GlobalPlatform Technology

SE Abstract Communication Layer Specification

Version 1.1

Public Release

July 2024

Document Reference: GPC_TEN_214

Please help us maintain the value of your membership and encourage recruitment by observing this restriction.**

Contents

Figures

Tables

1 Introduction

The aim of this document is to specify a protocol allowing to remotely access to a SE reader. This protocol is defined in the scope of the GlobalPlatform Secure Element Compliance Program and targets different SE form factors. Although hardware interfaces may differ from one SE to another, it tries to minimize the impacts on the protocol.

1.1 Audience

This document is intended primarily for GlobalPlatform Compliance participants which include SE manufacturers, Test Tool providers and Laboratories.

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://www.globalplatform.org/specificationsipdisclaimers.asp. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

Table 1: Normative References

Standard / Specification	Description	Ref
ISO/IEC 7816-2	Identification cards - Integrated circuit cards - Part 2: Cards with contacts - Dimensions and location of the contacts	[7816-2]
ISO/IEC 7816-4	Identification cards – Integrated circuit cards - Part 4: Organization, security and commands for interchange	[7816-4]
ISO/IEC 14443-3	Identification cards – Contactless integrated circuit cards – Proximity cards - Part 3: Initialization and anticollision	[14443-3]
ISO/IEC 14443-4	Identification cards – Contactless integrated circuit cards – Proximity cards - Part 4: Transmission protocol	[14443-4]
ISO/IEC 18092	Information technology – Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol (NFCIP-1)	[18092]
IETF RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	[RFC 2119]
IETF RFC 8259	The JavaScript Object Notation (JSON) Data Interchange Format	[JSON]
IETF RFC 793	Transmission Control Protocol	[TCP]
ETSI TS 102 622	Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)	[TS 102 622]

Standard / Specification	Description	Ref
GlobalPlatform HCI Extension for SE Certification	GlobalPlatform Mobile Task Force - HCI Extension for the Embedded Secure Element Certification	[GP HCI EXT]

1.4 Terminology and Definitions

The following meanings apply to SHALL, SHALL NOT, SHOULD, SHOULD NOT, and MAY, MAY NOT in this document (refer to [RFC 2119]):

- SHALL indicates an absolute requirement.

- SHALL NOT indicates an absolute prohibition.

- SHOULD and SHOULD NOT indicate recommendations.

- MAY and MAY NOT indicate an option.

Specific terms used in this document are included in Table 2.

Table 2: Terminology and Definitions

Term	Definition
SE Agent	Software component managed by SE vendor which implements the ACL protocol and which allows to access to SE. The SE Agent plays the role of interface between the SE and the Test Tool Agent.
Test Tool Agent	Software component managed by Test Tool vendor which implements the ACL protocol. The Test Tool Agent plays the role of interface between the Test Tool and the SE Agent.

1.5 Abbreviations and Notations

Table 3: Abbreviations and Notations

Abbreviation / Notation	Meaning
ACL	Abstract Communication Layer
AID	Application IDentifier
APDU	Application Protocol Data Unit
CLF	Contactless Front-end
HCI	Host Control Infrastructures
JSON	JavaScript Object Notation
NCI	NFC Controller Interface
NFC	Near Field Communication
RF	Radio Frequency
RFU	Reserved for Future Use
SE	Secure Element
SUT	System Under Test

Abbreviation / Notation	Meaning
SWP	Single Wire Protocol
TCP	Transmission Control Protocol
UICC	Universal Integrated Circuit Card
0 - 9	Decimal digits are not enclosed in quotation marks.
'0' - '9' and 'A' - 'F'	Hexadecimal values are enclosed in straight single quotation marks.

1.6   Revision History

GlobalPlatform technical documents numbered n.0 are major releases. Those numbered n.1, n.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered $n.n$.1, $n.n$.2, etc., are maintenance releases that incorporate errata and precisions; all non-trivial revisions are indicated, often with revision marks.

