GLOBALPLATFORM™

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

GlobalPlatform Card

Contactless Services

Card Specification v2.2 – Amendment C

Version 1.1.0.2

Public Review

April 2014

Document Reference: GPC_SPE_025

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Figures

Tables

1 Introduction

This document defines an extension of the GlobalPlatform Card Specification [GPCS] to facilitate deployment of contactless services in a Secure Element located in a mobile handset.

More specifically, this specification defines mechanisms, parameters, and interfaces to set up and maintain the configuration of applications and control their access to system resources like communication interfaces and memory. GlobalPlatform configuration(s) may require some or all of the capabilities specified in this document. Applications that are configured to be usable on a contactless interface (using mechanisms defined in Chapter 5: Communication Interface Access Configuration) are referred to as “Contactless Applications”.

The term “contactless” is used throughout this document and references primarily “proximity” technologies, e.g. ISO/IEC 14443. However, it may be possible that the mechanisms defined in this specification are useful to manage access to Applications over a variety of technologies, such as Bluetooth, Wi-Fi, and ISO/IEC 15693. Therefore, the term “contactless” shall not be understood as a limitation to strictly “proximity” technologies.

Certain application providers may choose to limit the accessibility of their Application(s) to specific technologies, and therefore, may define profiles specifying these industry-specific limitations.

Although the requirements motivating this amendment originated from a mobile/contactless services perspective, some of the concepts specified in this amendment are easily applicable beyond this core use case.

This document focuses on parameters and mechanisms required for Applications in card emulation mode. Applications using reader or peer-to-peer mode are beyond the scope of the current version of this specification, although such applications may in some cases consistently benefit from the mechanisms defined in this document.

Use Cases and Requirements

The following requirements shall be considered:

- The end-user shall be able to:
  - View and prioritize the list of services (implemented by a Standalone Application or an Application Group) based on a user-friendly service management application located in or outside of the secure element
  - Activate and deactivate contactless services (so that a contactless reader is able/unable to select a specific service)
  - Enable and disable communication over the contactless interface
- The Application provider shall be able to:
  - Provide technical meta-data (e.g. parameters for the contactless protocol) to clearly define the contactless environment of the Application
  - Provide end-user selection meta-data (e.g. logo, family…) to help the end-user to choose the intended service for activation, deactivation, etc.
- Additionally it shall be possible to:
  - Grant a Security Domain a specific amount of memory resources. All Applications and sub-Security Domains (if present) which are managed by this Security Domain shall not be allowed to consume more memory
  - Delete a root Security Domain and its hierarchy in a single operation

- Prohibit the use of a specific token on a Security Domain with the Token Verification Privilege

This specification addresses the following topics:

- Extensions of the OPEN and the GlobalPlatform Registry
- Application Groups
- Contactless Activation states
- Management of Communication Interface Access Configuration
- Application Selection
- CRS Application and CREL Application
- Conflicts between applications due to incompatible contactless parameters
- Privileges relative to Contactless Activation
- Memory resource management within a Security Domain Hierarchy
- Deletion of a Security Domain Hierarchy
- API for access to the extended GlobalPlatform registry
- API for interactions with CRS Application and CREL Application
- API extension to the `SecureChannel` interface enabling an Application to use the secure channel protocol when the APDU object is not available
- Token Identifier Blacklist for Delegated Management

1.1 Audience

This amendment is intended primarily for card manufacturers and application developers developing GlobalPlatform card implementations.

It is assumed that the reader is familiar with smart cards and smart card production, and in particular familiar with the GlobalPlatform Card Specification [GPCS].

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 Normative References

Table 1-1: Normative References

Standard / Specification	Description	Ref
GlobalPlatform Card Specification	GlobalPlatform Card Specification v2.2.1	[GPCS]
GPCS Amendment A	GlobalPlatform Card, Confidential Card Content Management, Card Specification v2.2 – Amendment A, v1.0.1	[GPCS-A]
ETSI TS 102 223	Smart Cards; Card Application Toolkit (CAT)	[102223]
ETSI TS 102 613	Smart Cards; UICC - Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics	[102613]
ETSI TS 102 622	Smart Cards; UICC – Contactless Front-end (CLF) Interface; Host Controller Interface (HCI) (Release 7)	[102622]
ETSI TS 102 705	Smart Cards; UICC Application Programming Interface for Java Card™ for Contactless Applications	[102705]
EMVCo AAUI	EMVCo Contactless Mobile Payment – Application Activation User Interface – Overview, Usage Guidelines, and PPSE Requirements, v1.0, December 2010	[AAUI]
IETF RFC 3629	UTF-8, a transformation format of ISO 10646	[3629]
ISO/IEC 10918-1	Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines	[10918-1]
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]
ISO/IEC 7816-6	Identification cards – Integrated circuit cards – Part 6: Interindustry data elements for interchange	[7816-6]
ISO/IEC 8825-1	Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)	[8825-1]
ISO/IEC 8859-1	Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No.1	[8859-1]
JCRE v3 Classic Edition	Runtime Environment Specification – Java Card™ Platform, v3.0.1 Classic Edition	[JCRE CE]
JIS X 6319-4	Japanese Industrial Standard X 6319-4: Specification of implementation for integrated circuit(s) cards – Part 4: High Speed proximity cards	[JIS6319-4]

1.4 Terminology and Definitions

Table 1-2 defines the expressions used within this specification that use an upper case first letter in each word of the expression. Expressions within this document that use a lower case first letter in each word take the common sense meaning. (Tagged data elements are also given an upper case first letter in each word of their names.)

Table 1-2: Terminology and Definitions

Term	Definition
Contactless Application	Applications that are explicitly or implicitly configured to be usable on a contactless interface (e.g. ISO/IEC 14443-based) are referred to as “Contactless Applications”.
Contactless Session	A contactless session starts when the card enters the RF field and ends when the card leaves the RF field.
Secure Element	A Secure Element (SE) is a tamper resistant component which is used in a device to provide the security, confidentiality, and multiple application environment required to support various business models. Such a Secure Element may exist in any form factor such as UICC, embedded SE, smartSD, smart microSD, etc.

1.5 Abbreviations and Notations

Table 1-3: Abbreviations

Abbreviation	Meaning
AID	Application Identifier
AFI	Application Family Indicator
APDU	Application Protocol Data Unit
API	Application Programming Interface
APSD	Security Domain of the Application Provider
CA	Controlling Authority
CASD	Controlling Authority Security Domain
CIA	Communication Interface Access
CL	Contactless
CLF	Contactless Front-end
CREL	Contactless Registry Event Listener
CRS	Contactless Registry Services
EMV	EuroPay MasterCard Visa; used to refer to Specifications for Payment Systems developed by EMVCo
GUI	Graphical User Interface
HCI	Host Controller Interface
IF	Interface
ISD	Issuer Security Domain
OPEN	GlobalPlatform Environment
OTA	Over-The-Air
P2P	Peer to Peer
PCD	Proximity Coupling Device
PICC	Proximity Integrated Circuit Card
PPSE	Proximity Payment System Environment
RAM	Remote Application Management
RF	Radio Frequency
SWP	Single Wire Protocol
TLV	Tag, Length, Value
UICC	Universal Integrated Circuit Card
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

1.6 Revision History

Table 1-4: Revision History

Date	Version	List of Modifications
Feb 2010	1.0	Initial Release
Feb 2012	1.0.1	Corrections and clarifications including: • Specified behavior when accessibility of the Head Application through the contactless (proximity) communication interface is disabled. • Expanded description of behavior of Member Applications regarding activation state changes. • Specified that if the AID of a Member Application is removed from the Group Authorization List, its Head Application field shall remain unmodified. • Corrected the name of the method to be called by the OPEN for each activation requested by an Application without the Contactless Self-Activation Privilege. • Adjusted the names associated with tags '8B' and '8C' in the Contactless Registry Data. • Changed the status bytes to be returned if template tag 'A0' was present in SET STATE response data. • Clarified and expanded warning conditions for SET STATUS. • Expanded the rules for computation on activation of the Current Protocol Parameters. • Expanded the rules for full computation of the Current Protocol Parameters when the Default Protocol Parameters are updated. • Expanded the instructions for conflict detection. • Corrected the normative reference that defines coding of the ATQA. • Expanded values for Length Operation Indicator. • Clarified the purpose and use of the Communication Interface Access Configuration mechanism. • Clarified that the method supporting promotion or demotion of an Application within the GlobalPlatform Registry is provided by the CRS API rather than the OPEN. • Clarified how the OPEN determines the default Application during Implicit Selection. • Specified that CGM amount shall not exceed the memory space currently available on a card. • Specified conditions under which the tag 'A2' may occur more than once in the Contactless Protocol Parameters. • Removed the restriction that tag '81' is used only with INSTALL [for registry update]. • Clarified the structure of the Contactless Protocol Parameters Profile. • Redefined the GET STATUS Reference Control Parameter P1. • Added tag 'CF' to the Filter Criteria (tag '5C') and to the GlobalPlatform Registry Data. *— continues —*

Date	Version	List of Modifications
Feb 2012 *(continued)*	1.0.1	• Specified that a card content management command containing a blacklisted Token Identifier shall be rejected. • Changed the version number of the Executable Load File AID of `org.globalplatform.contactless` and the version number of `org.globalplatform` to be used. • Corrected and expanded the Contactless Protocol Management example.
April 2013	1.1	• Added support for Type F protocol (parameters, anti-collision, implicit selection, etc.). • Added precisions for the support of non-APDU based Contactless Applications. • Added new mechanism to check beforehand whether a Contactless Application accepts being activated, based on a proprietary Application Activation Policy. • Added more diagnostic data in the response to the GlobalPlatform CRS Application's SET STATUS command. • Requires version 1.2 of the `org.globalplatform.contactless` package (to be published together with this document). This new version introduces the following changes: o Added new shareable interface `CLAppletActivationPolicy`. o Added new constant `GPCL_CL_APPLICATION_ACTIVATION_POLICY` in class `GPCLSystem` to access the shareable service (optionally provided by a Contactless Application). o Changed access conditions for some methods in the `GPCLRegistryEntry` interface. o Added new methods in `GPCLSystem` class: ▪ `getHostDeviceUserInterfaceState` ▪ `getSecureElementType` ▪ `launchHostDeviceApplication` o Added some constants in `CLAppletEvent` interface (new events). o Renamed some constants in `GPCLSystem` class and `CLAppletEvent` interface (previous constants are deprecated). • Modified the concept of Volatile Priority, which now implies temporary activation over the contactless interface, as well as the conditions to assign and reset it. These changes allow better and more suitable usage of the Volatile Priority mechanism. • Modified the way the Recognition Algorithm shall be applied to incoming commands to select an Application. Upon reset of the contactless interface, the Recognition Algorithm is now applied to every incoming command until an Application is eventually selected. • Added some precisions here and there.

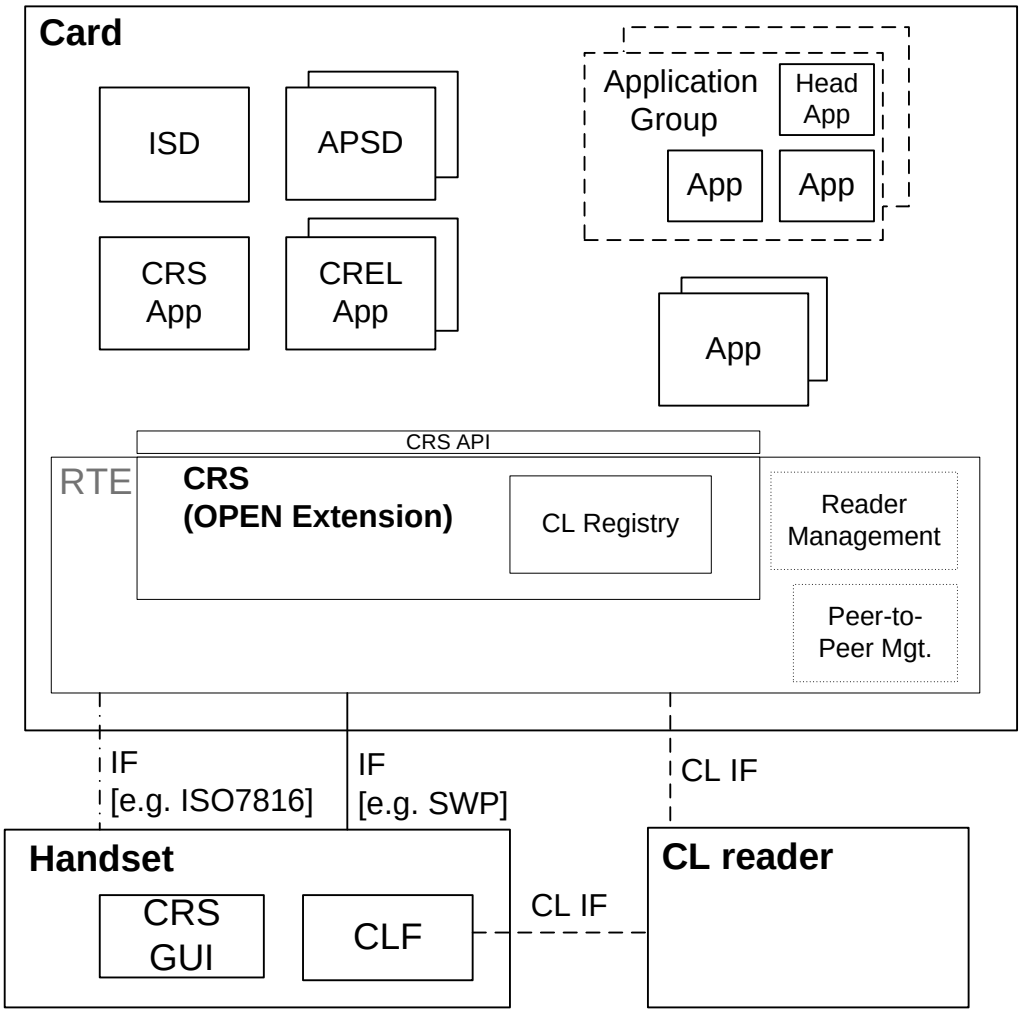
Date	Version	List of Modifications
April 2014	1.1.0.1	• Section 6.3.2.2. Added precision for the selection of Type F applications. • Section 6.5.2.2. Added precision regarding the usage of Recognition Algorithm for Type F applications. • Fixed references to section 8.3, Initial Activation State. • Added precision after Table 4-3: Protocol Data Type A and Table 4-5: Protocol Data Type B. • Sections 3.7.3, 8.1, and 8.3. Added precisions regarding possible Contactless Activation States when the Application is in the INSTALLED or LOCKED state.
April 2014	1.1.0.2	Public Review.

2 System Overview

2.1 High Level Architecture

Figure 2-1 provides a high level overview of the system architecture relevant to contactless services specified in this document.

Figure 2-1: High Level Architecture Overview



Some entities shown in the figure above are outside the scope of this document but are presented in the figure for a better understanding of the system context. Specifically, the APIs for Reader Mode, P2P mode, and the realization of the low level contactless connection between the card and the contactless terminal, (e.g. direct antenna connection, SWP, or other), are beyond the scope of this specification.

ISD

The Issuer Security Domain (ISD) is the primary, mandatory, on-card component representative of the Card Administrator, typically the Card Issuer (see [GPCS]).

APSD

The Application Provider Security Domain (APSD) is an optional on-card component representative of an Application Provider.

OPEN

This entity represents the GlobalPlatform Environment (OPEN). It provides an API to applications, command dispatch, Application selection, (optional) logical channel management, and Card Content management (see [GPCS]).

CL Registry

The Contactless Registry adds functions to the GlobalPlatform Registry Services for managing Contactless Applications (see section 2.2).

Application

An Application is an executable component providing end-user services after having been installed and made selectable by GlobalPlatform card management functions.

Application Group

An Application Group is a concept that allows group member Applications to be represented to the end-user by a dedicated group Head Application (see section 3.3).

Head Application

The Head Application (see section 3.3) is a Contactless Application that owns the Display Control Information and the Contactless Protocol Parameters of an Application Group.

CRS (OPEN Extension)

The Contactless Registry Service (CRS) is an extension of the OPEN to manage and provide access to contactless registry parameters (see section 2.2).

CRS Application

The CRS Application is an optional component designed for the management of Contactless Applications by the end user (see section 3.9).

CREL Application

A Contactless Registry Event Listener (CREL) Application is an Application interested in being notified of the changes occurring to one or more Contactless Application (see section 3.8).

CRS GUI

This handset-side component allows the end-user to interact with the on-card CRS Application.

CL Reader

The Contactless Reader represents the card external contactless communication interface allowing on-card and off-card Contactless Applications to interact with each other.

2.2 Contactless Registry Service (CRS)

The Contactless Registry Service (CRS) is an extension of the OPEN providing:

- The Contactless Registry; an extension of the GlobalPlatform Registry
- The CRS API; an extension of the GlobalPlatform API
- Services for:
  - Maintaining the Contactless Registry subsequent to the installation, update, or deletion of an Application
  - Maintaining the Contactless Registry subsequent to the activation, deactivation, or a change of priority of an Application
  - Providing the Contactless Registry information upon request from an authorized entity
  - Notifying Applications of changes made in the Contactless Registry
- Contactless protocol management
- Access control on Communication Interfaces
- Application selection rules on the contactless interface
- Contactless privileges

2.3 High-Level Communication Flows

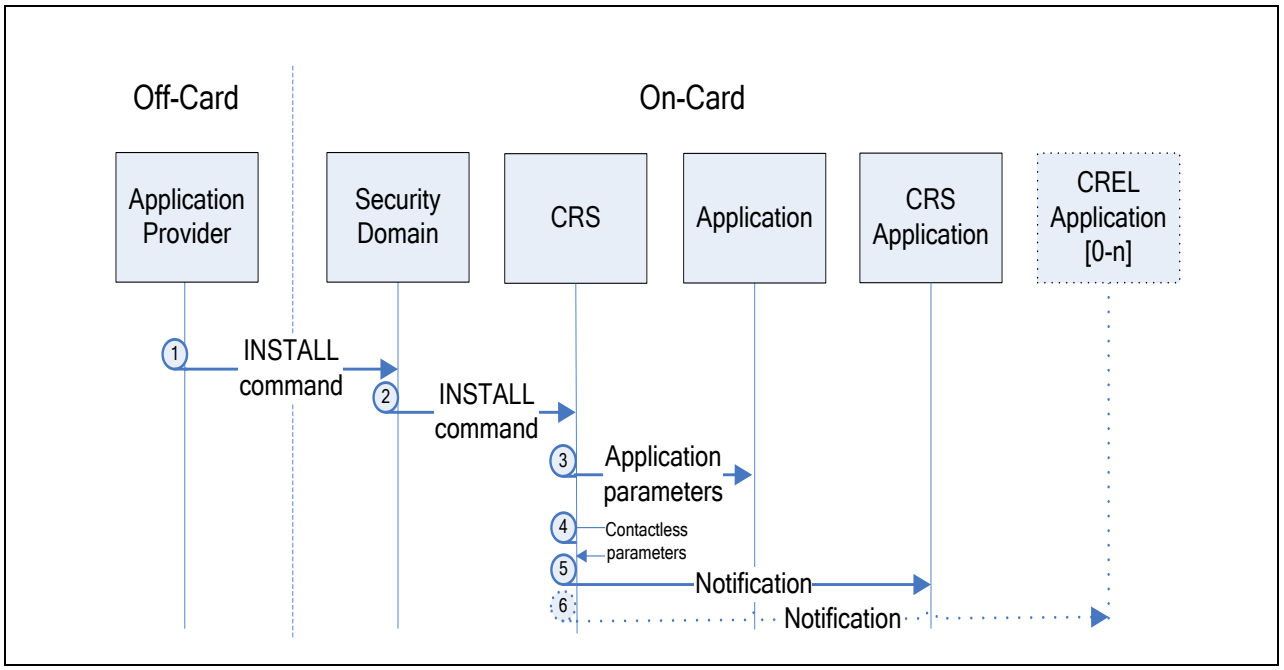
This section provides a high level overview of the main communication flows between the new system entities introduced in this specification. The flows described below show the interaction of the entities when executing the following new functions:

- Populating contactless registry parameters during Application installation
- Populating contactless registry parameters during Application personalization
- Removing contactless registry parameters during Application deletion
- Activation and deactivation of Contactless Applications (including conflict resolution)

2.3.1 Population and Update of Contactless Registry Parameters

Figure 2-2 shows the flow of populating the contactless registry parameters during the installation of an application. This flow also applies to the update of the contactless registry parameters using the INSTALL [for registry update] command.

Figure 2-2: Flow of Populating an Application's Contactless Registry Parameters during Installation



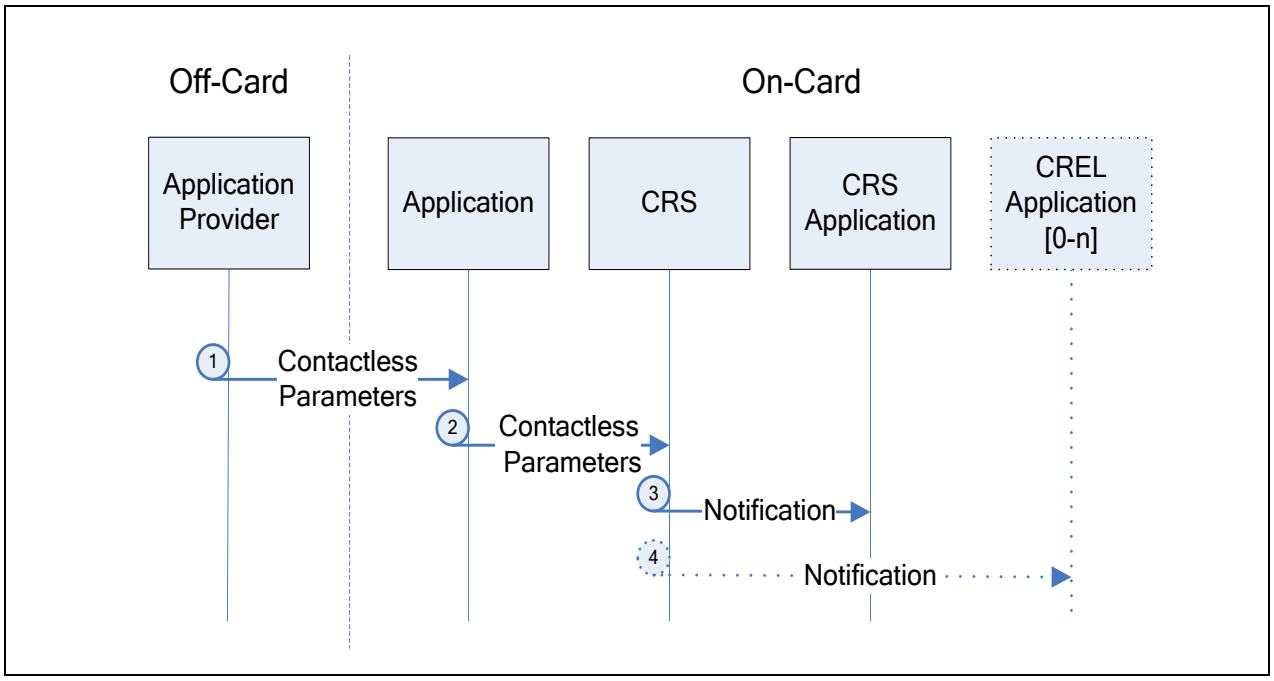
CRS shall send notifications about updates of the Contactless Registry to the CRS Application and registered CREL Applications.

The following steps are highlighted in the figure above:

1. The Application Provider supplies its Application with Contactless Registry Parameters through its Security Domain using the INSTALL command.
2. The Security Domain forwards the INSTALL command to the OPEN.
3. The OPEN populates the Application-specific Parameters.
4. The OPEN populates the Contactless Registry Parameters.
5. The CRS API notifies the CRS Application that a change has occurred in the Registry entry of that Application.
6. The OPEN notifies CREL Application(s) referenced by the Application.

Figure 2-3 shows the flow of populating or updating the contactless registry parameters of an application through the application itself.

Figure 2-3: Flow of Populating an Application's Contactless Registry Parameters Through Itself

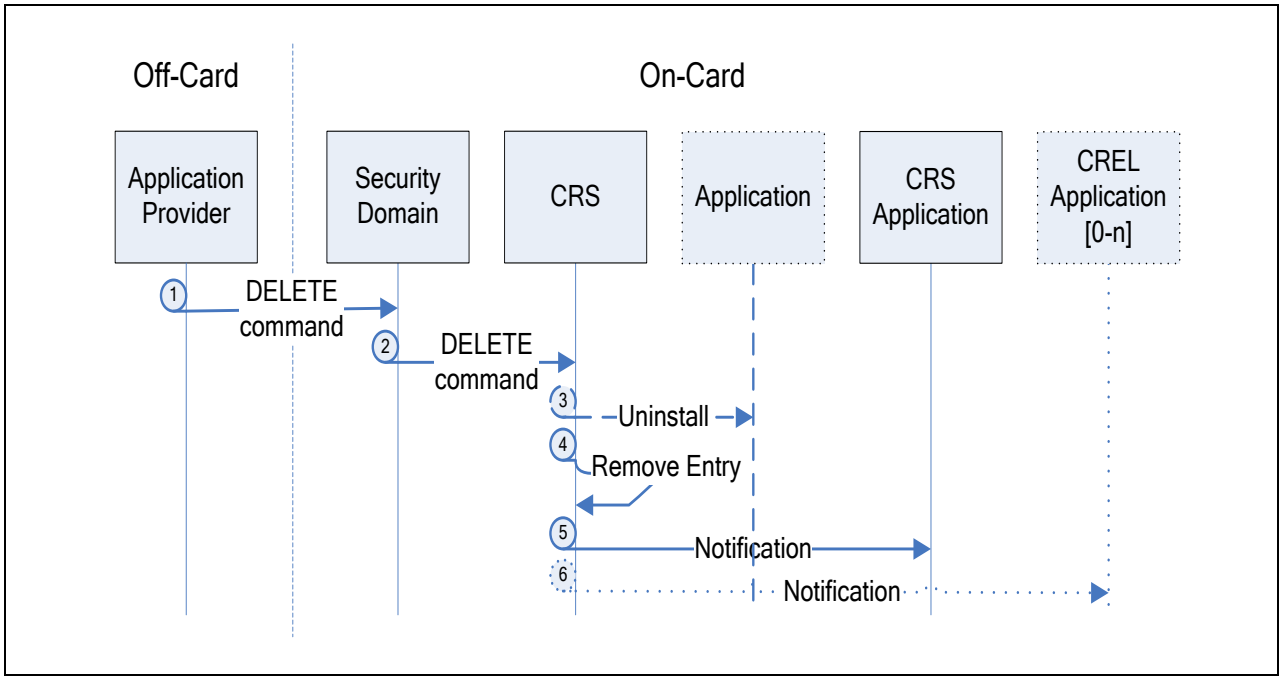


The following steps are highlighted in the figure above:

1. The Application Provider supplies its Application with Contactless Registry Parameters using a proprietary command.
2. The Application updates its Contactless Registry Parameters using the CRS API.
3. The CRS Application is notified of the changes that occurred to the Application.
4. CREL Application(s) referenced by the Application, if any, are notified of the changes that occurred to the Application.

