ETSI TS 102 226 V6.2.0 (2002-10)

Technical Specification

Smart Cards;
Remote APDU structure for UICC based applications
(Release 6)

Reference

RTS/SCP-000285r2

Keywords

protocol, smart card

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, send your comment to:
editor@etsi.org

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Project Smart Card Platform (SCP).

It is based on work originally done in the 3GPP in TSG-terminals WG3 and ETSI SMG.

The contents of the present document are subject to continuing work within EP SCP and may change following formal EP SCP approval. If EP SCP modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

0 early working draft;

1 presented to EP SCP for information;

2 presented to EP SCP for approval;

3 or greater indicates EP SCP approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document defines the remote management of files and applications on the UICC based on the secured packet structure specified in TS 102 225 [1].

It specifies the APDU format for remote management.

- Furthermore the document specifies:a set of commands coded according to this APDU structure and used in the remote file management on the UICC. This is based on TS 102 221 [2].
- A set of commands coded according to this APDU structure and used in the remote application management on the UICC. This is based on the Open Platform Card Specification [4].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] ETSI TS 102 225: "Smart Cards; Secured packet structure for UICC based applications (Release 6)".

[2] ETSI TS 102 221: "Smart cards; UICC-Terminal interface; Physical and logical characteristics (Release 6)".

[3] ETSI TS 102 223: "Smart cards; Card Application Toolkit (CAT) (Release 5)".

[4] "Open Platform Card Specification version 2.0.1" (see http://www.globalplatform.org/).

[5] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers (Release 5)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 102 225 [1] and TS 101 220 [5] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in TS 102 225 [1] and the following apply:

ADD	Access Domain Data
ADP	Access Domain Parameter
DAP	Data Authentication Pattern
KIK	Key Identifier for protecting KIc and KID
MSLD	Minimum Security Level Data

4 Overview of Remote Management

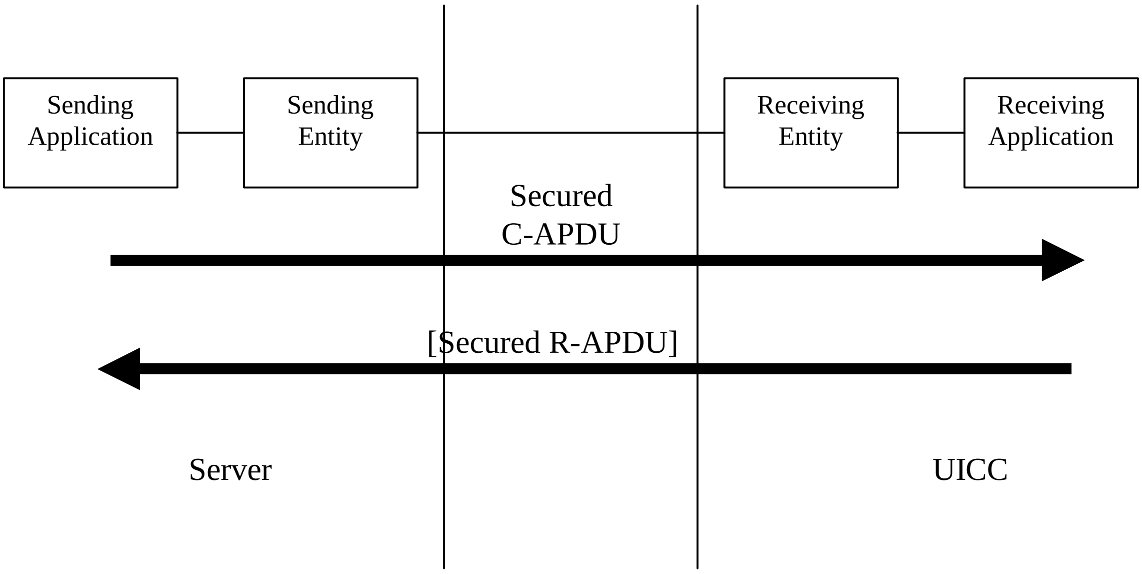

Figure 1: Remote Management

All data exchanged between the Sending Entity and Receiving Entity shall be formatted as "Secured data" according to TS 102 225 [1]:

1) The parameter(s) in the "Secured data" is either a single command, or a list of commands, which shall be processed sequentially.

