NFC SP Applet Development Guideline
Version 4.0
30 September 2015

This is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Non-confidential

Copyright Notice

Disclaimer

Antitrust Notice

Table of Contents

1 Introduction

1.1 Purpose and content of this document

The purpose of this document is to provide to Service Providers some rules, common to the Mobile Network Operators, in order to properly develop their mobile NFC service Applet (Basic Applications [7]). This document is for guidance only.

The document addresses the following topics:

- “Management Rules”: recommendations regarding the structure of the application and the usage by the application of the data, the resources or the sensitive functions and the development environment.
- “Usage of APIs and USIM Toolkit Commands and Events”: lists the APIs and the USIM Toolkit commands and events that are "Free of usage", "Limited" or "Forbidden". When an API, command or event is listed as "Limited", it means that it is sensitive and its usage by the application must be described in the documentation that the Service Provider provides to the MNO or to the Certification Entity.
- “Memory Management”: recommendations regarding the memory management (deletion, allocation etc.) because it is a limited resource on the UICC.
- “System Management”: description to the files’ access (reading, writing). The creation of files (except application files) is forbidden and the files authorised in restricted access are listed.
- “Exception and Error Management”: recommendations regarding the exception and error management.
- “Interactions”: recommendations on the interactions of the Applet with the mobile and with the external resources.
- “Developments Rules”: recommendations regarding the development of an Applet.

Warning: To adhere with the existing technologies on the field and associated specifications, it is recommended that NFC application developers comply at least with ETSI Java Card API specifications as described in Release 6. The exception is ETSI TS 102 705 (API HCI), where developers should comply with ETSI Java Card API specifications Release 9 and Java Card 2.2.2. Adherence to these specifications will ensure interoperability between cards (UICC’s) provided to different MNO’s and will also ensure backward compatibility.

If a developer needs to develop NFC applications that rely on a version of ETSI Java Card API which is higher than Release 6 and Java Card higher than 2.2.2, then this should be verified with the MNO.

1.2 Definition of Terms

Term	Description
APDU	Application Protocol Data Unit
API	Application Programming Interface

Term	Description
ATR	Answer To Reset
CAT	Card Application Toolkit
CREL	Contactless Registry Event Listener
CRS	Contactless Registry Services
MNO	Mobile Network Operator
NFC	Near Field Communication
OTA	Over The Air
SD	Security Domain
SIM	Subscriber Identity Module
SP	Service Provider
UICC	Universal Integrated Circuit Card
USAT	USIM Card Application Toolkit
USIM	Universal SIM

1.3 Normative References

Term	Description
[1]	GSMA SGP.03 NFC UICC Requirement Specification, Version 6.0
[2]	GSMA TS.27 NFC Handset APIs & Requirements, Version 7.0

1.4 Informative References

Term	Description
[3]	SIMalliance NFC Stepping Stones 2011 V1.0.0
[4]	SIMalliance Interoperability Stepping Stones Release 7 Version 1.0.0
[5]	GlobalPlatform Card - Contactless Services - Card Specification v2.2 - Amendment C - Version 1.11 or later
[6]	Java Card™ Platform, Version 3.0.1 Classic edition – Application Programming Interface
[7]	GlobalPlatform Card - Composition Model Security Guidelines for Basic Applications v2.0
[8]	ETSI TS 102 225: “Smart Cards; Secured packet structure for UICC applications” V.9.2.0 (2012-03) or later

2 Management Rules

- Rule 1: Interface management and Data management must be clearly separated.
    - Separate the handling of commands and the handling of content. The essential separation is between the interface and the actual processing of application data. This is particularly important when several interfaces are available, for instance an Application Protocol Data Unit (APDU) interface and a USIM Toolkit interface.
- Rule 2: The interactions between different interfaces must be clearly defined.

    - When several interfaces are used, their respective roles need to be clearly defined, and this needs to be reflected in the application code.

- Rule 3: Usage of the shareable interface is forbidden except with explicit MNO agreement.

2.1 State of the art

This document presents mandatory or recommended complementary rules to be taken into account for developing Java applets but applications developers should also comply with state of the art development practices.

