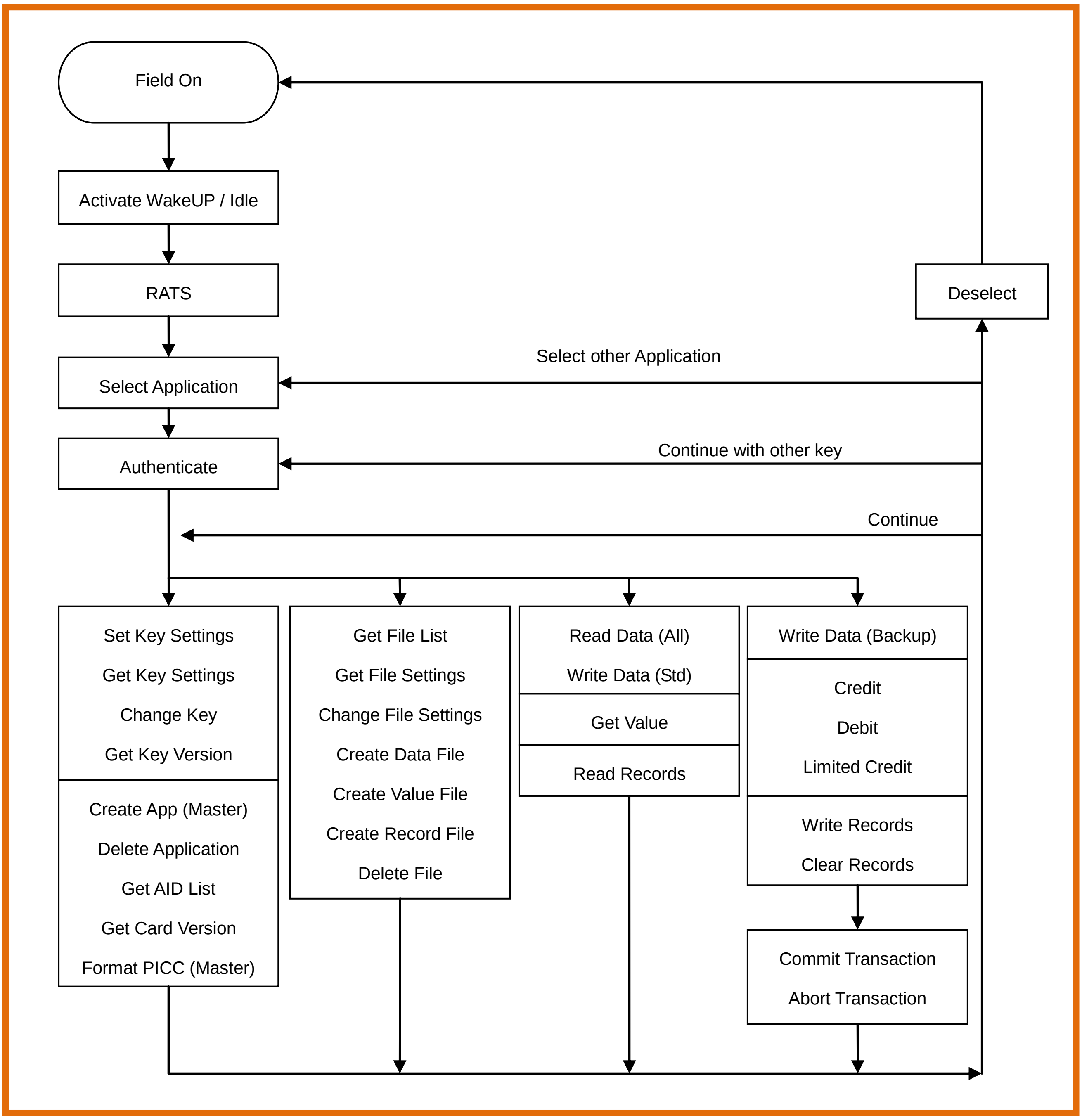
Mifare® Application Programming Guide

for DESFire®

REV.E
August 10, 2011

DESFire® Schema (In Command Mode)



MFAP Extend Functions Table for DESFire® (40h~5Fh)

Commands	Query (Master/Host)			Response (Slave/Device)	
	Func	Len	Parameters	Len	Data Bytes
T=CL Commands					
Activate WakeUp / Idle	40h	0		4/7	CSN (4 bytes) UID (7 bytes)
		1	Flag (byte)		
RATS (Request to Answer To Select)	41h	0		6	ATS
		1	CID (byte, 0~14)		
DESELECT	42	0		1	ACK, Successful NAK, TCL Error Code
Security related commands					
Authenticate	43h	17	Key#(byte) + Key(16 bytes)	2	ACK, Successful NAK, Error Code
		18	Key#(byte) + Crypto Type(byte) + Key(16 bytes)	2	ACK, Successful NAK, Error Code
		2	Key#(byte) + Crypto Type(byte)	2	ACK, Successful NAK, Error Code
		3	Key#(byte) on Reader + Crypto Type(byte) + Key#(byte) on Card	2	ACK, Successful NAK, Error Code
Set Key Settings	44h	1	Settings (byte)	2	ACK, Successful NAK, Error Code
Get Key Settings		0		2	Key Settings Number of Keys
Change Key	45h	17	Key# (byte), New Key (16 bytes)	1	ACK, Successful NAK, Error Code
		33	Key#(byte), New Key(16 bytes) Old Key(16 bytes)		
Change Key for AES	45h	18	Key# (byte), Key Version(byte) New Key (16 bytes)		
		34	Key#(byte), Key Version(byte) New Key(16 bytes) Old Key(16 bytes)		
Get Key Version	46h	1	Key# (byte)	1	ACK, Version NAK, Error Code
Save Key	6Bh	17 or 25	Key#(Byte) + Key Value(16 or 24 Bytes)	0	ACK, NAK, Error Code
Card Level Commands					
Create Application	47h	6	AID (long, MSB First) Key Settings(byte) Number of Keys(byte) + Crypto	1	ACK, Successful NAK, Error Code