Table 4: Revision History

Date	Version	Description
April 2022	1.0.0	First version
July 2024	1.1.0	Clarification of the Type F messages Adding information related to the PC/SC reader to be used when using the GitHub reference implementation

2   Architecture

2.1   General Overview

The following figure describes all components and how they connect to each other.
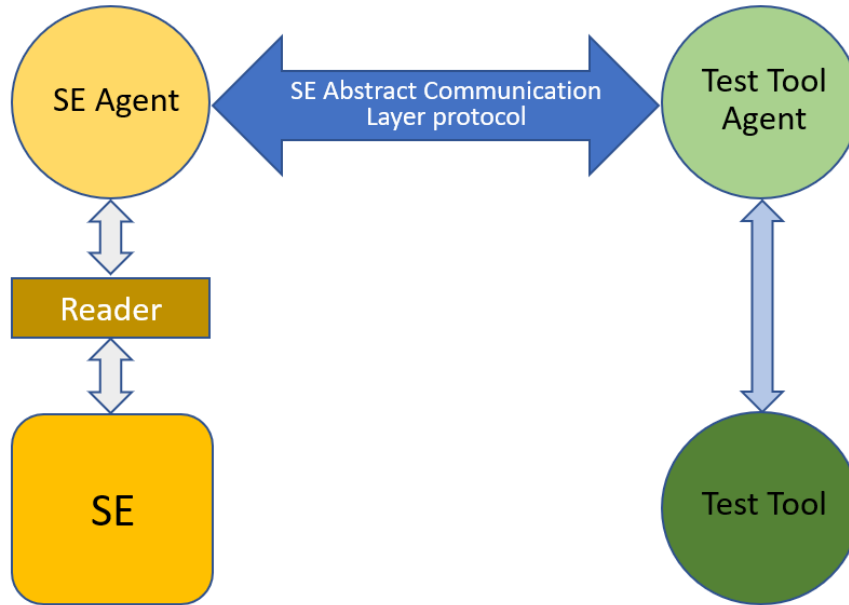
Figure 1: General Overview

2.2   Roles Definition

- SE: Secure Element acting as the SUT.
- SE Agent: Component which provides access to the SE and which implements the ACL. It plays the role of bridge between the Test Tool Agent and the SE.
- Test Tool: Software allowing to execute a specific SE Test Plan.
- Test Tool Agent: Component which plays the role of bridge between the Test Tool and the SE Agent over the ACL.

Note: The Test Tool Agent MAY be merged with the Test Tool through a specific library.

2.3   SE Interfaces

2.3.1   Multiple Interfaces

A SE MAY support multiple interfaces. As this specification describes the network protocol to be used for managing one interface, the SE Agent and Test Tool Agent SHALL open as many connections as the number of supported interfaces.
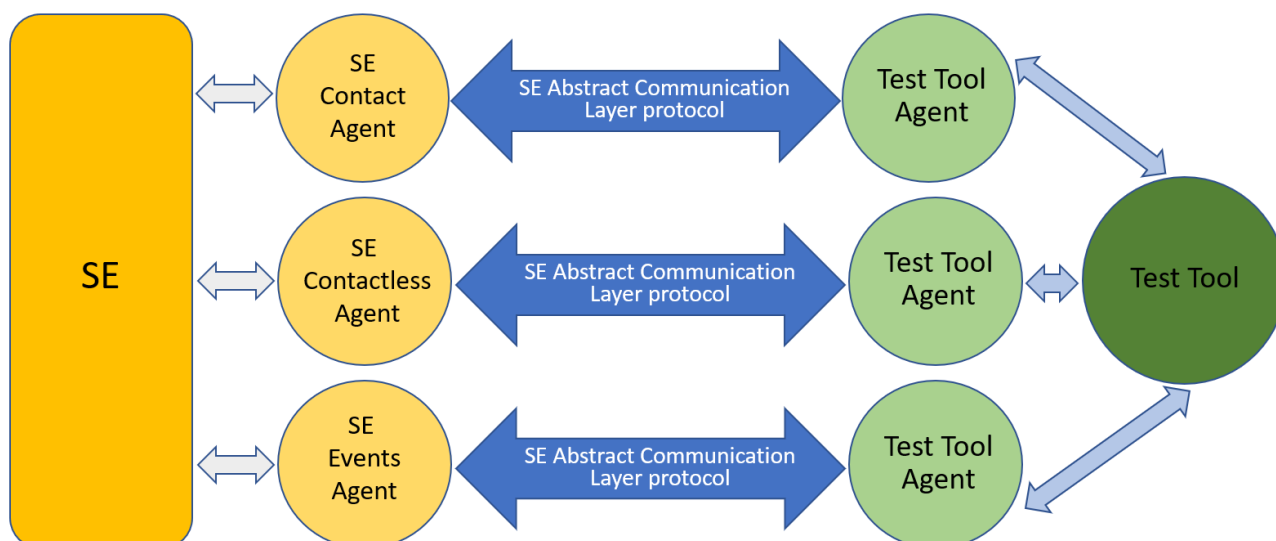
Figure 2: Example of Multiple Interfaces Architecture