Figure 2-4 shows the flow of removing the contactless registry parameters during deletion of an application.

Figure 2-4: Flow of Removing an Application's Contactless Registry Parameters during Deletion



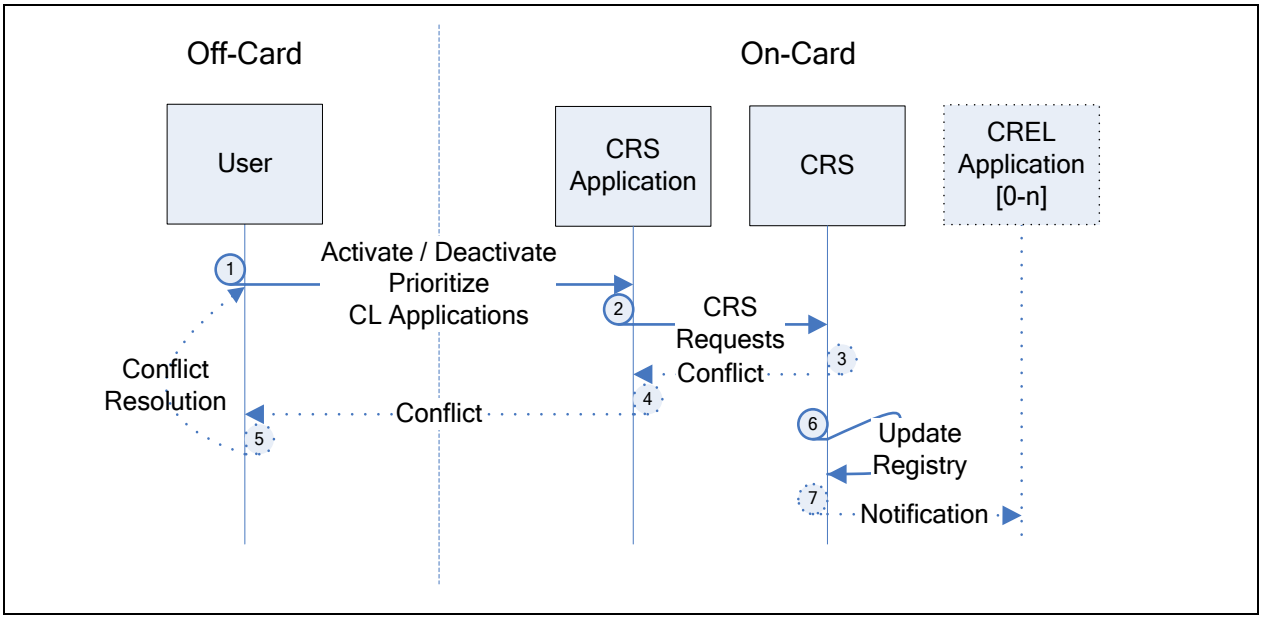
The following steps are highlighted in the figure above:

1. An Application Provider (off-card) sends the DELETE command to the associated Security Domain of the Application that is to be deleted.
2. The addressed Security Domain delegates the deletion command to the CRS (an extension of the OPEN) using proprietary interfaces.
3. The CRS/OPEN invokes the "Uninstall" method of the Application to be deleted as specified in [GPCS].
4. The CRS/OPEN removes all entries of the Application that is to be deleted in the GlobalPlatform Registry. The OPEN then deletes the data objects of the Application.
5. The CRS/OPEN sends the notification about the deletion of the Application to the CRS Application.
6. CREL Application(s) referenced by the Application, if any, are notified of the deletion of the Application.

2.3.2 User Interaction

Figure 2-5 shows the flow of the user interaction during activation, deactivation or change of priority of an application.

Figure 2-5: Flow of User Interaction



The following steps are highlighted in the figure above:

1. An Off-Card user interface provides the user with means to activate and deactivate a Contactless Application and to define the priority by which they are considered for selection through a contactless reader.
2. The Off-Card entity maps the user request to one or more command APDU defined for the CRS Application and transmits them to this Application. The CRS Application processes these commands with the help of an On-Card CRS API.
3. If an Application activation is requested, the CRS Application asks the CRS to determine whether the request conflicts with one (or more) of the already activated Applications. Two types of conflicts may occur:
   - Protocol parameter conflicts, which are detected by the OPEN as defined in section 4.5 and reported by the CRS Application.
   - Activation policy conflicts, which are application specific conflicts (e.g. shared memory is locked, a PIN must be validated before activation, etc.). During the activation process, such conflicts may be reported to the CRS Application by the Application being activated.
4. If a conflict is detected, a corresponding response is returned to the Off-Card entity to inform the user of the conflict.
5. The user can resolve the conflict by restarting at step 1 with deactivating the conflicting application(s) and then repeating the original request.
6. To deactivate, activate (without a conflict) or change the selection priority of an Application, the CRS updates the Registry entry of the respective Application.
7. CREL Applications possibly linked to the modified Application are notified about the registry change by the CRS.

2.4 Tag Encoding Rule

All TLVs defined in the Amendment C are encoded as specified in [GPCS]. The Identifier octet is context-specific class as described in ISO/IEC 8825-1 [8825-1]. The tag number is in the range [0,30] and is reserved for GlobalPlatform usage unless explicitly specified otherwise.

3 User Interaction Management

3.1 Definition and Scope

This section describes the roles and interaction between the entities involved in providing information that support the end-user to choose the Applications that are to be activated and deactivated and to prioritize the activated Applications on the contactless interface. These entities are the CRS Application, Standalone Applications, Application Groups, and Contactless Registry Event Listener (CREL) Applications.

This section also defines parameters that can be used to manage the visibility of Applications as seen by off-card entities. These User Interaction Parameters are accessed by Security Domains using the INSTALL commands as defined in section 11.1 or by the GlobalPlatform CRS Application using the GET STATUS or SET STATUS commands. The CRS API also provides similar functionality.

3.2 Display Control Information

Display Control Information is used to manage an off-card GUI by providing information for a Logo, a URL, or any related display information. The presence of Display Control Information is optional.

Display Control Information should be personalized for Applications to be visible to the end-user. Typically, Head Applications and Standalone Applications (see section 3.7) should be personalized with this parameter.

The Display Control Information shall be embedded within the Display Control Template (tag '7F20') and may include one or more of the following data elements:

- Tag '5F50': Uniform Resource Locator (see ISO/IEC 7816-6 [7816-6]). The purpose of this URL is out of scope of this document.
- Tag '6D': Application Image Template (see [7816-6])
  - Tag '5F44' Application Image
  - Tag '67' Authentication Data (optional; if this tag is not present, the Application Image format shall be according to ISO/IEC 10918-1 [10918-1])
- Tag '53' Image Encoding Format
  - '01' – JPG
  - '02' – JPG-2000
  - '03' – GIF
  - '04' – PNG-8
  - '05' – PNG-24
  - '06' – BMP
  - '07' to '7F' – RFU
  - '80' to '8F' – Proprietary usage
  - '90' to 'FF' – RFU
- Tag '5F45': Display Message (Data element containing a message to display) (see [7816-6]). The following formats are defined for the value of the information to be displayed to the user. A leading single byte identifier, given below, shall indicate the display message encoding format:
  - '01' – Format ASCII, which includes displayable characters (alphabetic, numerical, and special) and space in the range from '20' to '7E', coded on one byte and left-justified (see ISO/IEC 8859-1 [8859-1])
  - '02' – Format BCD, which includes only numerical digits, coded on a nibble (4 bits), left justified, and eventually padded on the right with an 'F' nibble if necessary (i.e. if the number of digits is odd)
  - '03' – Format HEX, which is equivalent to a transparent mode (“as is”) and includes all binary values coded on one byte
  - '04' – Format UTF-8, which is able to represent any character in the UNICODE character set (see IETF RFC 3629 [3629])

3.3 Policy Restricted Applications

The Policy Restricted Applications TLV provides one or more AIDs of Applications that should not be ACTIVATED on the contactless interface at the same time as the Application that has this TLV in its Registry. This TLV is not used by the OPEN for contactless protocol parameter conflict resolution or for any other purpose. This TLV may be used by the CRS Application and CREL Applications when applying their specific business logic.

Note: If an Application has the Contactless Self Activation privilege, the CRS Application will not be involved in the activation of the Application, although the CRS Application will be notified of this activation and may take action based on its own business logic and privileges.

Table 3-1 defines the value part of the “Policy Restricted Applications” TLV.

Table 3-1: Value Part of “Policy Restricted Applications” TLV

Tag	Length	Description	Presence
'4F'	5-16	AID of the Policy Restricted Application (e.g. PPSE)	Mandatory
'4F'	5-16	AID of the Policy Restricted Application	Optional
…	…	…	

When used as a search criterion (see section 3.11.2), the “Policy Restricted Applications” TLV shall contain only one AID.

3.4 Application Discretionary Data

This Application Discretionary data is accessible to the CRS Application and CREL Applications, if any, for proprietary usage.

3.5 Application Family

The Application Family indicates the sector of industry that a Contactless Application belongs to. This indicator may be used to group together the Applications of the same family for a similar process such as presenting only one family of Applications to the user.

The Application Family is coded as the AFI defined in ISO/IEC 14443-3 [14443-3]. The Application Family can be specified for any Contactless Application, regardless of its protocol type. Special rules defined in [14443-3] for filtering according to sub-family identifiers shall not be taken into account.

The OPEN shall not enforce that the Application Family and the AFI defined in section 4.7 (for applications using Type B protocol) are the same.

3.6 Display Required Indicator

The Display Required Indicator TLV is an optional TLV assigned to an Application so that it can indicate whether or not it is able to perform a contactless transaction even when the display is not available. This parameter is not evaluated by the OPEN. The CRS and/or CREL Applications may use this indicator for their own business purposes.

Table 3-2 defines the value part of the “Display Required Indicator” TLV.

Table 3-2: Value Part of “Display Required Indicator” TLV

Value	Description
'00'	Application requires a display
'01'	Application does not require a display

If this TLV is not present, the default value '00' applies: The Application indicates that it requires a display to perform the contactless transaction.

NOTE: See EMVCo AAUI [AAUI] for an example of how this indicator can be used.

3.7 Application Groups

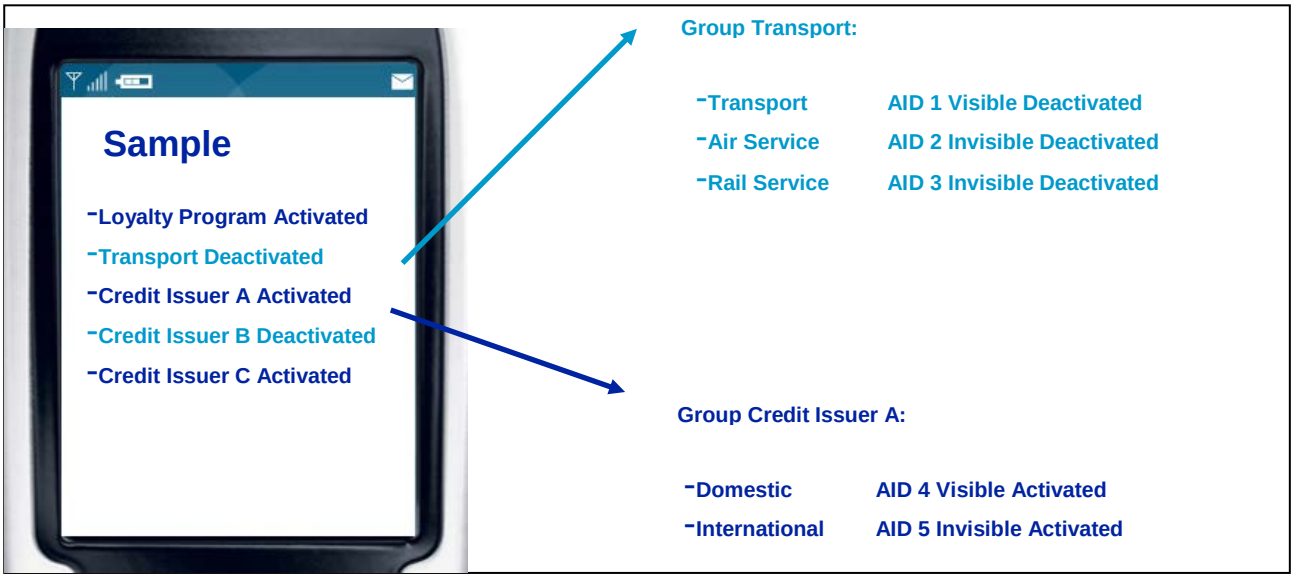
3.7.1 Definition and Scope

An Application Group is composed of one and only one Head Application and one or more Member Applications. The Head Application is associated with a Group Authorization List defining the possible membership of the group as a list of AIDs. Applications referenced (by AID) in this list may request to join the group.

An Application shall not be a member of more than one Application Group. An Application that is neither a Head Application, nor a member of an Application Group, is a Standalone Application. At any time, a Standalone Application may register itself as a Head Application, hence defining a new Application Group, or as the member of an Application Group.

The following Contactless Activation States are used throughout this specification: ACTIVATED, DEACTIVATED, NON_ACTIVATABLE. These states are defined in section 8.1.

Figure 3-1: Application Group



3.7.2 Head Application

The following rules apply to the Head Application:

- The Head Application establishes Contactless Protocol Parameters (see section 4.3) and prioritization for the Member Applications. It is the responsibility of the Application Provider of the Head Application to personalize the Head Application with Contactless Protocol Parameter values that support all of the Member Applications named in its Group Authorization List.
- When the Head Application is set to the ACTIVATED state, all the Member Applications shall be set to the ACTIVATED state unless they are in the NON_ACTIVATABLE state.
- When the Head Application is set to the DEACTIVATED state, all the Member Applications shall be set to the DEACTIVATED state, unless the Member Application is in the NON_ACTIVATABLE state.
- When the Head Application is set to the NON_ACTIVATABLE state, all the Member Applications shall be set to the DEACTIVATED state, unless the Member Application is in the NON_ACTIVATABLE state.
- When the Head Application is deleted, Applications previously belonging to this Application Group become Standalone Applications. All the former Member Applications shall be set to the DEACTIVATED state, unless the former Member Application is in the NON_ACTIVATABLE state.

See section 8.1 for more details on activation states.

When the accessibility of the Head Application through the contactless (proximity) communication interface is disabled (see Chapter 5 for details on Communication Interface Access Configuration), the Head Application shall be automatically set to the DEACTIVATED state, and the activation state of Member Applications shall be updated according to above rules. The Head Application cannot be set back to the ACTIVATED state until it becomes accessible through the contactless (proximity) communication interface again.

3.7.3 Member Applications

The following rules apply to Member Applications:

- Contactless Protocol Parameters (see section 4.3) and prioritization are provided by the Head Application, even if a Member Application was personalized with its own parameter values.
- If its Head Application is not in the ACTIVATED state, then a Member Application cannot be set to the ACTIVATED state.
- If a Member Application is neither in the INSTALLED state nor in the LOCKED state and its Head Application is in the ACTIVATED state, then the Member Application cannot be set to the DEACTIVATED state.
- When a Member Application is set to the NON_ACTIVATABLE state, only that Application becomes NON_ACTIVATABLE. Other Member Applications shall follow the activation state of the Head Application.
- When a Member Application is in the NON_ACTIVATABLE state:
  - The Member Application shall remain in the NON_ACTIVATABLE state even if the Head Application is set to the ACTIVATED state.
  - If the Member Application is set to the DEACTIVATED state:
    - If the Head Application is in the NON_ACTIVATABLE or DEACTIVATED state, then the Member Application shall be set to the DEACTIVATED state.
    - If the Head Application is in the ACTIVATED state, then the Member Application shall be set to the ACTIVATED state.

See section 8.1 for more details on activation states.

When the accessibility of a Member Application through the contactless (proximity) communication interface is disabled (see Chapter 5 for details on Communication Interface Access Configuration), the Member Application shall be automatically set to the DEACTIVATED state. This Member Application cannot be set back to the ACTIVATED state until it becomes accessible through the CL interface again. The activation states and accessibility of other Member Applications shall not be impacted.

3.7.4 Joining or Leaving an Application Group

The “Head Application” TLV allows an Application to request to be member of a group by providing the AID of the Head Application of the Group or to leave an Application Group using INSTALL [for install], INSTALL [for install and make selectable], or INSTALL [for registry update]. An Application may itself request to join or leave an Application Group using the CRS API.

Table 3-3 defines the value part of the “Head Application” TLV.

Table 3-3: Value Part of “Head Application” TLV

Tag	Length	Value Description
'4F'	5-16	AID of the group’s Head Application to join.

The following rules apply when joining an Application Group:

- The OPEN checks that the application does not yet belong to a group and is not itself a Head Application.
- If the requested Head Application is present, then the OPEN checks that the application requesting to join the group has its AID listed in corresponding Group Authorization List.
- If the requested Head Application is not present (i.e. specified Application Group does not exist), the installation does not fail. The association will be performed at the time the Head Application is installed and if the application AID is listed in its Group Authorization List.
- If the Member Application is in the NON_ACTIVATABLE state, it shall remain in the NON_ACTIVATABLE state.
- If the Head Application is in the DEACTIVATED or NON_ACTIVATABLE state, the Member Application shall be set to the DEACTIVATED state.
- If the Head Application is in the ACTIVATED state, the Member Application shall be set to the ACTIVATED state. The Current Protocol Parameters shall be recalculated as detailed in section 4.4.

The “Head Application” TLV shall not be in the same command data field as the “Add to the Group Authorization List” or “Remove from the Group Authorization List” TLVs.

The following rules apply when leaving an Application Group:

- The Head Application TLV shall be provided with empty length.
- The OPEN shall check that the Application belongs to an Application Group. If yes, the Application is removed from the Application Group, otherwise an error status is returned.
- The Application becomes a Standalone Application
- The Application shall be set to the DEACTIVATED state, unless it was previously in the NON_ACTIVATABLE state.
- The priority of the Application shall be set to the lowest priority.

3.7.5 Add to the Group Authorization List

When present in the INSTALL [for install], the INSTALL [for install and make selectable], or the INSTALL [for registry update] command, the “Add to the Group Authorization List” parameter (see Table 3-4) establishes the target application as a Head Application, by creating its Group Authorization List or adding AIDs to the existing one. A Head Application may itself request to update its Group Authorization List using the CRS API.

Table 3-4 defines the value part of the “Add to the Group Authorization List” TLV.

Table 3-4: Value Part of “Add to the Group Authorization List” TLV

Tag	Length	Description
'4F'	5-16	AID of an Application to be added to the Group Authorization List
'4F'	5-16	AID of an Application to be added to the Group Authorization List
…		

The following rules apply:

- The OPEN checks that the target Application is not a Member Application of an Application Group.
- The OPEN does not check for the uniqueness of an AID in multiple Group Authorization Lists.
- The installation does not fail if an AID in the Group Authorization List is already associated to another group or not present.

The “Add to the Group Authorization List” TLV shall not be in the same command data field as the “Head Application” TLV.

If this TLV contains the AID of an Application that already exists in the Group Authorization List of the Head Application, no error is reported and no notifications are made.

3.7.6 Remove from the Group Authorization List

When present in the INSTALL [for registry update] command, the “Remove from the Group Authorization List” parameter (see Table 3-5) is used to remove one or more AIDs from the Group Authorization List. A Head Application may itself request to update its Group Authorization List using the CRS API.

If an AID that was removed from the Group Authorization List refers to a Member Application, that Application:

- is removed from the group (but its Head Application field shall remain unmodified);
- becomes a Standalone Application;
- shall be set to the DEACTIVATED state, unless the former Member Application is in the NON_ACTIVATABLE state;
- and its priority shall be set to the lowest priority.

If the Group Authorization List becomes empty, this Head Application becomes a Standalone Application.

Table 3-5 defines the value part of the “Remove from the Group Authorization List” TLV.

Table 3-5: Value Part of “Remove from the Group Authorization List” TLV

Tag	Length	Description
'4F'	5-16	AID of an Application to be removed from the Group Authorization List
'4F'	5-16	AID of an Application to be removed from the Group Authorization List
…	…	…

The “Remove from Group Authorization List” TLV shall not be in the same command data field as the “Head Application” TLV.

If this TLV contains the AID of an Application that does not exist in the Group Authorization List of the Head Application, no error is reported and no notifications are made.

3.8 CREL Application

3.8.1 Definition and Scope

The Contactless Registry Event Listener (CREL) Application is an on-card Application implementing specific business rules for a specific set of Contactless Applications. The CREL Application is referenced by one or several Applications during their installation process. The CREL Application shall be notified by the OPEN after any change to the Registry entry of any referencing Application. The CREL Application shall implement and expose the `CRELApplication` interface. The actions, if any, taken by the CREL Application upon receipt of the notification are out of the scope of this document.

3.8.2 CREL Application Registration

During the installation process, during Registry update, or using the CRS API, any Application may indicate to the OPEN a list of CREL Applications that shall be notified of Registry events occurring to the Application, by using the CREL List TLV structure (see Table 3-6).

If a referenced CREL Application is not present or does not implement nor expose the `CRELApplication` interface, the operation does not fail: The Application is notified of the association with the CREL Application, however, this CREL Application will not receive the notifications. If a missing CREL Application is installed later, and it implements and exposes the `CRELApplication` interface, then it will receive notifications issued after its installation.

The CREL Application shall be able to retrieve the list of all the Applications that are referencing it. The CREL Application shall be able to retrieve the Registry entry of each Application that references it and read the content of the entry without any specific Privilege.

- The CREL Application may deactivate an Application referencing it (see section 8.1 for details on Contactless Activation State transition rules). The CREL Application shall not be able to make any other change to the Registry entries of referencing Applications.
- The CREL Application may reset the Volatile Priority (see section 6.2.2) if it was assigned to an Application referencing it.
- If the Application that the CREL Application wishes to deactivate is a Member Application (i.e. belonging to an Application Group; see section 3.7), it may not be possible to complete the deactivation due to the current activation state of its associated Head Application. In this case, if the Head Application is also referencing it, the CREL Application may try to deactivate the entire group by deactivating the Head Application.

3.8.3 Add to the CREL List

When present in the INSTALL [for install], the INSTALL [for install and make selectable], the INSTALL [for registry update] command, or using the CRS API, the "Add to the CREL List" parameter (see Table 3-6) provides the list of AIDs of CREL Applications that shall be notified by the OPEN upon any change in the `GPRegistryEntry` of the target Application. The OPEN shall be able to retrieve the list of all CREL Applications that an Application is referencing.

Table 3-6 defines the value part of the "Add to the CREL List" TLV.

Table 3-6: Value Part of "Add to the CREL List" TLV

Tag	Length	Description
'4F'	5-16	AID of a CREL Application that shall be added to the CREL List
'4F'	5-16	AID of a CREL Application that shall be added to the CREL List
…	…	…

The following rule applies to the notification list:

- The newly added CREL Application is notified.
- If the AID of a CREL Application in the CREL List was previously added, no error is reported and no notifications are made.

3.8.4 Remove from the CREL List

When present in the INSTALL [for registry update] command or using the CRS API, the "Remove from the CREL List" parameter (see Table 3-7) provides the list of AIDs of CREL Applications that shall be removed from the notification list of that Application.

Table 3-7 defines the value part of the "Remove from the CREL List" TLV.

Table 3-7: Value Part of "Remove from the CREL List" TLV

Tag	Length	Description
'4F'	5-16	AID of a CREL Application that shall be removed from the CREL List
'4F'	5-16	AID of a CREL Application that shall be removed from the CREL List
…	…	…

The following rule applies to the notification list:

- The removed CREL Application is notified.

If the AID of a CREL Application does not exist in the CREL List, no error is reported and no notifications are made.

3.9 CRS Application

3.9.1 Definition and Scope

The Contactless Registry Service (CRS) Application is responsible for providing the user with means to retrieve a list of all Application Groups and Standalone Applications, and to activate, deactivate, or change the Priority or Volatile Priority of these entities on the contactless interface.

When present, the CRS Application shall be accessible by the on-card Applications to request switching their own Activation State to ACTIVATED (see section 8.1).

The following requirements apply:

- Only one CRS Application shall exist per secure element.
- The Contactless Activation Privilege and the Global Registry Privilege must be assigned to the CRS Application. These privileges allow the CRS Application to successfully invoke the CRS API to
  - retrieve a list of all Applications,
  - activate or deactivate these Applications,
  - change the Priority or Volatile Priority of these Applications on the contactless interface (see Chapter 6)
  - globally switch on or off the contactless interface (see section 8.4)
- The CRS Application shall implement and expose the `CRSApplication` Interface. The `CRSApplication.processCLRequest()` method shall be called by the OPEN for each activation requested by an Application without the Contactless Self-Activation Privilege.
  - The reference to the CL registry entry of the targeted Application, which was passed as a parameter of `CRSApplication.processCLRequest()`, can be used by the CRS Application to retrieve the list of conflicting Applications.
  - The behavior of the `CRSApplication.processCLRequest()` method is out of the scope of this specification (may depend on the policy and business logic of the card issuer).

