ETSI TS 102 622 V12.2.0 (2018-12)

TECHNICAL SPECIFICATION

Smart Cards;
UICC - Contactless Front-end (CLF) Interface;
Host Controller Interface (HCI)
(Release 12)

Reference

RTS/SCP-THCIvc20

Keywords

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

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (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
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 (https://ipr.etsi.org/).

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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

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

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC 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 TC SCP for information;

2 presented to TC SCP for approval;

3 or greater indicates TC 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.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies a logical interface that enables contactless applications hosted on the UICC. The present document covers the configuration where the one host is embedded in the UICC which is connected to the host controller embedded in the CLF.

The interface is specified in two parts:

- the HCI core as described in the first part of the present document (clauses 4 to 8), specifies an application independent logical interface; and

- the contactless platform as described in the second part of the present document (clauses 9 to 11) specifies an implementation of the HCI core for contactless applications using the UICC and the CLF.

Lower layer protocols that support the HCI like the SWP specified in ETSI TS 102 613 [2] are out of the scope of the present document.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] Void.

[2] ETSI TS 102 613: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics".

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

[4] ISO/IEC 18092: "Information technology - Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol (NFCIP-1)".

[5] ISO/IEC 14443-2: "Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 2: Radio frequency power and signal interface".

[6] ISO/IEC 14443-3: "Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 3: Initialization and anticollision".

[7] ISO/IEC 14443-4: "Identification cards - Contactless integrated circuit cards - Proximity cards - Part 4: Transmission Protocol".

[8] ISO/IEC 7816-4: "Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange".

[9] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".

[10] ISO/IEC 7816-3: "Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols".

[11] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] JIS X 6319-4:2005: "Specification of implementation for integrated circuit(s) cards -- Part 4: High speed proximity cards".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

embedded secure element host: host implemented in a non-removable secure element

gate: entry point towards a service that is operated inside a host

host controller: host that is also responsible for managing a host network

host network: network of two or more hosts

host: logical entity that operates one or more service(s)

pipe: logical communication channel between two gates from different hosts

registry: data related to a gate that are stored as a collection of parameter - value pairs

service: collection of related atomic functions in a host for management purposes or for performing contactless actions

3.2 Symbols

For the purpose of the present document the following symbols apply:

G_{ID}	gate identifier
H_{ID}	host identifier
P_{ID}	pipe identifier
$PIPE_x$	a pipe with a P_{ID} value equal to x

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AFI	Application Family Identifier
AID	Application Identifier
APDU	Application Protocol Data Unit
ATR	Answer To Reset
BER-TLV	Basic Encoding Rules - Tag, Length, Value
C-APDU	Command APDU
CB	Chaining Bit
CLF	ContactLess Front-end
CLT	ContactLess Tunnelling
CRC	Cyclic Redundancy Code
HCI	Host Controller Interface
HCP	Host Controller Protocol
JIS	Japanese Industrial Standard
LEN	LENgth
NFC	Near Field Communication
PA	PreAmble
PCD	Proximity Coupling Device
PICC	Proximity Card
R-APDU	Response APDU
REF	REFerence data
RF	Radio Frequency
RFIN	RF data INcoming
RFU	Reserved for Future Use
RO	Read-Only
RST	ReSeT
RW	Read/Write
SAK	Select AcKnowledge
SD	Secure Digital
SWP	Single Wire Protocol
UI	User Interface
WO	Write-Only

3.4 Void

The content of this clause has been moved to clause 3A.

3A Coding conventions

For the purposes of the present document, the following coding conventions apply:

- All lengths are presented in bytes, unless otherwise stated. Each byte is represented by bits b8 to b1, where b8 is the most significant bit and b1 is the least significant bit. In each representation, the leftmost bit is the most significant bit.

- Hexadecimal values are specified between single quotes, e.g. '1F'. In a sequence of bytes, byte 1 is the leftmost byte.

- All bytes specified as RFU shall be set to '00' and all bits specified as RFU shall be set to 0.

4 HCI architecture

4.1 Overview

A valid host network has a star topology where one or more hosts physically connect to a host controller. The HCI defines the interface between hosts. More specifically the HCI has three levels:

- a collection of gates that exchange commands, responses and events; and
- an HCP messaging mechanism; and
- an HCP routing mechanism that may optionally segment messages when required.

Figure 1 illustrates the HCP stack in a possible host network.

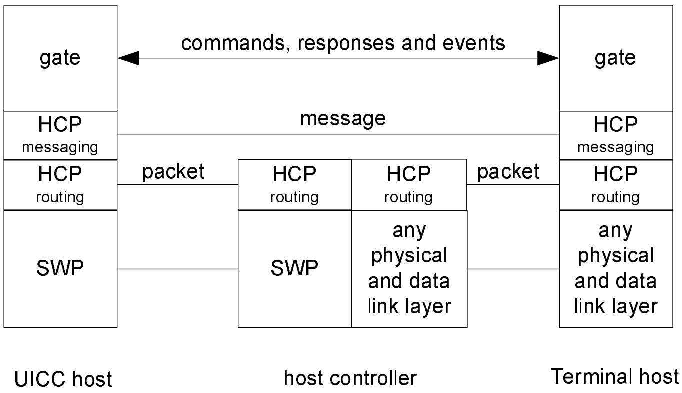


Figure 1

NOTE: For clarity only two gates are shown. In particular the host controller also has gates that connect via HCP to other hosts.

For proper operation, the HCP requires underlying data link layers to have the following properties:

- The data link layer (e.g. SWP) shall be error free and the order of the received/sent data shall be respected.
- The data link layer provides its own data flow control.
- The data link layer shall deliver packets of the upper layer up to a maximum size specific to the data link layer.
- The data link layer shall report the size of each received packet to its upper layer.

4.2 Hosts

The identity of a host is coded in a byte. Table 1 lists the reserved values for the host identifier, H_{ID}.

Table 1

Host	H_{ID}
host controller	'00'
terminal host	'01'
UICC host	'02'
RFU	'03' to '7F'
Dynamically allocated	'80' to 'BF'
Proprietary	'C0' to 'FF'
NOTE: If more than one UICC host is supported, see details below.	

In the present document, the generic term "host" is used to refer to any logical host (e.g. terminal host, UICC host) excluding the host controller.

The dynamically allocated range of values shall be used by the host controller to assign a host identifier to any host not identified in table 1. The host controller shall always assign the same H_{ID} to a given host throughout different sessions as long as there is no modification in the hardware configuration of the device. When the host controller uses a dynamically allocated H_{ID} for a host, the host will be able to retrieve its own H_{ID} upon pipe creation (Source H_{ID} in the answer to ADM_CREATE_PIPE command) or in the host controller administration gate registry.

If more than one UICC host is supported, the H_{ID} value '02' applies to pre-release 12 UICCs. Other UICCs if present will use dynamically allocated H_{ID} values. In the case of several pre-release 12 UICCs, the host controller should avoid having these UICCs active at the same time as they present the same $H_{ID.}$

NOTE: The Host controller can detect a UICC supporting release 12 or later by the fact that the UICC sets the HOST_TYPE in the Host Controller administration gate.

4.3 Gates

A gate provides an entry point to a service that is operated inside a host. The HCP enables gates from different hosts to exchange messages. There are two types of gates:

- Management gates that are needed for the management of the host network.
- Generic gates that are not related to the management of the host networks. Only the generic aspects of these gates are defined in the HCI core.

The type of a gate is identified by a gate identifier. Gate identifiers are listed in table 2 and are either unique within the scope of a host ('10' to 'FF'), or their values refer to the same gate type for every host ('00' to '0F').

Table 2

Gate	G_{ID}
reserved for proprietary use	'00' to '03'
loop back gate	'04'
identity management gate	'05'
RFU	'06' to '0F'
Host specific	'10' to 'EF'
reserved for proprietary use	'F0' to 'FF'

Host specific gates for hosts defined in the present document (including RFU ones) are reserved for use in the present document.

The following rules apply to hosts and gates:

- all hosts and the host controller shall have one administration gate (see clause 7.1.1); and
- all hosts may have one link management gate (see clause 7.1.2) and the host controller shall have one link management gate; and
- all hosts and the host controller shall have one identity management gate (see clause 7.1.3); and
- all hosts and the host controller shall have one loop back gate (see clause 7.1.4); and
- all hosts and the host controller may have one or more generic gates (see clause 7.2).

4.4 Pipes

A pipe is a logical communication channel between two gates. There are two types of pipes:

- static pipes that are always available, i.e. they do not need to be created and cannot be deleted; and
- dynamic pipes that can be created and deleted.

The state of a pipe is either open or closed. The state shall remain persistent if the hosts are powered down and up again. It shall also remain persistent if a host is temporarily removed from the host network and was not replaced by a different device in the meantime. The state of a dynamic pipe after creation and the initial state of a static pipe shall be closed.

The pipe identifier, P_{ID}, is 7 bits long. The value of P_{ID} is used in the header of HCP packets as routing information (see clause 5.1). For static pipes the pipe identifiers are predefined with values as defined in table 3. For dynamic pipes, pipe identifiers are dynamically allocated by the host controller.

Table 3

P_{ID}	Pipe ending at:	Pipe type
'00'	link management gate	static
'01'	administration gate	
'02' to '6F'	other gates	dynamic
'70' to '7F'	RFU	

The following rules apply to gates and pipes:

- a static pipe always connects a gate of a host to a gate of the host controller;
- a dynamic pipe connects two gates from different hosts; and
- static and dynamic pipes connect to different types of gates; see table 3 for the mapping; and
- dynamic pipe identifiers shall be unique in the host network.

Figure 2 illustrates hosts and gates in a valid host network.

NOTE: For clarity the identity management and loop back gates are not shown.

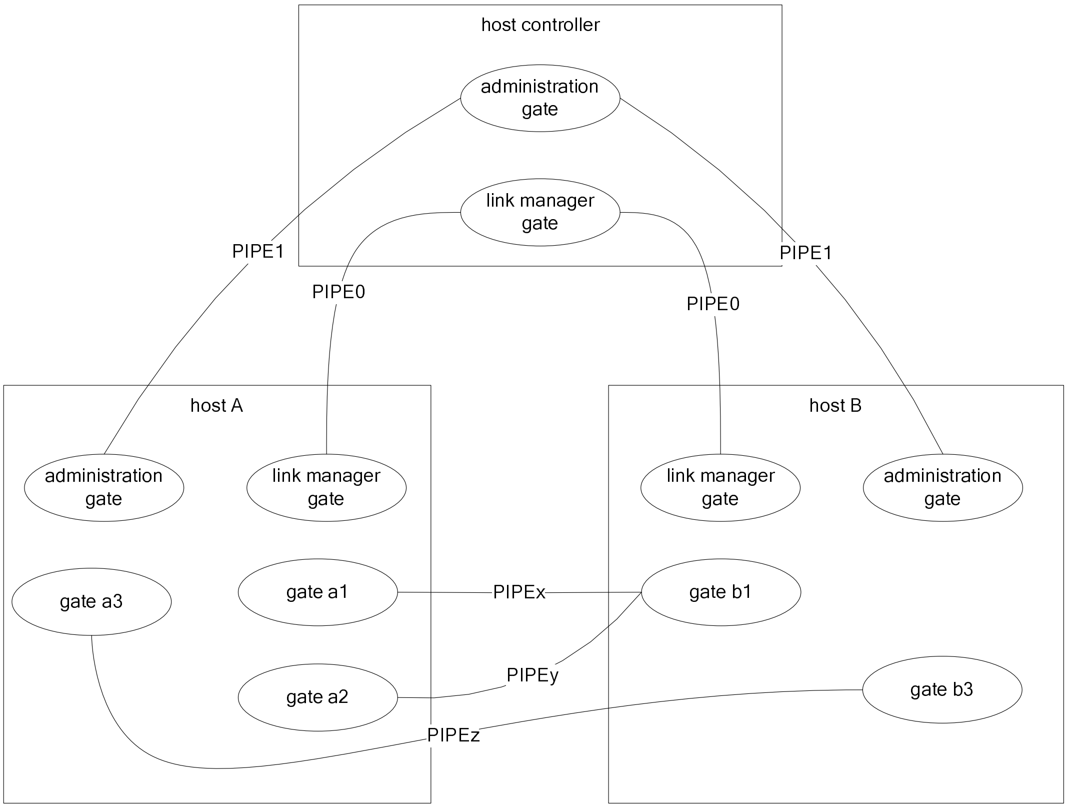


Figure 2

4.5 Registries

With every gate a registry template may be associated that defines parameters that are related to the gate. Parameters are identified by parameter identifiers consisting of one byte. Parameter identifiers are unique within the scope of the gate.

For all gates defined in the present document, parameter identifiers in the range of '00' to 'EF' are reserved for use in the present document. Parameter identifiers in the range of 'F0' to 'FF' may be used for proprietary purposes.

A new instance of the registry is created for every pipe that connects to the gate. Upon pipe creation all registry parameters with access rights Read-write (RW) or Write-only (WO) shall be set to their default values. Read-only (RO) parameters shall be set by the entity managing the registry to an appropriate value which may differ from the default values.

A host is responsible for managing its associated registries. The persistence of a registry and the default values for parameters shall be indicated with each registry description.

When a pipe is deleted its registry instance is also deleted.

5 HCP

5.1 HCP packets

Using the data link layer hosts exchange packets with the host controller. The format of a packet is defined in figure 3.

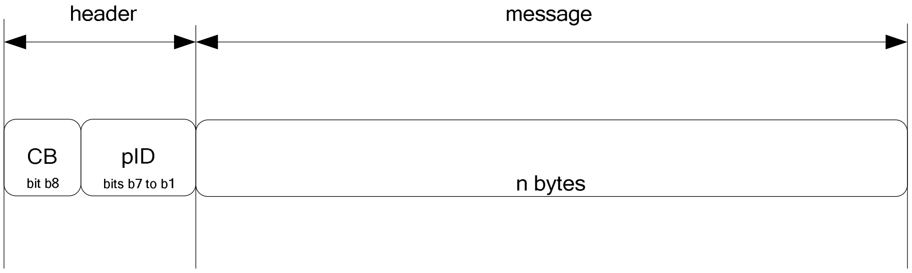


Figure 3

The interpretation of the fields in the packet header shall be the following:

- CB is the chaining bit and its value is equal to 1 except when message fragmentation is used (see clause 5.3); and
- P_{ID} specifies the pipe identifier.

The host controller uses the value of P_{ID} to forward a packet to the destination host. The destination host forwards the packet to the destination gate. Using these mechanisms any two gates that are connected by a pipe can exchange messages.

The host controller shall verify that the pipe identifier is used by a host involved in the creation of the pipe.

The size of a message is application specific. The message structure is described in clause 5.2.

5.2 HCP message structure

A message carries one instruction and optional data as defined in figure 4.

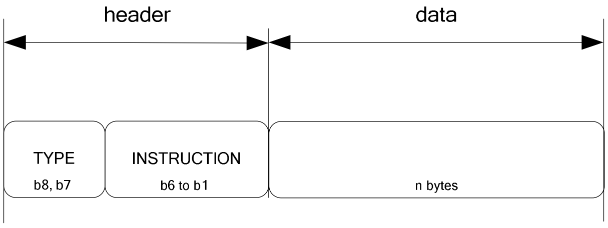


Figure 4

The meaning of the fields in the message header is the following:

- TYPE identifies the type of instruction; and
- INSTRUCTION identifies the instruction.

The following types of instructions are defined:

- commands (type value 0);
- events (type value 1); and
- responses to commands (type value 2).

Type value 3 is RFU. The instruction value further qualifies the command, event or response. All three types may carry data.

For instructions the following rules apply:

- An event is defined within the scope of the gate that accepts the events.
- A command is defined within the scope of the gate that accepts the command.
- A response is defined within the scope of its associated command.

A gate shall only accept a command or an event on a pipe when the state of that pipe is open unless otherwise stated. A gate shall not send a command or event on a pipe when it is waiting for a response to a previous command on that pipe unless otherwise stated. A gate shall interpret incoming events and commands even while it is waiting for a response to a previously sent command.

5.3 Message fragmentation

Message fragmentation shall be used when the size of the message is larger than supported by the underlying data link layer. Messages shall be fragmented according to the following rules:

- All message fragments shall have a packet header.
- The value of the chaining bit in the packet header is equal to 0, except for the packet with the last fragmented message where the value shall be 1.
- Only the first message fragment shall contain a message header.

Figure 5 illustrates how one message is split into two fragments.

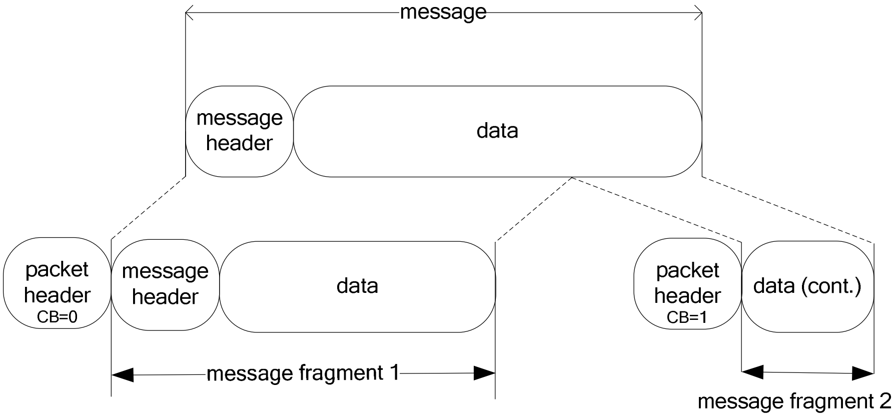


Figure 5

The source gate is responsible for fragmenting messages. The destination gate is responsible for rebuilding the message from the fragmented messages.