2.3.2   SE Form Factor and Interfaces

This specification describes three interfaces which MAY be used by the SE depending on its form factor and capacities:

- CONTACT
- CONTACTLESS
- EVENTS

Table 5: Interfaces Descriptions

Known SE	CONTACT Interface	CONTACTLESS Interface	EVENTS Interface
Contact Card (ID-1)	ISO 7816 Part 2 contact electric pins	NONE	NONE
Contactless Card (ID-1)	NONE	Radio Frequency (ISO 14443/18092)	NONE
Dual Card (ID-1)	ISO 7816 Part 2 contact electric pins	Radio Frequency (ISO 14443/18092)	NONE
UICC (SWP)	ISO 7816 Part 2 contact electric pins	SWP Pin	SWP Pin, HCI EVT_TRANSACTION
Embedded SE	HCI APDU Gate, Contact Context	HCI RF Gates, Contactless Context	HCI event EVT_TRANSACTION
Other SE	Contact channel	Contactless channel	Events

3   Network Protocol

3.1   Transport Layer

TCP/IP SHALL be used for the stream exchanges.

The Test Tool Agent SHALL act as a TCP/IP server and SHALL listen to any connection incoming in a given TCP port number.

The SE Agent SHALL act as a TCP/IP client and SHALL initiate a TCP connection to the Test Tool Agent.

Once the TCP connection is established between the SE Agent and the Test Tool Agent, the SE Agent SHALL send an 'Handshake' application message to the Test Tool Agent indicating which interface relates to the TCP connection. It is up to the SE Agent to choose the most relevant key words allowing the Test Tool Agent to deduce if the incoming TCP connection is linked to either the 'CONTACTLESS', 'CONTACT' or 'EVENTS' interface.

The 'Handshake' application message SHALL be composed of a 4 bytes big-endian length and a string payload. This payload SHOULD contain the name of the reader connected to the related SE Agent and eventually some other key words.

Table 6: Handshake Message Payload Examples

Length	Payload (UTF-8 string message)
'00000024'	client_contact – Contact Reader Name
'0000002D'	client_contactless - Contactless Reader Name
'0000001B'	client_events – Reader Name

4 Messages Payload

Messages are composed of a 4 bytes big-endian length and a JSON syntax presentation for payload.

Here are examples of message.

Table 7: Message Payload Examples

Length	Payload
'00000028'	{"data":"","request":10,"timeout":30000}
'00000103'	{"client_description":"OK","err_card_code":0,"err_card_description":"OK","err_client_code":0,"err_server_code":0,"err_server_description":"OK","err_terminal_code":0,"response":"E3164F0AA00000015100000001029F70020F01C5039EFE809000","terminal_description":"OK"}

JSON content is encoded in UTF-8.

4.1 Applicative Layer

4.1.1 Command-Response Protocol

This protocol is restricted to a command-response protocol on top of the TCP stream.

The Test Tool Agent (TCP server) SHALL only send commands, and the TCP client SHALL only return responses. There SHALL be one response for each command.

The SE Agent (TCP client) SHALL NOT send any response if no related command has been received.

The Test Tool Agent (TCP server) SHALL NOT send any further commands if response of previous command has not been received yet.

Any protocol violation of the command-response sequence SHALL result in a TCP disconnection.

The Test Tool Agent SHOULD define a proprietary timeout value for the TCP layer which is supposed to be greater than the 'timeout' set in the Command Payload (Table 8). This TCP timeout MAY be a configuration parameter of the Test Tool Agent.

4.1.2 Command Payload

The command is a JSON object containing the following properties.