- Rule 4: Package design: A basic applet shall never be in the same package as any sensitive applet [7].
- Rule 5: Entry points: All the different entry points shall be clearly defined at, for instance: APDU level, USIM Toolkit level, Over The Air (OTA) script level, Personalisation level, Shared methods level.
- Rule 6: Denial of Service: Developers will be especially careful to avoid practices leading any denial of service as, for instance:

    - Banning any Instantiation other than in install or applet's constructor;
    - Limitation of use of data heap resources;
    - Limitation of use of telecom file system resources;
    - Banning Infinite loop;
    - Banning repetitive NVM writing that could stress memory.

- Rule 7: Dead code or debug information must be deleted/removed from the code.
- Rule 8: Files imports: The file imports versions must fit to the versions (and method signatures) installed on the targeted cards, according to the binary compatibility rules described in Java Card 3.0.1 Classic Edition VM Architecture Specification.
- Rule 9: Service provider applications shall comply with the rules defined in Composition Model Security Guidelines for Basic Applications [7].
- Rule 10: Coding User Input: A basic application shall verify that each user entry is in the range of accepted values
- Rule 11: Coding NOP: A basic application bytecode shall not include any NOP (NO Operation) instruction.

2.2 Development Environment

The development environment shall be physically and logically separated from the production environment.

Access to the source code shall be protected and reserved to duly authorised persons.

A Software Configuration Manager (SCM) that allows drawing all the modifications made on the application shall be available.

- Rule 12: Version number increment: A basic application with the same AID as an already verified basic application but with different exported methods signatures shall have an incremented major version.

- Rule 13: ByteCode verification: In order to be protected against attacks on card resources, a basic application must be strictly in accordance with the Java Card specifications and must successfully pass the latest ByteCode verification tool from Oracle in JCDK latest version.
- Rule 14: Bytecode Verification Used Export Files: If a basic application imports some packages (sharing or libraries), the export file used for the verification shall correspond to the export file previously verified by the MNO or to the Certification Entity and shall not be provided by the Service Provider.

2.3 Management life cycle

The life cycle of the applications must be correctly managed: the execution of an action should be possible only in the state of the application to which it relates. The state of an application must be protected as sensitive data and thus some measures to protect its integrity shall be taken.

2.4 Applet Installation

- Rule 15: Persistent Objects (using EEPROM, File View Object and any other Persistent Objects) must be allocated during the application installation to avoid exceptions during runtime due to shortage of UICC memory.
- Rule 16: Applet instance registration should occur as late as possible.
- After applet instance registration it is possible to call only APIs which require registration.
- Rule 17: Menu entries shall be initialised only in install () method.
- Rule 63: STK applet, priority level setting. The Priority Level shall be set to 0xFF (the lowest priority level). Any use of a different priority level parameter value shall be agreed with the MNO.
- Rule 64 (**Limited**): Minimum security level.

  The Minimum Security Level (MSL) is used to specify the minimum level of security to be applied to Secured Packets sent to any Receiving Application as defined ETSI TS 102 225 [8].

Only the following values of the SPI byte, given in the table here under, are allowed:

b8	b7	b6	b5	b4	b3	b2	b1	Hexa
0	0	0	1	0	0	1	0	0x12
0	0	0	1	0	1	1	0	0x16
0	0	0	1	1	0	1	0	0x1A
0	0	0	1	1	1	1	0	0x1E