2) The application shall take parameters from the "Secured data" and shall act upon the files or applications according to these parameters.

3) A Command "session" is defined as starting upon receipt of the parameter/command list, and ends when the parameter list in the "Secured data" is completed, or when an error is detected which shall halt further processing of the command list.

4) At the beginning and end of a Command "session" the logical state of the UICC as seen from the terminal shall not be changed to an extent sufficient to disrupt the behaviour of the terminal. If changes in the logical state have occurred that the terminal needs to be aware of, the application on the UICC may issue a REFRESH command according to TS 102 223 [3]. However, this is application dependent and therefore out of scope of the present document.

5 Remote APDU Format

5.1 Remote command coding

A command string may contain a single command or a sequence of commands. Each command is coded according to the generalized structure defined below; each element other than the Data field is a single octet (see TS 102 221 [2]).

Class byte (CLA)	Instruction code (INS)	P1	P2	P3	Data

If a command has P3='00', then the UICC shall send back all available response parameters/data.

5.2 Response coding

If a proof of Receipt is required by the sending entity, the Additional Response Data sent by the Remote Management Application shall be formatted according to table 1.

Table 1: Format of additional response data

Length	Name
1	Number of commands executed within the command script (see note)
2	Last executed command status word
X	Last executed command response data if available (i.e. if the last command was an outgoing command)
NOTE:	This field shall be set to '01' if one command was executed within the command script, '02' if two commands were executed, etc.

6 Remote File Management

Access conditions for the files as seen by the UICC resident application, are not standardized. These access conditions may be dependent on the level of security applied to the "Secured Data".

6.1 Input Commands

The standardized commands are listed in table 2. The commands are as defined in TS 102 221 [2].

Table 2: Input Commands

Operational command
SELECT (see below)
UPDATE BINARY
UPDATE RECORD
SEARCH RECORD
INCREASE
VERIFY PIN
CHANGE PIN
DISABLE PIN
ENABLE PIN
UNBLOCK PIN
DEACTIVATE FILE
ACTIVATE FILE

The SELECT command shall not include the selection by DF name corresponding to P1='04' in the Command Parameters of SELECT (see TS 102 221 [2]).

6.2 Output Commands

The commands listed in table 3 are defined in TS 102 221 [2].

These commands shall only occur once in a command string and, if present, shall be the last command in the string. The Response Data shall be placed in the Additional Response Data element of the Response Packet.

Table 3: Output commands

Operational command
READ BINARY READ RECORD GET RESPONSE

7 Remote Application Management

Remote Application Management on a UICC card includes the ability to load, install, and remove applications. This management is under the responsibility of the Card Issuer or any entity with delegated management privileges.

The concept of embedding APDUs in a command packet is as defined in clause 5.

7.1 Remote Application Management Application behaviour

7.1.1 Load File Loading

The Load File Loading process allows to load new Load Files onto the UICC through a loading session with the card.

A loading session consists of the sequence of commands as described in figure 2.

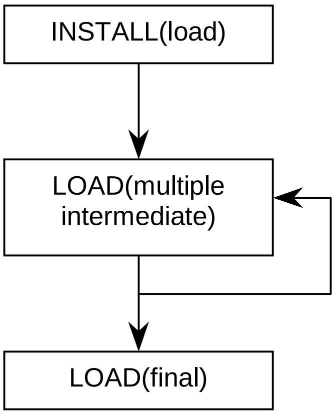

Figure 2: Loading session sequence of commands

Depending on the application size, several Command Packets might be used for the Load File loading.