A CRS Application which is installed with the reserved AID for the GlobalPlatform CRS Application shall support the APDU command set described in section 3.11.

3.10 Notification Rules

3.10.1 General Rules

When a power loss occurs, and not all Applications have been notified of the most recent Registry modification, the following rules apply:

- If no transaction was open at the time of the power loss, notifications for the most recent registry modification shall be issued again for all Applications upon the next card reset. Applications should be aware that they may be notified again in the case of power loss.
- If a transaction was open at the time of the power loss, previous modifications to the Registry are rolled back and the issuance of the notifications is not restarted. However, it is strongly recommended not to open such a transaction due to the operations potentially performed by notified Applications, which may exceed the system's ability to log operations during the transaction.

The order of notification is not defined.

WARNING: Application developers shall be aware of the risk of creating notification loops; e.g. an application could create such a loop by trying to undo a modification triggered by one of its associated CREL applications, for which it is notified.

3.10.2 CREL Notification

CREL Applications referenced by an Application, 'A', are notified when the Application 'A':

- switches to the Lifecycle State INSTALLED, SELECTABLE, or LOCKED
- is unlocked or deleted
- switches to the Activation State ACTIVATED, NON_ACTIVATABLE, DEACTIVATED
- has any of its Contactless Protocol Parameters (see Table 11-3) updated
- has any of its User Interaction parameters (see Table 11-5) updated
- has its partial selection order changed
- has its Volatile Priority changed

A CREL Application shall not be notified when it is the originator of an event.

See Annex A, where the details of the corresponding GlobalPlatform Events that are used for notifications are described.

3.10.3 CRS Notification

The CRS Application shall be notified about all Registry changes (as defined for CREL Applications, see section 3.10.2) occurring to any Application when it is not the originator for the event. Unlike a CREL Application, the CRS Application does not need to be registered for any Application in order to be notified about a Registry change for any Application.

3.10.4 Application Notifications

Applications that implement the `CLApplet` interface shall be notified of changes occurring to their Registry entry. However, an Application shall **not** be notified in any of the following cases:

- when it is the originator of an event
- when it is installed (EVENT_INSTALLED); however, it shall be notified the first time it becomes selectable (EVENT_SELECTABLE), i.e. upon successful processing of an INSTALL [for install & make selectable] or INSTALL [for make selectable] command
- when it is deleted (EVENT_DELETED)

3.11 GlobalPlatform CRS Application

3.11.1 Definition and Scope

The GlobalPlatform CRS Application, if present, shall implement the APDU set as defined in this section:

- an APDU to get information on registered Contactless Applications and their respective Contactless Registry parameters
- an APDU to:
  - activate/deactivate Contactless Applications and to respond with the list of conflicting Applications' AIDs
  - establish the order of Applications in the GlobalPlatform Registry. It shall be able to receive a list of Application AIDs (Head Applications, Member Applications, or Standalone Applications)
  - setup/discard the Volatile Priority (see section 6.2)

GlobalPlatform reserves the following AIDs for the purpose of installing the GlobalPlatform CRS Application:

- Executable Load File AID: 'A000000151435253'
- Executable Module AID: 'A00000015143525300'
- Application AID: 'A00000015143525300'

The standard GlobalPlatform CRS Application is defined in this specification as a possible implementation for a CRS Application. It shall be able to process the APDU commands described in sections 3.11.3 through 3.11.6, therefore providing the "user" with expected management capabilities as described in section 3.9.

The GlobalPlatform CRS Application cannot be a Security Domain because of the redefinition of the GET STATUS and SET STATUS commands in Chapter 11.

Parameters are accessed through the GET STATUS and SET STATUS commands.

3.11.2 TLV for Contactless Registry Data

The TLVs described below identify Contactless Registry Data that can be accessed by the GlobalPlatform CRS Application GET STATUS command (see section 3.11.3). These TLVs may be used as search or filter criteria. The description of each tag in Table 3-8 indicates which tag may be used with which command. Filter Criteria (FC) are provided within tag list '5C' and are used to build the response field of the GET STATUS command. Search Criteria (SC) apply to the incoming data field of the GET STATUS command.

Table 3-8: Contactless Registry Data Tag Usage

Contactless Registry Data	Tag	Length	GET STATUS		Section
			SC	FC	
Application AID	'4F'	5-16	Yes	Yes	3.11.2.1
Application Lifecycle State	'9F70'	2	Yes	Yes	3.11.2.2
Display Control Template	'7F20'	var	No	Yes	3.2
Uniform Resource Locator	'5F50'	var	No	Yes	3.2
Application Image Template	'6D'	var	No	Yes	3.2
Display Message	'5F45'	var	No	Yes	3.2
Application Update Counter	'80'	var	No	Yes	3.11.2.3
Selection Priority	'81'	1	Yes	Yes	3.11.2.4
Group Head Application	'A2'	var	Yes	Yes	3.11.2.5
Group Member's Application	'A3'	var	No	Yes	3.11.2.5
CREL Application AID List	'A4'	var	Yes	Yes	3.11.2.6
Policy Restricted Applications	'A5'	var	Yes	Yes	3.3
Application Discretionary Data	'A6'	var	No	Yes	3.4
Application Family	'87'	var	Yes	Yes	3.5
Display Required Indicator	'88'	1	Yes	Yes	3.6
Continuous Processing	'8A'	var	No	Yes	6.4
Recognition Algorithm for Implicit Selection	'8B'	var	No	Yes	6.5
Assigned Protocols for Implicit Selection	'8C'	var	No	Yes	6.6

These tags shall be used as search or filter criteria in a simple, non-constructed form. For example, '5F50' is a sub-tag of '7F20', however, only '5F50' shall appear in the requested tag list (see the example related to Table 3-14).

3.11.2.1 Application AID

The Application AID tag '4F' is a mandatory tag within the Contactless Registry and can be used both as a search criterion and in the tag list. Partial AID matching is supported when this tag is used as a search criterion in the GET STATUS command (see section 3.11.3.2).

3.11.2.2 Application Lifecycle State

The Application Lifecycle State tag '9F70' provides the Application Lifecycle state on the first byte as defined in [GPCS]. The Contactless Activation State is encoded on the second byte as defined in Table 8-1. The Application Lifecycle State tag can be used both as a search criterion and in a tag list.

When tag '9F70' is used as a search criterion:

- Application-specific lifecycle state bits in the first byte shall be ignored
- RFU bits in the second byte shall be ignored

3.11.2.3 Update Counters

The CRS maintains a counter in the range from 0 to 65535 for each registered Application. The counters initially shall be set to zero. For each update made to the Registry entry of an Application, the corresponding counter shall be incremented by one and managed cyclically.

Additionally, the CRS maintains a global counter in the range from 0 to 65535. This counter, which is initially set to zero, is incremented each time an Application is installed or deleted, and each time an Application counter is incremented. This counter is also managed cyclically.

These counters can be used by on-card or off-card entities to detect that changes have occurred in the Registry and to synchronize their own state. If an action leads to several changes in the Registry, counters may be incremented by 1 or more (implementation dependent).

The GlobalPlatform CRS Application allows retrieval of these parameters through an APDU Command response.

The Global Update Counter is returned in the response of the SELECT command. The encoding of the tag's class is SELECT-command specific (see section 3.11.5).

The Application Update Counter is retrieved using the GET STATUS command response data. A TLV can be used in the tag list as a filter criterion.

3.11.2.4 Selection Priority

The Selection Priority indicates the order in which Contactless Applications will be considered during partial selection over the contactless interface (see section 6.2). The Selection Priority is calculated based on the absolute position of the Contactless Application's entry in the GlobalPlatform Registry, and the Volatile Priority, if assigned. Contactless Applications that are assigned Volatile Priority are given the highest priority. When computing the absolute position, both Contactless and Non-Contactless Applications are taken into account.

The Selection Priority is coded as an unsigned byte and may range from 0 to 255. The value of 0 corresponds to the highest priority.

3.11.2.5 Application Group

Two TLVs are used to indicate the grouping relationship as follows:

- TLV Application Group Head (AID of the Head Application); this tag can be used as a search criterion to retrieve the Application Group Membership List.
- TLV Application Group Membership List; this tag can be used in the tag list. This is a list of AIDs of Applications that have successfully joined the group.

3.11.2.6 CREL Application AID List

The TLV CREL Application AID List provides information related to the CREL Application(s) referenced by an Application. The CREL Application AID List is a constructed TLV with at least one mandatory tag '4F' which provides the AID of a CREL Application. The TLV has the following structure:

- TLV CREL Application AID List (Template – constructed tag with sub-data elements) contains the AID(s) of the CREL Application(s) as sub-data elements.
  - Tag '4F': AID of 1st CREL Application
  - Tag '4F': AID of 2nd CREL Application
  - …

When used as a search criterion, the CREL Application AID List shall contain only one AID.

3.11.3 GET STATUS Command

3.11.3.1 Definition and Scope

The GET STATUS command is used to retrieve the CRS registered Contactless Applications display information, the Lifecycle status and other information according to the given match/search criteria. Data on Non-Contactless Applications is not available through this command. No authentication is required to process this command.

3.11.3.2 Command Message

The GET STATUS command message shall be coded according to Table 3-9.

Table 3-9: GET STATUS Command Message for CRS

Code	Value	Meaning
CLA	'80'	
INS	'F2'	GET STATUS
P1	'xx'	Reference control parameter P1
P2	'xx'	Reference control parameter P2
Lc	'xx'	Length of data field
Data	'xx...'	Search criteria
Le	'00'	

3.11.3.2.1 Reference Control Parameter P1

Reference control parameter P1 is used to select a subset of statuses to be included in the response message. It is coded as shown in Table 3-10.

Table 3-10: GET STATUS Reference Control Parameter P1

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
–	1	–	–	–	–	–	–	Applications
X	–	X	X	X	X	X	X	RFU

3.11.3.2.2 Reference Control Parameter P2

The reference control parameter P2 controls the number of consecutive GET STATUS commands and indicates the format of the response message. It shall be coded according to Table 3-11.

Table 3-11: GET STATUS Reference Control Parameter P2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
X	X	X	X	X	X	X	–	RFU
–	–	–	–	–	–	–	0	Get first or all occurrence(s)
–	–	–	–	–	–	–	1	Get next occurrence(s)

The “Get next occurrence(s)” request shall be rejected if no prior GET STATUS [get first or all occurrence(s)] was received within the current Application Session.

3.11.3.2.3 Data Field Sent in the Command Message

The data field is structured as shown in Table 3-12.

Table 3-12: GET STATUS Command for CRS Data Field

Tag	Length	Value Description	Presence
'4F'	0-16	Application AID	Mandatory
'xx' or 'xxxx'	0-n	Other search criteria	Optional
...	...	...	...
'5C'	1-n	Tag list	Optional

The GET STATUS command message data field shall contain at least one TLV-coded search qualifier: the AID (tag '4F'). It shall be possible to search for all the occurrences that match the selection criteria according to the reference control parameter P1 using a search criteria of '4F' '00'.

It shall be possible to search for all occurrences with the same RID.

Other search criteria may be added. In such cases, the additional search criteria shall be TLV coded and appended after the mandatory search criterion AID (tag '4F').

The search shall be limited to Applications accessible on a contactless interface according to the Communication Interface Access Configuration Parameters defined in Chapter 5. When tag '9F70' is used as a search criterion, Application-specific lifecycle state bits in the first byte shall be ignored.

The tag list (tag '5C') indicates to the CRS Application how to construct the response data for each on-card entity matching the search criteria. The value part of this TLV contains a concatenation of tags (without delimitation) indicating the data objects to include in the response.

3.11.3.3 Response Message

3.11.3.3.1 Data Field Returned in the Response Message

Based upon the search criteria of the GET STATUS command data field and the value of the reference control parameter P1 and P2, multiple occurrences of the data structure as described in the following paragraphs may be returned.

When no tag list (tag '5C') is present in the command data field, the data structure shall be set according to Table 3-13. For the purpose of this table, the following definitions of Mandatory, Conditional, and Optional, apply:

- Mandatory: Shall be present in the response (if its super-tag is present in the response)
- Conditional: Shall be present in response if at least one Application matches the search
- Optional: Shall be present if the tag exists in the Contactless Registry

Table 3-13: On-Card Registered Contactless Applications Data (TLV)

Tag	Length	Value Description					Presence
'61'	7-n	Application Template					Conditional
		Tag	**Length**	**Value Description**			
		'4F'	5-16	Application AID			Mandatory
		'9F70'	2	Application Lifecycle State			Mandatory
		'7F20'	Var	Display Control (Template – constructed tag with sub-data elements)			Optional
				Tag	**Length**	**Value Description**	
				'5F50'	var	Uniform Resource Locator	Optional
				'6D'	var	Application Image Template	Optional
				'5F45'	var	Display Message (Data element containing a message to display)	Optional
		'80'	2	Application Update Counter (big endian)			Mandatory
		'81'	1	Selection Priority			Mandatory
		'A2'	Var	Application Group Head (Head Application AID of the group that this application belongs to)			Optional
				Tag	**Length**	**Value Description**	
				'4F'	5-16	Application AID	Mandatory
		'A3'	Var	Application Group Members (Template – constructed tag with sub-data elements)			Optional
				Tag	**Length**	**Value Description**	
				'4F'	5-16	Application AID	Mandatory
				'4F'	5-16	Application AID	Optional
				'4F'	5-16	Application AID	Optional
		'A4'	Var	CREL Application AID List			Optional
				Tag	**Length**	**Value Description**	
				'4F'	5-16	AID of the CREL Application	Mandatory
				'4F'	5-16	AID of the CREL Application	Optional
		'A5'	Var	Policy Restricted Applications			Optional
				Tag	**Length**	**Value Description**	
				'4F'	5-16	AID of the Policy Restricted Application (e.g. PPSE)	Mandatory
				…	…	…	…
		'A6'	Var	Application discretionary data			Optional
		'87'	1	Application Family			Optional
		'88'	1	Display Required Indicator			Optional
		'8C'	Var	Assigned Protocols for Implicit Selection			Optional
		'8A'	1-2	Continuous Processing			Optional
		'8B'	Var	Recognition Algorithm for Implicit Selection			Optional

When a partial AID is provided as input, all corresponding Applications, regardless of being a Standalone Application, Head Application, or a member of a group, shall be returned provided that the other search criteria are fulfilled.

When tag list (tag '5C') is present in the command data field, the response shall contain tag '61' and a conditional set of TLV-coded Application related data. As an example, Table 3-14 indicates the format of this response data when the filter criteria are '4F9F707F20806D'. Tag '61' shall only contain tags that are listed in the tag list (tag '5C') of the command data field for the applications matching the search criteria. The order of the data objects within the application related data (template '61') is arbitrary. If none of the registry entries contain any of the tags listed in the tag list (tag '5C'), an error status message shall be returned.

For the purpose of Table 3-14, the following definitions of Conditional and Optional apply:

- Conditional: Shall be present in the response data if at least one Application matches the search
- Optional: The tag shall not be present if the content does not exist in the Contactless Registry

Table 3-14: Example of Response Data Returned when Filter Criteria Are Provided

Tag	Length	Value Description			Presence
'61'	Variable	Application related data			Conditional
		Tag	**Length**	**Value Description**	
		'4F'	5-16	AID	Optional
		'9F70'	2	Application Lifecycle State	Optional
		'7F20'	var	Display Control Template (including all available sub-tags and their values)	Optional
		'80'	2	Application Update Counter (big endian)	Optional
		'6D'	var	Application Image Template	Optional

3.11.3.3.2 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status word '9000'.

The command may return the warning condition shown in Table 3-15.

Table 3-15: GET STATUS Warning Condition

SW1 SW2	Meaning
'6310'	More data available

Following status word '6310', a subsequent GET STATUS [get next occurrence(s)] may be issued to retrieve additional data.

This command may return a general error condition as listed in section 11.1.3 of [GPCS], or one of the error conditions listed in Table 3-16.

Table 3-16: GET STATUS Error Conditions

SW1 SW2	Meaning
'6A88'	Referenced data not found
'6A80'	Incorrect values in command data

3.11.4 SET STATUS Command

3.11.4.1 Definition and Scope

The SET STATUS command shall be used to modify the Activation State of CRS registered Contactless Application(s), and their priority in the Application Selection process. In addition, this command can be used to globally switch on/off the contactless interface. Non-Contactless Application data cannot be modified through this command. No authentication is required to process this command.

3.11.4.2 Command Message

The SET STATUS command message is coded according to Table 3-17.

Table 3-17: SET STATUS Command Message for CRS

Code	Value	Meaning
CLA	'80'	
INS	'F0'	SET STATUS
P1	'xx'	Status type
P2	'xx'	Status value
Lc	'xx'	Length of data field
Data	'xxxxx…'	
Le	'00'	

3.11.4.2.1 Reference Control Parameter P1 – Status Type

The status type of the SET STATUS command message indicates the type of information that shall be updated. The status type shall be coded according to Table 3-18.

Table 3-18: SET STATUS – Status Type

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	1	Availability State over the Contactless Interface
0	0	0	0	0	0	1	0	Priority Order for Application Selection
0	0	0	0	0	1	0	0	Communication Interface Access
1	0	0	0	0	0	0	0	Remaining Response Data

If the status type indicates Remaining Response Data and there are no remaining response data, then an error condition shall be returned with status word '6A86' (incorrect P1 P2 parameters).

3.11.4.2.2 Reference Control Parameter P2 – Status Value

The meaning of the status value depends on the status type coded in the P1 parameter.

When the status type indicates Remaining Response Data, the status value shall be ignored.

When the status type indicates Availability State over Contactless Interface ('01'), the status value shall indicate the ACTIVATED or DEACTIVATED state and shall be coded as described in Table 8-1. A request to transition to the NON_ACTIVATABLE state shall be rejected with status word '6A86' (incorrect P1 P2 parameters). The Applications to which this setting applies are further specified in the Command Data Field.

When the status type indicates Priority Order for Application Selection ('02'), the status value shall be coded according to Table 3-19.

Table 3-19: SET STATUS – P2 Values for Priority Order for Application Selection

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	1	Assign Highest Priority
1	0	0	0	0	0	0	1	Assign Lowest Priority
0	0	0	0	0	0	1	0	Assign Volatile Priority
1	0	0	0	0	0	1	0	Reset Volatile Priority

See section 6.2 for a description of Priority and Volatile Priority. The Applications to which this setting applies are further specified in the Command Data Field.

When the status type indicates Communication Interface Access ('04'), the status value shall be coded according to Table 3-20.

Table 3-20: SET STATUS – P2 Values for Communication Interface Access

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	0	0	0	0	0	0	0	Communication interface switched ON
0	0	0	0	0	0	0	0	Communication interface switched OFF

In this case, the SET STATUS command is used to switch ON or OFF a communication interface globally. The Communication Interface(s) to which the new setting shall apply is further specified in the Command Data Field.

3.11.4.2.3 Data Field Sent in the Command Message

When the P1 parameter indicates Remaining Response Data, the content of the data field shall be ignored.

When the P1 parameter indicates Availability State over Contactless Interface ('01') or Priority Order for Application Selection ('02'), the data field shall contain the AID(s) of the target Application(s) for which the modification is requested.

Table 3-21: SET STATUS Command Data Field for Status Types P1='01' or P1='02'

Tag	Length	Value Description	Presence
'4F'	5-16	AID	Mandatory
'4F'	5-16	AID	Optional
	...	...	

When the data field indicates several Applications, the GlobalPlatform CRS Application shall perform the requested update for one Application at a time using the CRS API, starting with the first AID occurring in the command data field.

If an AID present in the data field does not identify any CRS-registered Contactless Application, this AID shall be ignored and other requests present in the command (if any) shall be processed. Warning information shall be returned as described in section 3.11.4.3.

When the P1 parameter indicates Availability State over Contactless Interface ('01')

- If the command is a request for activation, then all requested activations shall be performed altogether, or none. Processing of the command for activation shall be aborted after the first conflict is detected. In this case, all previous changes shall be reversed, and diagnostic information, including the AID of the Application in the command data field that conflicts, shall be returned as described in section 3.11.4.3.
- If activation is requested for an application currently in the NON_ACTIVATABLE state then the Application cannot be activated. This shall not be considered as an error and other requests present in the command (if any) shall be processed. Warning information shall be returned as described in section 3.11.4.3.

When the P1 parameter indicates Priority Order for Application Selection ('02')

- If an AID present in the data field identifies a Head Application, all Member Applications from corresponding Application Group (including the Head Application itself) shall have their priority modified accordingly, following their current order in the GlobalPlatform Registry. This rule applies both to modifications of the GlobalPlatform registry order (P2='01' or P2='81') and to modifications of the Volatile Priority (P2='02').
- If the P2 parameter indicates Assign Highest Priority then the last listed AID in the command data field shall receive the Highest Priority.
- If the P2 parameter indicates Assign Lowest Priority then the last listed AID in the command data field shall receive the Lowest Priority.
- If the P2 parameter indicates Assign Volatile Priority ('02'), then only a single AID shall be present in the data field, otherwise an error condition shall be returned with status word '6A80' (wrong data). If a Volatile Priority was already set up, then it is discarded and this command sets up a new one.
- If the P2 parameter indicates Reset Volatile Priority ('82'), then the content of the data field shall be ignored and the Volatile Priority is reset (i.e. emptied). It is not necessary to reset the Volatile Priority before assigning a new one.

When the P1 parameter indicates Communication Interface Access ('04'), then the data field shall be coded as follows:

Table 3-22: SET STATUS Command Data Field for Status Type P1='04'

Tag	Length	Value Description	Presence
'80'	1	Communication Interface Identifier (see Table 5-2)	Mandatory

If an attempt is made to change the availability of Contact-based communication, then an error condition shall be returned with status word '6A81' (function not supported).

3.11.4.3 Response Message

3.11.4.3.1 Data Field Returned in the Response Message

Response data may be present and returned depending on the result of the requested operation(s). When present, the response data field shall be formatted as follows:

Table 3-23: SET STATUS Response Data Field

Tag	Length	Value Description					Presence
'61'	Var	Application Template					Mandatory
		Tag	**Length**	**Value Description**			
		'4F'	5-16	AID of the application that could not be activated due to a conflict in protocol parameters.			Conditional
		'A0'	Var	List of Conflicting Applications			Conditional
				Tag	**Length**	**Value Description**	
				'4F'	5-16	AID of first or only Conflicting Application	Mandatory
				'4F'	5-16	AID of next Conflicting Application	Optional
				…	…		
		'A1'	Var	List of applications for which the operation failed for other reasons (e.g. not found or NON_ACTIVATABLE)			Conditional
				Tag	**Length**	**Value Description**	
				'4F'	5-16	AID of first or only non activated Application	Mandatory
				'4F'	5-16	AID of next non activated Application	Optional
				…	…		
		'A2'	Var	List of applications for which the operation failed for Application policy reasons.			Conditional
				Tag	**Length**	**Value Description**	
				'48'	2	Reason code describing why the activation failed	Mandatory
				'4F'	5-16	AID of first or only conflicting Application	Mandatory
				'4F'	5-16	AID of next conflicting Application	Optional
				'48'	2	Reason code describing why the activation failed	Optional
				'4F'	5-16	AID of first or only conflicting Application	Optional
				'4F'	5-16	AID of next conflicting Application	Optional
				…	…	…	...

The following statements apply when the P1 parameter indicates Availability State over Contactless Interface ('01') or Priority Order for Application Selection ('02'). (Note: A request to assign the Volatile Priority contains an implicit request for activation; see section 6.2.2.)

- If the processing of the command was aborted due to a conflict in protocol parameters, then the response shall contain the AID (tag '4F') of the application that could not be activated due to the conflict, as well as the list (tag 'A0') of the currently activated Applications conflicting with this Application.
- If the processing of the command was aborted due to a policy conflict reported by a Contactless Application (exposing the `CLAppletActivationPolicy` interface), then the response shall contain the AID (tag '4F') of the Application that rejected the activation request, as well as information about existing policy conflicts (tag 'A2'). To retrieve such information, the GlobalPlatform CRS Application shall call the `CLAppletActivationPolicy.getNextApplicationConflictInfo()` method to build the value of tag 'A2': the value of this tag may indicate several conflict reason codes (tag '48'), each one followed by a list of related Application AIDs (tag '4F').

  For the reason code (tag '48'), the range from '0000' to '7FFF' is reserved for GlobalPlatform. The range from '8000' to 'FFFF' is reserved for proprietary use.

  See section 8.2 for details on Application Activation Policy and related conflict detection procedures.

  The response may contain tag 'A0' and tag 'A2' if the Application could not be activated for both protocol parameters and policy conflicts.
- If no conflict was detected but some Applications could not be activated for other reasons (e.g. Application not found or Application was in the NON_ACTIVATABLE state), then the response shall contain the list (tag 'A1') of these Applications.

If response data do not fit in a single response data field, then warning status word '6310' shall be returned. In this case, one or more subsequent SET STATUS commands with P1='80' may be issued to retrieve remaining response data.

Following last response data,

- If template tag 'A0' and/or tag 'A2' was present in response data (i.e. a conflict was detected), then status word '6330' (warning: conflicts were detected) shall be returned. Otherwise,
- If template tag 'A1' was present in response data, then status word '6320' (warning: some Application(s) could not be operated) shall be returned.

3.11.4.3.2 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status word '9000'.

The command may return one of the warning conditions listed in Table 3-24.

Table 3-24: SET STATUS Warning Conditions

SW1 SW2	Meaning
'6310'	More response data available
'6320'	Operation failed for some Application(s)
'6330'	Operation failed due to Activation Conflict(s)

This command may either return a general error condition as listed in section 11.1.3 of [GPCS], or one of the error conditions listed in Table 3-25.