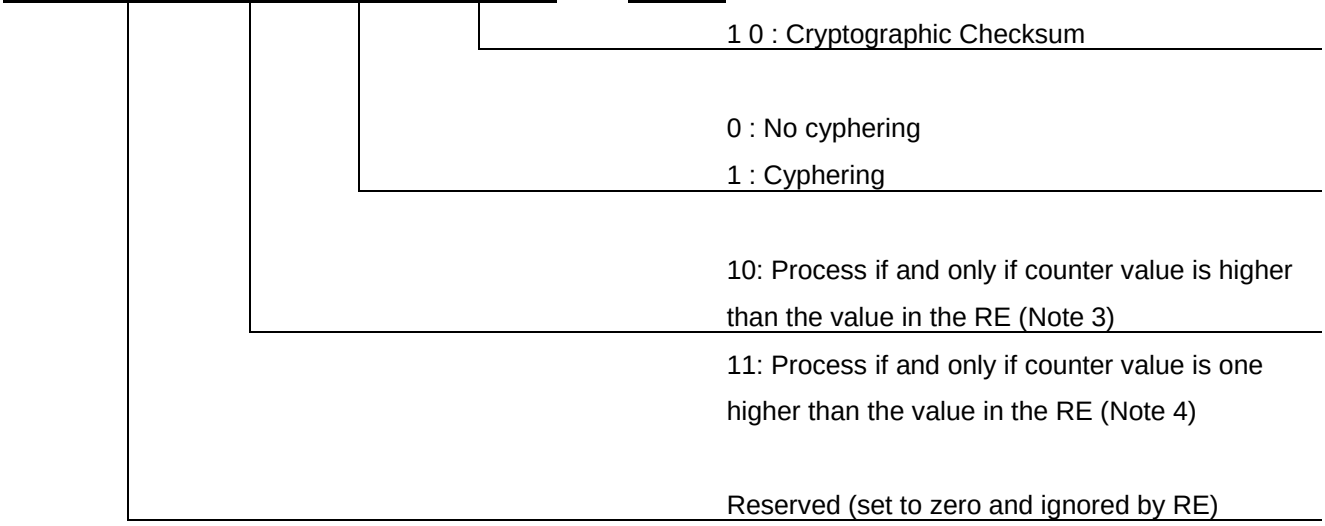

Table 1: SPI byte format

NOTE: The counter value is compared with the counter value of the last received Command Packet. This is tolerant to failures on the transport level (i.e. losses of Command Packets). A possible scenario is a global update.

NOTE: This provides strict control in addition to security indicated in Note 3.

2.5 Applet Deletion

- Rule 18: Implement the uninstall method in Java Card 3.0.1 classic edition applets.
    - The Java Card 3.0.1 classic edition specification defines the AppletEvent interface, which includes an uninstall method that is invoked when the deletion of an applet instance is requested. This method must be implemented by all applications and used to set all static object references to 'null'.

3 Usage of APIs and USIM Toolkit Commands and Events

This chapter lists the APIs and the CAT/USAT commands and events that are:

- "Limited" which means that this method, command or event is sensitive and its usage by the application must be described in the certification documentation provided to the MNO.
- "Forbidden" which means that this method must not be used.

3.1 Java Card

- Rule 19: The Applet must obey the following restrictions on the use of the Java Card API (see section 6 of GSMA SGP.03 NFC UICC Requirement Specification [1] for the list of implemented Java Card API):

Packages	Class/ Interfaces	Method	Rules
java.rmi	ALL	ALL	⛔ **Forbidden**
javacard.framework	APDU	ALL	Cf. process(APDU)
	Applet	getShareableInterfaceObject	**Limited**
	JCSystem	lookupAID()	**Limited**
		abortTransaction()	**Limited**
		getPreviousContextAID()	**Limited**
		getAppletShareableInterfaceObject()	**Limited**
	MultiSelectable		**Limited**
	OwnerPIN	setValidatedFlag(boolean)	⛔ **Forbidden**
		reset()	**Limited**
		update(byte[], short, byte)	**Limited**
		resetAndUnblock()	**Limited**
	Shareable	ALL	**Limited**
javacard.framework.service			⛔ **Forbidden**

3.2 GlobalPlatform

✓ Rule 20: The Applet must obey the following restrictions on the use of the GlobalPlatform API:

Packages	Class/Interface	Method	Rules
org.globalplatform.GPSystem	GPRegistryEntry	getRegistryEntry	**Limited**
		lockCard()	⛔ **Forbidden**
		setATRHistBytes	**Limited**
		setCardContentState	⛔ **Forbidden**
		terminateCard()	⛔ **Forbidden**
		deregisterService	**Limited**
	CVM	all except: - verify(), - getTriesRemaining(), - is*()	**Limited**
org.globalplatform.contactles	CRELApplicatio	notifyCLEvent(…)	**Limited**