7.1.2 Application Installation

The Application Installation process allows to install a new application onto the UICC. The installation may only be performed if the corresponding Load File has already been loaded onto the card. The Application Installation is performed using the INSTALL (install) command (see clause 7.2.1).

7.1.3 Load File Removal

The Load File Removal process is performed using the DELETE command (see clause 7.2.1). The Load File removal procedure shall be performed by the UICC as defined below:

1) If non-removed applications installed from this Load File remain, the card shall reject the removal with the corresponding status error code.

2) If the Load File is referred by other Load File(s), the card shall reject the removal with the corresponding status error code.

7.1.4 Application Removal

The Application Removal process shall be performed using the DELETE command (see clause 7.2.1). The UICC shall remove the components that make up the application.

7.1.5 Application Locking/Unlocking

The Application locking (and unlocking) procedure allows to disable (and enable) an application using the SET STATUS command (see clause 7.2.1). When an application is locked, it shall not be possible to be triggered or selected, and all of its menu entries will be disabled (i.e. removed from the SET UP MENU command).

7.1.6 Application Parameters Retrieval

The Application Parameters Retrieval procedure allows to remotely request the parameters of an application. This procedure is performed using the GET DATA command (see clause 7.2.2).

7.2 Commands coding

Commands are coded as for the Remote File Management procedure, each command is coded as an APDU.

Commands shall be executed by the Card Manager or a Security Domain depending on the TAR in the case of Remote Application Management. The messages for the Card Manager shall have a TAR value set to '000000' in hexadecimal.

The minimum security applied to a Secured Packet containing Application Management Commands shall be integrity using CC or DS, and in all cases, this security shall replace Data Authentication Patterns used in GlobalPlatform commands for secure messaging (This corresponds to the message DAP generated for secure messaging).

A complying card shall support at least the DES CBC algorithm for cryptographic computations.

Command status words placed in the Additional Response Data element of the Response Packet shall be coded according to the Open Platform Card Specification [4].

7.2.1 Input Commands

7.2.1.1 List of input commands

Table 4 extends table 2 defined in clause 6.1.

Table 4: Application Management input commands

Operational command
DELETE
SET STATUS
INSTALL
LOAD
PUT KEY

7.2.1.2 Description of the input commands

7.2.1.2.1 DELETE

The Delete command shall be coded according to the Open Platform Card Specification [4].

7.2.1.2.2 SET STATUS

The Set Status command shall be coded according to the Open Platform Card Specification [4].

7.2.1.2.3 INSTALL

The Install command shall be coded according to the Open Platform Card Specification [4].

7.2.1.2.3.1 Install (Load)

The Load File DAP field is a Cryptographic Checksum, a Digital Signature or a Redundancy Check calculated over the CAP file as transmitted in the subsequent Load commands. The exact generation of this field is outside the scope of the present document. If a DES algorithm in CBC mode is used the initial chaining value shall be zero. If padding is required, the padding octets shall be coded hexadecimal '00'.

The Load Parameter Field of the Install (Load) command shall be coded as follows.

Presence	Length	Name
Mandatory	1	Tag of System Parameters constructed field 'EF'
	1	Length of System Parameters constructed field
	4-n	System Parameters constructed value field

The System Parameters value field of the Install (Load) command shall be coded as follows.

Presence	Length	Name
Mandatory	1	Tag of non volatile memory space required for Load File loading field: 'C6'
	1	Length of non volatile memory space required for Load File loading field
	2	Non Volatile memory space (in bytes) required for Load File loading (see note)
Optional	1	Tag of non volatile memory requirements for installation field: 'C8'
	1	Length of non volatile memory requirements for installation
	2	Non volatile memory required for installation in byte
Optional	1	Tag of volatile memory requirements for installation field: 'C7'
	1	Length of memory requirements for installation
	2	Volatile memory required for installation in byte
NOTE: The memory space required indicates the minimum size that shall be available onto the card to download the application. The UICC must reject the application downloading if the required size is not available on the card.		