Table 8: JSON Command Payload

JSON property	Description	Presence
data	The command data as a hexadecimal string.	Mandatory
request	An integer identifying the request type (see Table 10 – Command ID).	Mandatory
timeout	An integer indicating the maximum delay for getting the response from the SE (in milliseconds)	Mandatory

The "data" parameter SHALL always be present in the Command payload, but MAY be empty depending on the Command type (see sections 4.1.4.x).

Here is an example of a request for a cold reset, with a timeout of 5 seconds:

```
{"data":"","request":10,"timeout":5000}
```

This is an example of a request to send a SELECT MF command APDU, with a timeout of 5 seconds:

```
{"data":"00A40004023F00","request":6,"timeout":5000}
```

4.1.3   Response Payload

The responses contain a JSON object which SHALL contain the following properties:

Table 9: JSON Response Payload

JSON property	Description	Presence
response	The response data as a hexadecimal string.	Mandatory
err_server_code	An integer identifying error or success on the server layer (see Table 18).	Mandatory
err_server_description	A string describing the error on the server layer, or "OK" in case of success.	Mandatory
err_client_code	An integer identifying error or success on the client layer (see Table 18).	Mandatory
client_description	A string describing the error on the client layer, or "OK" in case of success.	Mandatory
err_terminal_code	An integer identifying error or success on the terminal layer (see Table 18).	Mandatory
terminal_description	A string describing the error on the terminal layer, or "OK" in case of success.	Mandatory
err_card_code	An integer identifying error or success on the card layer (see Table 18).	Mandatory
err_card_description	A string describing the error on the card layer, or "OK" in case of success.	Mandatory

The "response" parameter SHALL always be present in the Response payload, but MAY be empty depending on the related Command type (see Table 10).

Example of response payload related to a cold reset request:

```
{
  "client_description": "OK",
  "err_card_code": 0,
  "err_card_description": "OK",
  "err_client_code": 0,
  "err_server_code": 0,
  "err_server_description": "OK",
  "err_terminal_code": 0,
  "response": "3B9F96803FC7828031E073F62157574A330581606100FA",
  "terminal_description": "OK"
}
```

4.1.4   Commands and Responses Description

Table 10: Commands and Responses

Command ID	Command Name	Related "response" field content
0	REQ_CONNECT	N/A (empty)
1	REQ_DIAG	A string containing diagnostic information.
2	REQ_DISCONNECT	N/A (empty)
3	REQ_ECHO	The data from the related command.
4	REQ_INIT	N/A (empty)
5	REQ_RESTART	N/A (empty)
6	REQ_COMMAND	The response APDU as a hexadecimal string.
7	REQ_COMMAND_A	The response APDU as a hexadecimal string.
8	REQ_COMMAND_B	The response APDU as a hexadecimal string.
9	REQ_COMMAND_F	The response APDU as a hexadecimal string.
10	REQ_COLD_RESET	The ATR as a hexadecimal string.
11	REQ_WARM_RESET	The ATR as a hexadecimal string.
12	REQ_POWER_OFF_FIELD	N/A (empty)
13	REQ_POWER_ON_FIELD	N/A (empty)
14	REQ_POLL_A	N/A (empty)
15	REQ_POLL_B	N/A (empty)
16	REQ_POLL_F	N/A (empty)
17	REQ_POLL_ALL_TYPES	N/A (empty)
18	REQ_DEACTIVATE_INTERFACE	N/A (empty)
19	REQ_ACTIVATE_INTERFACE	N/A (empty)
20	REQ_GET_NOTIFICATIONS	The notifications buffer as a hexadecimal string.
21	REQ_CLEAR_NOTIFICATIONS	N/A (empty)

Each command SHALL be sent to the relevant interface(s) as described in the Table below:

Table 11: Commands and Applicable Interfaces

Command Name	Related Applicable Interfaces
REQ_CONNECT	N/A
REQ_DIAG	CONTACT, CONTACTLESS, EVENTS
REQ_DISCONNECT	CONTACT, CONTACTLESS, EVENTS
REQ_ECHO	CONTACT, CONTACTLESS, EVENTS
REQ_INIT	N/A