Table 3-25: SET STATUS Error Conditions

SW1 SW2	Meaning
'6985'	Activation conflict detected
'6A80'	Wrong data in data field
'6A86'	Incorrect P1 P2 parameters

3.11.5 SELECT Command

3.11.5.1 Definition and Scope

The GlobalPlatform CRS Application SELECT command is identical to the SELECT command described in section 11.9 of [GPCS] except for the Response Message, as described below.

3.11.5.2 Response Message

The GlobalPlatform CRS Application SELECT response shall be coded as described in Table 3-26.

Table 3-26: GlobalPlatform CRS Application SELECT Response

<table>
<tr><th>Tag</th><th>Length</th><th colspan="5">Value Description</th><th>Presence</th></tr>
<tr><td rowspan="7">'6F'</td><td rowspan="7">Var</td><td colspan="5">FCI Template</td><td>Mandatory</td></tr>
<tr><td>Tag</td><td>Length</td><td colspan="3">Value Description</td><td></td></tr>
<tr><td>'84'</td><td>Var</td><td colspan="3">GlobalPlatform CRS Application AID (see section 3.11.1)</td><td></td></tr>
<tr><td rowspan="4">'A5'</td><td rowspan="4">Var</td><td colspan="3">FCI Proprietary Template</td><td></td></tr>
<tr><td>Tag</td><td>Length</td><td>Value Description</td><td></td></tr>
<tr><td>'9F08'</td><td>2</td><td>Version number (2 bytes), Value '01' '00'</td><td></td></tr>
<tr><td>'80'</td><td>2</td><td>Global Update Counter (big endian)</td><td></td></tr>
</table>

3.11.6 GET DATA Command

3.11.6.1 Definition and Scope

The GlobalPlatform CRS Application GET DATA command is used to retrieve the version number and global update counter.

3.11.6.2 Command Message

The GET DATA command message is coded according to Table 3-27.

Table 3-27: GET DATA Command Message for CRS

Code	Value	Meaning
CLA	'80'	
INS	'CA'	GET DATA
P1	'00'	P1 value
P2	'A5'	P2 value
Le	'00'	

3.11.6.3 Response Message

The GlobalPlatform CRS Application GET DATA response shall be coded as described in Table 3-28.

Table 3-28: GlobalPlatform CRS Application GET DATA Response

Tag	Length	Value Description		
'A5'	var	FCI Proprietary Template		
		Tag	**Length**	**Value Description**
		'9F08'	2	Version number (2 bytes), Value '01' '00'
		'80'	2	Global Update Counter (big endian)

4 Contactless Protocol Management

4.1 Overview and Scope

This section defines:

- the OPEN's responsibility for Contactless Registry parameters
  - Current Protocol Parameters
  - Conflict Detection rules
- Contactless parameter structure
  - Protocol Parameters
  - Protocol Parameters Profiles

4.2 The OPEN requirements

The OPEN owns the default parameter values. Only the Issuer Security Domain is able to update these default parameter values. To do so, the INSTALL [for registry update] command is used without providing an AID. The default parameter values may be retrieved through the CRS API.

The OPEN is also responsible for computation of the Current Protocol Parameters used in the interaction with the Proximity Coupling Device (PCD). These Current Protocol Parameters will be computed whenever an Application is activated or deactivated or Protocol Parameters are updated (see Figure 4-1), so that these parameters reflect the requirements of all activated applications. If an activation request would result in a conflict (see Figure 4-4), the OPEN shall reject this request.

The GlobalPlatform Registry is responsible for storage of the Application's Contactless Protocol Parameters. These parameters may be provided at application installation or using INSTALL [for registry update]. Additionally, an Application may update and retrieve its own Contactless Protocol Parameters using the CRS API. If no specific Protocol Data are provided for an Application, the Current Protocol Parameter settings are used.

4.3 Contactless Protocol Parameters

The OPEN supports the following protocols:

1. Protocol Type A: Defines the parameter values for initialization, anti-collision, and protocol activation, for Type A. See [14443-3] and ISO 14443-4 [14443-4].
2. Protocol Type B: Defines the parameter values for initialization, anti-collision, and protocol activation, for Type B. See [14443-3].
3. Protocol Type F: Defines the parameter values for initialization, anti-collision, and protocol activation, for Type F (see JIS X 6319-4 [JIS6319-4]). Please note that Type F is also referred to as ISO/IEC 18092 [18092] 212 kbps and 424 kbps passive mode in ETSI TS 102 613 [102613] and ETSI TS 102 622 [102622]; however, [JIS6319-4] provides more detailed information about anti-collision parameters and how APDU-based communications can be supported over Type F.

An Application is able to request specific values in the Protocol Data for a particular protocol type. If the Application has no mandatory value for a specific protocol type, it shall not provide Protocol Data parameters for this protocol type.

For each protocol type, Protocol Parameters consist of:

- a Protocol Data setting initialization and anti-collision values for transactions with the PCD;
- a Mandatory Mask identifying the specific fields and operations (e.g. XOR or MAX) that shall be used when combining Protocol Data for conflict detection and current protocol parameter computation.

The Protocol Parameters contain basic fields (byte or short values) or LV structured fields. For basic fields and both length and value parts of LV structured fields, the Mandatory Mask may be used to indicate that the value is not mandatory, or that it shall be strictly enforced or considered as a maximum value.

The OPEN is responsible for computing the Current Protocol Parameters for each supported protocol type. The Current Protocol Parameters are the result of combining the Default Protocol Parameters and the Protocol Parameters of activated Applications.

The OPEN owns the Default Protocol Parameters for all supported protocols. There are no mandatory bits within the Default Protocol Parameters.

4.4 Current Protocol Parameter Computation

The following procedure to calculate the Current Protocol Parameters applies for each protocol type independently. This procedure is triggered in the case of any one of the following events:

- a change to the Default Protocol Data
- a change to the Application-specific Protocol Data of an activated Application
- a transition of an Application to the ACTIVATED or to the DEACTIVATED state

If the change or transition occurs during a contactless session, the procedure is applied and the change will take effect after the current contactless session is finished.

The Current Protocol Parameters are the result of the reconciliation of the Default Protocol Data and the Application-specific Protocol Data of all the activated applications.

4.4.1 Current Protocol Parameter Initialization

The Current Protocol Parameters are initialized with the Default Protocol Parameters of the card.

- The Protocol Parameter Data is set equal to the Default Protocol Parameter data, and
- The Mandatory Mask is set to zero.

For Type F, the Default Protocol Parameter consists of one parameter indicating the maximum number of anti-collision parameter entries supported by OPEN. There is no Mandatory Mask for Type F.

4.4.2 Current Protocol Parameter: Computation on Activation

The Current Protocol Parameters shall be updated upon Application activation. To do so, the OPEN shall combine the Protocol Data computed over the currently activated Applications and the Protocol Data of the newly activated Application.

4.4.2.1 Computation on Activation for Type A and Type B

Note: If the Application Mandatory Mask contains mandatory bits in a byte index higher than the current maximum length, a conflict exists and the computation does not begin and the Current Protocol Parameters remain unchanged. For this reason, conflicts need not to be considered in the following description.

The “Protocol Data” of the Current Protocol Parameters is computed as follows, for each protocol parameter data field, in order to reflect the Application’s Mandatory Protocol Data. The first step is to pre-format the parameter data so that the computation of the Current Protocol Parameters can be executed.

For values with numeric representation (e.g. data rates), the following rules apply:

- If an Application requests (i.e. indicated by a Mandatory Mask) a maximum value, and the Current Protocol Parameter requests a different maximum value, then the minimum of these two values is taken as the result.
- If an Application does not request a maximum value, then the current Protocol Parameter is kept.

For the LV structured parameters, the following rules apply:

- If an Application requests an exact value, then this value is taken as the result for the value of L. For the next step of the processing of these parameters, the protocol data and the Mandatory Masks shall be aligned to this length (L). Alignment is accomplished either by truncation at the rightmost byte or by padding with '00's at the rightmost byte.

- If an Application requests a maximum value, and the Current Protocol Parameter requests a different maximum value, then the minimum of these two values is taken as the result for the value of L. For the next step of the processing of these parameters, the value part of the longer LV will be truncated to the length of the resulting L value, keeping the leftmost L bytes.
- If an Application does not request a maximum value, and the Current Protocol Parameter requests a maximum value, then the Application protocol data and the Mandatory Mask shall be aligned starting from the leftmost byte to the length of the Current Protocol Parameters. Alignment is accomplished either by truncation at the rightmost byte or by padding with '00's at the rightmost byte.
- If an Application requests a maximum value, and the Current Protocol Parameters do not request a maximum value, then the Current Protocol Parameters Data and Mandatory Mask shall be aligned starting from the leftmost byte to the length of the Application Protocol Parameters Data. Alignment is accomplished either by truncation at the rightmost byte or by padding with '00's at the rightmost byte.
- If neither an Application nor the Current Protocol Parameter requests a maximum value, then the protocol data and the Mandatory Masks shall be aligned to the maximum length of both. Alignment is accomplished by padding with '00's at the rightmost byte.

For the basic fields and value parts of LV structured fields, the following rules apply:

1. Current Protocol Parameter's Protocol Data is updated with the Application mandatory bits set to 1: "Intermediate Result" [C] = "Current Protocol Data" OR [A], where [A] is a mask forcing bits to 1 and is calculated as [A] = "Application Protocol Data" AND "Application Mandatory Mask".
2. The Intermediate Result is updated with the Application mandatory bits set to 0 [B] to obtain the new Protocol Data. "Current Protocol Parameter's Protocol Data" = "Intermediate Result" [C] AND [B], where [B] = NOT (NOT("Application Protocol Data") AND "Application Mandatory Mask").

Figure 4-1 illustrates the computation of the Current Protocol Parameters for basic fields and the value parts of LV structured fields having their Mandatory Masks set to 'FF', according to the rules given above.

Figure 4-1: Current Protocol Parameters: Protocol Data Computation (Type A/B)

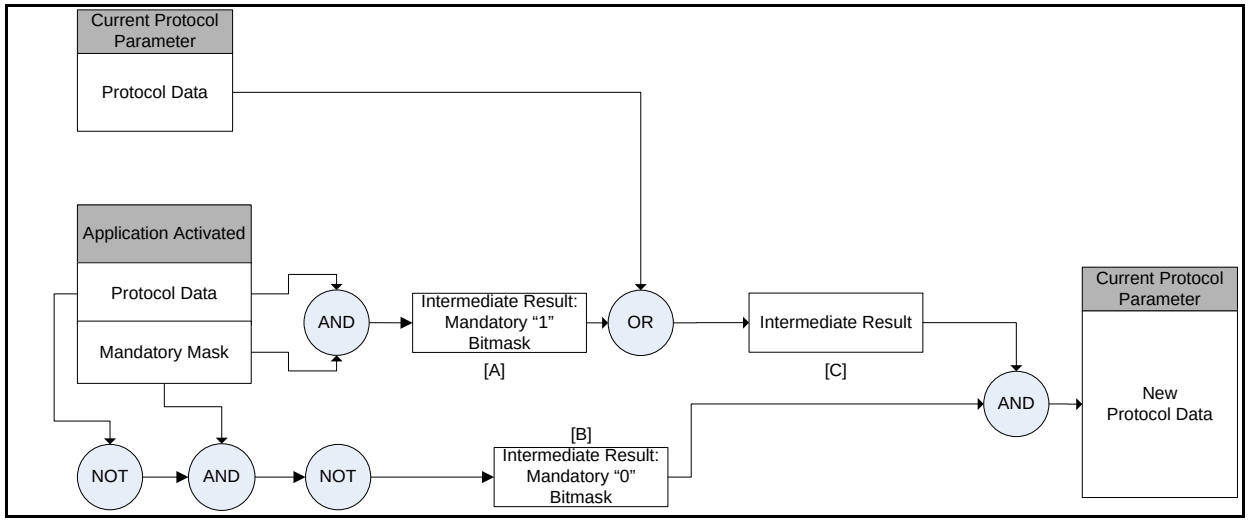
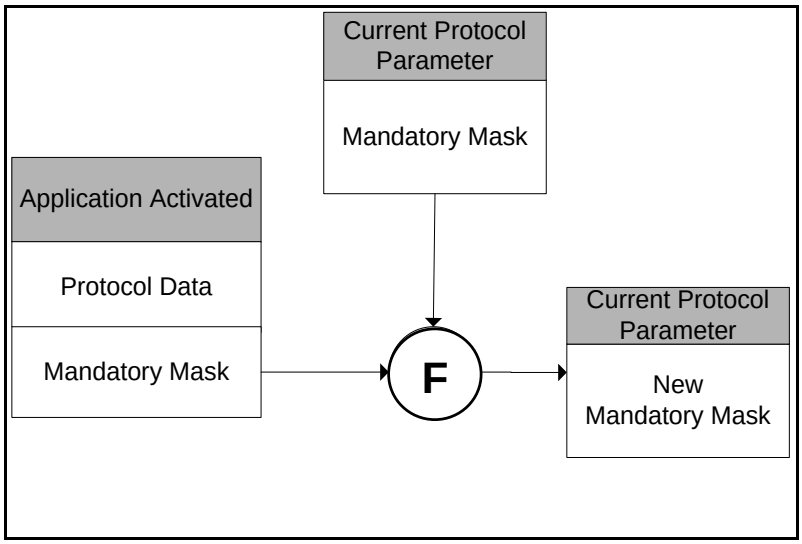


Figure 4-2 illustrates the computation of the Mandatory Mask of the Current Protocol Parameters, using a specific operator **F**.

Figure 4-2: Current Mandatory Mask Computation (Type A/B)



The Mandatory Mask of the Current Protocol Parameters is computed as follows:

- For basic fields and value parts of LV structured fields, the **F** operator is defined as the OR operator (Current Mandatory Mask ← Current Mandatory Mask OR Application Mandatory Mask)
- For length parts of LV structured fields, the **F** operator is defined by Table 4-1.

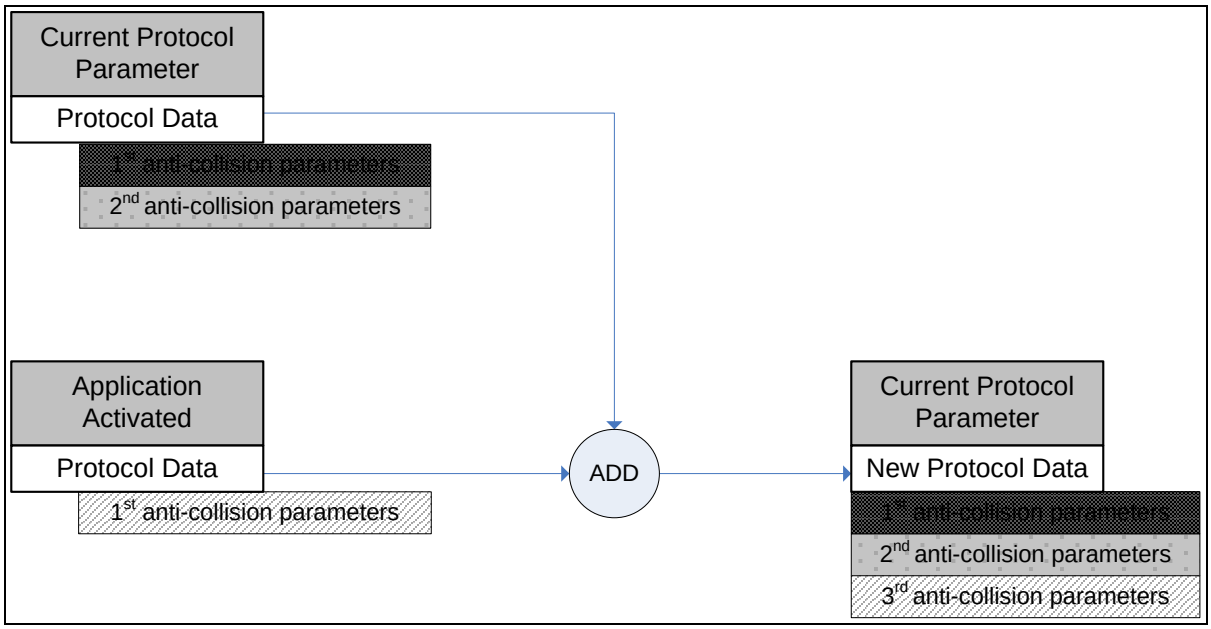
Table 4-1: F Operator in Current Mandatory Mask Computation (Type A/B)

F		**Current Mandatory Mask**		
		'00'	'0F'	'FF'
Application Mandatory Mask	'00'	'00'	'0F'	'FF'
	'0F'	'0F'	'0F'	'0F'
	'FF'	'FF'	'0F'	'FF'

4.4.2.2 Computation on Activation for Type F

For Type F, since the Protocol Parameters may consist of multiple entries of the Protocol Parameters of the activated Applications, the OPEN shall add the Protocol Data of the newly activated Application at the end of the Current Protocol Parameters if there is no conflict. Figure 4-3 illustrates an example of the computation of the Current Protocol Parameters for Type F applications.

Figure 4-3: Current Protocol Parameters: Protocol Data computation (Type F)



4.4.3 Current Protocol Parameter: Full Computation

The Current Protocol Parameters shall be fully recomputed.

1. When an Application is deactivated, the Current Protocol Parameters shall be computed as follows:
   a. The current protocol is set to the initial value, as specified in section 4.4.1.
   b. The current protocol shall be recomputed for each activated Application as specified in section 4.4.2.
2. When the Default Protocol Parameters are updated, the following process applies:
   a. All Applications are temporarily deactivated.
   b. For each of these Applications, the Conflict Detection is launched.
      i. If there is no conflict, the current protocol shall combine the new Application Protocol Data as specified in section 4.4.2.
      ii. If there is a conflict, the Application remains deactivated.
3. When the Protocol Data of one of the activated Applications is updated, the following process applies:
   a. The Application is temporarily deactivated.
   b. The current protocol is computed as for Application deactivation.
   c. The Conflict Detection is launched with the updated Application Protocol Data.
      i. If there is no conflict, the current protocol shall combine the new Application Protocol Data as specified in section 4.4.2.
      ii. If there is a conflict, the Application remains deactivated. The CRS Application may be requested to perform a conflict resolution action.

4.5 Protocol Parameter Conflict Detection Procedure

The conflict detection algorithm shall check that one set of Protocol Parameters does not conflict with another one.

The conflict detection shall be performed in each of the following situations:

- when an Application is activated
- when Protocol Parameters of an activated Application are set or updated
- during conflict resolution to retrieve the Applications that conflict with the one to be activated

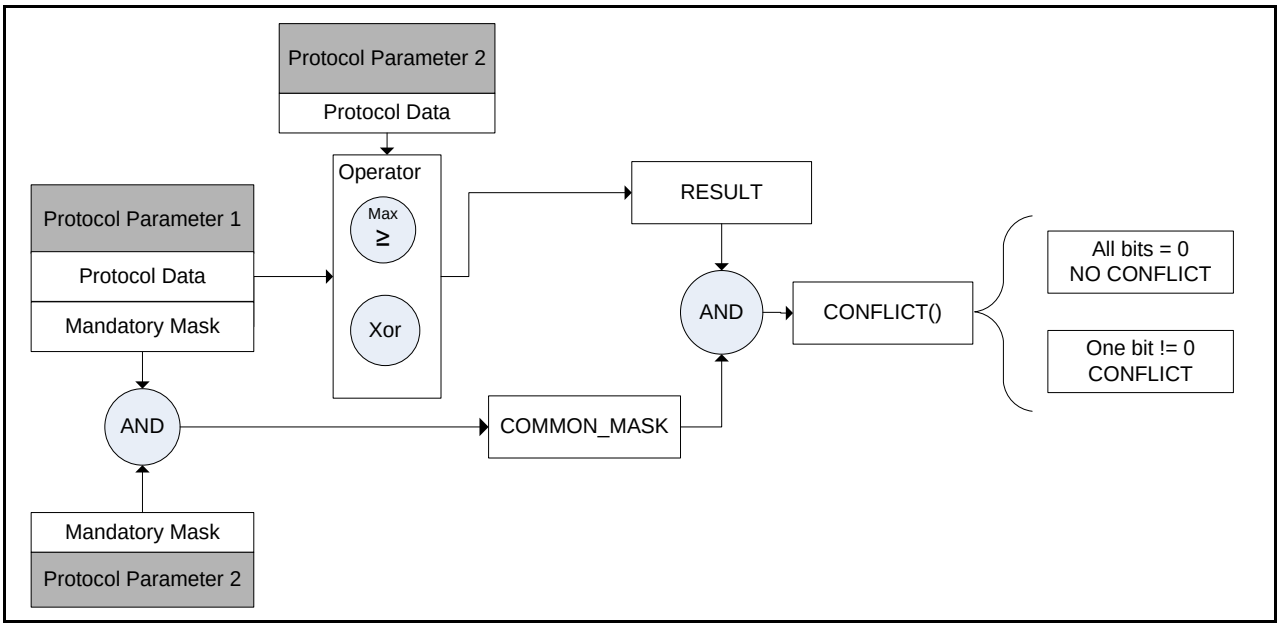
4.5.1 Conflict Detection Procedure for Type A and Type B

Figure 4-4 illustrates the Protocol Parameter Conflict Detection for Type A and Type B between Protocol Parameter 1 and Protocol Parameter 2.

The following assignment applies:

- Protocol Parameter 1 is the Protocol Data of the Application being evaluated upon activation or during conflict resolution.
- Protocol Parameter 2 is the Current Protocol Data or Application participating in the conflict resolution (e.g. current Activated Applications for the given protocol type).

Figure 4-4: Protocol Parameter Conflict Detection (Type A/B)



- For each field of the protocol parameter, the RESULT depends on the operator associated with the protocol data field being evaluated. For length parts of LV structured fields, the operator to be used is indicated in the corresponding Length Operation Indicator field of the Mandatory Mask.
  - For the length part of LV structured fields, a conflict exists if:
    - One of the protocol parameters requests strict length equality (i.e. the Length Operation Indicator is XOR) and the other contains mandatory bits in its mask of a byte index greater than the requested length value

        - One of the protocol parameters requests a MAX length value (i.e. the Length Operation Indicator is MAX) and the other contains mandatory bits in its mask of a byte index greater than the MAX length value
        - Both protocol parameters request strict length equality and request different length values
        - One of the protocol parameters requests a MAX length value lower than the strict length value requested by the other one
        - Both protocol parameters request a different MAX value and the parameter with a higher MAX value contains mandatory bits in its mask of a byte index greater than the smaller MAX value
    - If a length conflict is detected, the conflict detection procedure ends here.
- For length parts of LV structured fields, the RESULT is computed as follows:
    - If the Length Operation Indicator associated with the length part is MAX for any of Protocol Data Parameter 1 or Protocol Data Parameter 2, then the RESULT is '00'.
    - Otherwise, a logical XOR between the value of the Protocol Data Parameter 1 and Protocol Data Parameter 2 produces the value of RESULT.
- For basic fields (or any subparts of basic fields) associated with a MAX conflict detection operator, the RESULT is set to '00'.
- For basic fields (or any subparts of basic fields) associated with an XOR conflict detection operator, and for value parts of LV structured fields, the RESULT is computed as the logical XOR between the value of the Protocol Data Parameter 1 and Protocol Data Parameter 2.
- The COMMON MASK is computed by performing a logical AND between the Mandatory Mask of protocol parameter 1 and the Mandatory Mask of protocol parameter 2 of the same protocol type
- The CONFLICT is identified by applying a logical AND between the RESULT and the COMMON_MASK
- If one single bit of the CONFLICT is set to 1 then protocol parameter 1 is in conflict with protocol parameter 2. Therefore, the Application is conflicting and cannot be set to ACTIVATED at this time.

4.5.2 Conflict Detection Procedure for Type F

For Type F, a conflict exists if any of the following conditions is detected:

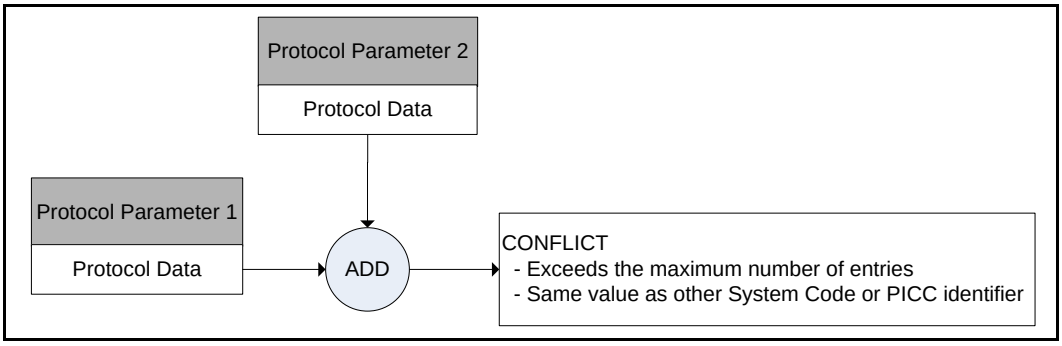
- The value of the System Code parameter or the PICC identifier of the Protocol Parameter Data (see section 4.8) of an application to be changed to ACTIVATED state is identical to a value in the OPEN Protocol Parameter Data of the activated Applications.
- The number of the Type F anti-collision parameters (see section 4.8) of the Application put into the ACTIVATED state plus the current number of already existing entries in the Current Protocol Parameters exceeds the maximum number of entries (see section 4.8) of the Current Protocol Parameters.

Figure 4-5 illustrates the Protocol Parameter Conflict Detection for Type F between Protocol Parameter 1 and Protocol Parameter 2.

The following assignment applies:

- Protocol Parameter 1 is the Protocol Data of the Application being evaluated upon activation or during conflict resolution.
- Protocol Parameter 2 is the Current Protocol Data.