If a reset of the underlying data link layer occurs, fragments of a partially received message shall be discarded and a partially sent message shall be re-sent from the beginning.

6 Instructions

6.0 Foreword

The structure of the data field for the instructions is defined in the tables in clause 6 as the sequence of fields in the tables.

6.1 Commands

6.1.1 Overview

Table 4 lists all the commands that will be described in clauses 6.1.2 and 6.1.3. The interpretation of all the commands shall be the same for all gates specified in clause 7. There are two groups of commands:

- generic commands are applicable to all gates; and
- administration commands needed for the management of the host network.

Table 4

Gate	Command	Description	Instruction
All gates	RFU		'00'
	ANY_SET_PARAMETER	the command to set a parameter in a registry	'01'
	ANY_GET_PARAMETER	the command to get a parameter from a registry	'02'
	ANY_OPEN_PIPE	the command to open a pipe	'03'
	ANY_CLOSE_PIPE	the command to close a pipe	'04'
	RFU		'05' to '0F'
Administration gate only	ADM_CREATE_PIPE	the administration command to create a dynamic pipe	'10'
	ADM_DELETE_PIPE	the administration command to delete a dynamic pipe	'11'
	ADM_NOTIFY_PIPE_CREATED	the administration command to notify the creation of a dynamic pipe	'12'
	ADM_NOTIFY_PIPE_DELETED	the administration command to notify the deletion of a dynamic pipe	'13'
	ADM_CLEAR_ALL_PIPE	the command to close/delete all pipes connected to a requesting host	'14'
	ADM_NOTIFY_ALL_PIPE_CLEARED	the command to notify the closing/deletion of all pipes connected to a requesting host	'15'
	RFU		'16' to '3F'
Other non-proprietary gates	Reserved		'10' to '3F'

For every command the following information is provided:

- a description of the command; and
- a list of parameters for the command; and
- the response on successful completion of the command (with optional parameters).

The mapping of possible response codes is given in clause 6.3.

6.1.2 Generic commands

6.1.2.1 ANY_SET_PARAMETER

The command ANY_SET_PARAMETER is used to write a parameter value into a gate registry.

The command parameters are as follows:

Table 5

Description	Length
index of the parameter in the registry	1
value of the parameter; the length and content of the parameter depend on the registry	N

When successful the host shall respond with ANY_OK without any parameters.

6.1.2.2 ANY_GET_PARAMETER

The command ANY_GET_PARAMETER is used to read a parameter value from a gate registry.

The command parameters are as follows:

Table 6

Description	Length
index of the parameter in the registry	1

When successful the host shall respond with ANY_OK with parameter as follows:

Table 7

Description	Length
value of the parameter; the length and content of the parameter depends on the registry	N

6.1.2.3 ANY_OPEN_PIPE

The command ANY_OPEN_PIPE allows a host to open a pipe.

NOTE: This command is sent over a closed pipe.

The command has no parameters.

If the destination host is the host controller, and if the pipe is successfully opened the host controller shall send an ANY_OK response without any parameter.

For hosts other than the host controller, when the pipe is successfully opened, the destination host shall send an ANY_OK response with a parameter as follows:

Table 8

Description	Length
Number of pipes already open on this gate before the execution of the command	1

6.1.2.4 ANY_CLOSE_PIPE

The command ANY_CLOSE_PIPE allows a host to close a pipe.

The command has no parameters.

When the pipe is successfully closed the destination host shall send an ANY_OK response without parameters.

6.1.3 Administration commands

6.1.3.1 ADM_CREATE_PIPE

With the command ADM_CREATE_PIPE a host can request the host controller to create a new dynamic pipe between two gates. The host requesting the pipe shall be the source host. When successful a pipe is created between the source host and a destination host. The host controller shall use the WHITELIST defined by the destination host in order to verify that the source host is authorized to create a pipe.

The service that can be used on this pipe is determined only by the destination gate. Any gate identifier can be used as source gate; however, only one pipe is allowed to be created for each combination of source host/source gate and destination host/destination gate.

The command parameters are as follows:

Table 9

Description	Length
source G_{ID}	1
destination H_{ID}	1
destination G_{ID}	1

When the pipe was successfully created, the host controller shall send the response ANY_OK with parameters as follows:

Table 10

Description	Length
source H_{ID}	1
source G_{ID}	1
destination H_{ID}	1
destination G_{ID}	1
P_{ID} of pipe	1

6.1.3.2 ADM_NOTIFY_PIPE_CREATED

The command ADM_NOTIFY_PIPE_CREATED is sent by the host controller to a destination host to notify of the creation of a dynamic pipe. The source host is the host that requested the creation of the pipe from the host controller.

The command parameters are as follows:

Table 11

Description	Length
source H_{ID}	1
source G_{ID}	1
destination H_{ID}	1
destination G_{ID}	1
P_{ID} of pipe	1

If the host accepts the pipe it shall send the response ANY_OK without parameters.

6.1.3.3 ADM_DELETE_PIPE

With the command ADM_DELETE_PIPE a host may request the host controller to delete a dynamic pipe. The host that requested the deletion of the pipe can only be the source host or destination host.

The command parameters are as follows:

Table 12

Description	Length
P_{ID} of pipe	1

When the pipe is successfully deleted, the host controller shall send the response ANY_OK without parameters.

6.1.3.4 ADM_NOTIFY_PIPE_DELETED

The command ADM_NOTIFY_PIPE_DELETED is sent by the host controller to a host to notify the deletion of a dynamic pipe.

The command parameters are as follows:

Table 13

Description	Length
P_{ID} of pipe	1

When the pipe has successfully been deleted, the host shall send the response ANY_OK without parameters.

6.1.3.5 ADM_CLEAR_ALL_PIPE

With the command ADM_CLEAR_ALL_PIPE a host requests the host controller:

- to delete all the dynamic pipes connected to the requesting host; and
- to close all static pipes connected to the requesting host; and
- to set all registry values related to static pipes to their default values.

For the data link layer specified in ETSI TS 102 613 [2], the command parameters are as follows:

Table 13a

Description	Length
Identity reference data	2

The identity reference data shall be used to initialize the reference data used by the host controller to check the UICC host identity. The identity reference data shall contain random elements.

When successful the host controller shall respond with an ANY_OK without parameters.

6.1.3.6 ADM_NOTIFY_ALL_PIPE_CLEARED

If the requesting host is not the host controller, the command ADM_NOTIFY_ALL_PIPE_CLEARED is sent by the host controller, following an ADM_CLEAR_ALL_PIPE command from the requesting host, to every host with at least one pipe to the requesting host to notify:

- that it deleted all dynamic pipes between the host and the requesting host.

If the requesting host is the host controller, the command ADM_NOTIFY_ALL_PIPE_CLEARED is sent by the host controller to a host to notify:

- that it deleted all dynamic pipes between the host controller and the host; and
- that it closed all static pipes between the host and the host controller.

The command parameters are as follows:

Table 14

Description	Length
H_{ID} of requesting host	1

The host shall respond with an ANY_OK without parameters.

6.2 Responses

For the commands specified in clause 6.1, table 15 lists the possible response codes and table 16 their mapping to each command. Unless otherwise stated these responses carry no additional parameters. A response shall be sent to all commands received even to those unknown to the receiving gate. Responses received out of order (i.e. if no command was sent previously) shall be discarded.

Table 15

Value	Response	Description
'00'	ANY_OK	command completed successfully (with optional parameters)
'01'	ANY_E_NOT_CONNECTED	the destination host is not connected
'02'	ANY_E_CMD_PAR_UNKNOWN	the format of the command parameters is wrong
'03'	ANY_E_NOK	command was rejected and/or not completed
'04'	ADM_E_NO_PIPES_AVAILABLE	no more dynamic pipes can be created
'05'	ANY_E_REG_PAR_UNKNOWN	the registry parameter identifier is either unknown to the registry or an optional registry parameter is not implemented
'06'	ANY_E_PIPE_NOT_OPENED	the pipe is not open
'07'	ANY_E_CMD_NOT_SUPPORTED	the command is not supported by the gate
'08'	ANY_E_INHIBITED	command is inhibited due to failure of lower layer identity check
'09'	ANY_E_TIMEOUT	an application level time-out occurred
'0A'	ANY_E_REG_ACCESS_DENIED	permission denied to write/read a value to/from a registry
'0B'	ANY_E_PIPE_ACCESS_DENIED	Permission denied to create a pipe due to a WHITELIST violation
'0C' to '3F'		RFU

Table 16

RESPONSE CODE COMMAND	ANY_OK	ANY__E_NOT_CONNECTED	ANY_E_CMD_PAR_UNKNOWN	ANY_E_NOK	ADM_E_NO_PIPES_AVAILABLE	ANY_E_REG_PAR_UNKNOWN	ANY_E_PIPE_NOT_OPENED	ANY_E_CMD_NOT_SUPPORTED	ANY_E_PIPE_ACCESS_DENIED	ANY_E_REG_ACCESS_DENIED	ANY_E_INHIBITED
ANY_SET_PARAMETER	•	•		•		•	•	•		•	•
ANY_GET_PARAMETER	•	•		•		•	•	•		•	•
ANY_OPEN_PIPE	•	•		•			•	•			•
ANY_CLOSE_PIPE	•	•		•			•				•
ADM_CREATE_PIPE	•	•	•	•	•		•	•	•		•
ADM_NOTIFY_PIPE_CREATED	•	•	•	•	•		•	•			•
ADM_DELETE_PIPE	•	•	•	•			•	•			•
ADM_NOTIFY_PIPE_DELETED	•	•	•	•			•	•			•
ADM_CLEAR_ALL_PIPE	•	•	•	•			•	•			
ADM_NOTIFY_ALL_PIPE_CLEARED	•	•	•	•			•	•			•
Command with RFU instruction value		•		•			•	•			•
• = This response code may result from this command.											

6.3 Events

The interpretation of the events listed in table 17 shall be the same for all gates specified in clause 7. Unknown events received shall be discarded.

Table 17

Value	Event	Description
'01'	EVT_HCI_END_OF_OPERATION	This event shall be sent by a host to the host controller when the host requires no more activity on the HCI interface and its underlying layer at the time the event is sent. This event has no parameters.
'02'	EVT_POST_DATA	This event shall be used to send and receive data with a parameter that contains a block of data of variable size
'03'	EVT_HOT_PLUG	This event shall be sent by the host controller to any other connected host to notify the connection or disconnection of a host to the host controller. This event has no parameters.

All other event values to non-proprietary gates that are not assigned in the present document are RFU.

7 GATES

7.0 Generic gate events and command support

The next clauses define all gates. For each gate the following information is provided:

- a brief description of the gate's responsibility; and
- a list of parameter - value pairs for the gate's registry.

The mapping of gates to the supported events and commands are given in tables 18 and 19, respectively.

Table 18

Gate Events	Host Link management	Host controller Link management	Host controller administration	Host administration	Identity management	Loop back	Generic
EVT_HCI_END_OF_OPERATION		•					▲
EVT_POST_DATA						•	▲
EVT_HOT_PLUG				•			

• = This command/event is supported by this Gate.
▲ = The host specific implementation of the generic gate defines if this command/event is supported.

Table 19

GATE COMMAND	Link management	Host controller administration	Host administration	Identity management	Loop back	Generic
ANY_SET_PARAMETER	•	•	•			▲
ANY_GET_PARAMETER	•	•	•	•		▲
ANY_OPEN_PIPE	•	•	•	•	•	•
ANY_CLOSE_PIPE	•	•	•	•	•	•
ADM_CREATE_PIPE		•				
ADM_NOTIFY_PIPE_CREATED			•			
ADM_DELETE_PIPE		•				
ADM_NOTIFY_PIPE_DELETED			•			
ADM_CLEAR_ALL_PIPE		•				
ADM_NOTIFY_ALL_PIPE_CLEARED			•			

• = This command/event is supported by this Gate.
▲ = The host specific implementation of the generic gate defines if this command/event is supported.

NOTE: Host specific implementations of the generic gates define commands for transmission of data and additional events.

7.1 Management gates

7.1.1 Administration gates

7.1.1.1 Host controller administration gate

The administration gate in the host controller provides access to services that manage the network of pipes in the HCI network. In addition, this gate provides access to services that allow the discovery of hosts at the first startup and when the configuration of the host network has changed (see clause 8). The registry shall be persistent.

Table 20 lists the entries in the gate registry.

Table 20

Type	Identifier	Parameter	Access Right	Comment	Length	Default
MANDATORY	'01'	SESSION_IDENTITY	RW	Session identifier that is used to detect if the connected host configuration changed.	8	'FFFFFFFFFFFFFFFF'
	'02'	MAX_PIPE	RO	Maximum number of created dynamic pipes supported by the host controller per host. The minimum value is '10' and the maximum value is '7D'.	1	'10'
	'03'	WHITELIST	RW	List of hosts that may communicate with the host connected to this administration gate.	N_0	empty
	'04'	HOST_LIST	RO	The list of the hosts that are accessible from this host controller including the host controller itself. The array contains a list of host identifiers.	N_1	'00'
MANDATORY, not present in Rel-11 and earlier	'05'	HOST_ID	RO	H_{ID} of the host connected on this pipe either statically or dynamically assigned.	1	No default value
	'06'	HOST_TYPE	RW	Type of host connected on this pipe. The first byte defines the host type family and the second byte defines variant of this type and is defined by the respective organizations.	2	'FFFF'
	'07'	HOST_TYPE_LIST	RO	The list of the host types that are accessible from this host controller including the host controller itself. The array contains a list of host types and follows the same order as the HOST_LIST.	$2*N_1$	'0000'

The session identity shall be modified by the host whenever a modification of the configuration is performed by the host. The default value of the session identity shall never be written by a host. The session identity shall use random values.

Every host writes its WHITELIST into the host controller administration gate in order to inform the host controller which hosts are allowed to communicate with it. The host controller shall reject create pipe requests if the source host is not listed in the WHITELIST of the destination host.

The WHITELIST shall not contain the host controller identifier and host identifier accessing the WHITELIST.

The WHITELIST is an array containing a list of host identifiers as defined in table 1.

Coding of the HOST_TYPE is defined as follow:

- '0000' = Host controller;
- '0100' = Terminal host;
- '0200' = UICC;
- '0300' = Embedded secure element;
- '04XX' = SD cards as defined by SD Association;
- 'FFFF' = Unknown host type;
- All other values are RFU.

The second byte of HOST_TYPE for SD cards is defined by SD Association.

The HOST_TYPE shall be written by the host into the Host Controller administration gate registry during the session initialization as described in clause 8.4, prior to any pipe creation, and shall not be further modified.

The host controller shall send a notification to each host administration gate, possibly including the host that triggered the connection notification, using an EVT_HOT_PLUG in the following cases:

- When a host is connected to the HCI network and has completed the session initialization as described in clause 8.4;
- When a host is disconnected from the HCI network.

7.1.1.2 Host administration gate

The administration gate at a host provides access to services involved in the management of the pipes towards that host.

The administration gate at a host has no registry entries defined.

7.1.2 Link management gate

7.1.2.1 Host controller link management gate

The host controller link management gate provides information about the underlying layer. The registry may not be persistent.

Table 21 lists the entries in the registry.

Table 21

Identifier	Parameter	Access Right	Description	Length	Default
'01'	REC_ERROR	RW	Number of invalid or lost frames previously sent by the host due to communication errors at the data link layer. This parameter can only be set to 0 in order to restart an error rate measure. When 'FFFF' is reached, the counter stops.	2	'0000'

7.1.2.2 Host link management gate

The host link management gate provides access to information related to the link layer. The registry may not be persistent.

Table 22 lists the entries in the registry.

Table 22

Identifier	Parameter	Access Right	Description	Length	Default
'01'	REC_ERROR	RW	Number of invalid or lost frames previously sent by the host controller due to communication errors at the data link layer. This parameter can only be set to 0 in order to restart an error rate measure. When 'FFFF' is reached, the counter stops.	2	'0000'

7.1.3 Identity management gate

The identity management gate provides software and hardware information about the host. The registry shall be persistent.

This gate shall be provided by all hosts and the host controller. As destination gate, the identity management gate shall accept at least one pipe from each host in its WHITELIST.

Table 23 lists the entries in the registry.