7.2.1.2.3.2 Install (Install)

Toolkit registration is only active if the toolkit application is at the state selectable, for example if the application is registered for the event Menu Selection it shall only appear in the menu if the application is in the selectable state.

The Install Parameter Field of the Install (Install) command shall be coded as follows.

Presence	Length	Name
Mandatory	1	Tag of System Parameters constructed field 'EF'
	1	Length of System Parameters constructed field
	15-n	System Parameters constructed value field.
Mandatory	1	Tag of Application specific parameters field: 'C9'
	1	Length of Application specific Parameters field
	0-n	Application specific Parameters

The System Parameters value field of the Install (Install) command shall be coded as follows.

Presence	Length	Name
Mandatory	1	Tag of non volatile memory requirements for installation field: 'C8'
	1	Length of non volatile memory requirement for installation (see clause 7.2.1.2.3.2.2)
	2	Non volatile memory required for installation in byte (see clause 7.2.1.2.3.2.2)
Mandatory	1	Tag of volatile memory requirements for installation field: 'C7'
	1	Length of volatile memory requirement for installation (see clause 7.2.1.2.3.2.2)
	2	Volatile memory required for installation in byte (see clause 7.2.1.2.3.2.2)
Mandatory	1	Tag of toolkit application specific parameters field: 'CA'
	1	Length of toolkit application specific parameters field
	6-n	Toolkit Application specific Parameters (see clause 7.2.1.2.3.2.1)

Even if the length of the non volatile or volatile memory is present in the Install(Load) command, the card shall use the values indicated in the Install(Install) command at instantiation, should these values differ.

The format of the install method buffer provided by the Install (Install) command shall be the one specified in the Open Platform Card Specification [4].

The application may invoke the register(bArray, bOffset, bLength) or the register() method: the application instance shall be registered with the instance AID present in the Install (Install) command.

If the register (bArray, bOffset, bLength) is invoked, the AID passed in the parameters shall be the instance AID provided in the install method buffer.

If the register() method is invoked the instance AID present in the Install(Install) command and the AID within the Load File, as specified in Open Platform Card Specification [4], should be the same.

7.2.1.2.3.2.1 Toolkit Application Specific Parameters

The toolkit application specific parameters field is used to specify the terminal and UICC resources the application instance can use. These resources include the timers, the Bearer Independent protocol channels, menu items for the Set Up Menu, the Minimum Security Level and the TAR Value(s) field. The Network Operator or Service Provider can also define the menu position and the menu identifier of the menus activating the application. The following format is used to code the application parameters.

Length	Name	Value
1	Length of Access Domain field	
1-p	Access Domain	
1	Priority level of the Toolkit application instance (see clause 7.2.1.2.3.2.5)	
1	Maximum number of timers allowed for this application instance	
1	Maximum text length for a menu entry	
1	Maximum number of menu entries allowed for this application instance	= m
1	Position of the first menu entry ('00' means last position)	\
1	Identifier of the first menu entry ('00' means do not care)	|
	….	| = 2*m bytes
1	Position of the last menu entry ('00' means last position)	|
1	Identifier of the last menu entry ('00' means do not care)	/
1	Maximum number of channels for this application instance	
1	Length of Minimum Security Level field	
0-q	Minimum Security Level (MSL) (see clause 7.2.1.2.3.2.2)	
1	Length of TAR Value(s) field	
3*y	TAR Value(s) of the Toolkit Application instance	

If the maximum number of timers required is greater than '08' (maximum numbers of timers specified in TS 102 223 [3], the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

If the maximum number of channels required is greater than '07' (maximum numbers of channels specified in TS 102 223 [3]), the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

The position of the new menu entries is an absolute position among the existing ones.

A part of the item identifier shall be under the control of the card system and the other part under the control of the card issuer. Item identifiers are split in two ranges:

- [1,127] under control of the card issuer;
- [128,255] under the control of the toolkit framework.