Figure 4-5: Protocol Parameter Conflict Detection (Type F)



4.6 Protocol Parameters for Type A (Card Emulation Mode)

The value part of the Protocol Data Type A TLV (see Table 11-3) when addressed to the OPEN is used to establish or update the default value. It has the coding shown in Table 4-2.

Table 4-2: OPEN – Value Part of "Protocol Data Type A" TLV

<table>
<tr><th>Tag</th><th>Len</th><th colspan="3">Value Description</th></tr>
<tr><td rowspan="11">'A0'</td><td rowspan="11">Var</td><td colspan="3">Protocol Parameter Data</td></tr>
<tr><td>Tag</td><td>Len</td><td colspan="2">Value Description</td></tr>
<tr><td>'80'</td><td>var</td><td colspan="2">Unique Identifier is LV structured</td></tr>
<tr><td></td><td></td><td>Length</td><td>Value Description</td></tr>
<tr><td></td><td></td><td>0, 4, 7, or 10</td><td>Unique Identifier values as defined in [102622]. There is no consistency check with UID size which is coded in ATQA.</td></tr>
<tr><td>'81'</td><td>1</td><td colspan="2">SAK: Preformatted Select AcKnowledge as defined in [14443-3], when the UID is complete</td></tr>
<tr><td>'82'</td><td>2</td><td colspan="2">ATQA: Coding of the ATQA as defined in [102622].</td></tr>
<tr><td>'83'</td><td>1-16</td><td colspan="2">ATS LV structure.</td></tr>
<tr><td></td><td></td><td>Length</td><td>Value Description</td></tr>
<tr><td></td><td></td><td>0-15</td><td>ATS Historical bytes in answer to select as defined in [14443-4].</td></tr>
<tr><td>'84'</td><td>1</td><td colspan="2">FWI, SFGI: Frame waiting time and Start-up frame guard time as defined in [14443-4] for Type A</td></tr>
<tr><td>'85'</td><td>1</td><td colspan="2">CID Support: Support of the Card Identifier field as defined in [14443-4].
'01' ATS indicates "CID Support"
'00' ATS may not indicate "CID Support"</td></tr>
<tr><td>'86'</td><td>3</td><td colspan="2">DATA_RATE_MAX: Maximum data rate supported as defined in [102622]</td></tr>
</table>

Table 4-3 describes the coding of the value part of the Protocol Data Type A TLV (see Table 11-3) requested by an Application.

Table 4-3: Application – Value Part of "Protocol Data Type A" TLV

<table>
<tr><th>Tag</th><th>Len</th><th colspan="6">Description</th></tr>
<tr><td rowspan="11">'A0'</td><td rowspan="11">Var</td><td colspan="6">Protocol Parameter Data</td></tr>
<tr><td></td><td>Tag</td><td>Len</td><td colspan="2">Value Description</td><td>Conflict Detection Operator</td></tr>
<tr><td></td><td>'80'</td><td rowspan="2">var</td><td colspan="2">Unique Identifier: Value is LV structured</td><td></td></tr>
<tr><td></td><td>Length</td><td>Value Description</td><td></td></tr>
<tr><td></td><td>0, 4, 7, or 10</td><td>Unique Identifier values as defined in [102622]. There is no consistency check with UID size is coded in ATQA.</td><td>MAX or XOR</td><td>XOR</td></tr>
<tr><td></td><td>'81'</td><td>1</td><td colspan="2">SAK: Preformatted Select AcKnowledge as defined in [14443-4], when the UID is complete</td><td>XOR</td></tr>
<tr><td></td><td>'82'</td><td>2</td><td colspan="2">ATQA: Coding of the ATQA as defined in [102622]</td><td>XOR</td></tr>
<tr><td></td><td>'83'</td><td rowspan="3">1-16</td><td colspan="2">ATS LV structure</td><td></td></tr>
<tr><td></td><td>Length</td><td>Value Description</td><td></td></tr>
<tr><td></td><td>0-15</td><td>ATS Historical bytes in answer to select as defined in [14443-4]</td><td>MAX or XOR</td><td>XOR</td></tr>
<tr><td></td><td>'84'</td><td>1</td><td colspan="2">FWI, SFGI: Frame waiting time and Start-up frame guard time as defined in [14443-4] for Type A</td><td>MAX</td><td>MAX</td></tr>
<tr><td></td><td>'85'</td><td>1</td><td colspan="2">CID Support: Support of the Card Identifier field as defined in [14443-4].
'01' ATS indicates "CID supported"
'00' ATS indicates "CID not supported"</td><td>XOR</td></tr>
<tr><td></td><td>'86'</td><td>3</td><td colspan="2">DATA_RATE_MAX: Maximum data rate supported as defined in [102622].</td><td>MAX</td><td>MAX</td><td>XOR</td></tr>
<tr><td rowspan="4">'A1'</td><td rowspan="4">Var</td><td colspan="6">Protocol Parameter Mandatory Mask</td></tr>
<tr><td>'80'</td><td>Var</td><td colspan="3">Mandatory Mask of LV UID</td></tr>
<tr><td></td><td></td><td>Length Operation Indicator</td><td>Value Mask</td></tr>
<tr><td></td><td></td><td>'00': Application doesn't care about UID length.
'0F': Application does care about UID length and the XOR operator (strict equality) shall be used.
'FF': Application does care about UID length and the MAX operator shall be used.</td><td>Mask for mandatory bits within the UID value</td></tr>
<tr><td></td><td>'81'</td><td>1</td><td colspan="3">Mask for mandatory bits within SAK</td></tr>
<tr><td></td><td>'82'</td><td>2</td><td colspan="3">Mask for mandatory bits within ATQA.</td></tr>
</table>

<table>
<tr><th>Tag</th><th>Len</th><th colspan="3">Description</th></tr>
<tr><td rowspan="4"></td><td rowspan="4"></td><td>'83'</td><td>Var</td><td>Mandatory Mask of LV ATS
Length Operation Indicator | Value Mask
'00': Application doesn't care about ATS historical bytes length.
'0F': Application does care about ATS historical bytes length and the XOR operator (strict equality) shall be used.
'FF': Application does care about ATS historical bytes length and the MAX operator shall be used.
| Mask for mandatory historical bit in answer to select as defined in [14443-4].</td></tr>
<tr><td>'84'</td><td>1</td><td>Operation Indicator for FWI, SFGI
'00': Application doesn't care about the FWI and SFGI values
'F0': Application requests FWI does not exceed the specified value, but does not care about the SFGI value.
'FF': Application requests that the values for FWI and SFGI do not exceed their specified values.
'0F': Application requests that the value of SFGI does not exceed the specified value and does not care about the value of FWI.</td></tr>
<tr><td>'85'</td><td>1</td><td>Mask for mandatory bits in CID_SUPPORT</td></tr>
<tr><td>'86'</td><td>3</td><td>Operation Indicator for mandatory bits within DATARATE_MAX.
Byte 1: '00': Application does not care about the Maximum bit rate in the PCD to PICC direction
'FF': Application requests that the indicated Maximum bit rate in the PCD to PICC direction is not exceeded
Byte 2: '00': Application does not care about the Maximum bit rate in the PICC to PCD direction
'FF': Application requests that the indicated Maximum bit rate in the PICC to PCD direction is not exceeded
Byte 3: '00': Application does not care about this parameter
'FF': Application requests that value of Byte 3 of DATARATE_MAX is respected</td></tr>
</table>

Each tag within tag 'A1' (Protocol Parameter Mandatory Mask) shall have the same length as its corresponding tag within 'A0' (Protocol Parameter Data). Some implementations may apply default strategies to handle the case where tags within tag 'A1' do not have the same length as their corresponding tags within tag 'A0'. Such strategies remain out of the scope of this document.

4.7 Protocol Parameters for Type B (Card Emulation Mode)

The value part of the Protocol Data Type B TLV (see Table 11-3) when addressed to the OPEN is used to establish or update the default value. It has the coding shown in Table 4-4.

Table 4-4: OPEN – Value Part of "Protocol Data Type B" TLV

<table>
<tr><th>Tag</th><th>Len</th><th colspan="3">Value Description</th></tr>
<tr><td rowspan="10">'A0'</td><td rowspan="10">Var</td><td colspan="3">Protocol Parameter Data</td></tr>
<tr><td>Tag</td><td>Len</td><td colspan="2">Value Description</td></tr>
<tr><td rowspan="3">'80'</td><td rowspan="3">1 or 5</td><td colspan="2">PUPI: Value is LV structure</td></tr>
<tr><td>Length</td><td>Value Description</td></tr>
<tr><td>0 or 4</td><td>Pseudo Unique PICC Identifier. Coding is as specified in [102622]</td></tr>
<tr><td>'81'</td><td>1</td><td colspan="2">AFI: Application Family Identifier as defined in [14443-3].</td></tr>
<tr><td>'82'</td><td>4</td><td colspan="2">ATQB: Coding of the ATQB as defined in [102622]</td></tr>
<tr><td rowspan="3">'83'</td><td rowspan="3">var</td><td colspan="2">Higher layer response in response to ATTRIB as defined in [102622]. Value is a LV structure</td></tr>
<tr><td>Length</td><td>Value Description</td></tr>
<tr><td>var</td><td>Application historical bytes in answer to select as defined in [102622]</td></tr>
<tr><td>'84'</td><td>3</td><td colspan="2">Maximum data rate supported as defined in [102622]</td></tr>
</table>

The OPEN does not enforce the consistency of the AFI and the Application Family defined in section 3.5.

Table 4-5 describes the value part of the Protocol Data Type B TLV (see Table 11-3) requested by an Application.

Table 4-5: Application – Value Part of "Protocol Data Type B"

<table>
<tr><th>Tag</th><th>Len</th><th>Value Description</th></tr>
<tr><td>'A0'</td><td>Var</td><td>Protocol Parameter Data
<table>
<tr><th>Tag</th><th>Len</th><th>Value Description</th><th>Conflict Detection Operator</th></tr>
<tr><td>'80'</td><td>1 or 5</td><td>PUPI: Value is LV structure
<table>
<tr><th>Length</th><th>Value Description</th><th></th></tr>
<tr><td>0 or 4</td><td>Pseudo Unique PICC Identifier. Coding is as specified in [102622]</td><td>XOR</td></tr>
</table></td><td></td></tr>
<tr><td>'81'</td><td>1</td><td>AFI: Application Family Identifier as defined in [14443-3].</td><td>XOR</td></tr>
<tr><td>'82'</td><td>4</td><td>ATQB: Coding of the ATQB as defined in [102622]</td><td>XOR</td></tr>
<tr><td>'83'</td><td>var</td><td>Higher Layer Response in response to ATTRIB as defined in [102622]. Value is a LV structure
<table>
<tr><th>Length</th><th>Value Description</th><th></th></tr>
<tr><td>var</td><td>Application historical bytes in answer to select as defined in [102622]</td><td>MAX XOR</td></tr>
</table></td><td></td></tr>
<tr><td>'84'</td><td>3</td><td>Maximum data rate supported as defined in [102622]</td><td>MAX MAX XOR</td></tr>
</table></td></tr>
<tr><td>'A1'</td><td>Var</td><td>Protocol Parameter Mandatory mask
<table>
<tr><th>Tag</th><th>Len</th><th>Value Description</th></tr>
<tr><td>'80'</td><td>1 or 5</td><td>Mandatory Mask for LV PUPI
<table>
<tr><th>Length Operation Indicator</th><th>Value Mask</th></tr>
<tr><td>'00': Application doesn't care about PUPI Length
'FF': Application does care about PUPI length and the XOR operator shall be used (strict equality).</td><td>Mask for mandatory bits within the PUPI</td></tr>
</table></td></tr>
<tr><td>'81'</td><td>1</td><td>Mask for mandatory bits within the AFI.</td></tr>
<tr><td>'82'</td><td>4</td><td>Mask for mandatory bits in ATQB</td></tr>
<tr><td>'83'</td><td>var</td><td>Mandatory Mask for LV Higher Layer Response
<table>
<tr><th>Length Operation Indicator</th><th>Value Mask</th></tr>
<tr><td>'00': Application doesn't care about Higher Layer Response Length.
'FF': Application does care about Higher Layer Response Length and the MAX operator shall be used.</td><td>Mask for mandatory bits in the Higher layer response.</td></tr>
</table></td></tr>
</table></td></tr>
</table>

Tag	Len	Value Description		
		'84'	3	Operation Indicator for mandatory bits within DATARATE_MAX. Byte 1: '00': Application does not care about the Maximum bit rate in the PCD to PICC direction 'FF': Application requests that the indicated Maximum bit rate in the PCD to PICC direction is not exceeded Byte 2: '00': Application does not care about the Maximum bit rate in the PICC to PCD direction 'FF': Application requests that the indicated Maximum bit rate in the PICC to PCD direction is not exceeded Byte 3: '00': Application does not care about this parameter 'FF': Application requests that value of Byte 3 of DATARATE_MAX is respected

Each tag within tag 'A1' (Protocol Parameter Mandatory Mask) shall have the same length as its corresponding tag within 'A0' (Protocol Parameter Data). Some implementations may apply default strategies to handle the case where tags within tag 'A1' do not have the same length as their corresponding tags within tag 'A0'. Such strategies remain out of the scope of this document.

4.8 Protocol Parameters for Type F (Card Emulation Mode)

The value part of the Protocol Data Type F TLV (see Table 11-3) when addressed to the OPEN is used to set or update the maximum number of anti-collision parameter entries. It has the coding shown in Table 4-6.

Table 4-6: OPEN – Value Part of "Protocol Data Type F" TLV

Tag	Len	Value Description		
'A0'	Var	Protocol Parameter Data		
		Tag	**Len**	**Value Description**
		'80'	1	Maximum Number of anti-collision parameter entries

Table 4-7 describes the value part of the Protocol Data Type F TLV (see Table 11-3) requested by an Application.

Table 4-7: Application – Value Part of "Protocol Data Type F" TLV

Tag	Len	Value Description				
'A0'	Var	Protocol Parameter Data				
		Tag	**Len**	**Value Description**		
		'A0'	24	Type F anti-collision parameters entry		
				Tag	**Len**	**Value Description**
				'80'	2	System Code: The System Code is an application specific code used during anti-collision
				'81'	8	PICC Identifier: The PICC identifier is used by the reader/writer to identify the "File System" associated with the System Code within a Type F card application.
				'82'	8	Response Time Descriptor: The Response Time Descriptor is used to calculate the response time for the commands defined in [JIS6319-4].
		…	...	...		
		'A0'	24	Type F anti-collision parameters entry		

Several occurrences of tag 'A0' (Type F anti-collision parameters entry) may be present to specify different anti-collision parameter entries. The number of such entries is limited to the maximum number specified at OPEN level (see Table 4-6).

4.9 Contactless Protocol Parameters Profiles

Contactless Protocol Parameters Profiles are introduced to facilitate initialization of the Contactless Application during instantiation or personalization of the Application. Instead of providing the values for each Contactless Protocol Parameter for the Application, a profile is provided to identify a collection of defined parameter values.

Contactless Protocol Parameters Profiles belong to the OPEN.

For initialization and update of profiles, the INSTALL [for registry update] command shall be issued to the ISD without an AID in the command data field. Within the command data field for each profile,

- Protocol Data shall be provided as described in Table 11-3 and,
- Contactless Protocol Parameters Profile shall be present and the data structure shall be as defined in section 11.2.2. Note that for each protocol type, only one Protocol Parameter Profile can be set in the same INSTALL [for registry update] command.

Each Contactless Protocol Parameters Profile shall provide the Mandatory Mask for each Protocol that it supports. The “Current Protocol Parameter” computation and conflict detection shall take into account the Contactless Protocol Parameter Profile, if present, for each Application. If a Contactless Protocol Parameters Profile is updated, then the new Contactless Protocol Parameter values shall be applied to each Application referencing that Profile the next time that conflict detection or computation of current protocol parameters occurs due to an attempt to activate that Application. Therefore, an Application referencing this profile will not use the new parameter values until the Application is deactivated and re-activated. Contactless Protocol Parameter Profiles cannot be deleted.

5 Communication Interface Access Configuration

5.1 Introduction, Overview, and Rationale

This section describes mechanisms that allow configuration of the access of an external entity to an application on the card for each communication interface. These mechanisms are provided in order to support various deployment scenarios where the access to a Standalone Application or to an Application Group within the same Security Domain hierarchy depends on the communication interface. In certain scenarios, a Standalone Application or an Application Group shall be available on a particular interface by default without providing specific installation parameters to the INSTALL [for install] command. In other scenarios it is essential that a particular Standalone Application or Application Group within the same Security Domain hierarchy are not available on a particular interface (e.g. the SIM/USIM/ISIM application on a telecommunication card should not be accessible on the contactless interface).

This section introduces three new System Install Parameters:

Communication Interface Access Restriction

This parameter may be provided for a Security Domain. It defines the set of communication interfaces that an Application may use.

Communication Interface Access Default

This parameter may be provided for a Security Domain. It defines the accessibility configuration of an Application which is installed under the hierarchy of the Security Domain and that does not have an Application specific “Communication Interface Access for Instance” parameter.

Communication Interface Access for Instance

This parameter may be provided for an Application (or a Security Domain). If supplied, this parameter defines the accessibility configuration of the Application. If missing, the default Communication Interface Access configuration of the associated Security Domain will be used.

The coding and the detailed rules associated with these parameters are explained in the following sections.

The Communication Interface Access Configuration mechanism is intended to configure the interface access of applications at the time of their installation. It is not intended to be used for the dynamic reconfiguration of the interface access and should be used with caution. In deployment scenarios using Application Groups, a change of the interface access of the Head Applications will affect the Contactless Activation State of Member Applications and the accessibility of the Head Application through the API.

Applications operating in the Java Card™ runtime environment may use `APDU.getProtocol()` to determine over which communication interface commands are being received and also which protocol type is being used.

5.2 Communication Interface Access Parameters

The value part of the Communication Interface Access Configuration TLV is coded as follows:

Table 5-1: Value Part of "Communication Interface Access Configuration" TLV

Length	Value Description				
Var	Communication Interface Access Configuration				
	Tag	**Length**	**Value Description**	**SD**	**Non-SD Application**
	'80'	1	Communication Interface Access Restriction	Optional	Not Present
	'81'	1	Communication Interface Access Default	Optional	Not Present
	'82'	1	Communication Interface Access per Instance	Optional	Optional

Tags '80', '81', and '82' in the table above shall indicate the set of communication interfaces that may be accessed. Communication Interface Identifiers shall be coded according to Table 5-2.

Table 5-2: Communication Interface Identifier

b8	b7	b6	b5	b4	b3	b2	b1	Communication Interface
1	–	–	–	–	–	–	–	Contact-based communication (e.g. ISO/IEC 7816) is supported.
–	1	–	–	–	–	–	–	Proximity-based communication (e.g. ISO/IEC 14443) is supported.
–	–	x	x	x	x	x	x	RFU

5.3 Security Domain Settings

The Security Domain installation parameter "Communication Interface Access Restriction" (CIA Restriction) defines the set of communication interfaces that an Application may use. If this parameter is not provided at the time of installation, the value shall be initialized as per the value of the associated Security Domain. The value of this parameter may be updated using the INSTALL [for registry update] command. The initial value of the "Communication Interface Access Restriction" parameter of the Issuer Security Domain is platform-dependent.

The Security Domain installation parameter "Communication Interface Access Default" (CIA Default) defines the set of communication interfaces that an Application will use by default unless otherwise specified at installation. The value of this parameter shall not be less restrictive than the "Communication Interface Access Restriction" of this Security Domain, otherwise an error status code shall be returned. If this parameter is not provided at the time of installation, the value shall be initialized according to the value of the "Communication Interface Access Restriction" of this Security Domain. For the Issuer Security Domain, the initial value of the "Communication Interface Access Default" parameter is platform-dependent. The value of this parameter may be updated using the INSTALL [for registry update] command.

It is recommended that the CIA Restriction and CIA Default always be updated together. When updating the CIA Restriction, if a conflict is detected between CIA Restriction and CIA Default (i.e. the CIA Default is less restrictive than the CIA Restriction), the card may reject the update or may update the CIA Default to the new value of CIA Restriction.

The capabilities of the Security Domain must be compatible with, i.e. not weaken the restrictions set up by its associated Security Domain. Restrictions set up by this Security Domain only apply to associated Applications.

5.4 Application Instance Settings

The parameter “Communication Interface Access per Instance” defines the set of communication interfaces that an Application is able to use.

When initializing or updating the value of this parameter, the OPEN shall check that the value of the parameter is not less restrictive than the “Communication Interface Access Restriction” of its associated Security Domain. If this is not the case, an error status code shall be returned.

The value of this parameter may be updated using the INSTALL [for registry update] command. The value of this parameter for the ISD is platform-dependent.

If the “Communication Interface Access per Instance” parameter is not provided for a new application at installation time, it shall be set up according to the value of the “Communication Interface Access Default” parameter of its associated Security Domain.

If proximity-based communications (i.e. accessibility through the contactless interface) are disabled for an Application, the OPEN shall attempt transitioning this Application to the DEACTIVATED state (see section 8.1, Contactless Activation State).

5.5 Rules for Extradition

In the case of extradition, the values of these parameters are not modified. NOTE: Therefore, the “Communication Interface Access per Instance” parameter of an Application may be in conflict with the parameter “Communication Interface Access Restriction” of the associated Security Domain after Extradition. Any updates following the extradition shall follow the restrictions of the newly associated Security Domain.

6 Application Selection

6.1 Reset Scenarios with Multiple Active Interfaces

If the runtime environment is based on the Java Card technology, the card shall conform to the behavior described in JCRE v3 Classic Edition [JCRE CE] with respect to a reset occurring on the contactless interface when multiple interfaces are active at the same time.

6.2 Application Selection Priority

The Application selection priority is based on the position of the Application's entry in the GlobalPlatform Registry. Additionally, for selection over the Contactless interface, the Volatile Priority, as defined below, shall be considered. The Volatile Priority may be assigned by the end user to an Application so that the Application shall temporarily receive the highest priority.

6.2.1 GlobalPlatform Registry Order

The GlobalPlatform Registry maintains a list of registered applications whose order is used by the selection by AID mechanism (and to order the response data returned by the GET STATUS command). This is a persistent list that can be reordered. The CRS API provides a method allowing to promote an Application to the first position or demote an Application to the last position in the GlobalPlatform Registry (see method `GPCLRegistryEntry.setPartialSelectionOrder()`). The CRS Application shall use this method to reorder the GlobalPlatform Registry list based on the user priority set up. For instance, if the user wants to set the priority of an Application to the third position then the CRS Application shall invoke this method to put this Application into the first position, then invoke this method to put the Application with priority 2 into the first position, and then again invoke this method to put the Application with priority 1 into the first position. When prioritizing an Application Group, the CRS Application shall only invoke this method for the Head Application. When a Head Application is moved, all Member Applications are moved. The relative order, within the group, of the Head Application and its Member Applications is not affected.

For example, let's consider a Head Application H, member Applications M1 and M2, and other Applications A1, A2, and A3. If the current order of the GlobalPlatform registry is { A1, M1, H, A2, M2, A3 }, then a request to give H the highest priority will reorder the GlobalPlatform registry as follows: { M1, H, M2, A1, A2, A3 }. Similarly, a request to give H the lowest priority will reorder the GlobalPlatform as follows: { A1, A2, A3, M1, H, M2 }.

6.2.2 Volatile Priority over the Contactless Interface

The Volatile Priority may be assigned to an Application Group or a Standalone Application that shall be considered first upon explicit or implicit selection.

An Application cannot be assigned the Volatile Priority if it cannot be activated (e.g. due to conflicts in RF parameters or because it is in the NON ACTIVATABLE state). An application in the DEACTIVATED state that is assigned the Volatile Priority is temporarily transitioned to the ACTIVATED state (see section 8.1 for details on contactless activation states). No notification of this temporary activation shall be sent (see section 3.10).

If the Volatile Priority is assigned to a Head Application, the Volatile Priority is assigned to the Head Application and all corresponding Member Applications in the same order as in the GlobalPlatform Registry. A Member Application cannot be assigned the Volatile Priority directly.

An Application (or Application Group) may be assigned the Volatile Priority even if the Volatile Priority is already assigned. In this case, the implementation shall first check that the Application can be activated and if so, the Volatile Priority shall be first reset before the Application gets the Volatile Priority. The behavior in the case the same Application already has and is assigned the Volatile Priority again remains implementation specific, in particular with respect to notifications.

The Volatile Priority may be assigned by the Application having the Contactless Activation privilege, or may be requested for itself by an Application having the Contactless Self-Activation Privilege. The Volatile Priority may also be requested for itself by an Application without any of these privileges, in which case the OPEN shall submit the request to the CRS Application (using the `CRSApplication.processCLRequest()` method; see section 3.9.1).

The CRS API provides the capability to assign and reset the Volatile Priority. The Volatile Priority is also discarded upon card reset or power-on.

When the Volatile Priority is reset, then any Application currently having the Volatile Priority shall have its contactless activation state reset to its previous value (i.e. before being assigned the Volatile Priority). No notification shall be sent of this change (if any) of the contactless activation state (see section 3.10).

The Volatile Priority may be reset by any Application currently having the Volatile Priority (possibly by a Member Application whose group was assigned the Volatile Priority), by the Application having the Contactless Activation privilege, and by any CREL Application referenced by an Application currently having the Volatile Priority.

In addition, when an Application currently has the Volatile Priority and it is explicitly transitioned to the DEACTIVATED state, then the Volatile Priority shall be reset.

6.3 Explicit and Implicit Selection over the Contactless Interface

6.3.1 Selection for APDU Based Applications

This section describes Explicit and Implicit Selection mechanisms over the Contactless Interface. In particular, it extends the Implicit Selection capabilities of [GPCS] with the addition of Implicit Selection by APDU pattern recognition and Assigned Protocol type on the Basic Logical Channel.

Card enters the RF field:

Upon explicit or implicit selection over the contactless interface, the OPEN first looks for the Applications assigned the Volatile Priority. If no Applications match, the OPEN parses the GlobalPlatform Registry to locate the Application.