			Type(bit6~7)(byte)		
Delete Application	48h	4	AID (long, MSB first)	1	ACK, Successful NAK, Error Code
Get AID List	49h	0		n	AID List (long, MSB first)
Select Application	4Ah	4	AID (long, MSB first)	1	ACK, Successful NAK, Error Code
Format PICC	4Bh	0		1	ACK, Successful NAK, Error Code
Get Card Version	4Ch	0		28	Version Info
Applications Level Commands					
Get FID List	4Dh	0		n	FID List
Get File Settings	4Eh	1	FID (byte)	n	fileSettings
Set File Settings		4	FID (byte), Communication Mode (byte), Access Right (int, MSB first)	1	ACK, Successful NAK, Error Code
Create Std Data File	4Fh	8	FID(byte), Communication Mode(byte), Access Right(int, MSB first), File Size(long, MSB first)	1	ACK, Successful NAK, Error Code
Create Back Data File	50h				
Create Value File	51h	17	FID(byte), Communication Mode(byte), Access Right(int, MSB first), Lower Limit(long, MSB first), Upper Limit(Long, MSB First), Initial Value(Long, MSB First), Limited Credit Enabled(byte)	1	ACK, Successful NAK, Error Code
Create Linear Record File	52h	12	FID(byte), Communication Mode(byte), Access Right(int, MSB first), Record Size(long, MSB first), Max. Num of Records(long, MSB first)	1	ACK, Successful NAK, Error Code
Create Cyclic Record File	53h				
Delete File	54h	1	FID(byte)		
File Level Commands					
Read Data	55h	9	FID(byte), Offset(long, MSB first), Length(long, MSB first) *Length=0~128	4/1	Length(long, MSB first) NAK, Error Code,
Write Data	56h				
Get Value	57h	1	FID(byte)	4/1	Value(long, MSB first) NAK, Error Code
Credit	58h	5	FID(byte), Amount(long, MSB first)		Amount(long, MSB first) NAK, Error Code
Debit	59h				
Limited Credit	5Ah				

Write Record	5Bh	9	FID(byte), Offset(long, MSB first), Length(long, MSB first)	4/1	Length(long, MSB first) NAK, Error Code
Read Records	5Ch	9	FID(byte), Record#(long, MSB first), NRecToRead (long, MSB first)	4/1	Count (long, MSB first) NAK, Error Code
Clear Record	5Dh	1	FID(byte)	1	ACK, Successful NAK, Error Code
Commit Transaction	5Eh	0		1	ACK, Successful NAK, Error Code
Abort Transaction	5Fh				
Additional					
Get File Buffer Addr.	60h	0		2	ACK, File Buffer Addr
Get UID	62h	0		4/2	ACK, 7 Bytes UID NAK, Error Code
Get Free Memory	64h	0		4/2	ACK, Free Memory (LONG) NAK, Error Code
LED & Buzzer Commands					
Control LED & Buzzer	3Ch	1	(00h)All LED Off, Buzzer Off (01h)Green LED ON (02h)Green LED OFF (03h)Red LED ON (04h)Red LED OFF (05h)Buzzer Beep 1 Time (06h)Buzzer Beep 3 Time (07h)Green LED ON with Beep 1 (08h)Red LED ON with Beep 3 (09h)All LED ON (Orange)	1	ACK, Successful
GNetPlus Base Commands					
Polling	00h	0		n	Return OEM Status
Get Version	01h	0		n	Return OEM Version String
Set Slave Addr	02h	1	New Address (1~255)	0	
Get Register	09h	3	Reg.Address2 + Reg.Lenght	n	Reg.Block
Set Register	0Ah	n	Reg.Address + Reg.Buffer	0	

Symbols and abbreviated terms

ACK	positive ACKnowledgement
AID	Application IDentifier
ATQ	Answer To reQuest
ATS	Answer To Select
CID	Card IDentifier
CRC	Cyclic Redundancy Check
CSN	Card Serial Number
DES	Data Encryption Standard
3DES	DES 3 times
FID	File IDentifier
GNet	Giga-tms Network protocol
GNetPlus	Giga-tms Network protocol Plus version
HLTA	HALT command, Type A
Int	16 bit (2 bytes) signed integer
Key#	Key Number (KeyNo)
Long	32 bit (4 bytes) signed integer
MAD	Mifare Application Directory
NAK	Negative AcKnowledgment
PCD	Proximity Coupling Device (Reader)
PICC	Proximity Card
RATS	Request for Answer To Select
REQA	REQuest command, type A
RFU	Reserved for Future Use
SAK	Select AcKnowledge
UID	Unique IDentification
WUPA	Wake-UP command, type A

Activate Wake-Up / Activate Idle (40H)

Func	Len	Parameters
40h	0 (WUPA)	
	1	flag (byte, 0x00=Activate Idle, 0x80=Activate Wakeup)

Following command set according to ISO14443-3:
Activate Wakeup (len=0) for PICC in Idle, Deselect or Halt state only.
Activate Idel (len=1, flag=0x80) for PICC in Idle state only.