Table 23

TYPE	Identifier	Parameter	Access Right	Description	Length	Default
OPTIONAL	'01'	VERSION_SW	RO	Version of the software defined by the vendor.	3	'000000'
	'03'	VERSION_HARD	RO	Version of the hardware defined by vendor	3	'000000'
	'04'	VENDOR_NAME	RO	Vendor name UTF8 coding. The maximum value for N shall be 20.	N_0	N_0=0
	'05'	MODEL_ID	RO	Model identifier assigned by the vendor.	1	'00'
MANDATORY not present in Rel-11 and earlier	'02'	HCI_VERSION	RO	Version of HCI supported by the host.	1	'02'
OPTIONAL	'08'	MAX_CURRENT	R0	The maximum current provided/required by the host controller/host during operation (mA).	1	10
MANDATORY	'06'	GATES_LIST	RO	The list of all gates that accept dynamic pipes as an array of gate identifiers.	N_1	N_1=0

A host according to the present document shall set the HCI_VERSION parameter to '02'. The HCI_VERSION parameter will be incremented for each release of this HCI specification that is not fully backward compatible with the previous release. Every host shall manage backward compatibility with previous HCI versions and use only commands and parameters defined in the present document n having the lower HCI version number exposed by the 2 hosts involved in a transaction. A host connected to a host with higher HCI version number shall operate according to its own version.

The MAX_CURRENT parameter of the host controller represent the maximum current that it is able to provide to a host during operation as defined for the different contactless mode of operation. The MAX_CURRENT parameter of a host represent the current required by this host to allow optimum performance during contactless mode operation. A host can use this maximum current only during the period defined in this document.

7.1.4 Loop back gate

The loop back gate provides access to services for testing the HCI network. As destination gate, the loop back gate shall accept at least one pipe from each host in its WHITELIST.

The loop back gate has no registry entries defined.

7.2 Generic gates

Generic gates are not related to the management of the host network, but provide functions in the context of their host. For the contactless platform the functions and registries are defined in clauses 9, 10 and 11.

8 HCI procedures

8.1 Pipe management

8.1.1 Pipe creation

Figure 6 illustrates how host A requests the creation of a dynamic pipe, between one of its gates and a gate in host B. All communications use PIPE$_1$.

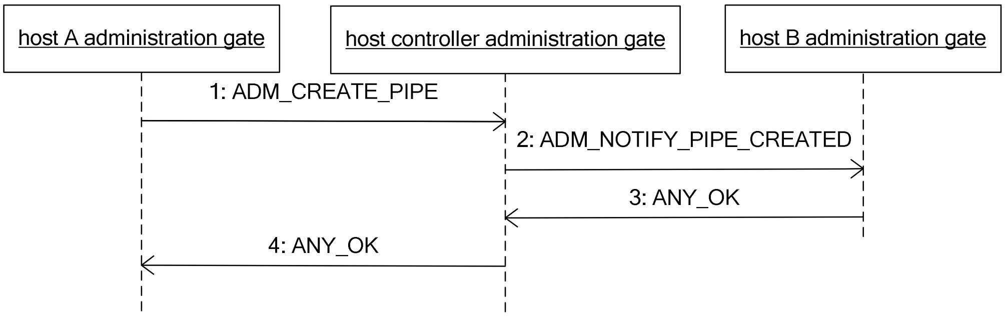

host A administration gate
host controller administration gate
host B administration gate
1: ADM_CREATE_PIPE
2: ADM_NOTIFY_PIPE_CREATED
3: ANY_OK
4: ANY_OK

Figure 6

The sequence flow is as follows:

1) Host A requests the host controller to create a pipe, PIPE$_x$. The host controller shall verify that Host B administration gate WHITELIST contains Host A identifier. If Host A is not part of the WHITELIST of Host B, the host controller shall send ANY_E_PIPE_ACCESS_DENIED response to Host A and stop any further processing of this command.

2) The host controller assigns an unused pipe identifier.

3) The host controller notifies host B that host A requested the creation of PIPE$_x$.

4) Host B responds to confirm that it accepts $PIPE_x$.

5) The host controller responds that $PIPE_x$ has been created.

6) If host B does not accept the creation of the pipe, it shall respond with an appropriate response code.

When the host controller wants to create a pipe then the pipe identifier is assigned and only steps 2 and 3 are needed.

When a pipe is created towards the host controller then only steps 1 and 4 are needed.

Figure 7 illustrates how the WHITELIST affects a request from a host to create a pipe. In the following, the WHITELIST associated with Host B does not initially contain the H_{ID} for Host A and so the request is rejected. Following updating of the WHITELIST by Host B a further request is accepted.

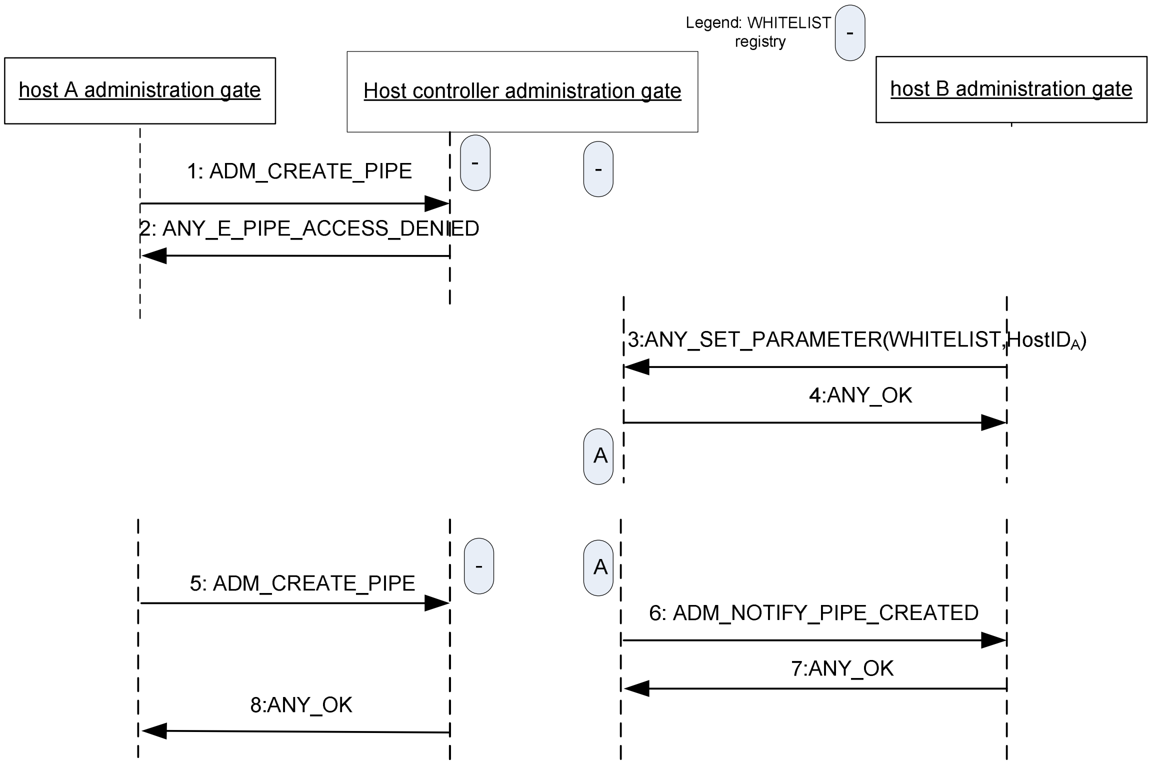

Legend: WHITELIST registry
-
host A administration gate
Host controller administration gate
host B administration gate
1: ADM_CREATE_PIPE
-
-
2: ANY_E_PIPE_ACCESS_DENIED
3:ANY_SET_PARAMETER(WHITELIST,HostIDA)
4:ANY_OK
A
5: ADM_CREATE_PIPE
-
A
6: ADM_NOTIFY_PIPE_CREATED
7:ANY_OK
8:ANY_OK

Figure 7

The sequence flow is as follows:

1) Host A requests the host controller to create a pipe, $PIPE_x$. The WHITELIST in the host controller administration gate connected to the Host B administration gate does not contain the host identifier of the host A. The Host A is not authorized by Host B.

2) The host controller shall send ANY_E_PIPE_ACCESS_DENIED response to Host A and stop any further processing of this command.

3) The Host B updates its WHITELIST register in the host controller administration gate by adding the Host A identifier.

4) The host controller returns ANY_OK.

5) Host A requests the host controller to create a pipe. The host controller assigns an available pipe identifier.

6) The host controller notifies host B that host A requested the creation of $PIPE_x$.

7) Host B responds to confirm that it accepts $PIPE_x$.

8) The host controller responds that PIPE_x has been created.

If host B does not accept the creation of the pipe, it shall respond with an appropriate response code.

8.1.2 Pipe deletion

Figure 8 illustrates how host A requests the deletion of a dynamic pipe, PIPE_x, between one of its gates and a gate in host B. All communications use PIPE_1.

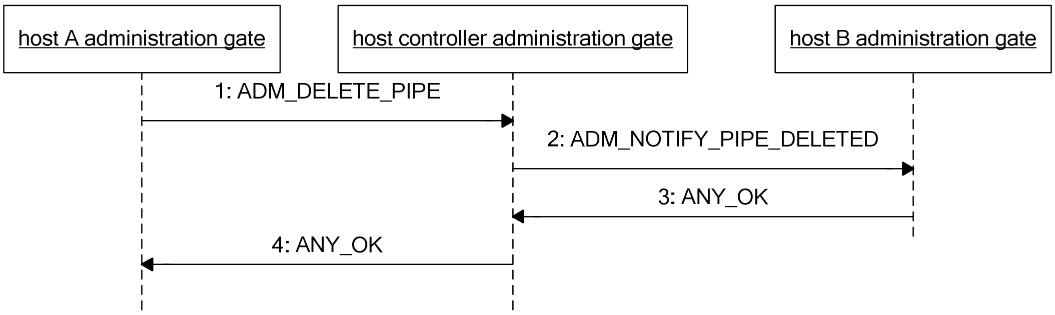

host A administration gate
host controller administration gate
host B administration gate
1: ADM_DELETE_PIPE
2: ADM_NOTIFY_PIPE_DELETED
3: ANY_OK
4: ANY_OK

Figure 8

The sequence flow is as follows:

1) Host A requests the host controller to delete PIPE_x.

2) The host controller notifies host B.

3) Host B confirms that it deleted PIPE_x.

4) The host controller confirms to host A that the PIPE_x is deleted.

Pipe deletion cannot be refused by host B.

A pipe created by host A can also be deleted by host B.

When PIPE_x connects to a gate at the host controller then:

- the connecting host may request the deletion and only steps 1 and 4 are needed; or
- the host controller may request the deletion and only steps 2 and 3 are needed.

8.1.3 Clear all Pipes

Figure 9 illustrates how host A requests the host controller to delete all of its dynamic pipes and close all of its static pipes. All communications use PIPE_1.

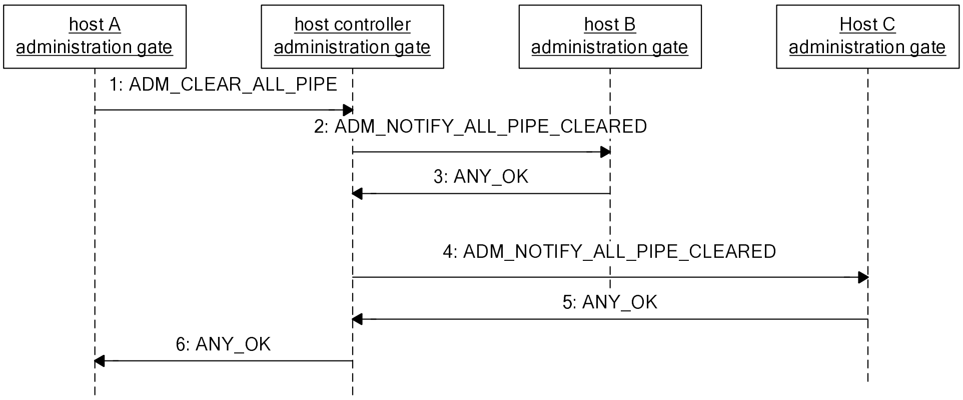


Figure 9

The sequence flow is as follows:

1) Host A requests the host controller to clear all pipes that connect to host A.

2) The host controller signals host B that all pipes are cleared towards host A.

3) Host B confirms.

4) The host controller signals host C that all pipes are cleared towards host A.

5) Host C confirms.

6) The host controller confirms to host A that pipes are cleared and host A closes its static pipes.

8.2 Registry access

Figure 10 shows how host A can read/write parameters in the registry of host B.

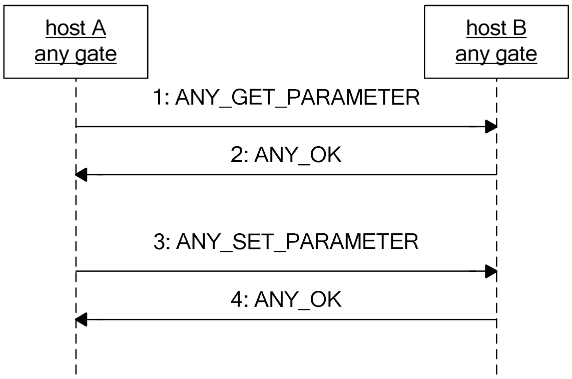


Figure 10

The sequence flow is as follows:

1) Host A requests a registry parameter from a gate of host B.

2) Host B replies with ANY_OK that includes the parameter value in its data.

3) Host A updates a registry parameter in a gate of host B.

4) Host B confirms that the parameter has been updated.

8.3 Host and Gate discovery

Figure 11 illustrates how a host may discover the hosts in a host network and the gates that a host supports.

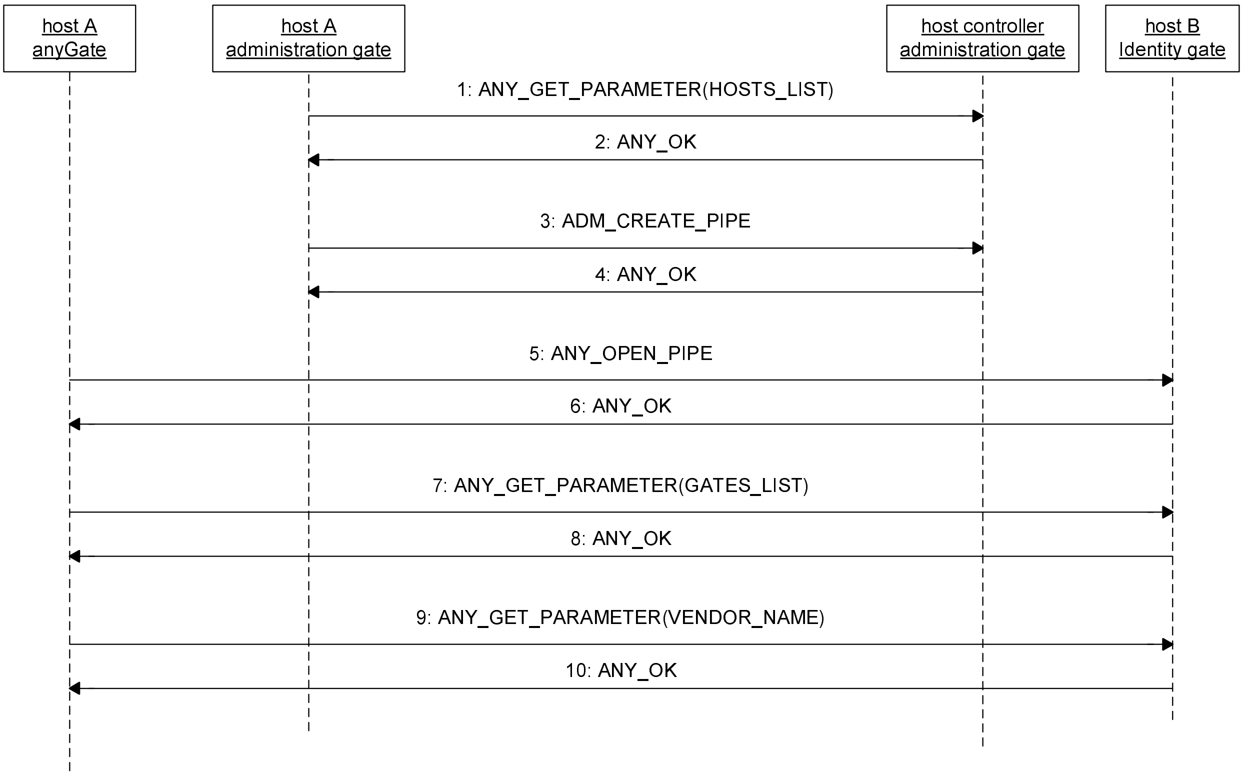

host A anyGate
host A administration gate
host controller administration gate
host B Identity gate
1: ANY_GET_PARAMETER(HOSTS_LIST)
2: ANY_OK
3: ADM_CREATE_PIPE
4: ANY_OK
5: ANY_OPEN_PIPE
6: ANY_OK
7: ANY_GET_PARAMETER(GATES_LIST)
8: ANY_OK
9: ANY_GET_PARAMETER(VENDOR_NAME)
10: ANY_OK

Figure 11

The sequence flow is as follows:

1) Host A uses the administration gate, over PIPE_1, to request the list of connected hosts.

2) The host controller returns the list.

3) Host A selects a host from the list, i.e. host B, and requests the host controller, over PIPE_1, to create a pipe towards the identity gate of host B.