Packages	Class/Interface	Method	Rules
s	n		
	CRSApplication	processCLRequest(…)	⛔ **Forbidden**
	GPCLSystem	getCardCLInfo(…)	⛔ **Forbidden**
		getGPCLRegistryEntry(…)	⛔ **Forbidden***
		getNextGPCLRegistryEntry(…)	⛔ **Forbidden**
		setCommunicationInterface(…)	⛔ **Forbidden**
		setVolatilePriority(…)	⛔ **Forbidden**

***Note: Limited when parameter is Null.

Rule 65 (Limited): GP Contactless Self-Activation privilege.

Whenever the usage of the privilege is agreed and recommended by the MNO, applets shall be set to support the GP Contactless Self-Activation privilege.

This implementation makes sense if the NFC Service customer experience expects, at the end of the installation process or after a lock / unlock procedure:

- Either, a first Mobile Application – Applet interaction, including the application selection on ISO interface, prior to any card emulation transaction;
- Or an APDU command sent OTA, in order to request the Applet to activate its contactless state by itself.

It is recommended that the SP retrieves the contactless state of the Applet at first, prior to setting the self-activation privilege:

GPCLSystem.getGPCLRegistryEntry(null).getCLState()

GPCLSystem.getGPCLRegistryEntry(null).setCLState(ctlessState)

NOTE: In accordance with Rule 20, use of all other parameter values in method “getGPCLRegistryEntry” is forbidden.

NOTE: During installation stage, if instantiation and activation are performed using 2 different commands (i.e. INSTALL [for install] then INSTALL [for make selectable]), the GP Contactless Self-Activation privilege has to be set.

This setting needs to be agreed between the service provider and the MNO, due to the implementation of some SIM cards among those actually distributed. This statement applies for example to SIM cards implemtingen the version 1.0 of GP Amendment C.

3.3 UICC API

- Rule 21: The Applet must obey the following restrictions on the use of the UICC API:

Packages	Class/interfaces	Method	Rules
uicc.access	FileView	ALL	**Limited**
	UICCSystem	ALL	**Limited**
uicc.access.bertlvfile	ALL	ALL	**Limited**
uicc.access.fileadministration	ALL	ALL	⛔ **Forbidden**
uicc.toolkit	ALL	ALL	See the tables below for Commands and Events
uicc.hci.framework	HCIService	requestCallBackNotification	⛔ **Forbidden**

✓ Rule 22: The Applet must obey the following restrictions on the use of the USIM Toolkit commands:

Proactive Commands	Usage
PRO_CMD_LAUNCH_BROWSER	**Limited**
PRO_CMD_PERFORM_CARD_APDU	**Limited**
PRO_CMD_POWER_OFF_CARD	**Limited**
PRO_CMD_PROVIDE_LOCAL_INFORMATION	**Limited**
PRO_CMD_RUN_AT_COMMAND	**Limited**
PRO_CMD_SEND_DATA	**Limited**
PRO_CMD_SEND_DTMF	**Limited**
PRO_CMD_SEND_SHORT_MESSAGE	**Limited**
PRO_CMD_SET_UP_CALL	**Limited**
PRO_CMD_CLOSE_CHANNEL	**Limited**
PRO_CMD_GET_CHANNEL_STATUS	**Limited**
PRO_CMD_OPEN_CHANNEL	**Limited**
PRO_CMD_RECEIVE_DATA	**Limited**
PRO_CMD_SEND_DATA	**Limited**
PRO_CMD_REFRESH	⛔ **Forbidden**

NOTE: All other proactive commands could be allowed; Nevertheless for ensuring the functional interoperability between the STK applets, the MNO shall validate the list of the Toolkit proactive commands that the SP wants to use for their application. MNO shall provide to SP a document to fulfil

- Rule 23: The Applet must obey the following restrictions on the use of the SIM Toolkit events:

Event	Usage
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_NAA	⛔ **Forbidden**
EVENT_CALL_CONTROL_BY_NAA	⛔ **Forbidden**
EVENT_EVENT_DOWNLOAD_MT_CALL	⛔ **Forbidden**
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED	⛔ **Forbidden**
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	⛔ **Forbidden**
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	⛔ **Forbidden**
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION	**Limited**
EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION	**Limited**
EVENT_EVENT_DOWNLOAD_BROWSING_STATUS	**Limited**
EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS	**Limited**
EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	**Limited**

NOTE: All other Events could be allowed; Nevertheless for ensuring the functional interoperability between the STK applets, the MNO shall validate the list of the Toolkit Event commands that the SP wants to use for their application. MNO shall provide to SP a document to fulfil.