Response:
ACK, UID (7 bytes) or CSN (4 bytes)

Example (Activate Wakeup):

```
:004000                     PCD send the Activate WakeUp Command
:000607044A5601366E10       PICC response the UID
```

Examples (Activate Idle):

```
:00400100                   PCD send the Activate Idle command
:000607044A5601366E10       PICC response the UID
```

RATS (41H)

Func	Len	Parameters
41h	0	
	1	CID (byte, 0~14)

Following command set according to ISO14443-4:
The response to the RATS command communicates the PICC capabilities to the PCD

CID : The logical number is in the range from 000 to 0x0E. This CID is used to distinguish several PICCs simultaneously selected by a single PCD. Default CID=0 when len=0.

```
Response:
ACK, ATS (6 Bytes)

Examples:
:004000                     PCD send Activate WakeUp command:000607041917795A1B80
        PICC response the UID
:004100                     PCD send RATS (Default CID=0)
:000606067577810280         PICC response the ATS (6 bytes)
```

DESELECT (42h)

Func	Len	Parameters
42h	0	

Following command set according to ISO14443-4:
To free the selected card after RATS.

```
Examples:
:004000                     PCD send Activate WakeUp command:000607041917795A1B80
        PICC response the UID
:004100                     PCD send RATS (Default CID=0)
:000606067577810280         PICC response the ATS (6 bytes)
:004200                     PCD send DESELECT command:0006020000
    PICC response ACK
```

Remark:
You have to use the Activate Wake-Up command to request the card after DESELECT.

Authenticate (43h)

Func	Len	Parameters
43h	17	Key# + Key (16 bytes)
43h	18	Key# + Crypto Type + Key (16 bytes for AES)
43h	26	Key# + Crypto Type + Key (24 Bytes for 3K3DES)
43h	2	EEPROM/Card Key# + Crypto Type
43h	3	EEPROM Key# + Crypto Type + Card Key#

Crypto Type:
0x00 – DES/3DES
0x40 – 3K3DES
0x80 – AES

In this procedure both, the PICC as well as the reader device, show in an encrypted way that they possesthe same secret which especially means the same key. This procedure not only confirms that both entitiescn trust each other but also generates a session key which can be used to keep the further communication successfully completed a new key for further cryptographic operations is obtained.

Depending on the configuration of the application (represented by its AID), an authentication has to be done to perform specific operations:

- Gather information about the application
- Change the keys of the application
- Create and delete files within the application
- Change access right
- Access data files in the authenticated application

Depending on the security configuration of the PICC, the following commands may require an authentication with the PICC (AID=0) master keys:

- Gather information about the application on the PICC
- Change the PICC master key itself
- Change the PICC key settings
- Create a new application
- Delete an existing application
- Format PICC

The authentication state is invalidated by

- Select an application
- Changing the key which was used for reaching the currently valid authentication status
- A failed authentication

Remark:
Master keys are identified by their key number 0x00. This is valid on PICC level (AID = 0x00) and on Application level (AID $\neq$ 0x00).

Set / Get Key Settings (44h)

Func	Len	Parameters
44h	1 (Set)	Settings (byte)
	0 (Get)	

This command set or get (len=0) the master key configuration settings depending on the currently selected AID. If AID=0x00 has been selected in advance, the change applies to the PICC key settings, otherwise (AID≠0x00) it applies to the application key settings of the currently selected application.

PICC (AID=0) Master Key Settings:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU	RFU	RFU	RFU	Configuration Changeable	PICC master key not required for create / delete	Free directory list access without PICC master key	Allow changing the PICC master key

On PICC Level (selected AID=0x00) the coding is interpreted as:

Bit 3: codes whether a change of the PICC master key settings is allowed:

- 0 = configuration no changeable anymore (frozen).
- 1 = this configuration is changeable if authenticate with PICC master key (default).

Bit 2: codes whether PICC master key authentication is needed before Create/Delete application

- 0 = Create/Delete application is permitted only with PICC master key authentication.
- 1 = Create application is permitted without PICC master key authentication (default).
  Delete application require an authentication with PICC master key or application master key[note].

Bit 1: codes whether PICC master key authentication is needed for application directory access:

- 0 = Successful PICC master key authentication is required for executing the “Get AID List” and “Get Key Settings” commands.
- 1 = “Get AID List” and “Get Key Settings” command succeed independently of a preceding PICC master key authentication (default).

Bit 0: codes whether the PICC master key is changeable:

- 0 = PICC master key is not changeable anymore (frozen).
- 1 = PICC master key is changeable (authentication with the current PICC master key necessary, default).

Note:

In case of usage of the application master key for deletion, the application which is about to be deleted need to be Selected and Authenticated with the application master key prior to the “Delete Application” command.

Application (AID≠0) Master Key Settings:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Change Key Access Right Bit 3	Change Key Access Right Bit 2	Change Key Access Right Bit 1	Change Key Access Right Bit 0	Configuration Changeable	Free create / delete without master key	Free directory list access without master key	Allow changing master key

On Application Level (selected AID≠0x00) the coding is interpreted as:

Bit 7-4: hold the Access Rights for changing application keys (Change Key command).

- 0x0 = Application master key authentication is necessary to change any key (default).
- 0x1…0xD = Authentication with the specified key is necessary to change any key.
- 0xE = Authentication with the key to be changed (same Key#) is necessary to change a key.
- 0xF = All keys (except application master key, see Bit 0) within this application are frozen.

Bit 3: codes whether a change of the application master key settings is allowed:

- 0 = configuration not changeable anymore (frozen).
- 1 = this configuration is changeable if authenticated with the application master key (default).

Bit 2: codes whether application master key authentication is needed before “Create File” / “Delete File”

- 0 = “Create File” / “Delete File” is permitted only with application master key authentication.
- 1 = “Create File” / “Delete File” is permitted also without application master key authentication (default).

Bit 1: codes whether application master key authentication is needed for file directory access:

- 0 = Successful application master key authentication is required for executing the “Get FID List”, “Get File Settings” and “Get Key Settings” commands.
- 1 = “Get FID List”, “Get File Settings” and “Get Key Settings” commands succeed independently of a preceding application master key authentication (default).

Bit 0: codes whether the application master key is changeable:

- 0 = Application master key is not changeable anymore (frozen).
- 1 = Application master key is changeable (authentication with the current application master key necessary, default).