REQ_RESTART	N/A
REQ_COMMAND	CONTACT, CONTACTLESS
REQ_COMMAND_A	CONTACTLESS
REQ_COMMAND_B	CONTACTLESS
REQ_COMMAND_F	CONTACTLESS
REQ_COLD_RESET	CONTACT, CONTACTLESS
REQ_WARM_RESET	CONTACT, CONTACTLESS
REQ_POWER_OFF_FIELD	CONTACTLESS
REQ_POWER_ON_FIELD	CONTACTLESS
REQ_POLL_A	CONTACTLESS
REQ_POLL_B	CONTACTLESS
REQ_POLL_F	CONTACTLESS
REQ_POLL_ALL_TYPES	CONTACTLESS
REQ_DEACTIVATE_INTERFACE	CONTACT, CONTACTLESS, EVENTS
REQ_ACTIVATE_INTERFACE	CONTACT, CONTACTLESS, EVENTS
REQ_GET_NOTIFICATIONS	EVENTS
REQ_CLEAR_NOTIFICATIONS	EVENTS

If the Test Tool Agent sends a command unapplicable for the targeted interface, the SE Agent SHALL send back an error.

4.1.4.1   REQ_CONNECT

This command is RFU.

4.1.4.2   REQ_DIAG

This command requests diagnostic information from the SE. The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.3   REQ_DISCONNECT

The SE Agent SHALL disconnect. The input parameter "`data`" SHALL be empty in the Command Payload. The output parameter "`response`" SHALL be empty in the Response payload.

4.1.4.4   REQ_ECHO

This command requests the SE Agent to send back the previously received command data. The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.5   REQ_INIT

This command is RFU.

4.1.4.6   REQ_RESTART

This command is RFU.

4.1.4.7   REQ_COMMAND

This command is used to transmit the string of bytes, present in the "`data`" field, to the SE. The format MAY be ISO 7816 APDU or another format (e.g. 1 byte for the command). The command format SHALL NOT be checked by the SE Agent and Test Tool Agent.

The command MAY be used to send stream over CONTACT or CONTACTLESS interface.

In case of CONTACTLESS interface, the command will be sent according ISO-14443-4 or 18092 (over type A or B according the current type of selection of the card done by the reader).

This command SHALL NOT be used for sending messages over the type F (see the REQ_COMMAND_F message for more details).

The output parameter "`response`" SHOULD contain the related R-APDU.

4.1.4.8   REQ_COMMAND_A

This command is used to transmit string of bytes over Contactless Type A interface.

This command SHALL be valid over CONTACTLESS interface only.

4.1.4.9   REQ_COMMAND_B

This command is used to transmit string of bytes over Contactless Type B interface.

This command SHALL be valid over CONTACTLESS interface only.

4.1.4.10   REQ_COMMAND_F

This command is used to transmit string of bytes over Contactless Type F interface.

This command SHALL be valid over CONTACTLESS interface only.

The format of the "`data`" field SHALL be as follow:

Table 12: Type F messages format

Information field	
LEN	**Data field**
1 byte	0 byte to 254 bytes
Length byte (LEN): 1 byte *Data field: (LEN – 1) bytes*	

Header field (preamble + sync code) and End field (CRC) SHALL NOT be present. It will be up to the SE Agent to add those two fields if required before sending the command to the SE.

4.1.4.11   Response contains the CRC.REQ_COLD_RESET

This command is used to ask the system to reset. Actual actions on SE depend on the interface context but it requires the addressed system to fully stop and restart. The input parameter "`data`" SHALL be empty in the Command payload.

Table 13: Cold Reset Actions

SE Interface	Context	SE Actions
ISO 7816-2	Contact	ISO-7816 COLD RESET (VCC OFF, VCC ON)
Radio Frequency (ISO 14443 or 18092)	Contactless	FIELD OFF FIELD ON
HCI APDU_Gate / RF Gate	Contact	SE POWER SUPPLY OFF SE POWER SUPPLY ON
	Contactless	HCI EVT_FIELD_OFF HCI EVT_FIELD_ON
UICC	Contact	DEACTIVATE SWIO ISO-7816 COLD RESET (VCC OFF, VCC ON) ACTIVATE SWIO
	Contactless	DEACTIVATE SWIO ACTIVATE SWIO
Other Board (e.g. using NCI)	Contact	VCC OFF VCC ON
	Contactless	FIELD OFF FIELD ON (Otherwise, similar behavior according to the connection characteristic)

4.1.4.12 REQ_WARM_RESET

This command is used to ask the system to reset by programmatic way. Actual actions depend on the interface context, but it does not require the addressed system to stop. The input parameter "`data`" SHALL be empty in the Command payload.