3.4 SIM API

- Rule 24: SIM API (TS 43.019) is obsolete and it is forbidden to use it.

4 Memory Management

4.1 Basic Recommendations

- Rule 25: Void.
- Rule 26: Applications should not rely exclusively on the object deletion feature because deletion:
    - It may be time-consuming, there is no guarantee to free instantly the memory, and the result is not guaranteed on some cards. Yet, it is foreseen that some applications will need to manage dynamically some information (such as transport ticket, e-coupon etc.). In case object deletion is necessary, it is recommended that the application requests to the operating system for object deletion at the end of its process.
- Rule 27: Use the scratch buffer shall be limited for local computations and shall be forbidden for exchanges between applets.

The scratch buffer accessible through UICCPlatform.getTheVolatileByteArray could be used to share data with applications belonging to different Java Card contexts; usage of this buffer should be restricted to local (not shared) computations.

- Rule 28: Allocate arrays with a determined size.

To prevent unexpected memory failures, applications should only allocate arrays of a fixed length, i.e., the array size should not be dynamically calculated. Thus, the usage of constant values is recommended.

- Rule 29: Void.

4.2 Memory Protection

- Rule 30: Void.
- Rule 31: Recursive code is forbidden.
    - Recursion may lead to stack overflow, especially in Java Card, and it should be avoided altogether.

4.3 Memory allocation/de-allocation (RAM, Flash, EEPROM)

- Rule 32: Objects allocated outside the installation phase shall be a singleton.
    - In case the application needs to allocate objects after applet registration either for functional or performance reasons, it is mandatory to protect the allocation with a singleton to avoid allocation repetition.

5 System Management

5.1 System File Access

- Rule 33: Files authorised in restricted accesses (subjected to the MNO agreement) are specified below; access to all other files is forbidden.

ID1	ID2	ID3	Name	Limited access	Conditions
3F00			MF	-	
2FE2			ICCID	Read	
7F10			TELECOM	-	
	6F3A		ADN	Read/Write	Under user control
	6F3C		SMS	Read	Under user control
	6F44		LND	Read	Under user control
	5F3A		PUBLIC PHONE BOOK	-	Under user control
		4F30	PBR	Read	Under user control
		4F3A	ADN	Read/Write	Under user control
		4F09	PBC	Read/Write	Under user control
		4F40	ANR1	Read/Write	Under user control
		4F60	SNE1	Read/Write	Under user control
		4F61	SNE2	Read/Write	Under user control
		4F50	EMAIL	Read/Write	Under user control

ID1	ID2	ID3	Name	Limited access	Conditions
		4F10	UID	Read/Write	Under user control
		4F22	PSC	Read/Write	Under user control
		4F23	CC	Read/Write	Under user control
		4F24	PUID	Read/Write	Under user control
7F20			GSM	-	
	6F07		IMSI	Read	
	6F14		CPHS ONS :	Read	
	6F46		SPN	Read	
7FFF (*)			USIM		
	6F07		IMSI	Read	
	6F3C		SMS	Read	Under user control
	6F46		SPN	Read	
	6F80		ICI	Read	Under user control
	6F81		OCI	Read	Under user control
	6F82		ICT	Read	Under user control
	6F83		OCT	Read	Under user control
	6FCE		MMSN	Read	Under user control
	6FD0		MMSICP	Read	Under user control
	6FD3		NIA	Read	Under user control