Change Key (45h)

Func	Len	Parameters
45h	17	Key#(byte) + New Key (16 bytes)
	33	Key#(byte) + New Key (16 bytes) + Old Key (16 bytes)
	18	Key#(byte) + Key Version(byte) + New Key (16 bytes)
	34	Key#(byte) + Key Version(byte) + New Key (16 bytes) + Old Key (16 bytes)

This command allows to change any key stored on the PICC.
If AID=0x00 is selected, the change applies to the PICC master key and therefore only KeyNo = 0x00 is valid (only one PICC master key is present on a PICC). In all other cases (AID ≠ 0x00) the change applies to the specified KeyNo within the currently selected application (represented by it's AID).

Remark:

- To Change any key (except Master Key and the "Change Key" Key), authentication with the "Change Key" is necessary.
- To Change the "Change Key" Key or the Master Key, authentication with the Master Key is necessary.
- Len=17 or 18, in case the Key# used for authentication is the SAME as the Key# parameter to be changed or if "Change Key Access Right" is set 0xE.
- Len=33 or 34, in case the Key# used for authentication is the DIFFERENT from the Key# parameter to be changed or if "Change Key Access Right" is set to a value ≠ 0xE.
- Len=18 or 34, in case the Key version only for AES.

Get Key Version (46h)

Func	Len	Parameters
46h	1	Key#(byte)

The Get Key Version command allows to read out the current key version of any key stored on the PICC.

If AID = 0x00 is selected, the command return the version of the PICC master key and therefore only KeyNo = 0x00 is valid (only one PICC master key is present on a PICC). In all other cases (AID≠0x00) the version of the specified KeyNo within the currently selected application (represented by it's AID) is returned.

Remark:

If you use 3DES or single DES:

Even if the first 8 byte key itself is the same as the second 8 byte, but the key version is coded into one half, it is used as 3DES key and not as a single DES key during authentication and session key generation, see example as below:

Example for single DES keys

```
------------------------------------------------------------------------------
key (hex) = 00  11  22  33  44  55  66  76  00  11  22  33  44  55  66  77
------------------------------------------------------------------------------
version   = 0   1   0   1   0   1   0   0   n/a n/a n/a n/a n/a n/a n/a n/a
------------------------------------------------------------------------------
```

Key version is coded in first 8 byte for 3DES and single DES, so the version would be : 0x54

Apart from AES key, where the key version is stored in the key settings, the key version for a DES key is stored in the so called parity bit. For single double or triple length keys, only 8 left bytes of the key stream are concerned for the key version.

Save Key (6Bh)

Func	Len	Parameters
6Bh	17	Key#(byte) + Key Value 16 Bytes (for AES)
	25	Key#(byte) + Key Value 24 Bytes (for 3K3DES)

The Save Key command allows to save key value to reader's EEPROM and manage by Key#.

Remark:

Key# : 0~13

Create Application (47h)

Func	Len	Parameters
47h	6	AID (long 4 bytes, MSB First) Key Settings (byte) Number of Keys(bit0~3) + Crypto Type(bit6~7) (byte)

The “Create Application” command allows to create new applications on the PICC.

- This command requires that currently selected AID=0x00 which references the card level.
- The 32 bits AID the first parameter of the command. (Range 1 ~ 16777215).
- The second parameter is the Application Master Key Settings as defined in Page 11.
- The last parameter “Number of Key” defines how many keys can be stored within the application for cryptographic purposes.

Bit7	Bit6	Crypto Type
0	0	DES/3DES
0	1	3K3DES
1	0	AES

Note:

All keys are initialized with a string consisting of sixteen 0x00 bytes and therefore.

Delete Application (48h)

Func	Len	Parameters
48h	4	AID (long 4 bytes, MSB First)

The “Delete Application” command allows to permanently deactivate applications on the PICC.

The application which will be deleted is represented by it’s AID, which is the only parameter of this command.

Either a preceding PICC master key authentication or an application master key authentication is required.

Get AID List (49h)

Func	Len	Parameters
49h	0	

The “Get AID List” command return the Application IDentifiers of all active applications on a PICC.

Response:

Func	Len	Parameters
ACK		first AID (long, MSB first) … N AID (long, MSB first)

len = (N) * 4

Example:

:004000 PCD send Activate WakeUp

command:00060704491B815A1B80 PICC response UID

:004100 PCD send RATS:000606067577810280

PICC response ATS

:004A0400000000 PCD select AID=0 (card level):00060100

PICC response ACK

:0043110000000000000000000000000000000000 PCD send Authentication command:0006020000

PICC response ACK

:00**49**00 PCD send “Get AID List”

command:00061000000001000000020000000300000004 PICC response AID List (total 4 AIDs in list)

PICC’s AIDs List as below:

0x00000000 (master, default)

0x00000001 (4 bytes)

0x00000002

0x00000003

0x00000004

Select Application (4Ah)

Func	Len	Parameters
4Ah	4	AID (long, MSB first)

The “Select Application” command allows to select one specific application for further access.

If this parameter is 0x00, the PICC level is selected and any further operations are related to this level.

If an application with the specified AID is found in the application directory of the PICC, the subsequent commands interact with this application.

Format PICC (4Bh)

Func	Len	Parameters
4Bh	0	

This command releases the PICC user memory.

Remark:

This command always requires a preceding authentication with the PICC master key.

Get Card Version (4Ch)

Func	Len	Parameters
4Ch	0	

This command returns manufacturing related data of the PICC.