The OPEN shall check that the contactless communication interface is switched on.

After card activation occurs according to [14443-3] or [JIS6319-4], the following rules shall be applied for each incoming command until an Application is eventually selected:

1. If the command is a SELECT [by Name] command:
   a. The OPEN locates the targeted Application according to the selection rules defined in [GPCS] (section 6.4).
   b. The located Application becomes a valid candidate for selection if it is ACTIVATED, is selectable according to [GPCS], and is configured to access the contactless interface (see Chapter 5).
   c. If the Application is not a valid candidate, the OPEN continues to look for another valid candidate.
   d. The OPEN shall select the Application. If the Application refuses the selection, the OPEN locates the next Application matching the given AID and restarts at step 1.b until the end of the Application list is reached.
2. Otherwise, if the command is not a SELECT [by Name] command and is received on the Basic Logical Channel, implicit selection by Recognition Algorithm is attempted as follows:
   a. The OPEN parses the list of applications to look for an application that:
      i. Is selectable according to [GPCS],
      ii. Is ACTIVATED and is configured to access the contactless interface (see Chapter 5),
      iii. The Recognition Algorithm succeeds on the received command (see sections 6.5 and 6.6).
   b. The OPEN shall select the Application. If the Application refuses the selection, the OPEN restarts at step 2.a until the end of the Application list.
3. Finally, if steps 1 and 2 have not identified a valid candidate, then:
   a. The OPEN searches for an Application that is a candidate for implicit selection on the contactless interface, as specified by tag 'CF' Implicit Selection Parameter described in section 11.1.7 of [GPCS] or, if no such applet can be found, the applet having the Card Reset privilege.
   b. The located Application becomes a valid candidate for selection if it is ACTIVATED, is selectable according to [GPCS], and is configured to access the contactless interface (see Chapter 5).
   c. If the Application is not a valid candidate, no Application is selected and the channel remains open.
   d. Otherwise, the OPEN shall select the Application. If the Application refuses the selection, no Application is selected and the channel remains open.

If no Application was selected, the above rules shall be applied to the next incoming command.

Card leaves the RF field:

When card leaves the RF field or is deactivated according to [14443-4] and the power is on, all Applications selected on the contactless interface are deselected and logical channels closed.

6.3.2 Selection for Non-APDU Based Applications

This section describes Implicit Selection mechanisms over the Contactless Interface for non-APDU based applications. Non-APDU based applications are expected to use a non-APDU communication channel such as the one enabled by the API specified in ETSI TS 102 705 [102705]. For completeness, [102705] is referenced in the procedures in this section; however, other implementations may be used.

6.3.2.1 Type A and Type B applications

For Type A and Type B applications, the following procedure shall apply:

Card enters the RF field:

The OPEN shall check that the contactless communication interface is switched on.

After card activation occurs according to [14443-3], the following rules shall be applied for each incoming command until an Application is eventually selected:

1. Implicit selection by Recognition Algorithm is attempted as described in step 2 of section 6.3.1. If a valid candidate is identified, the OPEN shall select and enable the candidate Application to receive commands over the API specified in [102705].
2. Otherwise, the OPEN shall look for an Application that shall be implicitly selected as described in step 3 of section 6.3.1. If a valid candidate is identified, the OPEN shall select and enable the Application to receive commands over the API specified in [102705].
3. Finally, if no valid candidate was identified, then no Application is selected, the command is discarded, and the OPEN waits for another command.

Card leaves the RF field:

When the card leaves the RF field and the power is on, the current non-APDU based applications, if any, shall be deselected according to the API specified in [102705].

6.3.2.2 Type F applications

For Type F applications, the following procedure shall apply:

Card enters the RF field:

The OPEN shall check that the contactless communication interface is switched on.

The Mode Flag parameter (described in section 6.3.3) of each Type F anti-collision parameters entry (consisting of System Code, PICC identifier, and Response Time Descriptor, as described in section 4.8) shall be initialized, i.e. set to TRUE.

The following rules shall be applied for each incoming command until an Application is eventually selected:

1. If the command is an anti-collision command, then the procedure described in section 6.3.3 shall apply.
2. Otherwise, implicit selection by Recognition Algorithm is attempted as follows:

   The OPEN parses the list of applications to look for an application that:

   i. Is selectable according to [GPCS],

   ii. Is ACTIVATED and is configured to access the contactless interface (see Chapter 5),

   iii. The Recognition Algorithm succeeds on the received command (see sections 6.5.2 and 6.6).
3. If step 1 or step 2 has identified a potential candidate, then the OPEN shall select and enable the Application to receive commands over the API specified in [102705].
4. Finally, if steps 1 and 2 have not identified a valid candidate or the Application refuses the selection in step 3, then no Application is selected, the command is discarded, and the OPEN waits for another command.

Card leaves the RF field:

When the card leaves the RF field and the power is on, the current non-APDU based Type F applications, if any, shall be deselected according to the API specified in [102705].

6.3.3 Anti-collision Command Handling for Non-APDU Based Type F Applications

If the card receives an anti-collision request via CLT mode according to [102613] and [102622], the anti-collision request shall be evaluated and shall be answered with an anti-collision response.

An anti-collision request consists of a command byte set to '00', System code, Request code, and Time slot number (see [JIS6319-4]).

To process the anti-collision request the Current Protocol Parameters (see section 4.8) shall be used. As discussed in section 4.4.2.2, only applications in the ACTIVATED state will have one or more entries in the Current Protocol Parameters. A transient Mode Flag shall be maintained for each anti-collision parameters entry (consisting of System Code, PICC identifier, and Response Time Descriptor) in the Current Protocol Parameters. The Mode Flag will be used to determine whether the anti-collision response is returned in response to the anti-collision request command as described below.

The Mode Flag may be modified by Type F applications during a contactless transaction using the `GPCLRegistryEntry.setInfo` method with the constant `INFO_PROTOCOL_TYPE_F_MODE_FLAG`. The modification of this Mode Flag shall be taken into account immediately by the anti-collision process.

1. The OPEN shall determine whether the requested System code contained in the anti-collision request matches a System code in each Current Protocol Parameters entry starting from the first entry. The OPEN shall use the first match for the anti-collision and stop the checking process after a match is found.
   - If the received System code is equal to 'FFFF', any System code matches the requested one.
   - If the first byte of the requested System Code is equal to 'FF', a System Code matches the requested one if the second byte is equal to the second byte of the requested one.
   - If the second byte of the requested System Code is equal to 'FF', a System Code matches the requested one if the first byte is equal to the first byte of the requested one.
   - Otherwise, a System code matches the requested one if the 2-byte value is equal to the requested value.
2. If an application having a matching System Code is found, the corresponding Mode Flag shall be checked.

   If the Mode Flag is True, then:
   - The anti-collision response shall be sent to the CLF in a CLT frame. The anti-collision response consists of a command byte set to '01', PICC identifier, Response time descriptor, and Request data if necessary (see [JIS6319-4]).
   - The Mode Flag of all Type F anti-collision parameters entries (of all Applications) shall be set to True.
   - The OPEN shall keep the Application having the matched System code as potential candidate for selection.

   If the Mode Flag is False, then:
   - The CLT frame with an empty data field shall be sent.
3. If no application having a matching System code is found, the CLT frame with an empty data field shall be sent.
4. After the CLT frame is sent to the CLF, the OPEN shall return to the receive mode in which it can receive another anti-collision request command or other command.

6.4 Continuous Processing

An application may be installed with a continuous processing indicator. When such an Application is selected on the contactless interface, the counter begins on the time-out and any command received on the contact interface is postponed until the contactless session ends or the time-out defined below expires and the current contactless APDU processing is completed. This applies to all protocols used on the contactless interface.

Table 6-1 describes the value part of the “Continuous Processing” tag (see Table 11-3) applicable to the OPEN.

Table 6-1: OPEN – Value Part of “Continuous Processing” TLV

Length	Value Description
2	Continuous processing timeout. Unsigned short value: Time in ms (default: 0 ms)

In the context of the OPEN, the value is the continuous processing timeout value. Note: Due to card internal inaccuracy, deviations of up to 10 ms may occur.

Table 6-2 describes the value part of the “Continuous Processing” tag (see Table 11-3) requested by an Application.

Table 6-2: Application – Value Part of “Continuous Processing” TLV

Length	Value Description
1	'01': Continuous processing is disabled (default if TLV is missing) '02': Continuous processing when this Application is selected on the contactless interface is enabled

This TLV controls the continuous processing requirements of the Application.

6.5 Recognition Algorithm

6.5.1 Recognition Algorithm for APDU Based Applications

The Recognition Algorithm provides the capability to identify and select, on the Basic Logical Channel of the contactless interface, a legacy Contactless Application not supporting the SELECT by AID command. The Recognition Algorithm does not apply for supplementary logical channels.

Table 6-3: Recognition Algorithm

Length	Value Description	
Var	**Data**	**Coding**
	Algorithm Identifier	'01' String recognition '02' Bitmap recognition
	Algorithm Parameter	• For string recognition: o Offset: 2 bytes o Pattern: X bytes The pattern associated with an Application is a string of X bytes to be compared against the APDU received on the contactless interface (on the Basic Logical Channel) starting from a given offset. Recognition is successful if there is a bitwise match (XOR) of the pattern. If the pattern exceeds the end of the APDU, the recognition is considered not successful. • For bitmap recognition: o Reference Data: X bytes o Mask: X bytes Reference Data and Mask have the same length. Recognition is successful if (DATAIN AND Mask) XOR Reference Data is equal to 0, where DATAIN is the APDU received on the contactless interface (on the Basic Logical Channel). If the length of the APDU does not match the length of the reference data/mask, the recognition is considered not successful.

6.5.2 Recognition Algorithm for Non-APDU Based Applications

6.5.2.1 Type A and Type B Applications

For non-APDU based Type A and Type B applications, the Recognition Algorithm defined in section 6.5.1 applies.

6.5.2.2 Type F Applications

For non-APDU based Type F applications, the Recognition Algorithm provides the capability to identify an application based on a non-anti-collision command recognized by the OPEN. In case the Recognition Algorithm fails, an empty EVT_SEND_DATA event as defined in [102622] shall be sent to the CLF, ensuring that the CLF is able to receive subsequent commands from the RF.

Table 6-4: Recognition Algorithm (Type F)

Length	Value Description	
Var	**Data**	**Coding**
	Algorithm Identifier	'03' ID recognition
	Algorithm Parameter	For ID Recognition: • Offset: 2 bytes • Pattern: 8 bytes • Mask: 8 bytes Recognition is successful if (DATAIN AND Mask) XOR Pattern is equal to 0, where DATAIN is 8 bytes data starting from the Offset in the command received on the contactless interface. The Offset should be set to '02' to indicate the start of the PICC identifier of the command unless specified otherwise. The OPEN shall implicitly increment this Offset by 1 if the upper 4 bits of the second byte (command code) in the received command is equal to (1100)b or (1101)b. The Mask should be set to '0FFFFFFFFFFFFFFF' unless specified otherwise. A Type F command format consists of 1 byte length, 1 or 2 bytes command code, and n bytes command parameters (see [JIS6319-4]).

6.6 Assigned Protocols for Implicit Selection

An Application can configure its ability to be implicitly selected (as specified above) on the contactless interface, using any of the protocol types A, B, or F. This mechanism is provided to support Applications that are not selected by the PCD with an explicit SELECT [by Name] command, and that support only a subset of these protocols.

The Assigned Protocols for Implicit Selection TLV defines the protocol types that are supported by an Application.

- If this TLV is not provided or is provided and is zero length, the Application is a valid candidate for implicit selection through any of these protocol types.
- If this TLV has nonzero length then the protocol types that are listed are usable for the application. The OPEN shall not perform implicit selection for any protocol type that is not listed.

Table 6-5: Value Part of "Assigned Protocol for Implicit Selection" TLV

Length	Value Description
0-3	List of protocol types assigned for the application: '81': Type A '82': Type B '84': Type F

6.7 Attempt to Select a Deactivated or Non Activatable Application

When an applet cannot successfully be selected because it is DEACTIVATED or NON_ACTIVATABLE, the selection process shall not return an error but shall try to find another partial AID match in the registry. If no other match can be found, then:

- If the command was sent to select the first or only occurrence (matching the search criteria), then the command shall be forwarded to the currently selected applet (on that logical channel); if there is no currently selected applet (on that logical channel), then a status word of '6999' shall be returned.
- If the command was sent to select the next matching occurrence (matching the search criteria), then a status word of '6A82' (application not found) shall be returned.

7 Contactless Privilege

7.1 Contactless Activation Privilege

The Contactless Activation privilege identifies the CRS Application. There shall be at most one Application in the secure element that is assigned this Privilege. This rule shall be enforced by the OPEN. In order to assign this privilege to another Application, the privilege shall first be removed from the current Application that is assigned this privilege or the Application with this privilege shall be deleted.

This Privilege allows:

- The Activation/Deactivation of Applications on the Contactless Interface
- The update of the Selection Priority
  - Manage the Volatile Priority
  - Reorder the GlobalPlatform Registry
- Notification by the OPEN when:
  - An application is INSTALLED, LOCKED, unlocked or deleted
  - The Activation State of an Application is changed between NON_ACTIVATABLE, ACTIVATED, or DEACTIVATED.
  - One of the Application's contactless registry parameters is updated.

7.2 Contactless Self-Activation Privilege

The Contactless Self-Activation Privilege allows an Application to activate itself (i.e. transition to the ACTIVATED state) or request Volatile Priority for itself without a prior request to the CRS Application. This Privilege can be assigned to any Application. Activation and/or Volatile Priority request fails if it would result in an RF conflict with the set of currently activated Applications. The CRS Application and associated CREL Applications, if any, are notified of the successful activation and/or Volatile Priority change. According to section 6.2.2, when Volatile Priority is assigned to a Group Head Application, it is also assigned to its associated Member Applications.

The Application with the Contactless Activation Privilege can also transition an Application with the Contactless Self-Activation Privilege to the ACTIVATED or DEACTIVATED states, or give it the Volatile Priority.

7.3 Privilege Coding

Table 7-1 updates and replaces Table 11-9, Privileges (Byte 3), of [GPCS].

Table 7-1: Privileges

b8	b7	b6	b5	b4	b3	b2	b1	Meaning	Privilege Number
1	–	–	–	–	–	–	–	Receipt Generation	16
–	1	–	–	–	–	–	–	Ciphered Load File Data Block	17
–	**–**	**1**	**–**	**–**	**–**	**–**	**–**	**Contactless Activation**	**18**
–	**–**	**–**	**1**	**–**	**–**	**–**	**–**	**Contactless Self-Activation**	**19**
–	–	–	–	X	X	X	X	RFU	–

8 Application Availability on the Contactless Interface

The ability of an Application to communicate through the contactless interface might be restricted in general, upon installation or registry update (see Chapter 5). The contactless interface might also be fully disabled as described in section 8.4 and section 3.11.4 (SET STATUS).

When no such restrictions exist, the ability of an Application to communicate through the contactless interface may still be temporarily activated or deactivated. To reflect this, the following per-application Contactless Activation States are defined: ACTIVATED, DEACTIVATED, and NON_ACTIVATABLE.

Interactions between Contactless Activation State, Communication Interface Access Configuration, and Contactless Interface Availability are described in section 5.4 and section 8.4.

8.1 Contactless Activation State

The Contactless Activation State exists concurrently with the existing Application Lifecycle State. The encoding of these new States is shown in Table 8-1.

Table 8-1: Contactless Activation State Byte Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	–	–	–	–	–	–	1	ACTIVATED
0	–	–	–	–	–	–	0	DEACTIVATED
1	–	–	–	–	–	–	0	NON_ACTIVATABLE
–	X	X	X	X	X	X	–	RFU (Reserved for future use)

The Contactless Activation State byte defined above is encoded independently of the Application Lifecycle State byte. The value of the Contactless Activation State byte can be retrieved independently using the GET STATUS command on the CRS Application or the associated Security Domain.

These states shall be interpreted as follows:

- An Application currently in the ACTIVATED state is able to communicate through the contactless interface.
- Conversely, an Application currently in the DEACTIVATED state is not able to communicate through the contactless interface.
- An Application in the NON_ACTIVATABLE state is implicitly DEACTIVATED, and due to some internal reason known by the Application or its provider (e.g. a possible attempt of fraudulent use) cannot be ACTIVATED. Any attempt to activate an Application that is currently in the NON_ACTIVATABLE state, shall fail.

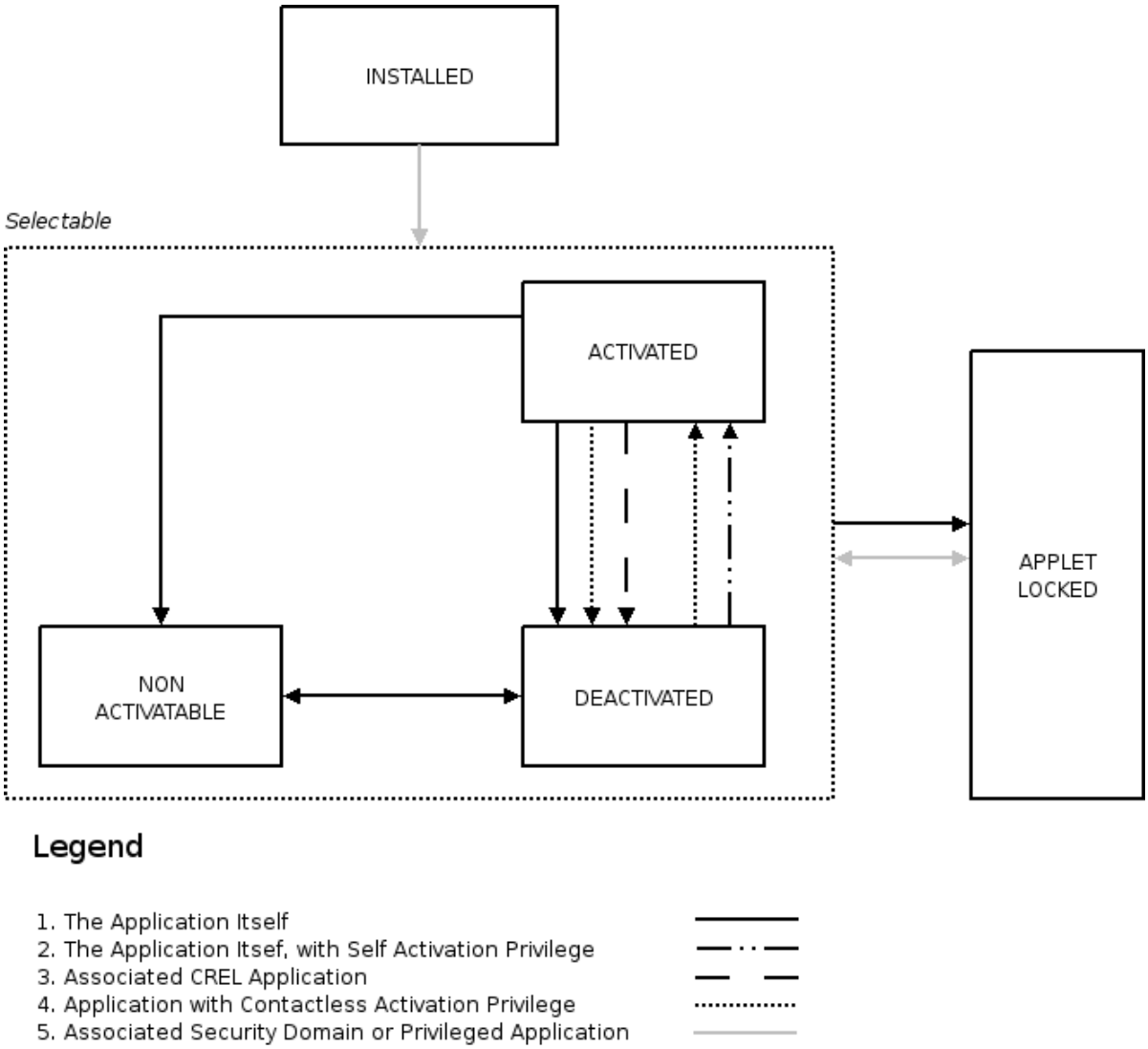
Transitions between these states abide by the following rules:

- An Application is able to transition itself into the DEACTIVATED state.
- An Application cannot transition itself into the ACTIVATED state, except if it was granted the Contactless Self-Activation Privilege.
- A unique Application may be granted the Contactless Activation privilege, allowing it to activate/deactivate any Application on the card.

- Applications registered as listeners for contactless events occurring to an Application (see section 3.8.2), are allowed to deactivate that particular Application.
- At any moment, an Application may transition itself into the NON_ACTIVATABLE state, therefore indicating to the OPEN that it is both deactivated and not in a suitable internal state to become ACTIVATED on the contactless interface.
- An Application is able to transition itself from the NON_ACTIVATABLE state to the DEACTIVATED state, hence indicating to the OPEN that it is in a suitable state to become ACTIVATED again, whenever this request would be made.

Figure 8-1 summarizes the rules described above.

Figure 8-1: Contactless Activation States



In addition, the following rules apply, which are not illustrated in Figure 8-1 above:

- When an Application belongs to an Application Group (i.e. is a Member Application), specific transition rules apply to preserve the consistency of activation states within the group (see section 3.7.3).
- When an Application is in the INSTALLED state, its Contactless Activation State is DEACTIVATED or NON_ACTIVATABLE. An Application may transition itself to the NON_ACTIVATABLE state during its installation.
- When an Application transitions from the INSTALLED state to the SELECTABLE state, the OPEN may attempt to activate the Application as described in section 8.3. However, this attempt shall fail if the activation of the Application conflicts with other currently activated Applications, or if the Application is in the NON_ACTIVATABLE state.
- When an Application is transitioned to the LOCKED state, then:
  - If it was ACTIVATED, then it becomes DEACTIVATED.
  - If it was DEACTIVATED, it remains DEACTIVATED.
  - If it was NON_ACTIVATABLE, it remains NON_ACTIVATABLE.
  - The Application cannot be activated again until the Application gets unlocked. The Application may transition from the DEACTIVATED state to the NON_ACTIVATABLE state, and vice versa.
- When an Application is unlocked, then:
  - If it was NON_ACTIVATABLE, it remains NON_ACTIVATABLE.
  - Otherwise the OPEN shall attempt to transition the Application to its Initial Contactless Activation State as defined during its installation (see section 8.3 for details). If the Application is to be activated but cannot be activated, then the Application transitions to the DEACTIVATED state.
- If the Contactless Activation State of a Head Application is modified as a result of one of the previous rules, then the state of associated Member Applications shall be modified according to the rules described in section 3.7.2.

8.2 Application Activation Policy

When the OPEN processes an activation request, it shall check whether the Contactless Application being activated does accept the activation request depending on its own business policy. By default, a Contactless Application will accept the activation request. To implement a specific business policy, a Contactless Application may implement and expose the `CLAppletActivationPolicy` interface. In this case, before processing the Protocol Parameter Conflict Detection Procedure described in section 4.5, the OPEN shall call the `acceptActivation()` method for that Contactless Application under the following conditions:

- The Contactless Application is not the originator of the activation request.
- The Contactless Application is in the DEACTIVATED state.

If the `acceptActivation()` method returns `false`, then the OPEN shall abort the processing of the activation request and is not required to process the conflict detection procedure. It is subsequently possible to retrieve policy conflict information by calling the `getNextApplicationConflictInfo()` method of the Contactless Application.

When a request is made to activate an Application Group, the OPEN shall call the `acceptActivation()` method of the Head Application only and shall not call the `acceptActivation()` of the Member Applications.

Depending on the business and deployment policies of application providers, the following behaviors may be implemented by the `CLAppletActivationPolicy` interface:

- If the Contactless Application is a Head Application, it may check whether some or all of its Member Applications (exposing the `CLAppletActivationPolicy` interface) report a conflict and/or retrieve conflict information from them.
- If the Contactless Application references CREL Applications, it may check whether some or all of these CREL Applications (exposing the `CLAppletActivationPolicy` interface) report a conflict and/or retrieve conflict information from them.

8.3 Initial Contactless Activation State

When an Application is installed and made selectable, its Initial Contactless Activation State can be specified using the “Initial Contactless Activation State” tag defined in Table 11-3.

If this parameter is not provided, a value owned by the OPEN shall apply. This value can only be modified by the Issuer Security Domain, using an INSTALL [for registry update] command with an empty AID field and the “Initial Contactless Activation State” tag defined in Table 11-3. The default value of this parameter is configuration-dependent.

The value part of the “Initial Contactless Activation State” TLV may have one of the following values:

- '00': Application shall not be activated.
- '01': Application shall be activated (if no conflict is detected).

During the installation process, if the Application is to be initially activated but has Activation Policy conflict (see section 8.2) or Protocol Parameter conflict (see section 4.5), the installation shall still be successful; however, the Application shall remain in its current Contactless Activation State (i.e. DEACTIVATED or NON_ACTIVATABLE) and a warning code (see section 11.2.4) shall be returned indicating that the Application was successfully installed but its Contactless Activation State remained unchanged. In this case, the CRS Application is notified of the installation, but not of the Contactless Activation State of the Application.

8.4 Contactless Interface Availability

The contactless interface may be switched globally on or off persistently by the Application that is granted the Contactless Activation Privilege (see sections 3.9.1 and 3.11.4). The technical means used to perform this global switch remain out of the scope of this document. Switching the contactless interface on or off shall not impact the Contactless Activation State or the Communication Interface Access Configuration (see Chapter 5) of an Application. When the contactless interface is switched off, a reset of the contactless interface shall be performed (see section 6.1, Reset Scenarios with Multiple Active Interfaces, for details on the expected behavior).