If the requested item identifier is already allocated, or in the range [128,255], then the card shall reject the install command. If the requested item identifier is '00', the card shall take the first free value in the range [128,255].

7.2.1.2.3.2.2 Coding of the Minimum Security Level

The Minimum Security Level (MSL) is used to specify the minimum level of security to be applied to Secured Packets sent to the application. The Receiving Entity shall check the Minimum Security Level before processing the security of the Command Packet. If the check fails, the Receiving Entity shall reject the messages and a Response Packet with the "Insufficient Security Level" Response Status Code (see table 5 of TS 102 225 [1]) shall be sent if required.

If the length of the Minimum Security Level field is zero, no minimum security level check shall be performed by the receiving entity.

If the length of the Minimum Security Level field is greater than zero, the Minimum Security Level field shall be coded according to the following table:

Length	Name
1	MSL Parameter
q-1	MSL Data

The MSL Data coding and length is defined for each MSL Parameter.

MSL Parameter

The possible values for the MSL Parameter are:

Value	Name	Support	MSL Data length
'00'	RFU	RFU	N/A
'01'	Minimum SPI1	Optional	1
'02' to '7F'	RFU	RFU	N/A
'80' to 'FE'	Reserved for Proprietary Mechanisms	Optional	N/A
'FF'	RFU	RFU	N/A

Minimum SPI1

The Minimum Security Level Data for the Minimum SPI1 MSL parameter shall use the same coding as the first octet of the SPI of a command packet (see clause 5.1.1 of TS 102 225 [1]).

The first octet of the SPI field in the incoming message Command Packet (SPI1) shall be checked against the Minimum Security Level Data (MSLD) byte by the receiving entity according to the following rules:

- If SPI1.b2b1 is equal to or greater than MSLD.b2b1; and
- if SPI1.b3 is equal to or greater than MSLD.b3; and
- if SPI1.b5b4 is equal to or greater than MSLD.b5b4,

then the Message Security Level is sufficient and the check is successful, otherwise the check is failed.

7.2.1.2.3.2.3 Memory space

The memory space required indicates the minimum size that shall be available on the card to download the application. The UICC shall reject the application downloading if the required size is not available on the card.

7.2.1.2.3.2.4 Access domain

The access domain is used to specify the UICC files that may be accessed by the application and the operations allowed on these files. The Access Domain field is formatted as follows.

Length	Name
1	Access Domain Parameter (ADP)
p-1	Access Domain Data (ADD)

The Access Domain Data coding and length is defined for each Access Domain Parameter.

- Access Domain Parameter

This parameter indicates the mechanism used to control the application instance access to the File System.

Value	Name	Support	ADD length
'00'	Full access to the File System	Mandatory	0
'01'	Reserved (for APDU access mechanism)	-	-
'02'	Reserved (for 3GPP access mechanism)	-	-
'03' to '7F'	RFU	RFU	RFU
'80' to 'FE'	Proprietary mechanism	-	-
'FF'	No access to the File System	Mandatory	0

The access rights granted to an application and defined in the access domain parameter shall be independent from the access rights granted at the UICC/Terminal interface.

NOTE: This implies in particular that the status of a secret code (e.g. disabled PIN1, blocked PIN2, etc.) at the UICC/Terminal interface does not affect the access rights granted to an application.

If an application with Access Domain Parameter 'FF' (i.e. No Access to the File System) tries to access a file the framework shall throw an exception.

If an application has Access Domain Parameter '00' (i.e. Full Access to the File System), all actions can be performed on a file except the ones with NEVER access condition.

If the Access Domain Parameter requested is not supported, the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

- UICC access mechanism

7.2.1.2.3.2.5 Priority level of the Toolkit application

The priority specifies the order of activation of an application compared to the other application registered to, the same event. If two or more applications are registered to the same event and have the same priority level, the applications are activated according to their installation date (i.e. the most recent application is activated first). The following values are defined for priority:

- '00': RFU
- '01': Highest priority level
- ...
- 'FF': Lowest priority level

7.2.1.2.3.2.6 Coding of TAR Value(s) field

The TAR is defined and coded according to TS 101 220 [5].