Response Version Info (28 bytes):

Field		Size (byte)	Value
H/W	vendor ID	1	0x04 for NXP
	type	1	0x01
	sub type	1	0x01
	major version	1	
	minor version	1	
	storage size	1	0x18 = 4096 bytes
	protocol type	1	0x05 = ISO14443-2 and -3
S/W	vendor ID	1	0x04 for NXP
	type	1	0x01
	sub type	1	0x01
	major version	1	
	minor version	1	
	storage size	1	0x18 = 4096 bytes
	protocol type	1	0x05 = ISO14443-3 and -4
UID		7	
production batch number		5	
calendar week of production		1	
year of production		1	

Get File List (4Dh)

Func	Len	Parameters
4Dh	0	

This command returns the File IDentifiers of all active files within the currently selected application.

Response:

Func	Len	Parameters
ACK	n	first FID (byte) … n FID (byte)

Set / Get File Settings (4Eh)

Func	Len	Parameters
4Eh	1 (Get)	FID (byte)
	4 (Set)	FID (byte) + Comm. Mode (byte) + Access Right (Int, MSB first)

The “Get File Settings” (len=1) command allows to get information on the properties of a specific file. The information provided by this command depends on the type of the file which is queried.

Response (for Get File Settings):

for file type = data file (0x00) or backup file (0x01)

ACK	Len=8	File Type	Comm. Mode	Access Right	File Size
		byte	byte	Int (MSB first)	Long (MSB First)

for file type = value file (0x02)

ACK	Len=17	File Type	Comm. Mode	Access Right	Lower limit	Upper limit	Limited credit value	Limited credit enabled
		byte	byte	Int (MSB first)	Long (MSB First)	Long (MSB First)	Long (MSB first)	byte

for file type = linear record file (0x03) or cyclic record file (0x04)

ACK	Len=16	File Type	Comm. Mode	Access Right	Record size	Max. number of records	Current number of records
		byte	byte	Int (MSB first)	Long (MSB First)	Long (MSB First)	Long (MSB first)

The “Set File Settings” (len=4) command changes the access parameters of an existing file.

The Comm. Mode byte:

Communication Mode	Bit 7~2	Bit 1	Bit 0
Plain communication	RFU	ignored	0
Plain communication secured by DES/3DES MACing	RFU	0	1
Fully DES/3DES enciphered communication	RFU	1	1

The Access Right Field (16 bits):

15 12	11 8	7 4	3 0
Read Access	Write Access	Read & Write Access	Change Access Right
MSB			LSB

Each of the Access Rights is coded in 4 bits, one nibble, Each nibble represents a link to one of the keys stored within the respective application’s key file.

One nibble (4 bits) allows to code 16 different values. If such a value is set to a number between 0 and 13 (max. 14 keys), this references a certain key within the application’s key file, provided that the key exists (selecting a non-existing key is not allowed).

If the number is coded as 14 (0xE) this means “free access”. Thus the regarding access is granted always with and without a preceding authentication, directly after the selection of the respective application.

The number 15 (0xF) defines the opposite of “free” access and has the meaning “deny” access. Therefore the respective linked Access Rights is always denied.

Create Std Data File (4Fh)

Func	Len	Parameters
4Fh	8	FID (byte) Comm. Mode (byte) Access Right (int, MSB First) File Size (long, MSB First)

The “Create Std Data File” command is used to create files for the storage of plain unformatted user data within an existing application on the PICC.

Remark:
The DESFire internally allocates NV-memory in multiples of 32 bytes. Therefore a file creation command with “File Size” parameter 0x00000001 (1 byte file size) will internally consume the same amount of NV-memory as a 32 bytes.

Create Backup Data File (50h)

Func	Len	Parameters
50h	8	FID (byte) Comm. Mode (byte) Access Right (int, MSB First) File Size (long, MSB First)

The “Create Backup Data File” command is used to create files for the storage of plain unformatted user data within an existing application on the PICC, additionally supporting the feature of an integrated backup mechanism.

Remark:

- The parameter FID only the first 8 files within an application feature the integrated backup mechanism. (only FID=0x00 to 0x07 is allowed).
- Due to the mirror image a Backup data file always consumes DOUBLE the NV-memory on the PICC compared to a Std Data File with the same specified “File Size”.

Create Value File (51h)

Func	Len	Parameters
51h	17	FID (byte) Comm. Mode (byte) Access Right (int, MSB First) Lower Limit(long, MSB first), Upper Limit(Long, MSB First), Initial Value(Long, MSB First), Limited Credit Enabled(byte)

The “Create Value File” command is used to create files for storage and manipulation of 32 bite signed integer values within an existing application on the PICC.

Remark:

- The upper limit has to be ≧ lower limit, otherwise an error message would be sent by the PICC and thus the file would not be created.
- The Limited Credit feature, see Limit Credit. Here 0x00 means that “Limited Credit Enabled” is disable and 0x01 enables this feature.
- Value File feature always the integrated backup mechanism. Therefore every access changing the value needs to be validated using the “Commit Transaction” command.

Create Linear Record File (52h)

Func	Len	Parameters
52h	12	FID(byte), Communication Mode(byte), Access Right(int, MSB first), Record Size(long, MSB first), Max. Num of Records(long, MSB first)

The “Create Linear Record File” command is used to create files for multiple storage of structural data, for example for loyalty programs, within an existing application on the PICC. Once the file is filled completely with data records, further writing to the file is not possible unless it is cleared, see command “Clear Record File”.

Remark:

- Thus the entire file size in the PICC NV-memory is given by “Record Size” x “Max. Num of Records”.
- Linear Record Files feature always the integrated backup mechanism. Therefore every access appending a record needs to be validated using the “Commit Transaction” command.

Create Cyclic Record File (53h)

Func	Len	Parameters
53h	12	FID(byte), Communication Mode(byte), Access Right(int, MSB first), Record Size(long, MSB first), Max. Num of Records(long, MSB first)

The “Create Cyclic Record File” command is used to create files for multiple storage of structural data, for example for loyalty programs, within an existing application on the PICC. Once the file is filled completely with data records, the PICC automatically overwrites the oldest record with the latest written one. This wrap is fully transparent for the PCD.