9 Cumulative Granted Memory

This amendment introduces a new install parameter that may be used to set up Cumulative Granted Memory (CGM) for a Security Domain and its sub-hierarchy. CGM amount may be setup for Volatile and Non-Volatile memory using the Cumulative Granted Volatile Memory and Cumulative Granted Non Volatile Memory tags in the Install Parameters field of the INSTALL [for install] and INSTALL [for registry update] as defined in Table 11-2. The usage of Cumulative Granted Memory may be limited in a specific configuration.

Note: The implementation of this feature may be limited, or otherwise impacted, by the specific runtime environment. Specifically, implementations of CGM for Volatile memory on the Java Card runtime environment may not be interoperable.

Cumulative Granted Memory (CGM), when assigned to a Security Domain, specifies the exact amount of memory granted to that Security Domain, to its associated Applications and its entire sub-hierarchy. This cumulative amount shall be interpreted both as a reserved memory amount, and as a memory quota. The sum of memory allocated to this set of applications (including the SD itself) must not exceed that cumulative amount. On the other hand, that cumulative amount is guaranteed to be available for these Applications. Any system data overhead shall be included in the calculation of the memory consumed by the Applications.

Cumulative Granted Memory applies to, and may be charged for any kind of memory allocation request, including card content management operations. In the case of extradition and deletion operations, Cumulative Granted Memory shall be charged back consistently.

When a Security Domain is assigned a CGM amount:

- The CGM amount shall not exceed the memory space currently available on the card.
- If an ancestor Security Domain is already assigned a CGM amount, then the newly assigned CGM amount shall be lower than, and is charged against, the CGM amount of the ancestor Security Domain.

When an Application is in the scope of a CGM amount (set up for its associated Security Domain, or one of its ancestor Security Domains), then the following rules shall apply:

When assigning Reserved Memory to an Application during its installation, Reserved Memory shall be charged against Cumulative Granted Memory.

Memory allocation requests performed by the Application during its lifetime

- Shall be charged in this order
  - Against Reserved Memory, if any was assigned to the Application
  - Against Cumulative Granted Memory, if Reserved Memory is exhausted, or no Reserved Memory was assigned to the Application
- Shall fail if the total amount of memory allocated by the Application would be
  - Greater than a Memory Quota, if any was assigned to the Application
  - Greater than remaining amount of Reserved Memory (if any) and remaining amount of Cumulative Granted Memory

When deleting an Application, the greater amount between Reserved Memory (if any was assigned to the Application) or the total amount of memory allocated by the Application, is credited back to the Cumulative Granted Memory.

When extraditing an Application to another Security Domain, the following rules shall apply (before the actual extradition):

- If the target Security Domain was assigned a CGM amount, or is in the scope of a CGM amount, then the greater amount between Reserved Memory (if any was assigned to the Application) and the total amount of memory allocated by the Application is charged against that CGM amount.
- If the origin Security Domain (associated with the Application) was assigned a CGM amount, or is in the scope of a CGM amount, then the greater amount between Reserved Memory (if any was assigned to the Application) and the total amount of memory allocated by the Application, is charged back to that CGM amount.

Once assigned to a Security Domain, a CGM amount can be increased or decreased, but cannot be removed. When updating the CGM amount for a Security Domain, the following rules shall apply:

- When decreasing CGM, the new value shall not be lower than the total amount of memory allocated or reserved by the hierarchy.
- The policy for updating the CGM amount is configuration dependent.

10 Cumulative Delete

10.1 Definition and Scope

Cumulative Delete is an option for a Security Domain with the Global Delete privilege (e.g. an Issuer) to delete a complete Security Domain hierarchy in case of technical problems or according to changing business relationships.

According to section 9.5, Content Removal, of [GPCS], a delete process is deemed to fail in case of open references (9.5.1, 9th bullet; 9.5.2, 12th bullet; 9.5.3, 9th, 10th, 14th and 15th bullet of [GPCS]) or remaining Security Domain associations (9.5.1, 10th bullet, of [GPCS]).

The Cumulative Delete process provides a solution to delete the root of a Security Domain hierarchy and all associated Executable Load Files or Applications. The GlobalPlatform and the underlying operating system dependencies are taken into account by logically deleting Executable Load Files and Applications with open references, but internally keeping the Executable Load Files and Applications until the references and relations are resolved. This allows extradited Applications to continue using the code of its Executable Load File.

A Security Domain with the Global Delete Privilege is able to retrieve a list of all logically deleted Executable Load Files.

10.2 Security Domain with Global Delete Privilege

This privilege (see section 9.1.3.4 of [GPCS]) provides the capability to remove any Executable Load File or Application from the card even if the Executable Load File or Application does not belong to this Security Domain (see section 9.5 of [GPCS]).

This privilege also provides the capability to delete a Security Domain hierarchy together with all associated Security Domains, Executable Load Files and Applications.

10.3 Security Domain Hierarchy Removal

A Security Domain associated with itself is the root of a hierarchy (see section 7.2, Security Domain Association, of [GPCS]). This optional feature allows the deletion of a hierarchy with all its associated objects.

When supported by the card, the following runtime behavior requirements apply during the Security Domain hierarchy removal process.

Runtime Behavior

On receipt of an Application deletion request (DELETE command), the Security Domain performing the deletion shall:

- Apply its own secure communication policy
- Apply its own security policy, e.g. check that its Lifecycle State is PERSONALIZED (only applicable to a Security Domain other than the Issuer Security Domain)
- If the Security Domain performing the deletion has the Delegated Management privilege or the Authorized Management privilege and the off-card entity at the origin of the delete request is not authenticated as its Security Domain Provider (see section 10.4, Entity Authentication, of [GPCS]), check that a Delete Token is present in the DELETE command

- If a Token is present in the DELETE command, request the OPEN to obtain verification of the Delete Token
- Request the OPEN to obtain a Delete Receipt

On receipt of a request to remove a Security Domain hierarchy, the OPEN shall:

- Check that the card Lifecycle State is not CARD_LOCKED or TERMINATED
- Check that the OPEN and the requesting on-card entity have no restriction for deletion
- Check that the requesting on-card entity is a Security Domain with the Global Delete privilege
- Check that the Security Domain to be deleted is the root of a hierarchy
- If the Security Domain performing the deletion has the Delegated Management privilege or the Authorized Management privilege and the off-card entity at the origin of the delete request is not authenticated as its Security Domain Provider, request the Security Domain with Token Verification privilege to verify the Delete Token
- Determine that the Security Domain and its associated Executable Load Files and Applications being deleted have entries within the GlobalPlatform Registry
- For each Application:
  - Determine whether the Application is currently selected on any logical channel.
  - If other Applications or Executable Load Files, not part of the same hierarchy, maintain references to any data within this Application, then the OPEN shall reset these references to null.
- For each Executable Load File:
  - Applications created from this Executable Load File shall be deleted, even if they were extradited to other hierarchies.
  - Determine if other Applications or other Executable Load Files present in the card maintain references to this Executable Load File.
    - If other Applications or other Executable Load Files that are part of the same hierarchy make references to this Executable Load File then the OPEN shall delete the referencing Applications, the referencing Executable Load Files and the Executable Load File itself.
    - If other Applications or other Executable Load Files not part of the same hierarchy make references to this Executable Load File, then the OPEN shall mark this Executable Load File as Logically Deleted with References (see section 10.4).
- Release and mark as available any Mutable Persistent Memory and when supported and apply the memory resource management rules described in section 9.7, Memory Resource Management, of [GPCS].
- At the request of the Security Domain performing the deletion, the Security Domain with the Receipt Generation privilege is requested to generate a Delete Receipt.

If the OPEN determines that any of the above verification steps have failed, the OPEN shall not initiate the delete process and shall inform the Security Domain to return the appropriate response. Once the delete processes begin, they shall all complete in the current Card Session or, in the event of an interruption, at least the updates to the GlobalPlatform Registry shall complete in a subsequent Card Session.

Only Mutable Persistent Memory is released and marked as available. Executable Load Files contained in Immutable Persistent Memory cannot be deleted but the entry for the Executable Load File and the entries for the Executable Modules present in the Executable Load File shall be deleted from the GlobalPlatform Registry or marked in the GlobalPlatform Registry as Logically Deleted with References. The OPEN can only guarantee that the hierarchy is logically deleted. A best effort shall be made to release Mutable Persistent Memory.

At the request of the OPEN, a Security Domain with the Token Verification privilege shall:

- Apply the issuer's policy to accept or reject a deletion authorization request without the presence of a Delete Token
- Verify the Delete Token

At the request of the OPEN, a Security Domain with the Receipt Generation privilege shall:

- Apply the issuer's policy to generate or not generate a Delete Receipt

10.4 Logically Deleted with References

Executable Load Files that are marked as Logically Deleted with References remain physically in the Mutable Persistent Memory and in the GlobalPlatform Registry. The following rules shall apply:

- There shall not be any Security Domain associated with Executable Load Files or Applications that are marked as "Logically Deleted with References".
- The OPEN shall not allow any Card Content Management to be performed on Executable Load Files that are marked as "Logically Deleted with References".
- The OPEN shall not allow loading of an Executable Load File which references one or more Executable Load Files marked as "Logically Deleted with References".
- Once all the references to an Executable Load File that is marked as "Logically Deleted with References" are resolved (e.g. referencing Application was deleted), the OPEN shall remove its entry from the GlobalPlatform Registry and release the Mutable Persistent Memory.
- The OPEN shall prevent the loading or installation of any load file or application with an AID that has been marked as "Logically Deleted with References".

The GET STATUS command (see section 11.4, GET STATUS (Logically Deleted with References) Command) can be used to retrieve a list of all Executable Load Files marked as Logically Deleted with References. The OPEN shall return this list to any Security Domain having the Global Registry privilege or Global Delete privilege.

11 Security Domain APDU Commands

11.1 Life Cycle State Coding

The following state for Load file is added to Table 11-3 of [GPCS] as follows:

The Executable Load File Life Cycle is coded on one byte as described in the following table.

Table 11-1: Executable Load File Life Cycle Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	LOGICALLY DELETED WITH REFERENCES
0	0	0	0	0	0	0	1	LOADED

11.2 INSTALL Command

The INSTALL command is coded and processed as described in [GPCS].

The TLV-structured values shown in Table 11-2 are added to the TLV System Specific Parameters.

Table 11-2: Contactless Specific Parameters

Tag	Length	Value Description			Presence
'EF'	var	System Specific Parameters			
		Tag	**Length**	**Value Description**	
		…	..	other parameters as defined in [GPCS]	
		'A0'	var	Contactless Protocol Parameters	Optional
		'A1'	var	User Interaction Parameters	Optional
		'B0'	var	(Assigned to ETSI)	Optional
		'82'	2 or 4	Cumulative Granted Volatile Memory (see Chapter 9)	Optional
		'83'	2 or 4	Cumulative Granted Non Volatile Memory (see Chapter 9)	Optional

The parameters defined above may be updated using INSTALL [for registry update] as defined in [GPCS]. The parameters above may also be used with the INSTALL [for install] and INSTALL [for install and make selectable] commands.

11.2.1 Contactless Protocol Parameters Structure

The Contactless Protocol Parameters are TLV structured values. They may be provided as part of system-specific INSTALL parameters.

Table 11-3 identifies the possible tags for managing an application on the contactless interface.

Table 11-3: Contactless Protocol Parameters

Tag	Length	Value Description	Presence
'80'	var	Assigned Protocols for implicit selection (see section 6.6)	Optional
'81'	1	Initial Contactless Activation State (see section 8.3 for specific usage of this TLV).	Optional
'A2'	var	Contactless Protocol Parameters Profile (see section 11.2.2)	Conditional
'83'	var	Recognition Algorithm (see section 6.5)	Optional
'84'	1-2	Continuous Processing (see section 6.4)	Optional
'A5'	var	Communication Interface Access Parameters (see section 5.2)	Optional
'86'	var	Protocol Data Type A (Card Emulation Mode) (see section 4.6)	Conditional
'87'	var	Protocol Data Type B (Card Emulation Mode) (see section 4.7)	Conditional
'88'	var	Protocol Data Type F (Card Emulation Mode) (see section 4.8)	Conditional

When configuring an application that has its own contactless parameters, the following rule shall apply:

- Tag 'A2' shall not be present.

When configuring an application that references a Contactless Protocol Parameters Profile, the following rules apply:

- Tag 'A2' shall be present, and may occur more than once if applying to different protocol types.
- Tags '86', '87', and '88' shall not be present.

When configuring a Contactless Protocol Parameters Profile, the following rules apply:

- Tag 'A2' shall be present, and may occur more than once if applying to different protocol types.
- Tag '86' shall be present if an occurrence of tag 'A2' is present and references Protocol Type A, otherwise it shall be absent.
- Tag '87' shall be present if an occurrence of tag 'A2' is present and references Protocol Type B, otherwise it shall be absent.
- Tags 'A2' and '88' shall not be present together. Profiles cannot be defined for Type F protocol.

11.2.2 Contactless Protocol Parameters Profile Structure

As described in section 4.9, Contactless Protocol Parameters may be managed by referencing Contactless Protocol Parameter Profiles.

The Profile Value (value part of the TLV 'A2') of the Contactless Protocol Parameters Profile structure is shown in Table 11-4.

Table 11-4: Contactless Protocol Parameters Profile

<table>
<tr><th>Length</th><th colspan="5">Value Description</th></tr>
<tr><td rowspan="4">Var</td><th>Tag</th><th>Length</th><th colspan="3">Value Description</th></tr>
<tr><td rowspan="3">'xx'</td><td rowspan="3">Var</td><td colspan="3">Profile Value for Protocol</td></tr>
<tr><th>Tag</th><th>Length</th><th>Name</th></tr>
<tr><td>'yy'</td><td>Var</td><td>Contactless Protocol Profile Identifier</td></tr>
</table>

Profile Value for Protocol Type Tag ('xx'):

- Tag 'A0' is used to identify the Profile Value for Protocol Type A
- Tag 'A1' is used to identify the Profile Value for Protocol Type B
- Tags 'A2' to 'AF' are reserved for assignment by GlobalPlatform of other Protocol Type profiles.
- Tags 'B0' to 'BE' are reserved for proprietary usage.

Protocol Profile Identifier Tag ('yy'):

- Tag '80' is reserved for use by EMV to identify EMV's profile (see [AAUI] for applicable profile identifiers) for all Protocol Types.
- Tags '81' to '8F' are reserved for assignment by GlobalPlatform to identify other organizations.
- Tags '90' to '9E' are reserved for proprietary usage.

Any of the tags above ('xx' and 'yy') shall be present only once within the Contactless Protocol Parameters Profile structure.

The presence of predefined Contactless Parameter Profiles is issuer-dependent.

11.2.3 User Interaction Parameters Structure

The User Interaction Parameters are TLV structured values. Table 11-5 defines the TLVs for encoding the parameters. They may be provided as part of system-specific INSTALL parameters.

Table 11-5: User Interaction Parameters

Tag	Length	Name	Presence
'7F20'	var	Display Control Template (see section 3.2)	Optional
'A0'	var	Head Application (see section 3.7.2)	Conditional
'A1'	var	Add to the Group Authorization List (see section 3.7.5)	Conditional
'A2'	var	Remove from the Group Authorization List (see section 3.7.6)	Conditional
'A3'	var	Add to the CREL List (see section 3.8.3)	Optional
'A4'	var	Remove from the CREL List (see section 3.8.4)	Optional
'A5'	Var	Policy Restricted Applications (see section 3.3)	Optional
'A6'	Var	Application discretionary data (see section 3.4)	Optional
'87'	1	Application Family (see section 3.5)	Optional
'88'	1	Display Required Indicator (see section 3.6)	Optional

A zero-length value for each of the tags above not beginning with “Add to” or “Remove from” shall delete any previous values for that tag on the target Application. A non-zero length value for each of the tags above not beginning with “Add to” or “Remove from” shall replace any previous values for that tag on the target Application.

11.2.4 Processing State returned in the Response Message

The INSTALL command may return the warning condition shown in Table 11-6.

Table 11-6: INSTALL Warning Condition

SW1 SW2	Meaning
'6200'	Applet could not be activated on the contactless interface

11.3 DELETE Command

The DELETE command is coded and processed as described in [GPCS].

Table 11-7 updates and replaces Table 11-21: DELETE Reference Control Parameter P2 of [GPCS]:

Table 11-7: DELETE Command Reference Control Parameter P2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	–	–	–	–	–	–	–	Delete object
1	–	–	–	–	–	–	–	Delete object and related object
0	1	–	–	–	–	–	–	Delete a root Security Domain and all associated Applications
–	–	X	X	X	X	X	X	RFU

If b7 = 1, then the DELETE command shall be rejected if one of the following additional error conditions is met:

- The AID specified in the Command Data does not refer to a root Security Domain.
- The entity processing this command does not have the Global Delete Privilege.

11.4 GET STATUS Command

The GET STATUS command is coded and processed as described in [GPCS].

Table 11-8 is an extension to Table 11-32: GET STATUS Reference Control Parameter P1 of [GPCS].

Table 11-8: GET STATUS Reference Control Parameter P1

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	–	–	–	0	–	–	–	Issuer Security Domain
–	1	–	–	0	–	–	–	Applications, including Security Domains
–	–	1	–	0	–	–	–	Executable Load Files
–	–	–	1	0	–	–	–	Executable Load Files and Executable Modules
0	0	0	0	1	0	0	0	Logically Deleted with References
–	–	–	–	–	x	x	x	RFU

Bit b4 “Logically Deleted with References” cannot be combined with any other option.

The following values of the reference control parameter shall be supported:

- '80' – Issuer Security Domain only. In this case the search criteria is ignored and the Issuer Security Domain information is returned
- '40' – Applications and Supplementary Security Domains only
- '20' – Executable Load Files only
- '10' – Executable Load Files and their Executable Modules only
- '08' – Executable Load Files that are marked as “Logically Deleted with References”

The ability to differentiate a Security Domain from an Application is achieved through privileges.

Only a Security Domain having the Global Registry privilege or Global Delete Privilege is able to retrieve a list of all Executable Load Files marked as Logically Deleted with References. These entries are provided by the OPEN and are not part of the GlobalPlatform Registry entries associated with the Security Domain receiving the GET STATUS command.

The current order of the GlobalPlatform Registry also defines the order in which data is returned in the response data of the GET STATUS command (see section 6.2.1).

11.4.1 Filter Criteria: Tag List (Tag '5C')

The tag list (tag '5C') indicates to the card how to construct the response data for each on-card entity matching the search criteria. The value part of this TLV contains a concatenation of tags (without delimitation) indicating the data objects to include in the response.

The possible criteria are listed in Table 11-9.

Table 11-9: Filter Criteria (Tag '5C')

Tag	Name
'4F'	AID
'9F70'	Application Life Cycle State on the first byte as defined in section 11.1. The Contactless Activation State is encoded on the second byte as defined in Table 8-1.
'C5'	Privileges (byte 1 – byte 2 – byte 3) (see section 11.1.2 of [GPCS])
'CF'	Implicit Selection Parameters (see section 11.1.7 of [GPCS])
'C4'	Application's Executable Load File AID
'CE'	Executable Load File Version Number
'84'	List of Executable Module AID
'CC'	Associated Security Domain's AID
'7F20'	Display Control Template (see section 3.2)
'5F50'	Uniform Resource Locator (see section 3.2)
'6D'	Application Image Template (see section 3.2)
'5F45'	Display Message (see section 3.2)
'A0'	Application Group Head Application (coded as in Table 3-3)
'A1'	Application Group Authorization List (coded as in Table 3-4)
'A2'	CREL Application AID List (coded as in Table 3-6)
'A3'	Policy Restricted Applications (see section 3.3)
'A4'	Application Group Members (coded as in Table 3-4) (excluding the Head Application)
'A5'	Application discretionary data (see section 3.4)
'86'	Application Family (see section 3.5)
'87'	Assigned Protocols for implicit selection (see section 6.6)
'88'	Initial Contactless Activation State (see section 8.3)
'A9'	Contactless Protocol Parameters Profile (see section 11.2.2)
'8A'	Recognition Algorithm (see section 6.5)
'8B'	Continuous Processing (see section 6.4)
'AC'	Communication Interface Access Parameters (see section 5.2)
'8D'	Protocol Data Type A (Card Emulation Mode) (see section 4.6)
'8E'	Protocol Data Type B (Card Emulation Mode) (see section 4.7)

Tag	Name
'8F'	Cumulative Granted Non-Volatile Memory (see Chapter 9)
'90'	Cumulative Granted Volatile Memory (see Chapter 9)
'91'	Cumulative Remaining Non-Volatile Memory (see Chapter 9)
'92'	Cumulative Remaining Volatile Memory (see Chapter 9)
'93'	Protocol Data Type F (Card Emulation Mode) (see section 4.8)

These tags shall be used as filter criteria in a simple, non-constructed form. For example, '5F50' is a sub-tag of '7F20', however, only '5F50' shall appear in the requested tag list (see example related to Table 3-14).

11.4.2 Response Message

The following GlobalPlatform Registry Data replaces Table 11-36 of [GPCS]: GlobalPlatform Registry Data (TLV) for cards compliant with this amendment.

Table 11-10: GlobalPlatform Application Registry Data (TLV)

Tag	Length	Value Description
'E3'	Variable	GlobalPlatform Registry related data
'4F'	5-16	AID
'9F70'	2	Application Life Cycle State on the first byte as defined in section 11.1.1 of [GPCS]. The Contactless Activation State is encoded on the second byte as defined in Table 8-1.
'C5'	0,1,3	Privileges (byte 1 – byte 2 – byte 3) see section 11.1.2 of [GPCS]
'CF'	1	First Implicit Selection Parameter. See section 11.1.7 of [GPCS].
…	…	…
'CF'	1	Last Implicit Selection Parameter. See section 11.1.7 of [GPCS].
'C4'	1-n	Application's Executable Load File AID
'CC'	5-16	Associated Security Domain's AID

Table 11-11: GlobalPlatform Load File Registry Data (TLV)

Tag	Length	Value Description
'E3'	Variable	GlobalPlatform Registry related data
'4F'	5-16	AID
'9F70'	1-2	Load File Life Cycle State on the first byte as defined in section 11.1. If present, the second byte shall be set to 0.
'CE'	1-n	Executable Load File Version Number
'84'	1-n	First or only Executable Module AID
…	…	…
'84'	1-n	Last Executable Module AID
'CC'	0, 5-16	Associated Security Domain's AID If the load file is marked as LOGICALLY DELETED WITH REFERENCES then the length shall be 0.

11.5 GET DATA Command

The GET DATA command is coded and processed as described in [GPCS], with the extensions described below.

11.5.1 Security Domain Manager URL

Security Domains shall support the following data object tag:

- Tag '5F50': Security Domain Manager URL

This data object provides an internet link to the Manager of the Security Domain. Its content and coding are defined in a specification of the systems committee [reference to be added once available]. This data object is different from and shall not be confused with the URL defined in section 3.2, Display Control Information.

11.5.2 Forwarded CASD Data

A mechanism is defined to enable retrieval of data forwarded from the Controlling Authority Security Domain (CASD; see [GPCS-A]). This feature allows a Service Provider to select (or target through an OTA RAM script) its own Security Domain to retrieve data actually owned by the CASD.

The command shall comply with the following requirements:

- The P1-P2 parameters shall be set to 'BF30'
- The command shall have a command data field encoding a request for one (and only one) of the following data:
  - CA Security Domain Recognition Data: '5C 01 66'
  - CA Certificate Store (containing a sequence of one or more CASD Public Key Certificates): '5C 02 7F21'

If the command data field is coded differently, a status word of '6A80' shall be returned.

If the CASD is not present or not yet able to provide such data, a status word of '6A88' shall be returned. Otherwise, the response shall contain the requested data object retrieved from the CASD, encapsulated in a data object with tag 'BF30'.

For example,

- The following GlobalPlatform GET DATA command:

  '80 CA BF 30 04 5C 02 7F 21'

  would have an answer of the following form:

  'BF 30' (L) '7F 21' (L) (V)

- The following ISO GET DATA command:

  '00 CA BF 30 04 5C 02 7F 21'

  would have an answer of the following form:

  '7F 21' (L) (V)

11.6 STORE DATA Command

The STORE DATA command is coded and processed as described in [GPCS], with the extensions described below.

11.6.1 Security Domain Manager URL

Security Domains shall support the following TLV-coded data object:

- Security Domain Manager's Uniform Resource Locator (tag '5F50').

For the content and encoding of this data object, see section 11.5.1. This data object is different from and shall not be confused with the URL defined in section 3.2, Display Control Information.

When the STORE DATA command uses DGI formatting, this data object shall be included within DGI '0070'.

12 Token Identifier Blacklist for Delegated Management

12.1 Definition and Scope

[GPCS] defines token formats for card content management; however, a means to prevent re-use of delegated management tokens is not provided. This section defines a mechanism to blacklist token use based upon the Token Identifier. Token Identifiers present in the blacklist cannot be used for delegated card content management. Each Security Domain with the Token Verification Privilege may own a blacklist. A card content management command containing a blacklisted Token Identifier shall be rejected.

12.2 Blacklist / Rehabilitate Using STORE DATA Command

The STORE DATA command may be used to add or remove a Token Identifier in the blacklist. Each operation is represented by a tag. The Token Identifier on which the operation applies is in the value body. The encoding format of data shall be as described in section 11.11.2.3 of [GPCS].

Every SD with the Token Verification Privilege may support the following TLV coded data objects:

- Add Token Identifiers to the blacklist
- Remove Token Identifiers from the blacklist

Upon receipt of these TLV, the Blacklist shall be updated and the Token Identifier(s) present in the command shall be Added / Removed from the list.

If a Token Identifier to be removed is not blacklisted, an error condition shall be returned according to [GPCS], and the entire operation shall fail.

If a Token Identifier to add to the blacklist is already present, no error is returned.

Any attempt to store, update or retrieve the blacklist on a SD that does not have the Token Verification privilege shall fail. An error condition code shall be returned according to [GPCS].