4) The host controller confirms the creation of the pipe, PIPE_x. Details of the pipe creation are shown in clause 8.1.1.

5) Host A opens the PIPE_x.

6) Host B confirms that PIPE_x is open.

7) Host A requests the list of gates of host B.

8) The list of gates is returned.

9) Host A may request the VENDOR_NAME parameter from the registry.

10) The VENDOR_NAME parameter is returned.

Steps 9 and 10 are optional.

Optionally host A may close and delete PIPE_x.

8.4 Session initialization

As pipe states are persistent, moving the host (e.g. UICC host) into a different terminal may result in inconsistent configurations. Using the session initialization procedure allows a host to detect that the host network has changed, either because the host controller itself changed or because the configuration has been changed. The host shall perform this procedure only at startup and only if no contactless transaction is pending. In case of an underlying data link layer according to ETSI TS 102 613 [2] this procedure shall only be performed after the initial interface activation in full-power mode and only if no contactless transaction is pending.

Figure 12 illustrates how a host A detects that the host controller has changed and the subsequent recovery mechanism.

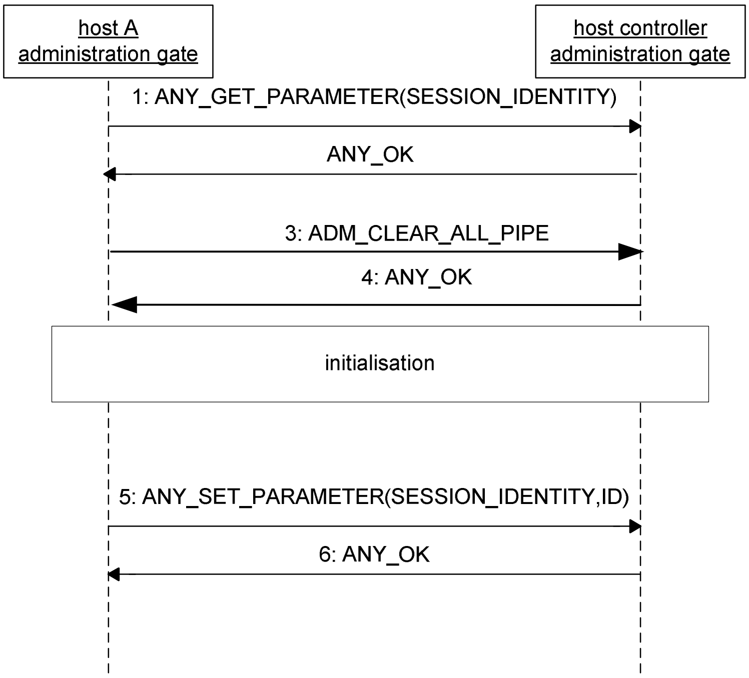

host A administration gate
host controller administration gate
1: ANY_GET_PARAMETER(SESSION_IDENTITY)
ANY_OK
3: ADM_CLEAR_ALL_PIPE
4: ANY_OK
initialisation
5: ANY_SET_PARAMETER(SESSION_IDENTITY,ID)
6: ANY_OK

Figure 12

The flow is as follows:

1) Host A queries over PIPE_1 the SESSION_IDENTITY parameter in the administration gate of the host controller (an ANY_OPEN_PIPE command may precede this command).

2) If the returned value equals the previous value stored in the host then the procedure stops.

3) Otherwise host A needs to reinitialize and it requests the host controller to clear all pipes (see clause 8.1.3).

4) The host controller confirms and host A shall set its HOST_TYPE (if its HCI_VERSION is '02' or greater) and can perform any further initializations.

5) Host A generates a new session identity and stores its value and stores it in the host controller registry.

6) The host controller confirms.

If the host controller stores a configuration (registry values and pipe states) of host A that is used before HCI initialization and need to be protected, an appropriate identity check mechanism shall be implemented by the lower layers. This identity check mechanism is out of the scope of the present document. In case the lower layer identity check fails, the host controller shall:

- execute only the following commands: ANY_OPEN_PIPE, ADM_CLEAR_ALL_PIPE, ANY_GET_PARAMETER, and only if these are sent on $PIPE_1$. All the other commands shall return ANY_E_INHIBITED and all events sent to the host controller shall be ignored; and
- return the default value of the SESSION_IDENTITY (see table 20). However the value of the SESSION_IDENTITY in the registry remains unchanged.

This inhibited state shall be terminated after processing a valid ADM_CLEAR_ALL_PIPE command.

Additional behaviour of the host controller in case of card emulation is defined in clause 9.4.6.

NOTE: It is up to host controller implementation to maintain the inhibited state after a subsequent successful identity check at the lower layer or not.

8.5 Loop back testing

Figure 13 shows how host A can verify the pipe connectivity to host B.

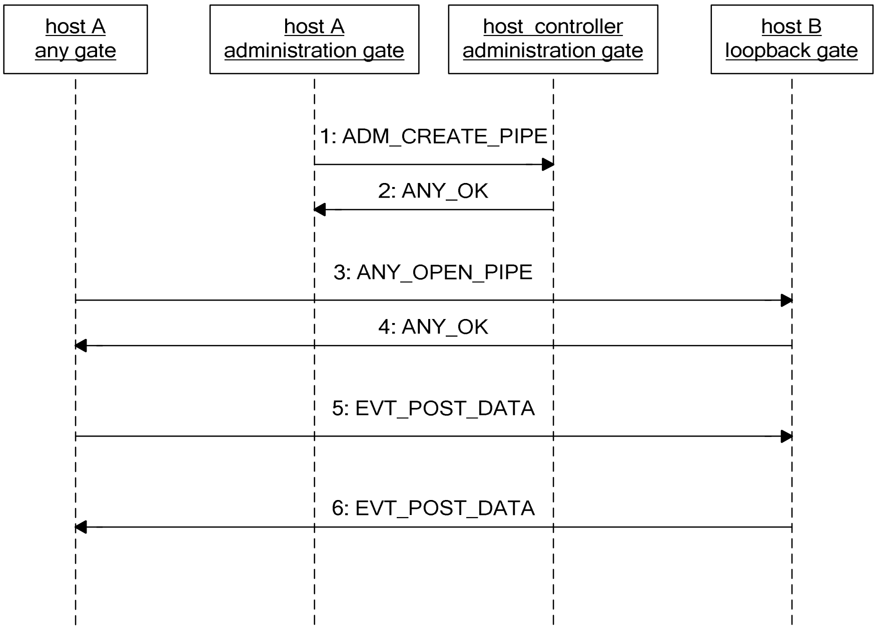


Figure 13

The sequence flow is as follows:

1) Host A requests over $PIPE_1$ the creation of a pipe between any gate and the loop back gate in host B.

2) The host controller confirms the creation of the pipe, $PIPE_x$. Details of the pipe creation are shown in clause 8.1.1.

3) Host A opens $PIPE_x$.

4) Host B accepts the opening.

5) Host A sends the event EVT_POST_DATA with some data as parameter. The loopback gate shall support messages with a size of up to 250 bytes and may also support longer messages.

6) Host B sends back the event EVT_POST_DATA with the same data as received in the previous step. Upon reception host A may compare the received data against the original data.

Optionally host A may close and delete PIPE_x.

9 Contactless card emulation

9.1 Overview

This clause specifies how a contactless card supporting multiple RF technologies is emulated on top of HCI core described in the previous clauses.

The RF technologies that are supported are:

- type A as defined in ISO/IEC 14443 parts 2 [5], 3 [6] and 4 [7]; or
- type B as defined in ISO/IEC 14443 parts 2 [5], 3 [6] and 4 [7]; or
- type B' compliant with ISO/IEC 14443-2 [5] and with a standard framing as defined in ISO/IEC 14443-3 [6];
- type F as defined in ISO/IEC 18092 [4] as 212 kbps and 424 kbps passive mode.

NOTE 1: Type F technology, framing and protocol is also described in Japanese Industrial Standard JIS X 6319-4 [i.1]. The JIS terminology is different from ISO/IEC 18092 [4] but the technology is the same.

The contactless platform for card emulation mode is realized as a host network with two hosts; the host controller and the host (e.g. the UICC host). The CLF shall run the host controller of the host network. Additional hosts using the services of the CLF are out of the scope of the present document.

The host operates contactless card applications and the CLF handles the RF communication layers to the external contactless reader.

The host controller has one card RF gate for each RF technology it supports. For each card RF gate it wants to use, the host has one card application gate.

For the contactless platform for card emulation mode the pipes to card RF gates shall be created, opened, closed and deleted by the host. The RF technology of a card RF gate is active when there is an open pipe connected to it. The host shall not create more than one pipe to each RF gate.

For ISO/IEC 14443-4 [7] compliant contactless card applications the processing of the transmission protocol (i.e. up to the layer specified in ISO/IEC 14443-4 [7]) is done in the CLF. The card application gate and card RF gate exchange APDUs over their pipe.

For type B' contactless card applications, the host controller and the host exchange raw frames. The raw frame structure is as defined in ISO/IEC 14443-3 [6]; but the data are transferred without the CRC_B.

In case of an underlying data link layer according to ETSI TS 102 613 [2] for type A contactless card applications: If the external reader does not use the transmission protocol defined in ISO/IEC 14443-4 [7] the CLF and the UICC exchange data using the CLT as defined in ETSI TS 102 613 [2].

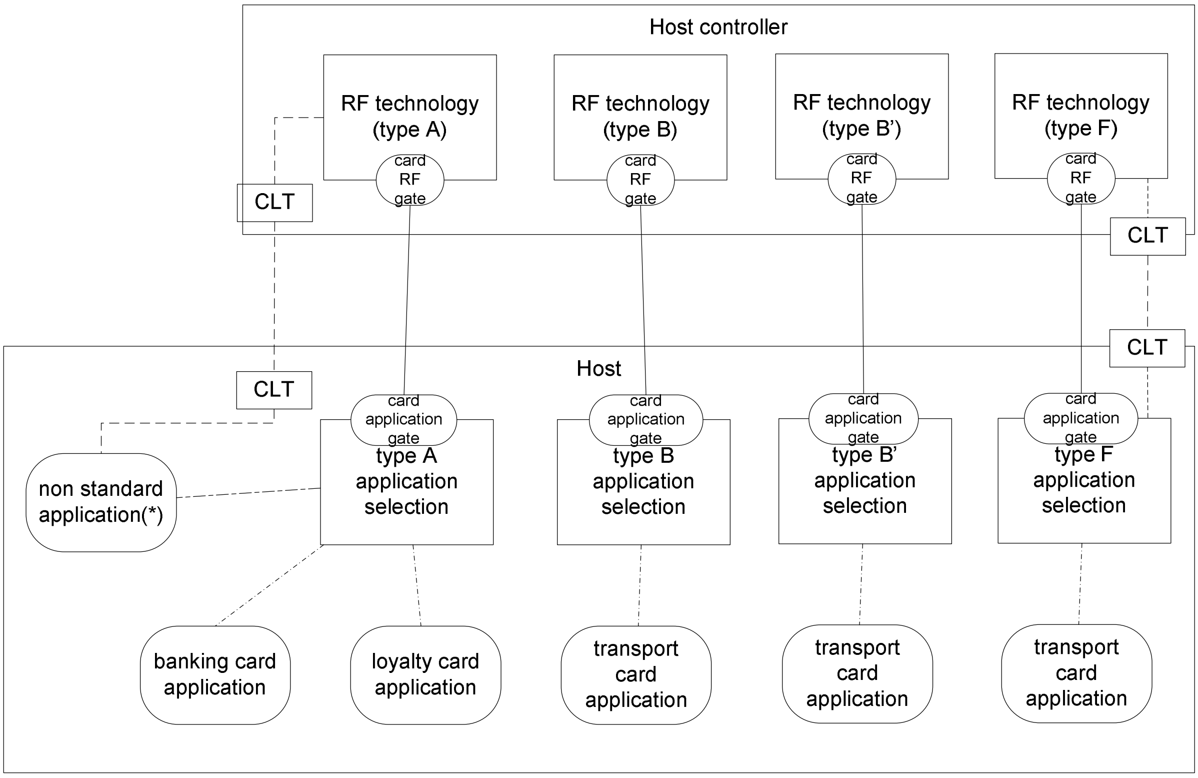
For Type F contactless card applications, the host controller and the host exchange ISO/IEC 18092 [4] 212 kbps/424 kbps frames. The frame structure is as defined in ISO/IEC 18092 [4] but the data are transferred without PA and SYNC fields and E2.

The host is allowed to consume a current up to the maximum defined by the host controller in its identity management gate registry between the appearance and the disappearance of the RF unless restricted by the underlying layers e.g. ETSI TS 102 613 [2] where the restrictions for low-power mode and power saving mode still apply.

The host controller activates one or more RF technologies as requested by the host to the external reader. Further details of this activation process are out of scope for the present document.

NOTE 2: Application selection of a contactless card application (using e.g. AID) is out of scope for the present document.

Figure 14 illustrates a valid contactless platform.



NOTE: Non-standard applications refer to type A contactless card applications that that do not use the transmission protocol defined in ISO/IEC 14443-4 [7].

Figure 14

9.2 Void

9.3 Gates

9.3.1 Void

Table 24: Void

9.3.2 Identity management gate

The following parameter shall be added to the identity management gate of the host controller as defined in clause 7.1.3.

Table 25

Identifier	Parameter	Access Right	Description	Length	Default
'07'	LOW_POWER_SUPPORT	RO	Indicates the power mode that is supported; where '01' indicates that low power mode is supported and '00' that it is not supported.	1	N/A

9.3.3 Card RF gates

9.3.3.1 Overview

Card RF gates are located in the host controller. A card RF gate controls the behaviour of a RF technology. Protocol and parameter settings related to the RF technology are available in the RF gate's registry.

The following clauses define the commands and events that a card RF gate supports in addition to those of a generic gate as defined in clause 7.2.

Table 26 lists the G_{ID} for the defined card RF gates.

Table 26

Gate	G_{ID}
Type B card RF gate	'21'
Type B' card RF gate	'22'
Type A card RF gate	'23'
Type F card RF gate	'24'
Reserved for proprietary use	'25' to '29'

9.3.3.2 Commands

There are no additional commands defined.

9.3.3.3 Events

9.3.3.3.0 Supported events

The card RF gates supports the event listed in table 27. The events are described in the following clauses.

Table 27

Value	Event
'10'	EVT_SEND_DATA

9.3.3.3.1 EVT_SEND_DATA

This event allows the sending of data to the CLF.

The event has parameters as follows:

Table 28

Description	Length
data that are sent	N

9.3.3.4 Registry

9.3.3.4.0 Registries persistence

All registries shall be persistent.

9.3.3.4.1 RF technology type A

Table 29 defines the registry entries for a card RF gate for RF technology type A.

Table 29

Identifier	Parameter	Access Right	Comment	Length	Default
'01'	MODE	RW	Type A card emulation enabled indicator	1	'FF'
'02'	UID_REG	WO	UID as defined in ISO/IEC14443-3 [6] for type A	N_0=0, 4, 7, 10	N_0=0
'03'	SAK	RW	Preformatted SAK as defined in ISO/IEC 14443-3 [6] for type A	1	'00'
'04'	ATQA	RW	Preformatted ATQA; byte 1 shall contain b8..1, byte 2 b16..9 of the ATQA coded as defined in ISO/IEC 14443-3 [6] for type A	2	'0000'
'05'	APPLICATION_DATA	RW	The historical bytes as defined in ISO/IEC 14443-4 [7] for type A	N_1	N_1=0
'06'	FWI, SFGI	RW	Frame waiting time and Start-up frame guard time as defined in ISO/IEC 14443-4 [7] for type A	1	'EE'
'07'	CID_SUPPORT	RW	support for CID as defined ISO/IEC 14443-4 [7]	1	'00'
'08'	CLT_SUPPORT	RO	Support of tunnelling mode for ISO/IEC 14443-4 [7] non compliant protocols	1	'00'
'09'	DATARATE_MAX	RW	Maximum data rate supported	3	'030300'

Coding for the MODE is as follows:

- 'FF' = Type A card emulation not enabled; and
- '02' = Type A card emulation enabled; and
- All other values are RFU.

Coding for UID_REG is as follows:

- If Length equals 0 then the CLF shall generate a single size UID with uid0 = '08' and uid1 to uid3 as random number. The random number shall be generated only on state transitions from POWER-OFF to IDLE state (state definitions according to ISO/IEC 14443-3 [6]). The CLF shall interpret the absence of an RF-field as POWER-OFF state.

- If Length equals 4, 7 or 10 then the CLF shall use UID_REG as UID.

Coding for the CID_SUPPORT parameter:

- '01' = the support of the CID is required; the CLF shall set CID support in the ATS; and
- '00' = the support of the CID is not required; the CLF may or may not set CID support in the ATS.
- All other values are RFU.

Coding for the CLT_SUPPORT parameter:

- '01' = the CLF contains a tunnelling mode capability for type A ISO/IEC 14443-4 [7] non compliant protocol support; and
- '00' = the CLF does not contain any tunnelling mode capability for type A ISO/IEC 14443-4 [7] non compliant protocol support.