Remark:

- Thus the entire file size in the PICC NV-memory is given by “Record Size” x “Max. Num of Records”.
- Cyclic Record Files feature always the integrated backup mechanism. Therefore every access appending a record needs to be validated using the “Commit Transaction” command.
- As the backup feature consumes one record, the “Max. Num Of Records” needs to be one largerthen the application requires.

Delete File (54h)

Func	Len	Parameters
54h	1	FID(byte),

The “Delete File” command permanently deactivates a file within the file directory of the currentlhy selected application.

Remark:

- Depending on the application master key settings, a preceding authentication with the application master key is required.
- Allocate memory blocks associated with the deleted file are not set free. The FID of the deleted file can be re-used to create a new file within that application.
- To release memory blocks for re-use, the whole PICC user NV-memory needs to be erased using the “Format PICC” command.

Read Data (55h)

Func	Len	Parameters
55h	9	FID(byte), Offset(long, MSB first), Length(long, MSB first),

The “Read Data” command allows to read data from Std Data File or Backup Data File.

The Offset parameter is of 4 byte length and codes the starting position for the read operation within the file. This parameter has to be in the range from 0x00000000 to file size-1.

The Length parameter is also 4 byte long and specifies the number of data bytes to be read. This parameter has to between in the range from 0x00000000 to 0x00000080. If the Length is coded as 0x00000000, the entire data file, starting from the position specified in the offset value, is read.

Write Data (56h)

Func	Len	Parameters
56h	9	FID(byte), Offset(long, MSB first), Length(long, MSB first),

The “Write Data” command allows to write data to Std Data File or Backup Data File.
Each time the max data size written is 128 bytes.

Remark:

- The “Write Data” command requires a preceding authentication either with the key specified for “Write” or “Read&Write” access.
- If the “Write Data” operation is performed on a Backup Data File, it is necessary to validate the written data with a “Commit Transaction” command. An “Abort Transaction” command will invalidate all changes.
- If data is written to Std Data Files (without integrated backup mechanism), data is directly programmed into the visible NV-memory of the file. The new data is immediately available to any following “Read Data” command performed on that file.

Get Value (57h)

Func	Len	Parameters
57h	9	FID(byte),

The “Get Value” command allows to read the currently stored value from Value File.

Response:
Value (long, MSB first)

Remark:

- The “Get Value” command requires a preceding authentication with the key specified for Read, Write or Read&Write access.
- After updating a value file’s value but before issuing the “Commit Transaction” command, the “Get Value” command will always retrieve the old, unchanged value which is still the valid one.

Credit (58h)

Func	Len	Parameters
58h	5	FID(byte), Amount(long, MSB first),

The Credit command allows increasing a value in a Value File.

Remark:

- It is necessary to validate the updated value with a “Commit Transaction” command, an “Abort Transaction” command will invalidate all changes.
- The value modifications of Credit, Debit and Limit-Credit commands are cumulated until a “Commit Transaction” command is issued.
- The Credit commands do NEVER modify the Limited Credit Value of a Value File. However, if the Limited Credit value needs to be set to 0, a Limited-Credit with value 0 can be used.
- The Credit command requires a preceding authentication with the key specified for “Read&Write” access.

Debit (59h)

Func	Len	Parameters
59h	5	FID(byte), Amount(long, MSB first),

The Debit command allows decreasing a value in a Value File.

Remark:

- It is necessary to validate the updated value with a “Commit Transaction” command, an “Abort Transaction” command will invalidate all changes.
- The value modifications of Credit, Debit and Limit-Credit commands are cumulated until a “Commit Transaction” command is issued.
- The Credit command requires a preceding authentication with the key specified for Read, Write and Read&Write access.
- If the usage of the Limited-Credit feature is enabled, the new limit for a subsequent “Limit Credit” command is set to the sum of Debit commands within one transaction before issuing a “Commit Transaction” command. This assures that a “Limit Credit” command can not re-book more values than a debiting transaction deducted before.

Limited Credit (5Ah)

Func	Len	Parameters
5Ah	5	FID(byte), Amount(long, MSB first),

The “Limited Credit” command allows a limited increase of a value stored in a Value File without having full Read&Write permissions to the file. This feature can be enabled or disabled during value file creation.

Remark:

- It is necessary to validate the updated value with a “Commit Transaction” command, an “Abort Transaction” command will invalidate all changes.
- The value modifications of Credit, Debit and Limit-Credit commands are cumulated until a “Commit Transaction” command is issued.
- The Limited-Credit command requires a preceding authentication with the key specified for Read, Write and Read&Write access.
- The value for “Limited Credit” is limited to the sum of the Debit commands on this value file within the most recent transaction containing at least one Debit. After executing the “Limited Credit” command the new limit is set to 0 regardless of the amount which has been re-booked. Therefore the “Limited Credit” command can only be used once after a Debit transaction.

Write Record (5Bh)

Func	Len	Parameters
5Bh	9	FID(byte), Offset (Long, MSB first), Length (Long, MSB First)

The “Write Record” command allows writing data to a record in a Cyclic or Linear Record File.

The Offset parameter offset within one single record (in byte). This parameter has to be in the range from 0x000000000 to record size-1.

The Length parameter has to be in the range from 0x00000000 to record size - Offset.

Remark:

- The “Write Record” command appends one record at the end of the linear record file, it erases and overwrites the oldest record in case of a cyclic record file if it is already full. The entire new record is cleared before data is written to it.
- If no “Commit Transaction” command is sent after a “Write Record” command, the next “Write Record” command to the same file writes to the already created record. After sending a “Commit Transaction” command, a new “Write Record” command will create a new record in the record file. An “Abort Transaction” command will invalidate all changes.
- After issuing a “Clear Record File” command, but before a “Commit Transaction” / “Abort Transaction” command, a “Write Record” command the same record file will fail.
- The “Write Record” command requires a preceding authentication either with the key specified for “Write” or “Read&Write” access.