12.3 Add to Blacklist

The content of the TLV is coded as follows:

Table 12-1: Add Token Identifiers to the Blacklist TLV

Tag	Length	Value Description		
'A0'	Var	Add Token Identifier(s) to the blacklist		
		Tag	**Length**	**Value Description**
		'93'	Var	Token Identifier
		…		
		'93'	Var	Token Identifier

12.4 Remove from Blacklist

The content of the TLV is coded as follows:

Table 12-2: Remove Token Identifiers from the Blacklist TLV

Tag	Length	Value Description		
'A1'	Var	Remove Token Identifier(s) from the blacklist		
		Tag	**Length**	**Value Description**
		'93'	Var	Token Identifier
		…		
		'93'	Var	Token Identifier

12.5 Read Blacklist

The list of the blacklisted Token Identifiers can be retrieved from an SD with the Token Verification privilege using the GET DATA command with tag 'A0'. The SD shall return the list of all blacklisted Token Identifiers it has.

The content of the TLV is coded as follows:

Table 12-3: Token Identifier Blacklist TLV

Tag	Length	Value Description		
'A0'	Var	Token Identifier blacklist		
		Tag	**Length**	**Value Description**
		'93'	Var	Token Identifier
		…		
		'93'	Var	Token Identifier

12.6 Processing State Returned in the Response Message

A successful execution of the command shall be indicated by status word '9000'.

The following error conditions may occur and are indicated by the status bytes given in Table 12-4.

Table 12-4: Error Conditions

SW1 SW2	Meaning
'6A80'	Incorrect values in command data
'6A88'	Referenced data not found

Annex A GlobalPlatform Java Card API

The following requirements apply for cards implementing the mechanisms described in this document:

- The package `org.globalplatform` shall be implemented in version 1.5 (or above), ensuring that the `secureChannelx2` interface is defined.
- The package `org.globalplatform.contactless` shall be implemented in version 1.2 (or above).

Export files and HTML documentation for the GlobalPlatform Java Card API may be downloaded at:

http://www.globalplatform.org/specificationscard.asp.

The figures in this annex describe the expected logic flow for certain activities defined elsewhere in this specification, as viewed at the API level.

Figure A-1: Application Policy Conflict: Detection and Resolution

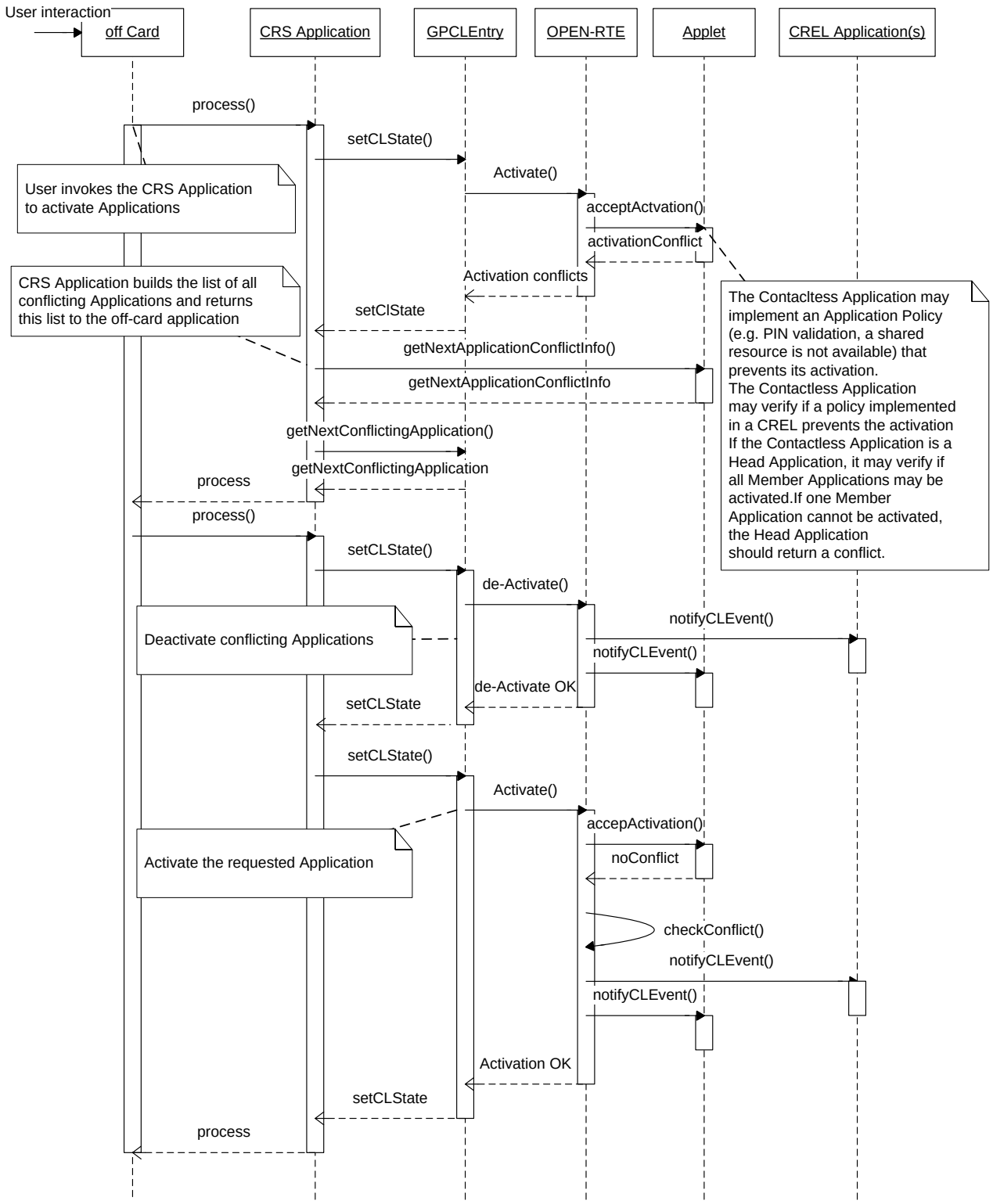


Figure A-2: Application Protocol Parameter Conflict: Detection and Resolution

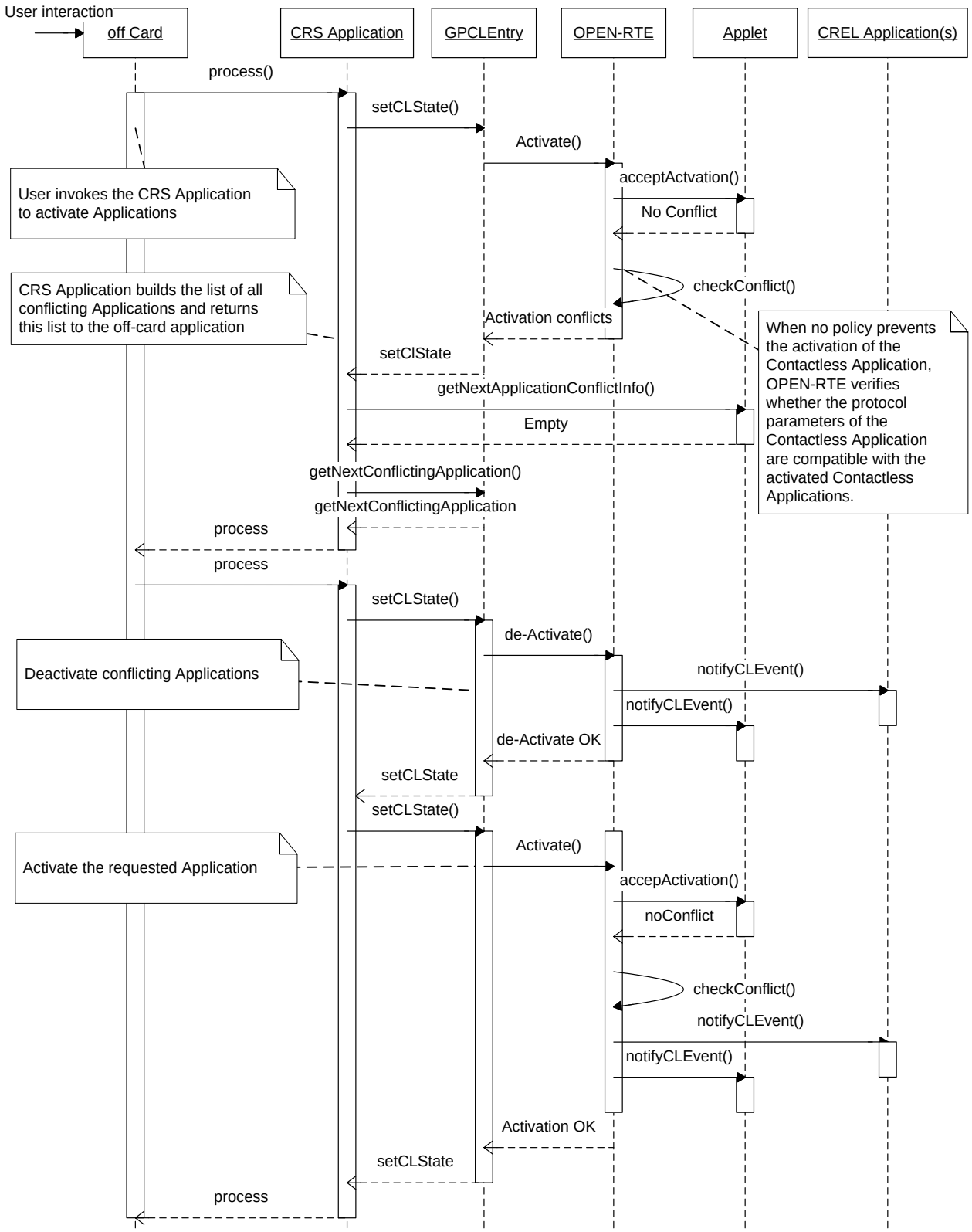


Figure A-3: Contactless Self-Activation: No Protocol Parameter Conflict

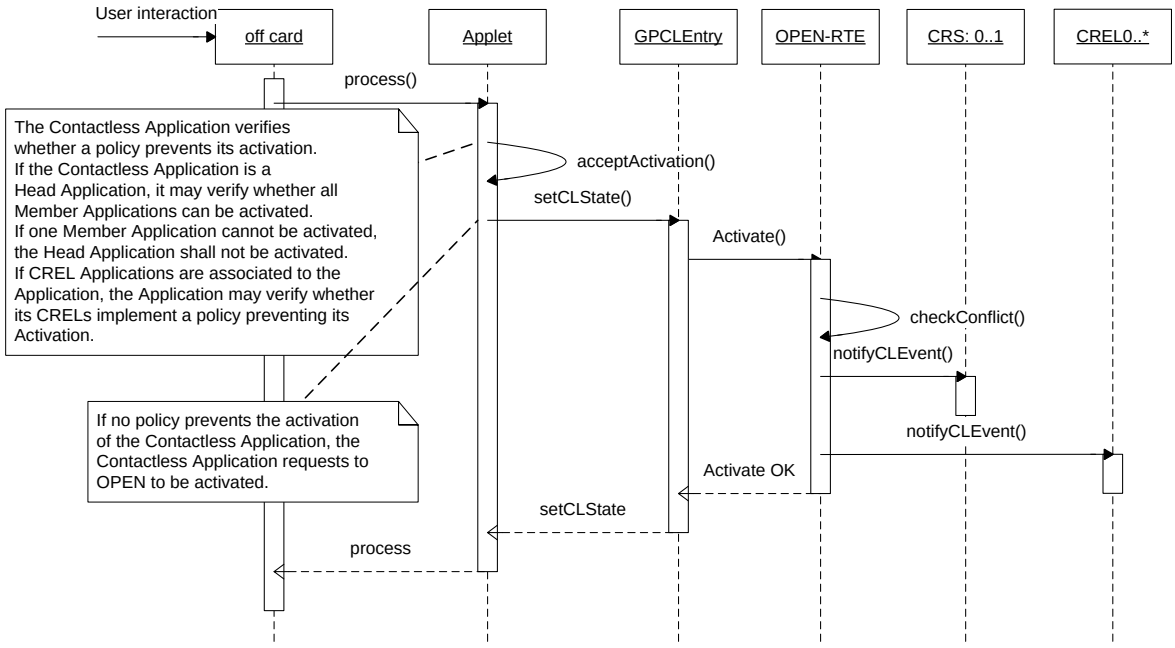
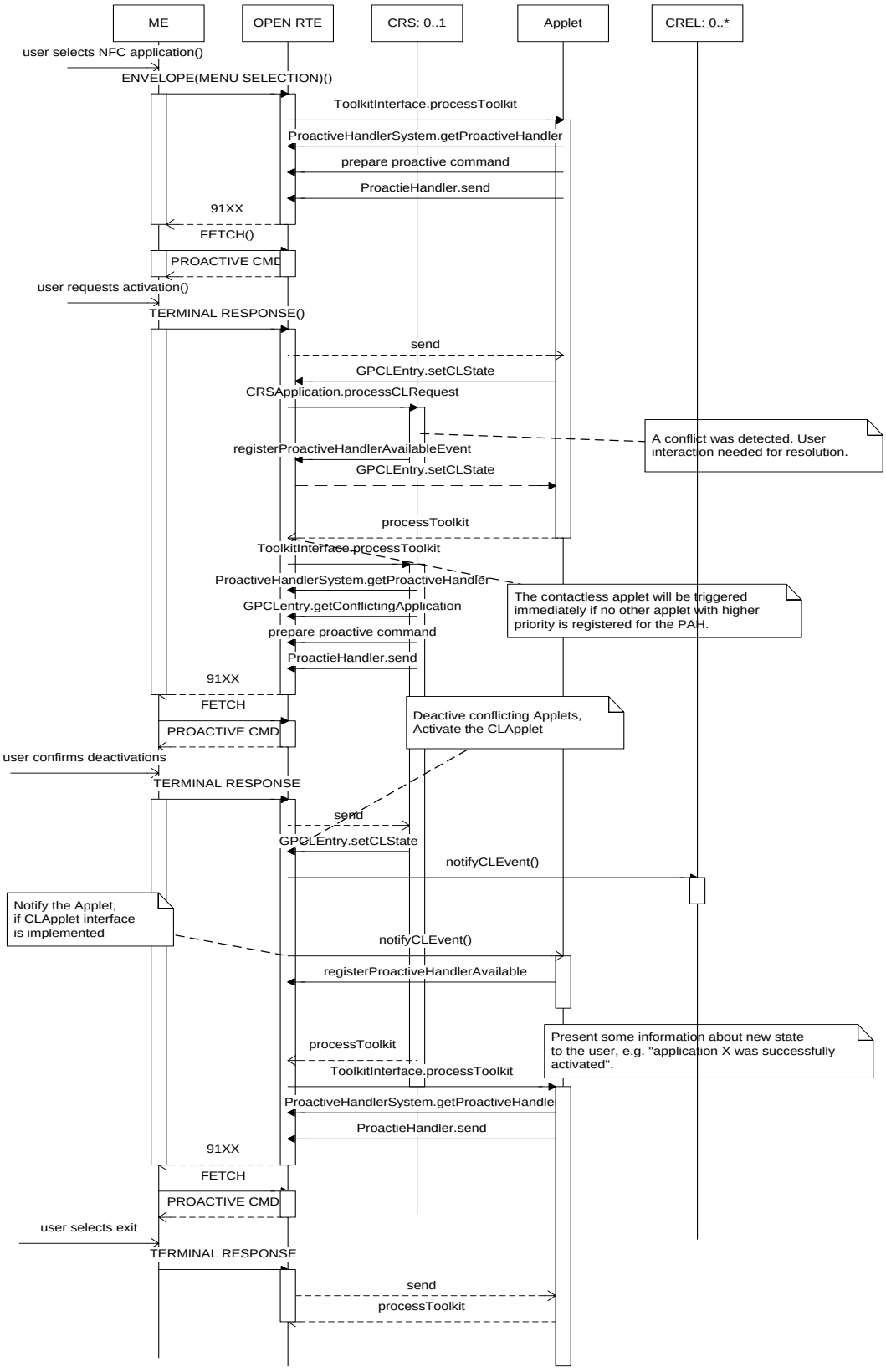


Figure A-4: No Contactless Self-Activation: Conflict Detection and Resolution using CAT Framework[1]



[1] See ETSI TS 102 223 [102223].

Annex B Contactless Protocol Management: Example

B.1 Current Protocol Parameters for Type A Computation

The computation of the Current Protocol Parameters is detailed in section 4.4.2.1.

B.1.1 Value Definition

The current Protocol Parameters for Type A is:

Table B-1: Type A Computation: Current Protocol Parameters

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'00'	'0000'	'03010203'	'EE'	'01'	'030300'
Mandatory Mask						
'FF'	'81'	'00FF'	'FF00FF00'	'FF'	'FF'	'FFFFFF'

The Protocol Parameter for Type A of the Activated Application is:

Table B-2: Type A Computation: Activated Application Protocol Parameters

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'21'	'8200'	'020102'	'EE'	'01'	'030300'
Mandatory Mask						
'FF'	'24'	'FF00'	'FFFF00'	'FF'	'FF'	'FFFFFF'

B.1.2 Protocol Data Computation: Intermediate Result [A]

The length of the ATS Historical bytes of the current protocol data is reduced from 3 to 2 since the Application is requesting that the length of the ATS historical bytes shall not exceed 2.

Table B-3: Type A Computation: Intermediate Result: Mandatory '1' Bit Mask [A]

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'20'	'8200'	'020100'	'EE'	'01'	'030300'

B.1.3 Protocol Data Computation: Intermediate Result [B]

Table B-4: Type A Computation: Intermediate Result: Mandatory '0' Bit Mask [B]

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'FB'	'82FF'	'0201FF'	'EE'	'01'	'030300'

B.1.4 Protocol Data Computation: Intermediate Result [C]

Table B-5: Type A Computation: Intermediate Result [C]

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'20'	'8200'	'020102'	'EE'	'01'	'030300'

B.1.5 Current Protocol Parameter of Type A: Result

Table B-6: Type A Computation: New Current Protocol Parameter

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'20'	'8200'	'020102'	'EE'	'01'	'030300'
Mandatory Mask						
'FF'	'A5'	'FFFF'	'FFFFFF'	'FF'	'FF'	'FFFFFF'

B.2 Protocol Parameter for Type A: Conflict Detection

The conflict detection algorithm is specified in section 4.5.

B.2.1 Value Definition

The current Protocol Parameter for Type A is:

Table B-7: Type A Conflict Detection: Current Protocol Parameter

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'20'	'8200'	'020102'	'EE'	'01'	'030300'
Mandatory Mask						
'FF'	'A5'	'FF00'	'FF00FF'	'FF'	'FF'	'FFFFFF'

The Protocol Parameter for Type A of the Application being activated is:

Table B-8: Type A Conflict Detection: Activated Application Protocol Parameter

Protocol Data						
LV UID	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'	'**00**'	'8200'	'03010203'	'EE'	'**00**'	'030300'
Mandatory Mask						
'FF'	'**24**'	'00FF'	'FFFF00**FF**'	'FF'	'**FF**'	'FFFFFF'

In this example, the computation stops at this point. The conflicts are highlighted in the table directly above (bold text) and described below:

- Two different SAK values are requested.
- The LV ATS Historical bytes conflict due to a requested third value in the Historical bytes, even though the requested MAX length does not conflict.
- Different CID Support values are requested

B.3 Protocol Parameter for Type A: UID Computation

B.3.1 Value Definition

The current Protocol Parameters for Type A is:

Table B-9: Type A UID Computation: Current Protocol Parameters

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'00'		'00'	'0000'	'03010203'	'EE'	'01'	'030300'
Mandatory Mask							
'00'		'81'	'00FF'	'FF00FF00'	'FF'	'FF'	'FFFFFF'

Application demands that the UID is exactly 7 bytes but does not mandate a value. As a default value, it suggests to use the first UID that is defined in the chip or OPEN. The Protocol Parameter for Type A of the Activated Application is as follows:

Table B-10: Type A UID Computation: Activated Application Protocol Parameters

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'07'	'00000000000001'	'00'	'4400'	'03010203'	'EE'	'01'	'030300'
Mandatory Mask							
'0F'	'00000000000000'	'81'	'FF00'	'FF00FF00'	'FF'	'FF'	'FFFFFF'

B.3.2 Protocol Data Computation: Intermediate Result [A]

The UID with index 1 is retrieved from OPEN because the first byte of the UID is '00'. In this example, the UID with index 1 is '12345678901234'.

Table B-11: Type A UID Computation: Intermediate Result [A]

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'07'	'00000000000000'	'00'	'4400'	'03000200'	'EE'	'01'	'030300'

B.3.3 Protocol Data Computation: Intermediate Result [B]

Table B-12: Type A UID Computation: Intermediate Result [B]

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'07'	'FFFFFFFFFFFFFF'	'7E'	'44FF'	'03FF02FF'	'EE'	'01'	'030300'

B.3.4 Protocol Data Computation: Intermediate Result [C]

Table B-13: Type A UID Computation: Intermediate Result [C]

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'07'	'00000000000000'	'00'	'4400'	'03010203'	'EE'	'01'	'030300'

B.3.5 Current Protocol Parameter of Type A: Result

The UID length result is 7 bytes and the UID value as suggested by the Activated Application.

Table B-14: Type A UID Computation: New Current Protocol Parameter

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'07'	'00000000000000'	'00'	'4400'	'03010203'	'EE'	'01'	'030300'
Mandatory Mask							
'0F'	'00000000000000'	'81'	'FFFF'	'FF00FF00'	'FF'	'FF'	'FFFFFF'

B.4 Protocol Parameter for Type A: Conflict Detection in UID

B.4.1 Value Definition

The current Protocol Parameter for Type A is:

Table B-15: Type A Conflict Detection in UID: Current Protocol Parameters

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'07'	'12345678901234'	'00'	'4400'	'03010203'	'EE'	'01'	'030300'
Mandatory Mask							
'0F'	'00000000000000'	'81'	'FF00'	'FF00FF00'	'FF'	'FF'	'FFFFFF'

The Protocol Parameter for Type A of the Application being activated is:

Table B-16: Type A Conflict Detection in UID: Activated Application Protocol Parameters

Protocol Data							
Length UID	**Value UID**	**SAK**	**ATQA**	**LV ATS Historical Bytes**	**FWI,SFGI**	**CID Support**	**Max Data Rate**
'04'	'87654321'	'00'	'4400'	'03010203'	'EE'	'01'	'030300'
Mandatory Mask							
'0F'	'00000000'	'81'	'FF00'	'FF00FF00'	'FF'	'FF'	'FFFFFF'

In this example, the UID lengths are conflicting. The current Protocol Parameter requires a 7 byte UID while the Protocol Parameter of the Application being activated requires 4 bytes and the UID length value of both Parameter Protocols shall be treated as the exact value to be used.

B.5 Current Protocol Parameters for Type F Computation

B.5.1 Value Definition

The Current Protocol Parameter for Type F is as shown in Table B-17.

Table B-17: Current Protocol Parameters for Type F

Protocol Data				Mode Flag[2]
	System Code	PICC Identifier	Response Time Descriptor	
1st anti-collision parameters	'12FC'	'02FE010101010101'	'FFFFFFFFFFFFFFFF'	TRUE
2nd anti-collision parameters	'AA22'	'02FE020202020202'	'FFFFFFFFFFFFFFFF'	TRUE

The Protocol Parameter for Type F of the Application being activated is:

Table B-18: Activated Application Protocol Parameter for Type F

Protocol Data			
	System Code	PICC Identifier	Response Time Descriptor
1st anti-collision parameters	'AA33'	'02FE030303030303'	'FFFFFFFFFFFFFFFF'

B.5.2 Current Protocol Parameter of Type F: Result

Table B-19: New Current Protocol Parameter for Type F

Protocol Data				Mode Flag
	System Code	PICC Identifier	Response Time Descriptor	
1st anti-collision parameters	'12FC'	'02FE010101010101'	'FFFFFFFFFFFFFFFF'	TRUE
2nd anti-collision parameters	'AA22'	'02FE020202020202'	'FFFFFFFFFFFFFFFF'	TRUE
3rd anti-collision parameters	'AA33'	'02FE030303030303'	'FFFFFFFFFFFFFFFF'	TRUE

The 3rd anti-collision parameters entry of the Current Protocol Data represents the 1st anti-collision parameters of the activated Application.

[2] The Mode Flag value is encoded using the values of constants PROTOCOL_TYPE_F_MODE_FLAG_TRUE and PROTOCOL_TYPE_F_MODE_FLAG_FALSE of the `GPCLRegistryEntry` class (see GlobalPlatform Contactless API v1.2).

B.6 Protocol Parameter for Type F: Conflict Detection

B.6.1 Value Definition

In this example it is assumed that the Maximum Number of anti-collision parameter entries is set to 4.

The current Protocol Parameters for Type F are as shown in Table B-20.

Table B-20: Current Protocol Parameters for Type F

Protocol Data				Mode Flag
	System Code	PICC Identifier	Response Time Descriptor	
1st anti-collision parameters	'12FC'	'02FE010101010101'	'FFFFFFFFFFFFFFFF'	TRUE
2nd anti-collision parameters	'AA22'	'02FE020202020202'	'FFFFFFFFFFFFFFFF'	TRUE
3rd anti-collision parameters	'AA33'	'02FE030303030303'	'FFFFFFFFFFFFFFFF'	TRUE

The Protocol Parameter for Type F of the Application being activated is:

Table B-21: Protocol Parameters for Type F Application Being Activated

Protocol Data			
	System Code	PICC Identifier	Response Time Descriptor
1st anti-collision parameters	'AA44'	'02FE040404040404'	'FFFFFFFFFFFFFFFF'
2nd anti-collision parameters	'AA55'	'02FE050505050505'	'FFFFFFFFFFFFFFFF'

In this example, the maximum number of entries is conflicting. The current Protocol Parameter has already stored three entries while the Protocol Parameter of the Application being activated has two entries to be added. Since the total number of entries in the Current Protocol Parameter exceeds the maximum number if two entries are added, the Application will not be activated and stays in the DEACTIVATED state.