It is possible to define several TAR Values at the installation of a Toolkit Application.

The TAR Value(s) field shall be coded according to the following table:

Bytes	Description	Length
1-3	TAR Value 1	3
4-6	TAR Value 2	3
…	…	…
3*y-2 to 3*y	TAR Value y	3

If the length of TAR Value(s) is zero, the TAR may be taken out of the AID if any.

If the length of the TAR Value(s) is greater than zero then the application instance shall be installed with the TAR Value(s) field defined above and the TAR indicated in the AID if any shall be ignored.

If a TAR Value(s) is already assigned on the card for a Toolkit Application instance or if the length of TAR Value(s) field is incorrect, the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

7.2.1.2.4 LOAD

The Load command shall be coded according to the Open Platform Card Specification [4].

EXAMPLE: The load block data is created by taking successive blocks of the data from the Java Card CAP file components in the order described in the Java Card specification.

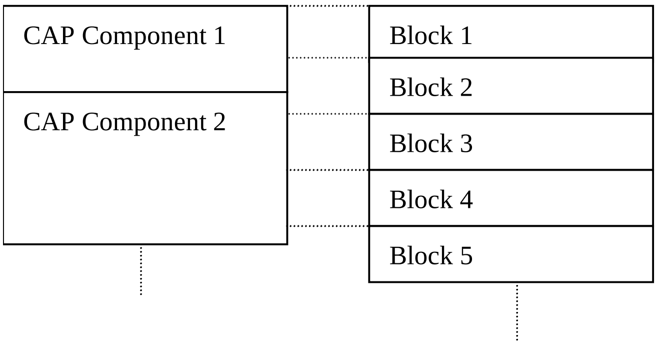

Figure 3: Relationship between CAP File components and Load File Blocks

7.2.1.2.5 PUT KEY

The Put Key command shall be coded according to the Open Platform Card Specification [4].

The keys which may be updated by the PUT KEY command refer to the transport security keys, i.e. KID and KIc in a Secured Packet. In addition, a third key type is defined: KIK. This key is used to encrypt the key data value of the PUT KEY command.

One or several keys within an existing key set version may be replaced using the Put Key command.

Keys within a key set are structured in the following way:

	Key Set Version 0	Key Set Version 1	….	Key Set Version n (maximum 'F')
Key Index 1	Reserved	KIc 1		KIc n
Key Index 2	Reserved	KID 1		KID n
Key Index 3	Reserved	KIK 1		KIK n

A card may have up to 15 key set versions each containing 3 key indexes. These versions numbers represent the indication of keys to be used in bits 8 to 5 in the coding of KIc and KID (see clauses 5.1.2 and 5.1.3 of TS 102 225 [1]). Each key index represents:

- Key Index 1: KIc
- Key Index 2: KID
- Key Index 3: KIK

Key Sets can only be changed with the PUT KEY command once the card has been issued. New Key Sets cannot be created using PUT KEY command at post issuance. Key used for securing the PUT KEY command is the key index 3 of the same key set version as the changed key. Key Set version 0 defined in OP for the creation of keys is not relevant for the present document.

A key set version number shall never be updated using the PUT KEY command.

7.2.2 Output Commands

7.2.2.1 List of the output commands

The following table extends table 3 defined in clause 6.2.

Table 5: Application Management output commands

Operational command
GET STATUS
GET DATA

7.2.2.2 Description of the output commands

7.2.2.2.1 GET STATUS

The Get Status command shall be coded according to the Open Platform Card Specification [4].

7.2.2.2.2 GET DATA

The Get Data command shall be coded according to the Open Platform Card Specification [4].