Read Records (5Ch)

Func	Len	Parameters
5Bh	9	FID(byte), Record#(long, MSB first), NRecToRead (long, MSB first),

Example: File ID =1

Record#	Record Data
0	4444444444
1	3333333333
2	2222222222
3	1111111111

Read Record (FID=1 , Record#=1 , NRecToRead=2)
Result= 3333333333 and 2222222222

The "Read Record" command allows reading out a set of complete records from a Cyclic or Linear Record File.

The Record# parameter offset of the newest record which is read out. In case of 0x00000000 the latest record is read out. The Record# value must be in the range from 0 to number of exist records-1.

The "NRecToRead" parameter is another 4 bytes which code the Number of Records. To be Read from the PICC. Records are always transmitted by the PICC in chronological order (= starting with the oldest, which is number of records – 1 before the one addressed by the given offset). If this parameter is set to 0x00000000 then all records, from the oldest record up to and including the newest record are read. The allowed range for the number of records parameter is from 0x00000000 to number of existing records – Record#.

Response:
Return ACK with Record Size (unit: byte).

Remark:

- In cyclic record files the maximum number of stored valid records is one less than the number of records specified in the "Create Cyclic Record File" command.
- A "Read Records" command on any empty records file will result in an error.
- The "Read Records" command requires a preceding authentication either with the key specified for "Read" or "Read&Write" access.

Clear Record File (5Dh)

Func	Len	Parameters
5Dh	1	FID(byte)

The “Clear Record File” command allows resetting a Cyclic or Linear Record File to the empty state.

After executing the “Clear Record File” command but before “Commit Transaction”, all subsequent “Write Record” commands will fail. The “Read Records” command will return the old still valid records.

After the “Commit Transaction” command is issued, a “Read Records” command will fail, “Write Record” command will be successful. An “Abort Transaction” command will invalidate the clearance.

Remark:
Full “Read&Write” permission on the file is necessary for executing this command.

Commit Transaction (5Eh)

Func	Len	Parameters
5Eh	0	

The “Commit Transaction” command allows validating all previous write access on Backup Data Files, Value Files and Record Files within one application.

The “Commit Transaction” command validates all write access to files with integrated backup mechanisms:

- Backup Data Files
- Value Files
- Linear Record Files
- Cyclic Record Files

Remark:
The “Commit Transaction” is typically the last command of a transaction before the ISO 14443-4 DESELECT command or before proceeding with another application.

Abort Transaction (5Fh)

Func	Len	Parameters
5Fh	0	

The “Abort Transaction” command allows invalidating all previous write access on Backup Data Files, Value Files and Record Files within one application.

This is useful to cancel a transaction without the need for re-authentication to the PICC, which would lead to the same functionality.

The “Abort Transaction” command invalidates all write access to files with integrated backup mechanisms without changing the authentication status:

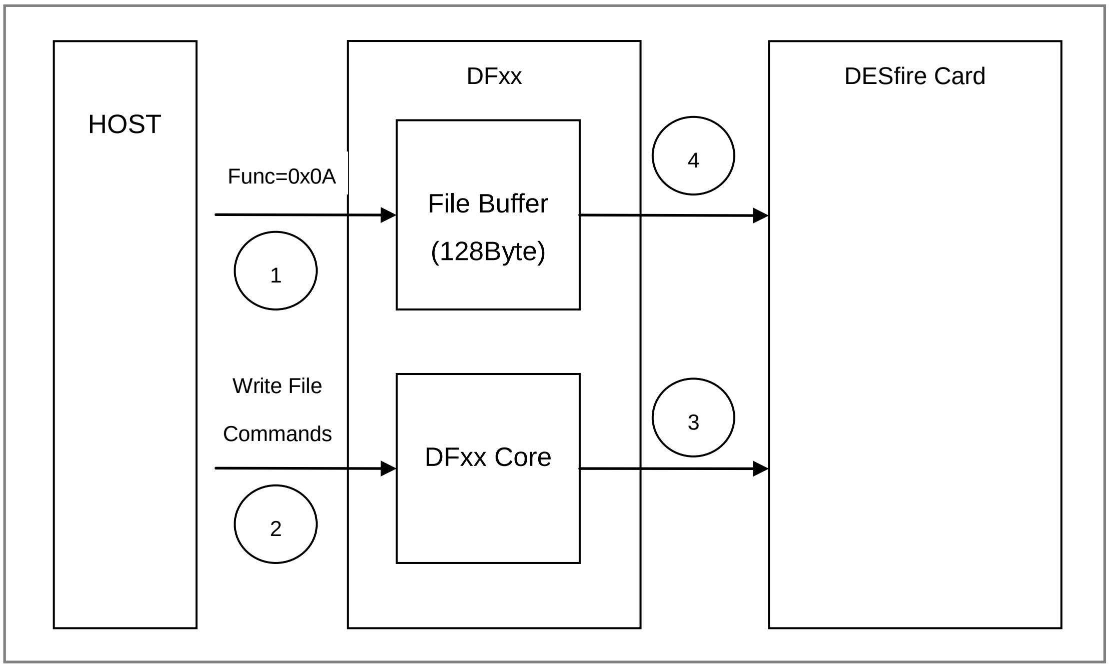
- Backup Data Files
- Value Files
- Linear Record Files
- Cyclic Record Files