Table 14: Warm Reset Actions

SE Interface	Context	SE Actions
ISO 7816-2	Contact	ISO-7816 WARM RESET
Radio Frequency (ISO 14443 or 18092)	Contactless	FIELD OFF FIELD ON
HCI APDU Gate / RF Gate	Contact	EVT_SOFT_RESET
	Contactless	EVT_FIELD_OFF EVT_FIELD_ON
UICC	Contact	ISO-7816 WARM RESET
	Contactless	DEACTIVATE SWIO ACTIVATE SWIO
Other Board (e.g. using NCI)	Contact	SIMULATE WARM_RESET (otherwise VCC OFF – VCC ON)

Note: for Contactless, there is no other way to proceed.

4.1.4.13   REQ_POWER_OFF_FIELD

This command shuts down the contactless field or CLF. The input parameter "`data`" SHALL be empty in the Command payload.

Table 15: Power Off Field Actions

SE Interface	Context	SE Actions
ISO 7816-2	Contact	ERROR
Radio Frequency (ISO 14443 or 18092)	Contactless	FIELD OFF
HCI APDU Gate / RF Gate	Contact	ERROR
	Contactless	EVT_FIELD_OFF
UICC	Contact	ERROR
	Contactless	DEACTIVATE SWIO

4.1.4.14   REQ_POWER_ON_FIELD

This command enables the contactless field or CLF. The input parameter "`data`" SHALL be empty in the Command payload.

Table 16: Power On Field Actions

SE Interface	Context	SE Actions
ISO 7816-2	Contact	ERROR
Radio Frequency (ISO 14443 or 18092)	Contactless	FIELD ON
HCI APDU Gate / RF Gate	Contact	ERROR
	Contactless	EVT_FIELD_ON
UICC	Contact	ERROR
	Contactless	ACTIVATE SWIO

4.1.4.15   REQ_POLL_A

This command enables exclusive Contactless Type A polling.

Contactless type SHALL NOT be impacted by any reset (REQ_WARM_RESET nor REQ_COLD_RESET) command.

The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.16   REQ_POLL_B

This command enables exclusive Contactless Type B polling. The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.17   REQ_POLL_F

This command enables exclusive Contactless Type F polling. The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.18   REQ_POLL_ALL_TYPES

This command enables all Contactless Types supported by the contactless reader (e.g. A, B, F). The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.19   REQ_DEACTIVATE_INTERFACE

This command informs the SE Agent that the addressed interface will no more process commands other than a REQ_ACTIVATE_INTERFACE.

The SE Agent MAY discard this information depending on its requirement. It is designed for exclusive access interfaces.

The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.20   REQ_ACTIVATE_INTERFACE

This command informs the SE Agent that the addressed interface is to be activated to process further commands.

The SE Agent MAY discard this information depending on its requirement. It is designed for exclusive access interfaces.

The input parameter "`data`" SHALL be empty in the command payload.

4.1.4.21   REQ_GET_NOTIFICATIONS

Whenever a notification is sent by the SE, it is stored by the SE Agent in a notification buffer until the use of this command.

This command is a request to get the notification buffer for a given applet AID. It returns the notification buffer which contains all the collected notifications associated to a specified AID and then clears the notification buffer.

Format of the "`response`" SHALL be a sequence of:

- Length (2 bytes) of notification payload
- Notification payload

If there is no notification available, the "`response`" SHALL contain an empty value.

The input parameter "`data`" SHALL be empty in the Command payload.

4.1.4.21.1   Response Payload: Example of two Notifications ('010203' and '01020304')

`"response":"0003010203000401020304"`

Table 17: Notifications Buffer

1st Notification		2nd Notification	
Length	**Notification Payload**	**Length**	**Notification Payload**
'0003'	'010203'	'0004'	'01020304'

4.1.4.21.2   HCI Notification Case

Notification type is context dependent. If the SE supports HCI and the SE Agent maps the HCI events to the REQ_GET_NOTIFICATIONS response, this section applies.