The command Get Data is extended to retrieve specific card information with tag values in P1 and P2. The following values have been defined:

- 'FF 1F': Menu Parameters Tag, this retrieves the menu parameters of an application;
- 'FF 20': Card Resources Tag, this retrieves information on the card resources used and available;
- 'FF 21' to 'FF 7F': reserved for allocation in the present document.

7.2.2.2.2.1 Menu Parameters

The following format is used to code the command data.

Bytes	Description	Length
1	Application AID tag = '4F'	1
2	Application AID length	1
3 - (X+2)	Application AID	X = 5 - 16

After the successful execution of the command, the following data is returned by a GET RESPONSE command.

Bytes	Description	Length
1	First item position	1
2	First item identifier	1
…	…	…
X – 1	Last item position	1
X	Last item identifier	1

7.2.2.2.2.2 Card Resources Information

After the successful execution of the command, the following data is returned:

Bytes	Description	Length
1-2	Free E^2PROM	2
3	Number of installed applications	1

7.3 Security of messages sent to the Remote Management Applications

7.3.1 Minimum Security Level

In order to control the access to the Remote Management Applications (Remote File Management and Remote Application Management applications), a Minimum Security Level as defined in clause 7.2.1.2.3.2.2 shall be assigned to each one of these applications. This Minimum Security Level shall be checked for all Secured Packet sent to one of these applications.

The Receiving Entity shall manage this Minimum Security Level as described in clause 7.2.1.2.3.2.2.

7.3.2 Remote File Management Access Conditions

In order to control the access conditions of the Remote File Management Applications, an Access Domain as defined in clause 7.2.1.2.3.2.4 shall be assigned to each Remote File Management Application.

Annex A (informative): Change History

This annex lists all changes made to the present document.

Change History								
Date	**Meeting**	**Tdoc**	**CR**	**Rv**	**Cat**	**Changes**	**Old**	**New**
2001-07	T3 API #7	T3a010114				Initial version is based on 3GPP TS 23.048 v4.0.0. Submitted to EP SCP#7 for information.	-	0.0.0
2001-10	SCP2#1	SCP2-010004				Alignment with 3GPP TS 23.048 v5.1.0 and editorial enhancements.	0.0.0	0.1.0
2001-10	T3 API #9	T3a010196				Updated to include the results of EP SCP WG2 #1.	0.1.0	0.2.0
2001-11	T3#21/ SCP#8	T3-010670/ SCP-010346				Submitted to 3GPP T3#21 - EP SCP#8. Editorial changes.	0.2.0	0.2.1
2001-11	SCP#8	SCP-010377				Editorial and formatting enhancements. Version number raised to 1.0.0 in line with decision at EP SCP #8.	0.2.1	1.0.0
2002-01	SCP2#2	SCP2-020019				Updated to include the results of TSG-T#14 and editorial changes.	1.0.0	1.1.0
2002-03	SCP#9	SCP-020049				Updated to include the results of TSG-T#15 and editorial changes. Submitted to SCP#9 for approval.	1.1.0	2.0.0
2002-03	SCP#9	SCP-020057				Editorial changes after discussion at SCP#9. This version has been sent to the ETSI secretariat for publication in March 2002 as TS 102 226 v6.0.0. No technical changes compared to v2.0.0.	2.0.0	6.0.0
2002-06	SCP#10	SCP-020169	001	1	B	Definition of the TAR Value(s) parameter in the Application Specific Parameters of the Install(Install) command.	6.0.0	6.1.0
2002-09	SCP#11	SCP-020232	003		A	Toolkit Access with modified secret code status	6.1.0	6.2.0
			004		A	Minimum Security Level for the Remote Management Applications and Access conditions for Remote File Management Application.		
			005		A	Clarification on Put Key command		
			006		A	Maximum number of channels allowed for this applet instance		
		SCP-020237	007		A	Clarification on letter 'n' describing the length of parameters of the Install(Install) command		

History

Document history		
V6.0.0	April 2002	Publication
V6.1.0	July 2002	Publication
V6.2.0	October 2002	Publication