DATARATE_MAX codes the maximum divisor supported, with the coding as follows:

- Byte 1 defines the maximum divisor supported in direction PCD to PICC:
  - '00' = the maximum divisor supported is 1 (~106 kbit/s);
  - '01' = the maximum divisor supported is 2 (~212 kbit/s);
  - '02' = the maximum divisor supported is 4 (~424 kbit/s);
  - '03' = the maximum divisor supported is 8 (~848 kbit/s);
  - All other values are RFU.
- Byte 2 defines the maximum divisor supported in direction PICC to PCD:
  - '00' = the maximum divisor supported is 1 (~106 kbit/s);
  - '01' = the maximum divisor supported is 2 (~212 kbit/s);
  - '02' = the maximum divisor supported is 4 (~424 kbit/s);
  - '03' = the maximum divisor supported is 8 (~848 kbit/s);
  - All other values are RFU.
- Byte 3 defines a limitation to support different divisors for each direction:
  - '00' = different divisors for each directions (PICC to PCD and PCD to PICC) is supported;
  - '01' = only the same divisor in both direction (PICC to PCD and PCD to PICC) is supported;
  - All other values are RFU.

NOTE: The resulting actual maximum supported divisor indicated by the interface byte TA(1) as defined in ISO/IEC 14443-4 [7] is given by the minimum of:

- the value and limitation as indicated in this registry; and
- the maximum divisor implemented in the CLF.

9.3.3.4.2 RF technology type B

Table 30 defines the registry entries for a RF gate for RF technology type B.

Table 30

Identifier	Parameter	Access Right	Comment	Length	Default
'01'	MODE	RW	Type B card emulation enabled indicator	1	'FF'
'02'	PUPI_REG	WO	PUPI as defined in ISO/IEC 14443-3 [6] for type B.	N_0	N_0=0
'03'	AFI	RW	Application family identifier as defined in ISO/IEC 14443-3 [6] for type B	1	'00'
'04'	ATQB	RW	Additional data for ATQB	4	See table 31
'05'	HIGHER_LAYER_RESPONSE	RW	Higher Layer response in answer to ATTRIB command as defined in ISO/IEC 14443-3 [6] for type B	N_2	N_2=0
'06'	DATARATE_MAX	RW	Maximum data rate supported	3	'030300'

Coding for the MODE parameter:

- 'FF' = Type B card emulation not enabled; and
- '02' = Type B card emulation enabled; and
- All other values are RFU.

Coding for the PUPI_REG is as follows:

- In case N=0 then the CLF shall generate the PUPI as dynamically generated number. The PUPI shall only be generated by a state transition from the POWER-OFF to the IDLE state (state definitions according to ISO/IEC 14443-3 [6]). The CLF shall interpret the absence of an RF-field as POWER-OFF state.
- In all other cases, the CLF shall use the PUPI _REG as PUPI.

Structure of the ATQB parameter.

Table 31

Parameter	Comment	Length	Default
CRC_AID	Compressed CRC AID as described in the ISO/IEC 14443-3 [6]	2	'0000'
NUMBER_APLI	Number of applications as described in the ISO/IEC 14443-3 [6]	1	'00'
PROTO_INFO	Third byte of the Protocol information as described in the ISO/IEC 14443-3 [6]	1	'E4'
NOTE: PROTO_INFO also encodes FWI and support for CID, which are separate registry parameters in case of Type A.			

DATARATE_MAX codes the maximum bit rates supported, with the coding as follows:

- Byte 1 defines the maximum bit rate supported in direction PCD to PICC:
    - '00' = the maximum bit rate supported is fc/128 (~106 kbps);
    - '01' = the maximum bit rate supported is fc/64 (~212 kbps);
    - '02' = the maximum bit rate supported is fc/32 (~424 kbps);
    - '03' = the maximum bit rate supported is fc/16 (~848 kbps);
    - All other values are RFU.
- Byte 2 defines the maximum bit rate supported in direction PICC to PCD:
    - '00' = the maximum bit rate supported is fc/128 (~106 kbps);

- '01' = the maximum bit rate supported is fc/64 (~212 kbps);
- '02' = the maximum bit rate supported is fc/32 (~424 kbps);
- '03' = the maximum bit rate supported is fc/16 (~848 kbps);
- All other values are RFU.

- Byte 3 represents the limitation of having the bit rate in both direction compulsory:
  - '00' = different bit rates in both directions (PICC to PCD and PCD to PICC) are supported;
  - '01' = only the same bit rate in both directions (PICC to PCD and PCD to PICC) is supported;
  - All other values are RFU.

NOTE: The resulting actual maximum bit rate supported is indicated in the first byte of the protocol information as defined in ISO/IEC 14443-3 [6] is given by the minimum of:

- the value and limitation as indicated in this registry; and
- the maximum divisor implemented in the CLF.

9.3.3.4.3 RF technology type B'

Table 32 defines the registry entries for a RF gate for the RF technology type 'B'.

Table 32

Identifier	Parameter	Access Right	Comment	Length	Default
'01'	MODE	RW	Type B' card emulation enabled indicator	1	'FF'
'02'	PAT_IN	RW	Pattern to match with the incoming data	N_0	N_0=0
'03'	DAT_OUT	RW	Data to reply if the incoming data matches with PAT_IN pattern	N_1	N_1=0

Coding for the MODE parameter:

- 'FF' = Type B' card emulation not enabled; and
- '02' = Type B' card emulation enabled; and
- All other values are RFU.

9.3.3.4.4 RF technology Type F (ISO/IEC 18092 212 kbps/424 kbps card emulation only)

Table 33 defines the registry entries for a RF gate for the RF technology Type F.

Table 33

Identifier	Parameter	Access Right	Comment	Length	Default
'01'	MODE	RW	Type F card emulation enabled indicator	1	'FF'
'02'	SPEED_CAP	RO	CLF Speed Capabilities for Type F	1	'01'
'03'	CLT_SUPPORT	RO	Support of tunnelling mode for anti-collision data exchange in type F card emulation	1	'00'

Coding for the MODE parameter:

- 'FF' = Type F card emulation not enabled; and
- '02' = Type F card emulation enabled; and
- All other values are RFU.

Table 34 defines the coding for the SPEED_CAP parameter.

Table 34

SPEED_CAP	Meaning
b8	1: auto bit rate detection supported 0: auto bit rate detection not supported
b7..b3	RFU
b2	1: 424 kbps bit rate supported 0: 424 kbps bit rate not supported
b1	1: 212 kbps bit rate supported 0: 212 kbps bit rate not supported

Coding for the CLT_SUPPORT parameter:

- '01' = the CLF contains a tunnelling mode capability for type F card emulation anti-collision support; and
- '00' = the CLF does not contain any tunnelling mode capability for type F card emulation anti-collision support.

9.3.4 Card application gates

9.3.4.1 Overview

A card application gate controls the access towards contactless card applications.

The next clauses define the commands and events that a card application gate supports in addition to those of a generic gate as defined in clause 7.2.

The G_{ID} for the card application gates are dynamically assigned by the host.

9.3.4.2 Commands

No additional command defined.

9.3.4.3 Events

9.3.4.3.0 Supported events

The card application gates support the events listed in table 35. The events shall be described in the following clauses.

Table 35

Value	Event
'10'	EVT_SEND_DATA
'11'	EVT_FIELD_ON
'12'	EVT_CARD_DEACTIVATED
'13'	EVT_CARD_ACTIVATED
'14'	EVT_FIELD_OFF

9.3.4.3.1 EVT_FIELD_ON

This event signals that the CLF detected an RF carrier from an external reader. This event shall be sent by the host controller within 2ms after the detection of an RF field if the card RF gate is open with the MODE parameter set to '02'.

In case of an underlying data link layer according to ETSI TS 102 613 [2], the following exception applies: if SWP is in DEACTIVATED state, the CLF shall activate the interface instead of sending the EVT_FIELD_ON.

The event has no parameters.

9.3.4.3.2 EVT_CARD_DEACTIVATED

This event signals that the CLF is deactivated as defined in ISO/IEC 14443-3 [6] for type B or ISO/IEC 14443-4 [7] for type A.

The event has no parameters.

9.3.4.3.3 EVT_CARD_ACTIVATED

This event signals that the CLF is activated as defined in ISO/IEC 14443-3 [6] for type B or ISO/IEC 14443-4 [7] for type A.

The event has no parameters.

9.3.4.3.4 EVT_FIELD_OFF

This event signals that the CLF detected the loss of the RF carrier from the external reader.

The event has no parameters.

9.3.4.3.5 EVT_SEND_DATA

This event allows the sending of data to the host from the CLF. The event has parameters as follows:

Table 36

Description	Length
data that are sent	N
RF error indicator	1

Coding for the RF error indicator:

- '00' = no error; and
- '01' = error; and
- All other values are RFU.

NOTE: The end of the data and the beginning of the RF error indicator can unambiguously be derived from the message size. The RF error indicator is the last parameter, as the first segments of a long segmented message may already have been sent on the interface to the host before the sender (CLF) can do the integrity check of the original message received on the contactless interface.

The event shall be discarded by the host when the error indicator is set to '01'.

9.3.4.4 Registry

The card application gate does not have registry entries defined.

9.4 Procedures

9.4.0 Overview and basic rule

The procedures in clause 9.4 are described for full power mode. If the procedures are different in low power mode then these exceptions are described.

The CLF shall only send events defined in table 35 to the Card application gate or open a CLT session as defined in ETSI TS 102 613 [2] for the corresponding RF technology if the MODE parameter in the associated Card RF gate registry is set to '02' (enabled).

9.4.1 Use of contactless card application

Figure 15 illustrates the messages that type A and type B card RF gates and a card application gate exchange when entering a reader field.

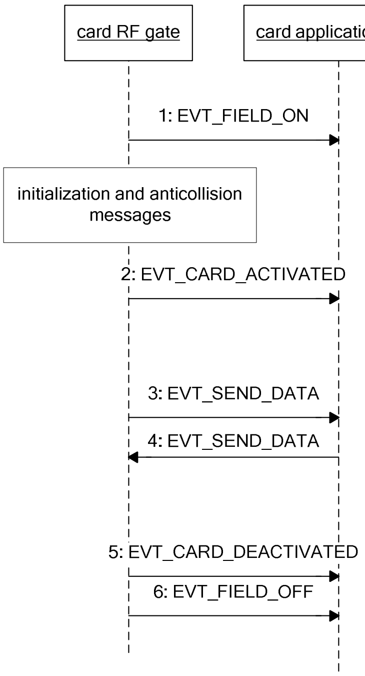


Figure 15

The sequence flow is as follows:

1) When the CLF detects a RF field, the card RF gate shall send the event EVT_FIELD_ON to the card application gate as specified in clause 9.3.4.3.1 if the card RF gate is open with the MODE parameter set to '02'. When there are multiple open card RF gates with the MODE parameter set to '02', the CLF shall send the EVT_FIELD_ON to the corresponding card application gate with the lowest G_{ID}. Next the CLF starts the initialization and anti-collision process as defined in ISO/IEC 14443-3 [6].

   In low power mode, the event EVT_FIELD_ON may be omitted.

2) At the end of the activation sequence as defined ISO/IEC 14443-4 [7] the card RF gate shall send EVT_CARD_ACTIVATED to the card application gate to indicate that APDUs may be exchanged.

3) The card RF gate shall forward the C-APDUs from the external contactless reader to the card application gate using the EVT_SEND_DATA.

4) The host shall reply by sending the R-APDU in an EVT_SEND_DATA event to the card RF gate. The CLF may forward an empty C-APDU. The host shall reply to this with either an empty R-APDU or an R-APDU containing an error code.

5) The steps 3 and 4 may be repeated.

6) At the end of the deactivation sequence the card RF gate shall send an EVT_CARD_DEACTIVATED to signal that no more APDUs can be exchanged. This event will only be sent when the external contactless reader supports it.

When the CLF detects at any time during the sequence above that the RF field is off, the card RF gate shall send EVT_FIELD_OFF to the card application gate if the card RF gate is open with the MODE parameter set to '02'. When there are multiple open card RF gates with the MODE parameter set to '02', the CLF shall send the EVT_FIELD_OFF to the card application gate used during the transaction or to the corresponding card application gate with the lowest G_{ID}. In low power mode the event EVT_FIELD_OFF may be omitted if the host is powered down instead.

9.4.2 Non ISO/IEC 14443-4 type A applications

For type A contactless card, where the external reader does not use the transmission protocol defined in ISO/IEC 14443-4 [7], the flow is as follows:

- The card RF gate shall send the EVT_FIELD_ON as specified in clause 9.4.1. The CLF handles initialization and anti-collision.
  In low power mode the event EVT_FIELD_ON may be omitted.

- Any other communications are done using the CLT mode as defined in ETSI TS 102 613 [2].

When the CLF detects at any time during the sequence above that the RF field is off, the card RF gate shall send EVT_FIELD_OFF as specified in clause 9.4.1. In low power mode the event EVT_FIELD_OFF may be omitted if the host is powered down instead.

The rules for establishing CLT mode are defined in ETSI TS 102 613 [2].

9.4.3 Type B' RF technology

For type B' contactless card applications the flow is as follows:

- The card RF gate shall send the EVT_FIELD_ON as specified in clause 9.4.1. In low power mode the event EVT_FIELD_ON may be omitted.

- The RF incoming raw frames are matched with the pattern in the parameter PAT_IN. The rules for matching are as follows:

  - PAT_IN shall have an even number of bytes. The first half is the reference data REF, the second half is the mask M. If RFIN represents the received frame (the data bytes between SOF and the CRC), matching is successful if the following bitwise calculation has an all ones result:

    - (RFIN XOR REF) OR M.

- If the matching is successful then the CLF shall prefix the first byte of the received frame to the bytes of the parameter DAT_OUT, add the CRC and send the result to the RF reader. The time between the end of the RF incoming frame and the beginning of the answer to the reader shall be less than 1 ms.

- Before the successful matching the incoming raw frames are not transferred to the host.

When the CLF detects at any time during the sequence above that the RF field is off, the card RF gate shall send EVT_FIELD_OFF to the card application gate as specified in clause 9.4.1. In low power mode the event EVT_FIELD_OFF may be omitted if the host is powered down instead.

9.4.4 Type F RF technology

For ISO/IEC 18092 [4] based contactless card emulation applications in 212 kbps/424 kbps passive mode as used in existing widespread infrastructure, the following applies:

1) The card RF gate sends the EVT_FIELD_ON as specified in clause 9.4.1. In low power mode the event EVT_FIELD_ON may be omitted.

2) In case SWP as defined in ETSI TS 102 613 [2] is used as a data link layer, if an initialization command is received, then the initialization data exchange is performed using CLT as defined in ETSI TS 102 613 [2] The host provides information for the initialization.

3) The card RF gate shall forward the ISO/IEC 18092 [4] 212 kbps/424 kbps frames from the external reader to the card application gate using the EVT_SEND_DATA. The host and the host controller shall be able to receive the ISO/IEC 18092 [4] 212 kbps/424 kbps frames without a previous initialization in the step before. The CLF shall not forward RF frames to the host if a response from the host is pending. If an RF frame was received but the response from the host is still pending the received RF frame shall be discarded.

   The resulting HCP packet has the following structure:

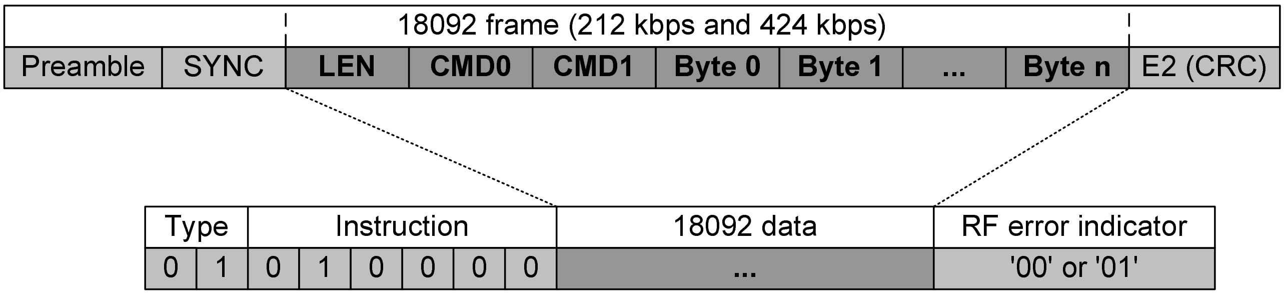


Figure 15a

4) The host sending a response shall encapsulate the ISO/IEC 18092 [4] 212 kbps/424 kbps frames in an EVT_SEND_DATA event and send it to the card RF gate. The host shall send an empty EVT_SEND_DATA event to the card RF gate if there is no application data to be sent. The CLF shall not send anything to RF but shall be able to receive RF frames after receiving an empty EVT_SEND_DATA from the host.

5) Steps 3 and 4 may be repeated.

When the CLF detects at any time during the sequence above that the RF field is off, the card RF gate shall send EVT_FIELD_OFF to the card application gate as specified in clause 9.4.1. In low power mode the event EVT_FIELD_OFF may be omitted if the host is powered down instead.