5.2 File event and management

- Rule 34: Applications shall only access ADF's and files with determined identifiers.
- Rule 35: Constants must be used for AID values (except application instance AID).
- Rule 36: Do not access files controlled by the operator.
    - Accesses to the file system are handled using FileView objects: UICC view or an ADF view. The UICC view provides access to files that are under the control of the operator, these files should not be accessed by applications, except for the few exceptions listed below: files involved in the management of the phone book, and files involved in the management of messaging.
- Rule 37: Only register for file update events on accessible files.
    - This guideline extends the file access restriction guidelines to the usage of file events with the toolkit registry. The usage of method ToolkitRegistry.registerFileEvent, should be limited to events on application-specific files (under its ADF). Applications must not listen to event on files controlled by the operator.
- Rule 38: The application should verify the content written in phone book files.

- The phone book structure is based on several additional elementary files, to fully manage contacts instead of simple couple's name/dialling number.
    - EFADN: Abbreviated Dialling Number
    - EFANR: Additional Number
    - EFAAS: Additional number Alpha String
    - EFEXT1: Extension 1
    - EFCCP1: Capability Configuration Parameters 1
    - EFEMAIL: Email Address
    - EFSNE: Second Name Entry
    - EFGRP: Grouping File
    - EFGAS: Grouping information Alpha String etc.

The framework does not provide any specific command for the management of these files and the data manipulation inside a file record is difficult. In addition, the information in these files should only consist of standard characters, without any control characters. Applications should verify the content written in phone book files to prevent erroneous or malformed data to be written.

- Rule 39: Do not resize files after their creation.
    - The size of files should not be modified after their creation. Applications should not use the FileAdminView.resize method; instead the application should inform the end user that no more space is available.

6 Exception and Error Management

- Rule 40: Exceptions should be used to handle errors and exceptional situations.
    - The Java Card specific exception mechanism (throwIt methods and reason arguments) should be used systematically.
- Rule 41: Catch all exceptions in library code.
    - No exception must reach the JCRE, apart from ISOException instances that are thrown in response to an APDU. The only exception in ISOException instances are the one with the specific status word REPLY_BUSY, which can be sent in response to some events.
- Rule 42: Never throw runtime exceptions explicitly.
    - Applications must only explicitly throw non-runtime exceptions (in Java Card, it is recommended to use the UserException), except for ISOException. Non-runtime exceptions need to be caught by the application, which is a good practice.
- Rule 43: Always throw and catch specific exception types.
- Rule 44: Avoid defining application-specific exceptions.
    - If necessary, use the standard pattern, and catch them all.

    - The application behaviour shall be specified and implemented in all possible erroneous conditions.

7 Interactions

7.1 Handset (USAT, USB)

- Rule 45: Do not use ISOException (REPLY_BUSY) as a conclusion to event processing.

    - Exceptions thrown by toolkit applications are all caught by the CAT Runtime Environment, except instances of ISOException, with reason REPLY_BUSY, which are propagated to the terminal. Because of interoperability issues, this mechanism should not be used by applications.
    - The Java Card requirement also applies to any developed library code. Applications should avoid using transaction features or at least should strictly ensure that:

- Rule 46: Don’t use potential hidden channels.

    - All communication mechanisms used for application communication shall be standard.

- Rule 47: The usage of static fields should be strictly limited to the definition of “public final static” primitive constants.
- Rule 48: Do not use objects that implement system interfaces shared by another application.
- Rule 49: Register Events and initialize USAT menu during installation.

7.2 External Interaction (RMI, network)

- Rule 50: Do not use Java Card RMI.

    - The usage of the Java Card RMI mechanism is prohibited, because it lacks security related features (e.g. authentication and secure channels).

8 Development Rules

- Rule 51: GP Privileges: a basic application must not have the following GlobalPlatform privileges:

    - SD and associated privileges
    - Card lock
    - Card terminate
    - Card reset
    - CVM Management
    - Global Delete
    - Global lock
    - Global registry
    - Final application

- Rule 52: Telecom Applets: Either in 2G or 3G domains, a basic application shall never be assigned an access domain (for UICC file system or for any ADF, in normal access or administrative access) that gives more right than needed (“least privilege” principle). A basic application not accessing any filesystem shall be installed with "no access" as access domain parameter as specified in ETSI TS 102 226 clause 8.2.1.3.2.5.1.
- Rule 53: Always include a default case in a switch statement.
    - No case should be ignored. If there is no specific handling to do, it is possible to declare a default clause at the end of the switch statement and to simply include a comment indicating that there is nothing to do.
    - Rule 54: Follow the ISO7816 specifications.