The SE Agent MAY return only relevant events which are context dependent (e.g. only EVT_TRANSACTION).

To be able to know which notification event has been stored by the SE Agent, the "Notification Payload" described in section below SHALL be preceded with the HCI Event TYPE and HCI Event INSTRUCTION as defined in [TS 102 622] section 5.2.

So the notification buffer, present in the "`response`" field, is a sequence of:

- Length (2 bytes) of notification payload

- 1-byte of HCI TYPE / HCI INSTRUCTION

- HCI payload (e.g. including AID and Payload as in [TS 102 622] EVT_TRANSACTION, Table 52)

Table 18: HCI Notifications Buffer

1st Notification		2nd Notification	
Length	**Notification Payload**	**Length**	**Notification Payload**
'000D'	'52 81 05 A000000001 82 03 010203'	'000C'	'52 81 05 A000000002 82 02 0102'

4.1.4.21.3   Other Cases

This is RFU.

4.1.4.22   REQ_CLEAR_NOTIFICATIONS

This command requests to clear all notifications collected for a given applet AID. It SHALL clear the notifications buffer associated to the specified applet AID.

The input parameter "`data`" SHALL be empty in the Command payload.

4.1.5   Error Codes

Table 19: Error Codes

Value	Name	Description
0	SUCCESS	The command was successfully executed.
-1	ERR_TIMEOUT	The command has not been executed within the given time.
-2	ERR_NETWORK	A network error occurred.
-3	ERR_CLIENT_CLOSED	The client disconnected.
-4	ERR_INVALID_STATE	The command cannot currently be executed.
-5	ERR_INVALID_REQUEST	The command was not understood by the SE Agent.
-6	ERR_JSON_PARSING	The command (or response) could not be parsed.
-7	ERR_INVALID_TERMINAL	The terminal was not available.

5   Additional requirements

5.1   Initialization Sequence

A specific commands sequence SHALL be performed prior to any exchange in case the SE provides both CONTACT and CONTACTLESS interfaces. The following sequence SHALL be run by the SE Agent and the Test Tool Agent whenever a new TCP connection to an interface is needed:

1.   SE Agent connects with TCP to the Test Tool Agent through the CONTACTLESS interface.

     1.1 Test Tool Agent sends REQ_ACTIVATE_INTERFACE.

     1.2 Test Tool Agent sends REQ_DEACTIVATE_INTERFACE.

2.   SE Agent connects with TCP to the Test Tool Agent through the CONTACT interface.

     2.1 Test Tool Agent sends REQ_ACTIVATE_INTERFACE.

     2.2 Test Tool Agent sends REQ_DEACTIVATE_INTERFACE.

3.   Optionally, SE Agent connects with TCP to the Test Tool Agent through the EVENTS interface.

     3.1 Test Tool Agent sends REQ_ACTIVATE_INTERFACE.

     3.2 Test Tool Agent sends REQ_DEACTIVATE_INTERFACE.

This sequence is summarized in the table below:

Table 20: Initialization Sequence

Interface	Actions
CONTACTLESS	TCP Connect REQ_ACTIVATE_INTERFACE REQ_DEACTIVATE_INTERFACE
CONTACT	TCP Connect REQ_ACTIVATE_INTERFACE REQ_DEACTIVATE_INTERFACE
EVENTS [optional]	TCP Connect REQ_ACTIVATE_INTERFACE REQ_DEACTIVATE_INTERFACE

Then, the Test Tool Agent MAY run arbitrary commands on any interface.

The purpose of this sequence is to allow the SE Agent implementation to enable automatic access on both interfaces with a dual reader. So, when the first REQ_ACTIVATE_INTERFACE is received on CONTACTLESS interface, the SE Agent SHALL configure the reader so that it will be able to switch interface without having to manually eject the card.

Prior to any initialization sequence, it is assumed that the Test Tool Agent knows the capacities of the targeted SE and all its supported interfaces so that it can associate each new TCP connection with its related functional interface.

5.2 Interface Activation

5.2.1 State

An interface MAY be:

- ACTIVATED after using REQ_ACTIVATE_INTERFACE command on the related network interface.