ISO/IEC 18092 [4], 212 kbps/424 kbps frames, except initialization command and response (command code '00' and '01), shall be exchanged using the appropriate gate depending on the command code of the frame as described in table 37. The command code (CMD0) is the byte after LEN as defined in ISO/IEC 18092 [4]. The command codes reserved for the NFCIP-1 protocol shall not be forwarded.

Table 37

Command Code (CMD0)	Send via
'00' (POLLING_REQUEST), '01' (POLLING_RESPONSE)	CLT mode
'D4' (NFCIP-1 REQ), 'D5' (NFCIP-1 RES)	Reserved for ISO/IEC 18092 [4] NFCIP-1
Other	ISO/IEC 18092 [4] 212 kbps/424 kbps card emulation application gate

9.4.5 Update RF technology settings

To avoid inconsistencies in parameter settings, the following procedure is recommended when updating the settings in the CLF for the different RF technologies:

1) The host first sets the MODE parameter for the RF technology to be updated to 'FF' to disable the use of this technology by the CLF for the host.

2) Now all other parameters of the RF technology in the CLF can be updated by the host.

3) Finally, the host can enable the RF technology again by setting the MODE parameter in the CLF to '02'.

9.4.6 Identity check

For the data link layer specified in ETSI TS 102 613 [2] the following shall apply:

In addition to what is defined in clause 8.4, in case the lower layer identity check fails, the host controller shall inhibit card emulation mode for the UICC host, i.e. the host controller shall not respond to an external contactless reader with any parameter from the card emulation registries related to the UICC host.

10 Contactless reader

10.1 Overview

This clause specifies how a contactless reader supporting multiple RF technologies is emulated on top of HCI core.

The RF technologies that are supported are:

- type A as defined in ISO/IEC 14443 parts 2 [5], 3 [6] and 4 [7]; or
- type B as defined in ISO/IEC 14443 parts 2 [5], 3 [6] and 4 [7].

The contactless platform for reader mode is realized as a host network with two hosts; the host controller and the host (e.g. the UICC host). The CLF shall run the host controller of the host network. Additional hosts using the services of the CLF are out of the scope of the present document.

The platforms for reader and card emulation mode can coexist.

The host operates contactless reader applications and the CLF handles the RF communication layers towards the external contactless card(s).

The host controller has one reader RF gate for each RF technology it supports. For each reader RF gate it wants to use the host has one reader application gate. The host shall not create more than one pipe to each reader RF gate.

The CLF handles the RF layers of the communications as defined in ISO/IEC 14443-2 [5].

The anti-collision and activation as defined in ISO/IEC 14443-3 [6] is handled by the CLF under the control of the host.

The RF protocol as defined in ISO/IEC 14443-4 [7] is handled by the CLF.

The reader RF gate and reader application gate exchange APDUs defined in ISO/IEC 7816-4 [8] over their pipe.

The host is allowed to consume a current up to the maximum defined by the host controller in its identity management gate registry between the notification of a target discovery and the end of operation on this target unless restricted by the underlying layers e.g. ETSI TS 102 613 [2] where the restrictions for low-power mode and power saving mode still apply.

Figure 16 illustrates a valid contactless platform.

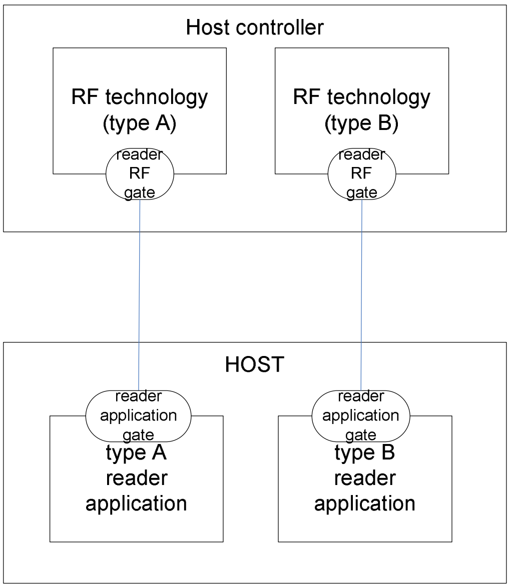


Figure 16

10.2 Reader RF gates

10.2.1 Overview

A reader RF gate controls the behaviour of a RF technology. Protocol and parameter settings related to the reader RF technology are available in the reader RF gate's registry. The reader RF gates may accept a single target in the field of the reader irrespective of the RF technology.

The following clauses define the commands and events that a reader RF gate supports in addition to those of a generic gate as defined in clause 7.2.

Table 38 lists the G_{ID} for the defined reader RF gates.

Table 38

Gate	G_{ID}
ISO/IEC 14443 Type B reader RF gate	'11'
reserved for proprietary use	'12'
ISO/IEC 14443 Type A reader RF gate	'13'
reserved for proprietary use	'14' to '1B', '7E', '7F'

10.2.2 Command

10.2.2.1 WR_XCHG_DATA

This command allows the sending of data from a reader application gate to the reader RF gate; the data sent in this command is intended for transmission over the RF. A reply from an external device is returned to the reader application gate in the response to this command.

Table 39

Value	Command
'10'	WR_XCHG_DATA

This command has following parameters:

Table 40

Description	Length
CTR	1
data sent	N

The CTR field is defined as follows:

- b8 to b6: RFU.
- b5: Time-out activation: If b5 is set to zero, application level time-out is deactivated. If b5 is set to one, the time-out is given in b4 to b1.
- b4 to b1: Time-out. This Value (0 up to 14) is used to calculate the application level time-out by the following formula:

Time-out = (256 × 16 / 13,56 MHz) × 2 ^ Value.

If b5 of the CTR field is set to one and no response is received by the CLF from a target within the time-out period commencing after the error free transmission of this command by the CLF to the target, the CLF shall respond to the host with ANY_E_TIMEOUT with no parameters and shall discard data received from the target thereafter.

If the CLF detects an RF transmission error and the error detection and recovery procedure defined by ISO/IEC 14443-4 [7] does not succeed then the CLF shall respond to the host with ANY_OK with the error indicator set to '01'.

When successful the host shall respond with ANY_OK with parameter as follows:

Table 41

Description	Length
Data received	N
RF error indicator	1

Coding for the RF error indicator:

- '00' = no error; and
- '01' = non-recoverable error; and
- All other values are RFU.

NOTE: The first segment of a long segmented message may already have been sent on the interface to the host before the host controller can do the integrity check of the original message received on the contactless interface. The same response is to be used if the transmission of the response over the contactless interface is interrupted.

10.2.3 Registries

10.2.3.1 Type A reader RF gate

Table 42 defines the registry for the RF gate ISO/IEC 14443 Type A reader RF gate. The registry is not persistent. The values '02' to '06' are updated after each target activation. The value '07' is updated when the operating status changes.

Table 42

Type	Identifier	Parameter	Access Right	Comment	Length	Default
MANDATORY	'02'	UID	RO	UID as defined in ISO/IEC 14443-3 [6] for type A	4, 7, 10	'08000000'
	'04'	ATQA	RO	Preformatted ATQA; byte 1 shall contain b8..1, byte 2 b16..9 of the ATQA coded as defined in ISO/IEC 14443-3 [6] for type A	2	'0000'
	'05'	APPLICATION_DATA	RO	The historical bytes as defined in ISO/IEC 14443-4 [7] for type A	N	N=0
	'03'	SAK	RO	Preformatted SAK as defined in ISO/IEC 14443-3 [6] for type A	1	'00'
	'06'	FWI, SFGT	RO	Frame waiting time and start-up frame guard time as defined in ISO/IEC 14443-4 [7] for type A	1	'EE'
	'01'	DATARATE_MAX	RW	Maximum data rate supported	1	'00'
MANDATORY, not present in Rel-10 and earlier	'07'	OPERATING_STATUS	RO	Indication of the current operating status of the reader	1	'00'
	'08'	STATUS_EVENT_EN	RW	Enable for reader status events	1	'00'

Coding for the DATARATE_MAX is as follows:

- only the same divisor (ISO/IEC 14443-4 [7]) for both directions (PICC to PCD and PCD to PICC) is supported:
  - '00' = the maximum divisor supported is 1.
  - '01' = the maximum divisor supported is 2.
  - '02' = the maximum divisor supported is 4.
  - '03' = the maximum divisor supported is 8.

NOTE: The actual maximum supported divisor is given by the minimum of:

- ▪ the value as indicated in this registry; and
- ▪ the maximum divisor implemented in the CLF.

- All other values are RFU.

Coding of the OPERATING_STATUS is as follows:

b8-2 : RFU

b1 : Reader status

The CLF shall set b1 to 0 when the Type A reader mode is not available for the host (e.g. the CLF is exclusively active for another host).

The CLF shall set b1 to 1 when the Type A reader mode is available for the host.

Coding of the STATUS_EVENT_EN is as follows:

b8-2 : RFU

b1 : Reader status event enable

The Reader status event enable (b1) indicates to the CLF that the host requests events to be sent when the reader status changes and by the same method may indicate to the CLF that it does not wish to receive such events as specified in detail in clause 10.3.4.2.

10.2.3.2 Type B reader RF gate

Table 43 defines the registry for the RF gate ISO/IEC 14443 Type B reader RF gate. The registry is not persistent. The values '01', '03', '04' and '02' (read value) are updated after each target activation. The value '07' is updated when the operating status changes.

Table 43

Type	Identifier	Parameter	Access Right	Comment	Length	Default
MANDATORY	'03'	PUPI	RO	PUPI as defined in ISO/IEC 14443-3 [6] for type B	N_0	N_0=0
	'04'	APPLICATION_DATA	RO	Application data as defined in ISO/IEC 14443-3 [6] for ATQB	N_1	N_1=0
	'02'	AFI	RW	Application family identifier as defined in ISO/IEC 14443-3 [6] for type B used by the reader to poll the target	1	'00'
	'01'	HIGHER_LAYER_RESPONSE	RO	Higher layer response as defined in ISO/IEC 14443-3 [6] for answer to ATTRIB command	N_2	N_2=0
	'05'	HIGHER_LAYER_DATA	RW	Higher layer data as defined in ISO/IEC 14443-3 [6] for ATTRIB command	N_3	N_3=0
MANDATORY, not present in Rel-10 and earlier	'07'	OPERATING_STATUS	RO	Indication of the current operating status of the reader	1	'00'
	'08'	STATUS_EVENT_EN	RW	Enable for reader status events	1	'00'

Coding of the AFI is as follows:

- The value written to the registry shall be used by the reader to poll the target.
- Once a target is activated, the registry value shall reflect the AFI value of the target.

Coding of the OPERATING_STATUS is as follows:

b8-2 : RFU

b1 : Reader status

The CLF shall set b1 to 0 when the Type B reader mode is not available for the host (e.g. the CLF is exclusively active for another host).

The CLF shall set b1 to 1 when the Type B reader mode is available for the host.

Coding of the STATUS_EVENT_EN is as follows:

b8-2 : RFU

b1 : Reader status event enable

The Reader status event enable (b1) indicates to the CLF that the host requests events to be sent when the reader status changes and by the same method may indicate to the CLF that it does not wish to receive such events as specified in detail in clause 10.3.4.2.

10.2.4 Events

10.2.4.0 Supported events

The reader RF gates support the events listed in table 44. The events shall be described in the following clauses.

Table 44

Value	Event
'10'	EVT_READER_REQUESTED
'11'	EVT_END_OPERATION

10.2.4.1 EVT_READER_REQUESTED

With this event the host signals to the CLF that a reader application gate is requesting to activate the RF polling (turn on the RF carrier). The method and the algorithm used by the reader RF gates to achieve target polling are implementation dependant and out of the scope of the present document. This event has no parameters and may be sent over any open pipe of any reader RF gate.

10.2.4.2 EVT_END_OPERATION

The host sends this event to inform the CLF Controller of the end of operation. Upon reception of the event the CLF controller shall turn the RF field OFF if the EVT_TARGET_DISCOVERED has been previously sent to that specific host, otherwise the CLF may turn the RF field OFF and move forward in its internal processing.

10.2.5 Responses

For the command specified in clause 10.2.2.1, table 45 lists the possible response codes. Unless otherwise stated these responses carry no additional parameters.

Table 45

Value	Response	Description
'00'	ANY_OK	command completed successfully (with optional parameters)
'03'	ANY_E_NOK	command was rejected and/or not completed
'09'	ANY_E_TIMEOUT	an application level time-out occurred
'10'	WR_RF_ERROR	The target has returned an RF error

10.3 Reader application gates

10.3.1 Overview

A reader application gate controls the access to contactless reader applications.

The following clauses define the commands and events that a reader application gate supports in addition to those of a generic gate as defined in clause 7.2.

The G_{ID} for the reader application gates are dynamically assigned by the host.

10.3.2 Command

No additional command defined.

10.3.3 Registry

The reader application gate does not have registry entries defined.

10.3.4 Events

10.3.4.0 Supported events

The reader application gates support the events listed in table 46. The event shall be described in the following clause.

Table 46

Value	Event
'10'	EVT_TARGET_DISCOVERED
'11'	EVT_READER_STATUS

10.3.4.1 EVT_TARGET_DISCOVERED

This event signals to the reader application gate the existence of an RF target in the field of the activated RF technology. The event notifies the reader application gate about the availability of the RF information in the registry. The activation of the target is completed and the associated registry may inform the application about the RF feature of the application.

This event has one parameter:

Table 47

Description	Length
STATUS	1

Coding for the status field:

- '00' = when there is a single target in the reader field and the activation of the target is completed; and
- '03' = when there are several targets in the field irrespective of the RF technology.

10.3.4.2 EVT_READER_STATUS

This event indicates to a host that the availability status of the reader RF gate for a particular RF technology has changed.

The CLF shall send this event (see also Annex A)

- once after the reader status event enable bit (b1) in the STATUS_EVENT_EN is changed to 1, and
- as long as the reader status event enable bit (b1) in the corresponding STATUS_EVENT_EN remains at 1 whenever the availability status of a reader RF gate has changed.

Even if multiple pipes from one host to reader RF gates are open, this event shall be sent over one of these pipes only.

If different types of the reader mode become available/unavailable for the host at different points in time, the CLF shall send one event for each change.

The CLF shall not send the EVT_READER_STATUS if the reader status event enable bit (b1) in the corresponding STATUS_EVENT_EN is set to 0.

This event has one parameter.

Table 47a

Description	Length
READER_ STATUS	1

Coding for the reader status field:

b8-3 : RFU

b1 : Type A reader status

0 = reader mode for Type A targets is not available for this host.

1 = reader mode for Type A targets is available for this Host.

b2 : Type B reader status

0 = reader mode for Type B targets is not available for this host.

1 = reader mode for Type B targets is available for this host.

10.4 Procedures

10.4.0 Introduction

The following procedures are described for the reader mode.

10.4.1 Use of contactless reader application

Figure 17 illustrates the messages that type A and type B reader RF gates and a reader application gate exchange when detecting a target.

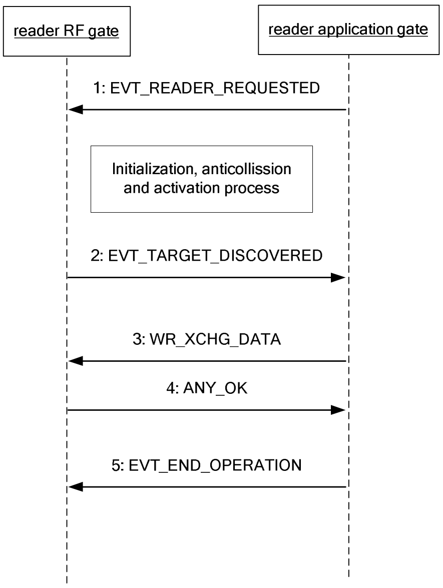


Figure 17

The sequence flow is as follows:

1) When the host sends the EVT_READER_REQUESTED to the CLF to enable the RF polling. The CLF starts the detecting of a target according to all reader RF gates of the host that have an open pipe. The host shall send the EVT_READER_REQUESTED on a single pipe only.

2) A target has been detected and activated. The CLF notifies the host via the event EVT_TARGET_DISCOVERED. If there are several targets in the field then the procedure shall stop. The host may read the reader RF gate registry to get RF specific information (e.g. AFI, etc.).

3) The reader application gate may send C-APDU over the pipe by using the command WR_XCHG_DATA.

4) The reader RF gate replies in sending back an R-APDU.

The steps 3 and 4 may be repeated.

5) When the operation is completed, the host shall send the EVT_END_OPERATION to turn off the RF carrier as defined above. The host shall send the EVT_END_OPERATION on a single pipe only.

10.4.2 Contactless reader not available

Figure 17a illustrates the messages that type A and type B reader RF gates and a reader application gate may exchange when the reader is not available.