An application shall only accept commands and only return status words that are valid according to ISO7816-4 specification.

- CLA should not be 0xFF.
- INS most significant nibble should not be 0x6 or 0x9.
- The most significant byte of status word should be in ranges 0x[62..6B], 0x[6D..6F] and 0x[91..9E].

- Rule 55: Mask the low-order CLA bits.
    - Since the Java Card 2.2 specification, different applets can be selected at the same time on different logical channels. As defined in ISO7816-4 specification, the two least significant bits of the CLA byte are used to indicate the channel number. To prevent any command recognition issue, applications need to ignore the channel bits, for instance masking it with byte 0xFC.

- Rule 56: Don’t register to call and SMS control events.
    - In addition to the fact that these operations may be crucial for the operator, the behaviour of devices with respect to these events (EVENT_CALL_CONTROL_BY_NAA and EVENT_-MO_SHORT_MESSAGE_CONTROL_BY_NAA) is not interoperable. Applications should therefore not register to these two events.

This information is already presented in the tables of Chapter 3.3.

- Rule 57: Don’t register to proprietary events.
    - This is a basic portability guideline: applications should not manage proprietary events.

- Rule 58: Void.
- Rule 59: The application should protect all accesses to handlers by an exception handler.

    - Because handlers are not always available (e.g., re-entrance case) and a ToolkitException with reason HANDLER_NOT_AVAILABLE may be thrown, implementations should protect all accesses to handlers in a try-catch block.

- Rule 60: The application should properly declare the attributes of classes, fields, and methods.

    - All classes, fields, and methods should be declared as privately as possible, and final whenever possible.

- Rule 61: All constants should be named and declared as static final fields.

    - Naming constants is important for readability and maintenance, and declaring them properly is important for size optimization and performance.

- Rule 62: An identifier shall be used in parameter of the function calls.

    - Identifier usage allows easy tracking of applications calls. This is useful if a new attack path is discovered to identify the pieces of code that could be vulnerable.

Annex A Synthesis of rules

Description	Mandatory	Recommended
Rule 1: Interface management and Data management must be clearly separated.	X	
Rule 2: The interactions between different interfaces must be clearly defined.	X	
Rule 3: Usage of the shareable interface is forbidden except with explicit MNO agreement.	X	
Rule 4: Package design: A basic applet shall never be in the same package as any sensitive applet [7].	X	
Rule 5: Entry points: All the different entry points shall be clearly defined at, for instance: APDU level, USIM Toolkit level, OTA script level, Personalisation level, Shared methods level.	X	
Rule 6: Denial of Service: Developers will be especially careful to avoid practices leading any denial of service as, for instance:	X	
Rule 7: Dead code or debug information must be deleted/removed from the code.	X	
Rule 8: Files imports: The file imports versions must fit to the versions (and method signatures) installed on the targeted cards, according to the binary compatibility rules described in Java Card 3.0.1 Classic Edition VM Architecture Specification. Usage of any custom component (i.e. a component which is neither standardised nor part of the Installation package) is not allowed.	X	
Rule 9: Service provider applications shall comply with the mandatory rules defined in Composition Model Security Guidelines for Basic Applications [7].	X	
Rule 10: Coding User Input: A basic application shall verify that each user entry is in the range of accepted values.	X	
Rule 11: Coding NOP: A basic application bytecode shall not include any NOP (NO Operation) instruction.	X	
Rule 12: Version number increment: A basic application with the same AID as an already verified basic application but with different exported methods signatures shall have an incremented major version.	X	
Rule 13: ByteCode verification: In order to be protected against attacks on card resources, a basic application must be strictly in accordance with the Java Card specifications and must successfully pass the latest ByteCode verification tool from Oracle in JCDK latest version.	X	
Rule 14: Bytecode Verification Used Export Files: If a basic application imports some packages (sharing or libraries), the export file used for the verification shall corresponds to the export file previously verified by the MNO or to the Certification Entity and shall not be provided by the Service Provider.	X	