Error Code

Hex Code	Status	Descriptions
0x00	OPERATION_OK	Successful operation
0x0C	NO_CHANGES	No changes done to backup files, Commit Transaction / Abort Transaction not necessary
0x0E	OUT_OF_EEPROM_ERROR	Insufficient NV-Memory to complete command
0x1C	ILLEGAL_COMMAND_CODE	Command code not supported
0x1E	INTEGRITY_ERROR	CRC or MAC does not mach data Padding bytes not valid
0x40	NO_SUCH_KEY	Invalid key number specified
0x7E	LENGTH_ERROR	Length of command string invalid
0x9D	PERMISSION_DENIED	Current configuration / status does not allow the requested command
0x9E	PARAMETER_ERROR	Value of the parameter(s) invalid
0xA0	APPLICATION_NOT_FOUND	Requested AID not present on PICC
0xA1	APP_INTEGRITY_ERROR	Unrecoverable error within application, application will be disabled *
0xAE	AUTHENTICATION_ERROR	Current authentication status does not allow the requested command
0xAF	ADDITIONAL_FRAME	Additional data frame is expected to be sent
0xBE	BOUNDARY_ERROR	Attempt to read/write data from/to beyond the file's/record's limits. Attempt to exceed the limits of a value file
0xC1	PICC_INTEGRITY_ERROR	Unrecoverable error within PICC, PICC will be disabled
0xCA	COMMAND_ABORTED	Previous Command was not fully completed Not all Frames were requested or provided by the PCD
0xCD	PICC_DISABLED_ERROR	PICC was disabled by an unrecoverable error
0xCE	COUNT_ERROR	Number of Applications limited to 28, no additional CreateApplication possible
0xDE	DUPLICATE_ERROR	Creation of file/application failed because file/application with same number already exists
0xEE	EEPROM_ERROR	Could not complete NV-write operation due to loss of power, internal backup/rollback mechanism activated
0xF0	FILE_NOT_FOUND	Specified file number does not exist
0xF1	FILE_INTEGRITY_ERROR	Unrecoverable error within file, file will be disabled

How to Read/Write a File

Write A File



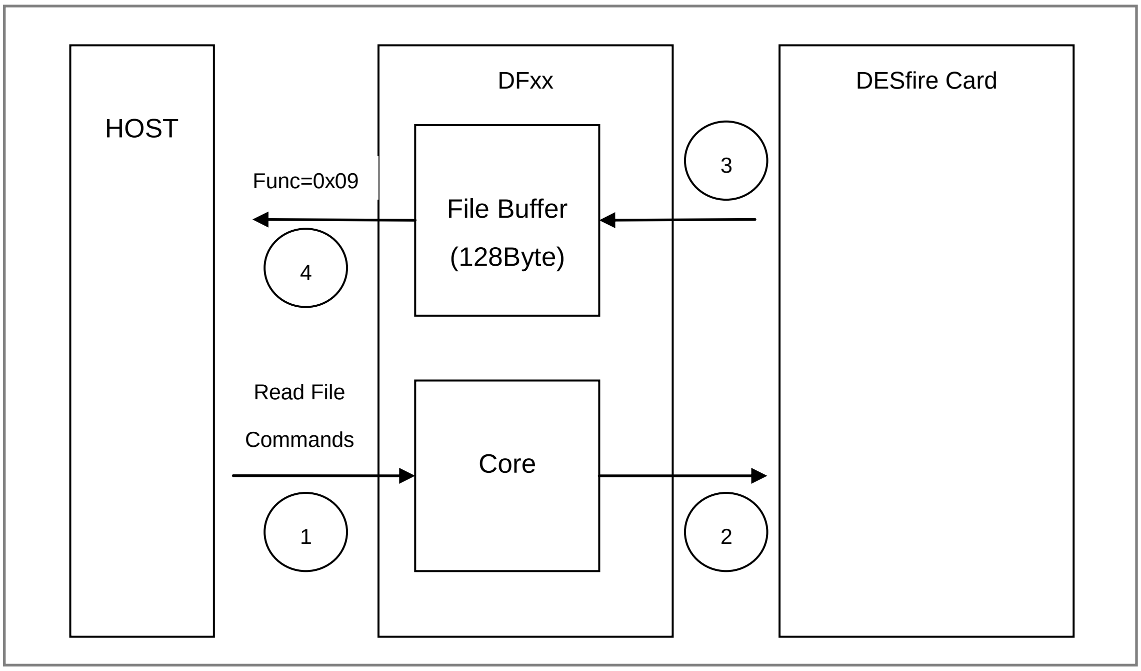
1. Prepare the file data to File Buffer (use func=0x0A Set Register with File Buffer Address)

2. Send Write File (or Record) commands to DFxx.

3. DFxx send the DESfire RF commands to Card.

4. DFxx send file data form buffer when card accept above commands.

Read A File



1. Host send the Read File (or Record) commands to DFxx.

2. DFxx send the DESfire RF commands to Card.

3. Card accept and response the file contents to File Buffer.

4. Host send func=0x09 (Get Registers) with File Buffer Address to get the file data from File Buffer.

Remark:

1. Using func 0x60 to get the File Buffer Address of module (or reader); the File Buffer Address will be used in Get/Set Register commands for File buffer use.
2. Using func=0x09 with File Buffer Address and Length=Zero to get the Maxima File Buffer Size for our feature module.

History

Rev. A (Jason)

August 5, 2008 First Edition

Rev.B (Jason)

January 12, 2010 Add EV1 Support

Rev.C (Terry)

November 4, 2010 Fix some mistake.

Rev.D (Jason)

April 13, 2011 Add Save DESfire Key Command Set. (Page 10, Page 14)

Rev.E (Jason)

Fixed Crypto Type Code. (Page 10)

Add How to Read/Write a File (Page 42~43)

Add Key Version remark (Page 14)

PROMAG®

GIGA-TMS INC.

http://www.gigatms.com.tw

mailto:promag@gigatms.com.tw

TEL : +886-2-26954214

FAX : +886-2-26954213

Office: 8F, No. 31,Lane 169, Kang-Ning St.,Hsi-Chih, Taipei, Taiwan