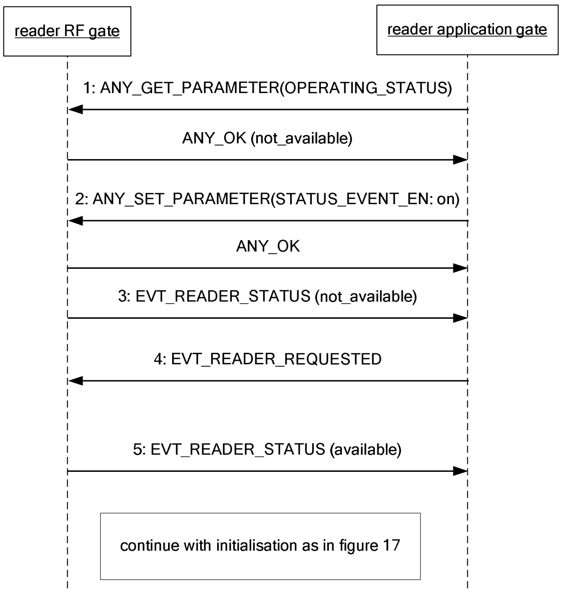

reader RF gate
reader application gate
1: ANY_GET_PARAMETER(OPERATING_STATUS)
ANY_OK (not_available)
2: ANY_SET_PARAMETER(STATUS_EVENT_EN: on)
ANY_OK
3: EVT_READER_STATUS (not_available)
4: EVT_READER_REQUESTED
5: EVT_READER_STATUS (available)
continue with initialisation as in figure 17

Figure 17a

The sequence flow is as follows:

1) Before starting reader mode, the host checks if the reader is available.

2) If the reader is not available, the host enables the reader status event.

NOTE: Step 1 is optional. The host may also start directly with enabling of the reader status event.

3) In response the CLF reports that the reader is (still not) available.

4) The host sends the event reader requested and waits for the reader to become available.

5) Once the reader becomes available for the host, the CLF activates the reader for the host and the reader RF gates signals this via the event reader status.

A host may decide to always activate the reader status event and skip checking of the status in step 1. If the reader is available when the event reader requested is sent, the host will then receive the event reader status as a confirmation immediately.

10.4.3 Error management

Figure 17b illustrates the sequence the Host should follow in case the CLF reports an RF error in the response to a WR_XCHG_DATA command.

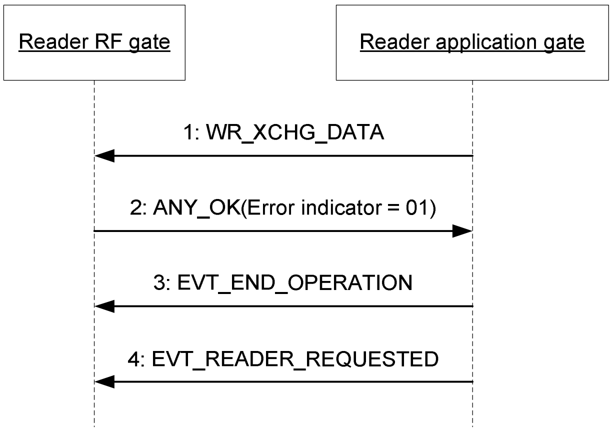


Figure 17b

The sequence flow is as follows:

1) Assuming that the Host has triggered the reader RF gate by sending the EVT_READER_REQUESTED and that the CLF has notified the host via the event EVT_TARGET_DISCOVERED, the Host can send the WR_XCHG_DATA command.

2) The CLF detected a non-recoverable RF error and the reader RF gate replies in sending the response ANY_OK with parameter "Error indicator" set to '01'.

3) The Host stops the reader operation by sending the EVT_END_OPERATION.

4) The Host may want to restart the reader operation by sending the EVT_READER_REQUESTED.

11 Connectivity

11.1 Overview

This clause defines:

- A mechanism that allows a UICC to start a proactive session as defined in ETSI TS 102 223 [3] whenever this is required in the context of a contactless transaction.

- A mechanism for a host to launch an application in the terminal host that is related to the application running in the host.

- Commands and events needed by the terminal host in order to perform power management of the system.

- A registry indicating the availability of the user interface of the terminal.

From the host perspective these commands and events are addressed to the connectivity gate in the terminal host. However it is implementation dependent whether there is a real HCI to the terminal or whether it is implemented virtually in the CLF and proprietary mechanisms between the CLF and the terminal are used to trigger the sending of commands and storing of registry information.

Support of the connectivity gate is optional for the terminal host.

11.2 Connectivity gate

11.2.0 Overview

The connectivity gate is a generic gate as defined in clause 7.2. As destination gate, the connectivity gate shall accept at least one pipe from each host in its WHITELIST.

Table 48 lists the G_{ID} for the connectivity gate.

Table 48

Gate	G_{ID}
Connectivity gate	'41'

The command and the events defined in the following clauses shall be sent to this gate.

11.2.1 Commands

11.2.1.1 PRO_HOST_REQUEST

This command allows a host to request the terminal host to activate during a limited time one or several other hosts in order to interact with them. This procedure shall not apply to request activation of the host controller or terminal host.

Table 49

Value	Command
'10'	PRO_HOST_REQUEST

This command has following parameters:

Table 50

Description	Length
Activation Duration (ms)	2
List of host identifiers	N

When successful for all requested hosts, the terminal host shall respond with ANY_OK with no parameters. The host shall not interact with another host after the expired activation duration time.

11.2.2 Events

11.2.2.0 Supported events

The connectivity gate supports the events listed in table 51. The events shall be described in the following clauses.

Table 51

Value	Event
'10'	EVT_CONNECTIVITY
'12'	EVT_TRANSACTION
'13'	EVT_OPERATION_ENDED

11.2.2.1 EVT_CONNECTIVITY

This event notifies the terminal host that it shall send a "HCI connectivity event" as defined in ETSI TS 102 223 [3].

This event has no parameters.

11.2.2.2 Void

11.2.2.3 EVT_OPERATION_ENDED

This event notifies the terminal host that the interaction with the hosts listed in the command PRO_HOST_REQUEST is ended before the activation duration time limit.

This event has no parameters.

11.2.2.4 EVT_TRANSACTION

This event notifies the terminal host that it shall launch an application which is associated to a UICC-hosted application identified by the AID on the parameter list.

NOTE: This method to launch an application does not replace the launch application supported by a proactive session as defined in ETSI TS 102 223 [3].

This event sends a list of parameters coded in BER-TLV fields as defined in ETSI TS 101 220 [9]. The tag and length of these fields are defined in the table 52.

Table 52

Description	Tag	Length
AID	'81'	5 to 16
PARAMETERS	'82'	0 to 255

The terminal host shall make the PARAMETERS available to the launched application.

11.2.3 Registry

The connectivity gate provides information about the terminal's user interface status (i.e. the availability of the screen to display a message to the end user and the availability of the touchpad or keypad for the end user to input information).

Table 52a lists the entries in the registry.

Table 52a

Identifier	Parameter	Access Right	Description	Length	Default
'01'	UI_STATE	RO	Terminal's User Interface availability	1	'00'

Coding for the UI_STATE is as follows:

- '00' = UI availability unknown.
- '01' = The UI is fully available, i.e. the screen is currently active and the terminal application can display a message to the user and/or ask for a user input.
- '02' = The UI is locked and the user cannot be notified.
- '03' = The UI is locked and the user can be notified.
- '04' = The UI is unlocked but the user cannot be notified.
- All other values are RFU.

The information provided by the terminal is only accurate at the time it is delivered by the terminal. The terminal should update this information whenever the availability state of the UI is modified.

11.3 Connectivity application gate

11.3.0 Overview

Clause 11.3 defines the commands and events that a connectivity application gate supports in addition to those of a generic gate as defined in clause 7.2.

The G_{ID} for the connectivity application gates are dynamically assigned by the host.

11.3.1 Commands

There are no additional commands.

11.3.2 Events

11.3.2.0 Supported events

The connectivity application gate support the event listed in table 53. The event shall be described in the following clause.

Table 53

Value	Event
'10'	EVT_STANDBY

11.3.2.1 EVT_STANDBY

This event should be used by the terminal host or the host controller to notify the host that it shall stop any ongoing communication with the other hosts and the host controller after 100 ms. The terminal host shall take care about not interrupting ongoing contactless transactions.

This event has no parameters.

11.3.3 Registry

There are no registry entries defined.

11.4 Connectivity gate procedures

11.4.1 Use of connectivity gate

Figure 18 illustrates the messages between a UICC host and a terminal host using the EVT_CONNECTIVITY event. Same shall apply for other events.

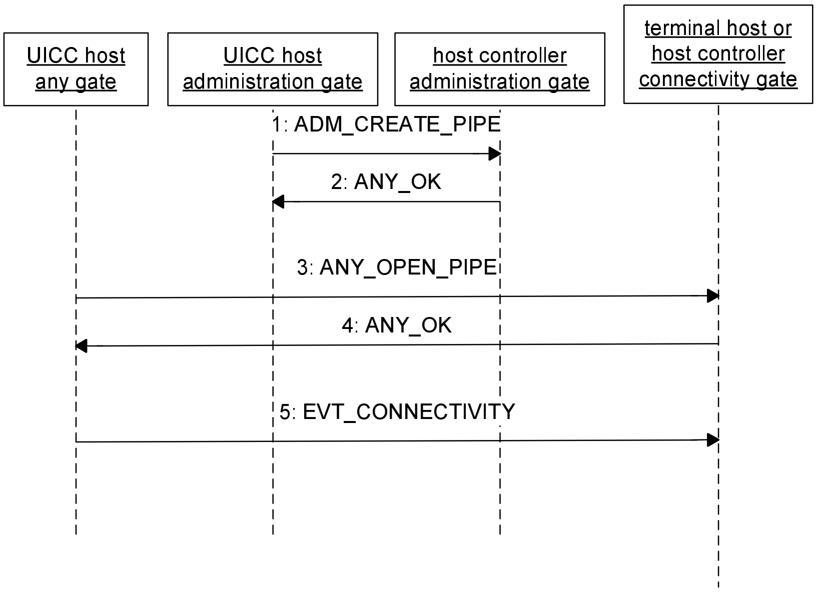

UICC host any gate
UICC host administration gate
host controller administration gate
terminal host or host controller connectivity gate
1: ADM_CREATE_PIPE
2: ANY_OK
3: ANY_OPEN_PIPE
4: ANY_OK
5: EVT_CONNECTIVITY

Figure 18

The sequence flow is as follows:

1) The UICC host requests the creation of a pipe to the connectivity gate in the terminal host.

2) The terminal host confirms as specified in clause 8.1.1.

3) UICC host opens the pipe.

4) Terminal host accepts the opening.

5) UICC sends the EVT_CONNECTIVITY to the terminal host.

12 APDU Transport

12.1 Overview

This clause defines a mechanism that allows an HCI host to send APDU commands to another HCI host and to receive an answer.

The host sending the APDU command is called the "client APDU host" and the host receiving and processing the APDU commands is called the "server APDU host".

A client APDU host shall not create more than one pipe to the APDU gate of a server APDU host.

A client APDU host should close the pipe to the APDU gate of a server APDU host after having sent the event EVT_END_OF_APDU_TRANSACTION.

Support of the APDU gate and APDU application gate are optional for any host.

APDUs are supported as defined in ISO/IEC 7816-4 [8]. Usage of APDUs with extended length fields is optional and only allowed if the value of the MAX_C-APDU_SIZE registry parameter exceeds the default value indicated in clause 12.2.3. Usage of both the basic logical channel and further logical channel(s) is allowed.

12.2 APDU gate

12.2.0 Description

Clause 12.2 defines the commands and events that an APDU gate of a server APDU host supports in addition to those of a generic gate as defined in clause 7.2.

The APDU gate may accept only an implementation specific maximum concurrent number of pipes from other hosts.

Table 54 lists the G_{ID} for the APDU gate.

Table 54

Gate	G_{ID}
APDU gate	'30'

The command and the events defined in the following clauses shall be sent to this gate.

12.2.1 Commands

There are no additional commands.

12.2.2 Events

12.2.2.0 Supported events

The APDU gate supports the events listed in table 55. The events are described in the following clauses. The states in which each event can be sent are defined in clause 12.4.

Table 55

Value	Event
'10'	EVT_C-APDU
'11'	EVT_ABORT
'21'	EVT_END_OF_APDU_TRANSACTION

12.2.2.1 EVT_C-APDU

This event shall be used by the client APDU host in order to send an APDU command to be processed by the server APDU host.

This event has the following parameter.

Table 56

Description	Length
Command APDU	N

The length of the Command APDU field in the event shall not exceed the length specified in the gate's registry.

12.2.2.2 EVT_ABORT

Upon reception of this event the server APDU host shall reset the state of the APDU gate to "INIT" and shall logically behave as if the signal input (RST) was reset on the ETSI TS 102 221 [11] interface. In particular, only the application context(s) of the pipe connected to the APDU server gate shall be affected, and there shall not be any impact on any application context(s) on other HCI pipe or other interfaces.

After processing of this event, the server APDU host shall send an EVT_ATR event to the client APDU host.

This event has no parameters.

12.2.2.3 EVT_END_OF_APDU_TRANSACTION

This event shall be sent by the client APDU host when no more activity is requested on the sever APDU host.

NOTE: A host receiving this event may then enter a power saving mode, informing the Host Controller by sending the event EVT_HCI_END_OF_OPERATION.

This event has no parameters.

12.2.3 Registry

The APDU gate provides the following registry. The registry shall be persistent.

Table 57

Identifie r	Parameter	Access Right	Description	Length	Default
'01'	MAX_C-APDU_SIZE	RO	Maximum acceptable total length of the Command APDU field in EVT_C-APDU event.	2	261
'02'	MAX_WAIT_TIME	RO	Maximum waiting time for the execution of a command APDU in ms.	2	1000

NOTE: It is strongly recommended to have a MAX_C-APDU_SIZE value of at least 261 (to support an APDU with full payload) if interoperability is required.

12.3 APDU application gate

12.3.0 Description

Clause 12.3 defines the commands and events that an APDU application gate of a client APDU host supports in addition to those of a generic gate as defined in clause 7.2.

The G_{ID} for the APDU application gate is dynamically assigned by the host.

12.3.1 Commands

There are no additional commands.

12.3.2 Events

12.3.2.0 Supported events

The APDU application gate support the events listed in table 58. The events are described in the following clause. The states in which each event can be sent are defined in clause 12.4.

Table 58

Value	Event
'10'	EVT_R-APDU
'11'	EVT_WTX
'12'	EVT_ATR

12.3.2.1 EVT_R-APDU

This event shall be sent by the server APDU host after processing of APDU command received in an EVT_C-APDU.

EVT_R-APDU shall be sent by the server APDU host not later than the maximum waiting time specified in the APDU gate registry after receiving EVT_C-APDU or after sending the last EVT_WTX.

This event has parameters as follows.

Table 59

Description	Length
Response APDU	N

12.3.2.2 EVT_WTX

This event shall be sent by the server APDU host during processing of APDU command if the processing of the command exceeds the maximum waiting time specified in the APDU gate registry. Upon reception of this event, the APDU application gate which received the request for additional time shall restart its timer with the value set to the maximum waiting time. The event can be sent repeatedly if more time is required for the processing.

This event has no parameters.

12.3.2.3 EVT_ATR

This event shall be sent by the server APDU host to indicate to the client APDU host the availability of the server for APDU processing. EVT_ATR shall be sent not later than 100 ms after the occurrence of the following events:

- opening of the pipe by the client APDU host; or
- reception of an EVT_ABORT; or
- the end of the session initialization as described in clause 8.4, in case the pipe is already open.
  EVT_ATR should not be sent before the end of the session initialization.
  The server APDU host should take care to send EVT_ATR before sending EVT_HCI_END_OF_OPERATION (if sent).

If the client APDU host has control of the availability of the underlying data link layer (e.g. activation and deactivation), it should take care to keep the underlying data link layer available to the server APDU host for at least 100 ms after the events indicated above or until EVT_ATR is received.

This event has the following parameter:

Table 60

Description	Length
ATR	N

The value of ATR is encoded as defined in ISO/IEC 7816-3 [10]. The Interface Characters may be present but have no meaning in the context of the present document.

NOTE 1: It is expected that the ATR Characters will not be used by any application except for the identification of the type of the server APDU host.

NOTE 2: Although the Interface Characters have no meaning in the context of the present document, it is recommended to set them to a value reflecting the actual capabilities of the server APDU host.

12.3.3 Registry

There are no registry entries defined.

12.4 State diagram for the APDU gate

Figure 19 specifies the states and the transitions that an APDU gate shall take during its operation.