- DEACTIVATED after using REQ_DEACTIVATE_INTERFACE command on the related network interface.

Default state value of an interface SHALL be considered by the Test Tool Agent as DEACTIVATED.

5.2.2 Activate

Whenever an interface needs to be used, the Test Tool Agent SHALL activate it by sending a REQ_ACTIVATE_INTERFACE prior to sending any other command.

5.2.3 Deactivate

On the opposite, whenever an interface is no more used, the Test Tool Agent SHALL deactivate it by sending a REQ_DEACTIVATE_INTERFACE.

5.2.4 Concurrency

This ACL has been designed for having access to one interface at a time.

Nevertheless, it SHOULD be possible to keep the EVENTS interface ACTIVATED at the same time as the CONTACT or the CONTACTLESS interface.

5.3 Contactless Polling

5.3.1 State

Polling type is a state attribute of a CONTACTLESS interface. It is set using the related commands.

The SE Agent SHALL have a default interface contactless polling state with all contactless types supported by the reader.

The SE Agent SHALL provide a persistent state that survives to the reset commands (REQ_COLD_RESET / REQ_WARM_RESET).

The SE Agent SHALL reset the polling interface to its default value whenever an REQ_ACTIVATE_INTERFACE command is received.

5.3.2 Commands

It is possible to select the polling type of the contactless reader by sending a command through the CONTACTLESS interface

- REQ_POLL_A: enables Type A polling only and executes actions described at REQ_COLD_RESET

- REQ_POLL_B: enables Type B polling only and executes actions described at REQ_COLD_RESET

- REQ_POLL_F: enables Type F polling only and executes actions described at REQ_COLD_RESET

- REQ_POLL_ALL_TYPES: enables all Types (e.g. A, B, F) and reactivates all available protocols supported by the reader

5.4   Predefined Sequences

Every action in the sub-sections below SHALL be executed line by line in the order that they appear. Interface order SHALL also be taken into account.

5.4.1   Sequence for Starting a Contact Exchange

Before executing the sequence below, it is assumed that the currently selected interface on SE (if any) has already been deactivated.

Table 21: Sequence for Starting a Contact Exchange

Interface	Actions
CONTACT	REQ_ACTIVATE_INTERFACE REQ_COLD_RESET

5.4.2   Sequence for Starting a Contactless Exchange

Before executing the sequence below, it is assumed that the currently selected interface on SE (if any) has already been deactivated.

Table 22: Sequence for Starting a Contactless Exchange

Interface	Actions
CONTACTLESS	REQ_ACTIVATE_INTERFACE [Optional] REQ_POLL_<X>

5.4.3   Sequence for Switching from Contact to Contactless

Table 23: Sequence for Switching from Contact to Contactless Interface

Interface	Actions
CONTACT	REQ_COLD_RESET REQ_DEACTIVATE_INTERFACE
CONTACTLESS	REQ_ACTIVATE_INTERFACE [Optional] REQ_POLL_<X>

5.4.4   Sequence for Switching from Contactless to Contact

Table 24: Sequence for Switching from Contactless to Contact Interface

Interface	Actions
CONTACTLESS	REQ_COLD_RESET REQ_DEACTIVATE_INTERFACE
CONTACT	REQ_ACTIVATE_INTERFACE REQ_COLD_RESET

5.4.5   Sequence for Switching from Contactless or Contact to Events

The EVENTS interface MAY be ACTIVATED right after the Initialization Sequence (section 5.1) and remain ACTIVATED independently from the activation of another interface.

Annex A   GitHub repository

A reference implementation of the SE Agent and Test Tool Agent is available on GitHub:

https://github.com/GlobalPlatform/SE_Abstract_Communication_Layer_Over_TCP_IP

Any adaptation of the SE Agent MIGHT be done by SE vendor when necessary.

It is recommended to use the following PC/SC reader when using this reference implementation:

- Identiv uTrust 4701 F with latest firmware and drivers:
    - See https://support.identiv.com/4701f/
    - Firmware: v1.06 (Sept 2016)
    - Windows 10 drivers: v1.9.0.7 (Feb 2023)
    - Windows 11 drivers: v1.11.00 (Feb 2023)