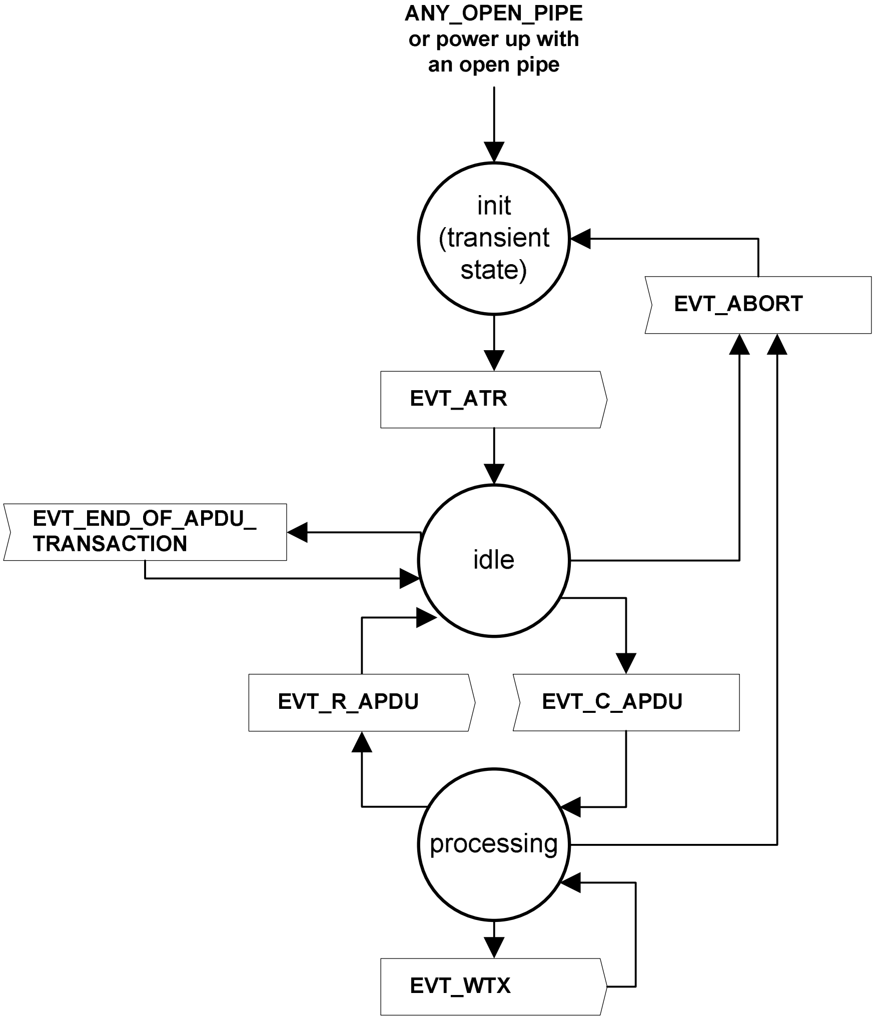


Figure 19

Other events or commands received on the APDU gate shall not change its state, with the exception of ANY_CLOSE_PIPE which can be received in any state.

Upon power up of the server APDU host, if the pipe was already open, the initial state shall be "INIT", meaning that an EVT_ATR shall be sent by the server APDU host as soon as it is ready to process APDUs.

Annex A (normative): State diagram for contactless reader mode

Figure A.1 shows the state changes and related events for the operation of reader mode in the CLF.

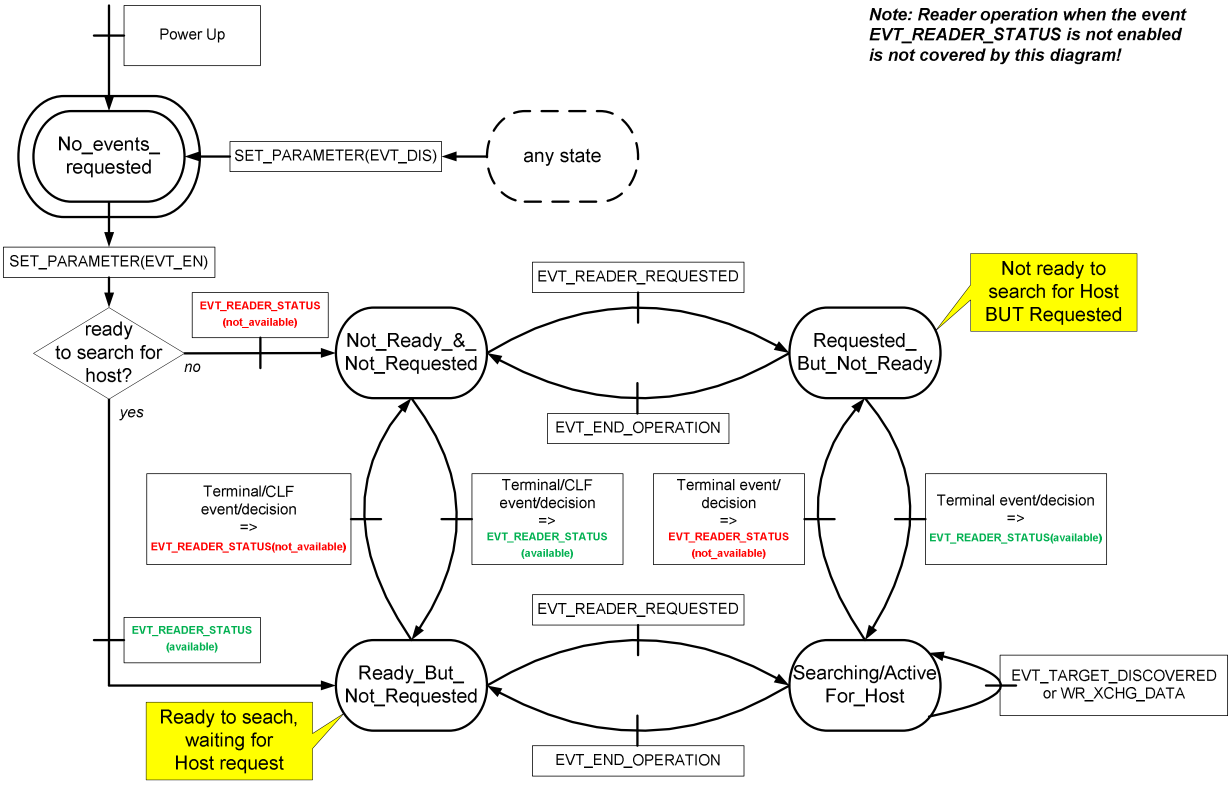

Power Up
Note: Reader operation when the event EVT_READER_STATUS is not enabled is not covered by this diagram!
No_events_ requested
SET_PARAMETER(EVT_DIS)
any state
SET_PARAMETER(EVT_EN)
ready to search for host?
no
yes
EVT_READER_STATUS (not_available)
Not_Ready_&_ Not_Requested
EVT_READER_REQUESTED
EVT_END_OPERATION
Requested_ But_Not_Ready
Not ready to search for Host BUT Requested
Terminal/CLF event/decision => EVT_READER_STATUS(not_available)
Terminal/CLF event/decision => EVT_READER_STATUS (available)
Terminal event/ decision => EVT_READER_STATUS (not_available)
Terminal event/decision => EVT_READER_STATUS(available)
EVT_READER_STATUS (available)
Ready_But_ Not_Requested
EVT_READER_REQUESTED
EVT_END_OPERATION
Searching/Active For_Host
EVT_TARGET_DISCOVERED or WR_XCHG_DATA
Ready to seach, waiting for Host request

Figure A.1

Annex B (informative): Change history

This annex lists all Changes Requests (CR) applied to the present document.

Change history								
Date	**Meeting**	**Plenary Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
2008-04	SCP #37	SCP-080214	001	-	F	Definition of the reaction to a reset of the underlying data link layer	7.0.0	7.1.0
		SCP-080214	002	-	F	Correction and clarification of card emulation registry parameters	7.0.0	7.1.0
		SCP-080214	003	-	F	Remove UID length 1 for reader mode	7.0.0	7.1.0
		SCP-080214	004	-	D	Correction of the Fig 1, 14 and 16 in order to add consistency in the naming of the hosts.	7.0.0	7.1.0
		SCP-080214	005	-	F	Clarification of the reader mode procedure	7.0.0	7.1.0
		SCP-080214	006	-	F	Clarification of consequences of failure in identity check in lower layers and update of identity reference	7.0.0	7.1.0
2008-07	SCP #38	SCP-080318	008	-	D	Editorial corrections	7.1.0	7.2.0
		SCP-080318	009	-	C	Clarification in Identity management gate	7.1.0	7.2.0
		SCP-080318	010	-	F	Correction in the reader mode gate	7.1.0	7.2.0
		SCP-080322	011	-	F	Clarification of connectivity	7.1.0	7.2.0
		SCP-080318	012	-	C	Remove EVT_END_OF_TRANSACTION	7.1.0	7.2.0
		SCP-080365	014	1	F	Clarification and correction of Type F card emulation	7.1.0	7.2.0
		SCP-080366	015	1	D	Correction of spelling mistakes and improved wording	7.1.0	7.2.0
		SCP-080318	017	-	F	Clarification of application level timeout in reader mode	7.1.0	7.2.0
		SCP-080367	018	1	F	Correction of EVT_FIELD_OFF	7.1.0	7.2.0
		SCP-080318	019	-	F	Modification of the Administration gate and the Link management gate	7.1.0	7.2.0
		SCP-080318	020	-	F	Clarification on UID & PUPI values	7.1.0	7.2.0
2008-10	SCP #39	SCP-080438	013	1	C	Remove "battery on" event	7.2.0	7.3.0
		SCP-080432	016	1	F	Clarify behaviour for unknown commands and events	7.2.0	7.3.0
		SCP-080432	021	-	C	Correction of table 2	7.2.0	7.3.0
		SCP-080432	022		D	Editorial change to table markings	7.2.0	7.3.0
		SCP-080432	023		D	Explanation of Figure 7	7.2.0	7.3.0
		SCP-080432	024	-	F	Parameter length for AID in EVT_TRANSACTION	7.2.0	7.3.0
		SCP-080432	025		F	Clarification for EVT_POST_DATA	7.2.0	7.3.0
		SCP-080432	026		F	Clarification of host for connectivity	7.2.0	7.3.0
		SCP-080432	027		F	Change in the WR_XCHGDATA for the RF Reader gate	7.2.0	7.3.0
		SCP-080432	028	-	C	Usage of the EVT_FIELD_ON event in the Card emulation procedure	7.2.0	7.3.0
		SCP-080432	029		F	Extending parameters for type A and B card mode	7.2.0	7.3.0
		SCP-080432	030	-	C	Correction of Table 16	7.2.0	7.3.0
		SCP-080468	031	-	F	Addition of hot unplug	7.2.0	7.3.0
		SCP-080483	033	-	C	Usage of the EVT_FIELD_ON event in the Card emulation procedure (addendum)	7.2.0	7.3.0
2009-01	SCP #40	SCP-090061	034	-	F	Correction of EVT_FIELD_ON	7.3.0	7.4.0
		SCP-090061	035	-	F	Clarification of HCI registry default values	7.3.0	7.4.0
2009-05	SCP #41	SCP-090161	037	-	F	Correction of issues raised by STF	7.4.0	7.5.0
		SCP-090159	036	-	F	Addition of a HCI end of operation event	7.4.0	7.5.0
2009-10	SCP #43	SCP-090331	041	-	F	Correction of CID support description	7.5.0	7.6.0
		SCP-090331	040	-	F	Correction on use of EVT_END_OPERATION	7.5.0	7.6.0
		------	---	-	-	Removal of a bullet introduced in table 18 due to a CR implementation error (EVT_HCI_END_OF_OPERATION/Loopback)	7.5.0	7.6.0
		------	---	-	-	Creation of release 8 of the specification in order to create a release 9 version.	7.6.0	8.0.0
		SCP-090331	039	-	D	Several editorial corrections.	8.0.0	9.0.0
		SCP-090356	038	1	D	Clarification on event and command reception	8.0.0	9.0.0

Change history

Date	Meeting	Plenary Doc	CR	Rev	Cat	Subject/Comment	Old	New
2010-03	SCP#44	SCP(10)0027	044	-	A	Correction of ATQB Parameter length	8.0.0	9.0.0
2010-07	SCP#45	SCP(10)0135	047	-	A	Correction of contactless reader operation	9.0.0	9.1.0
		SCP(10)0135	052	-	A	Definition of source gate assignment	9.0.0	9.1.0
		SCP(10)0135	048	-	C	Correction of procedure for contactless card emulation	9.1.0	10.0.0
		SCP(10)0172	053	1	C	Collection of clarifications	9.1.0	10.0.0
2010-10	SCP#46	SCP(10)0285r1	057	1	A	Clarification of EVT_HCI_END_OF_OPERATION	10.0.0	10.1.0
2011-01	SCP#47	SCP(11)0051r1	061	1	A	Clarification of handling of empty frame	10.1.0	10.2.0
2011-03	SCP#48	SCP(11)0158	065	-	A	Correction of a reference to the ISO/IEC 14443 specification	10.2.0	10.3.0
2011-03	SCP#48	SCP(11)0154	069	-	A	Processing of RF error indicator in card emulation mode	10.2.0	10.3.0
2011-05	SCP#49	SCP(11)0199	070	-	D	Clarification of the use of WR_XCHG_DATA	10.3.0	11.0.0
2011-09	SCP#52	SCP(11)0287r1	071	1	C	Clarification of error response after timeout for WR_XCHG_DATA command	11.0.0	11.1.0
2011-09	SCP#52	SCP(11)0288r1	072	-	C	Multiple pipes to one gate	11.0.0	11.1.0
2012-03	SCP#54	SCP(12)000014	073	-	B	Support of power indication for contactless interface	11.0.0	11.1.0
2012-06	SCP#55	SCP(12)000124	074	-	B	Addition of reader gate status indications and events	11.0.0	11.1.0
2012-12	SCP#57	SCP(12)000270	075	-	D	Correction of REC_ERROR for host link management gate	11.1.0	11.2.0
2012-12	SCP#57	SCP(12)000271	076	-	D	Editorial correction in the Card emulation architecture figure	11.1.0	11.2.0
2012-12	SCP#57	SCP(12)000272	077	-	F	Correction of reader gate status indications and events	11.1.0	11.2.0
2012-12	SCP#57	SCP(12)000273	078	-	F	Clarification of the RF errors handling in reader mode	11.1.0	11.2.0
2012-12	SCP#57	SCP(12)000274	079	-	C	Clarification of the impact of the Card RF gate MODE parameter	11.1.0	11.2.0
2013-07	SCP#60	SCP(13)000150	081	-	C	Clarification of Type F no response case	11.2.0	12.0.0
2013-10	SCP#61	SCP(13)000242	080	1	C	CR 102 622 R12 #080r1: Clarification for Type F command/response sequence	12.0.0	12.1.0
2014-02	SCP#62	SCP(14)000044r1	082	1	B	CR 102 622 R12 #082r1: Addition of User Interface availability information	12.0.0	12.1.0
2014-08	SCP#65	SCP(14)000220	083	1	B	CR 102 622 R12 #083r1: Support of APDU connectivity for HCI hosts	12.0.0	12.1.0
2014-08	SCP#65	SCP(14)000221r1	090	1	B	CR 102 622 R12 #090r1: Addition of Identification of multiple HCI hosts	12.0.0	12.1.0
2014-08	SCP#65	SCP(14)000223	092	-	A	CR 102 622 R12 #092:Corrections of Event Names in Annex A	12.0.0	12.1.0
2014-08	SCP#65	SCP(14)000219	089	-	A	CR 102 622 R12 #089: Clarification of the behaviour of contactless operation for non-ISO/IEC14443-4 applications	12.0.0	12.1.0
2014-12	SCP#66	SCP(14)000282	093	-	D	CR 102 622 R12 #093: Clarification of the consequences of EVT_END_OF_APDU_TRANSACTION	12.1.0	12.2.0
2014-12	SCP#66	SCP(14)000283	094	-	B	CR 102 622 R12 #094: Additional rule for a client APDU host	12.1.0	12.2.0
2014-12	SCP#66	SCP(14)000284	095	-	C	CR 102 622 R12 #095: Correction of conditions for sending EVT_HOT_PLUG	12.1.0	12.2.0
2015-02	SCP#67	SCP(15)000047	096	-	C	CR 102 622 R12 #096: Clarifications for EVT_TRANSACTION	12.1.0	12.2.0
2015-02	SCP#67	SCP(15)000048	097	-	F	CR 102 622 R12 #097: Correction of implementation of CR083r1	12.1.0	12.2.0
2016-04	SCP#73	SCP(16)000093	101	1	A	CR 102 622 R12 #101r1: Clarification of usage of MODE parameter on contactless card application sequences	12.1.0	12.2.0
2016-08	SCP#74	SCP(16)000136r1	105	1	F	Clarifications related to APDU Gate	12.1.0	12.2.0
2016-10	SCP#75	SCP(16)000191	107	-	F	Alignment between APDU Gate and APDU Card Application gate	12.1.0	12.2.0
2016-10	SCP#75	SCP(16)000192r1	108	1	F	Change of Host ID attribution	12.1.0	12.2.0
2016-10	SCP#75	SCP(16)000206r1	109	1	F	Clarifications related to MAX_C- APDU_SIZE in APDU Gate	12.1.0	12.2.0
2016-12	SCP#76	SCP(16)000236r1	113	1	F	Clarifications related to MAX_WAIT_TIME	12.1.0	12.2.0
2016-12	SCP#76	SCP(16)000238r1	115	1	F	Review the method to detect a UICC supporting REL12 or later	12.1.0	12.2.0
2017-02	SCP#77	SCP(17)000020	118	-	F	Correction on terminal host	12.1.0	12.2.0
2017-06	SCP#80	SCP(17)000127	121	-	F	EVT_HOT_PLUG Sending conditions for backward compatibility	12.1.0	12.2.0

History

Document history		
V12.0.0	October 2013	Publication
V12.1.0	October 2014	Publication
V12.2.0	December 2018	Publication