Description	Mandatory	Recommended
Rule 15: Persistent Objects (using EEPROM) must be allocated during the application installation to avoid exceptions during runtime due to shortage of UICC memory.	X	
Rule 16: Applet instance registration should occur as late as possible.		X
Rule 17: Menu entries shall be initialised only in install() method.	X	
Rule 18: Implement the uninstall method in Java Card 3.0.1 classic edition applets.	X	
Rule 19: The Applet must obey the following restrictions on the use of the Java Card API:	X	
Rule 20: The Applet must obey the following restrictions on the use of the GlobalPlatform API:	X	
Rule 21: The Applet must obey the following restrictions on the use of the UICC API:	X	
Rule 22: The Applet must obey the following restrictions on the use of the USIM Toolkit commands:	X	
Rule 23**:** The Applet must obey the following restrictions on the use of the SIM Toolkit events:	X	
Rule 24: SIM API (TS 43.019) is obsolete and it is forbidden to use it.	X	
Rule 25: Void.		
Rule 26: Applications should not rely exclusively on the object deletion feature because deletion:		X
Rule 27: Use the scratch buffer for local computations, not for exchange between applets.	X	
Rule 28: Allocate arrays with a determined size.	X	
Rule 29: Void.		
Rule 30: Void.		
Rule 31: Recursive code is forbidden.	X	
Rule 32: Only allocate objects in the installation phase or with a singleton.	X	
Rule 33: Files authorised in restricted accesses (subject to the MNO agreement) are specified below; access to all other files is forbidden.	X	
Rule 34: Only access ADFs and files with determined identifiers.	X	
Rule 35: Constants must be used for AID values (except application instance AID).	X	
Rule 36: Do not access files controlled by the operator.	X	
Rule 37: Only register for file update events on accessible files.	X	
Rule 38: The application should verify the content written in		X

Description	Mandatory	Recommended
phone book files.		
Rule 39: Do not resize files after their creation.	X	
Rule 40: Exceptions should be used to handle errors and exceptional situations.		X
Rule 41: Catch all exceptions in library code.	X	
Rule 42: Never throw runtime exceptions explicitly.	X	
Rule 43: Always throw and catch specific exception types.	X	
Rule 44: Avoid defining application-specific exceptions.		X
Rule 45: Do not use ISOException (REPLY_BUSY) as a conclusion to event processing.	X	
Rule 46: Do not use potential hidden channels.	X	
Rule 47: Thus, no shared objects neither public static fields should be defined in library code.	X	
Rule 48: Do not use objects that implement system interfaces shared by another application.	X	
Rule 49: Register Events and initialize USAT menu during installation	X	
Rule50: Do not use Java Card RMI.	X	
Rule 51: GP Privileges: a basic application must not have the following GlobalPlatform privileges:	X	
Rule 52: Telecom Applets: Either in 2G or 3G domains, a basic application shall never be assigned an access domain (for UICC file system or for any ADF, in normal access or administrative access) that gives more right than needed (“least privilege” principle).	X	
Rule 53: Always include a default case in a switch statement.	X	
Rule 54: Follow the ISO7816 specifications.	X	
Rule 55: Mask the low-order CLA bits.	X	
Rule 56: Don’t register to call and SMS control events.	X	
Rule 57: Don’t register to proprietary events.	X	
Rule 58: Void.		
Rule 59: The application should protect all accesses to handlers by an exception handler.		X
Rule 60: The application should properly declare the attributes of classes, fields, and methods.		X
Rule 61: All constants should be named and declared as static final fields.		X
Rule 62: An identifier shall be used in parameter of the function calls.	X	
Rule 63: STK applet, priority level setting.	X	
Rule 64: Minimum security level.		X

Description	Mandatory	Recommended
Rule 65 : GP Contactless Self-Activation privilege.		X

Annex B User Interaction Parameters for CRS (Normative)

The GlobalPlatform Contactless Services defined in [5] introduces in section 3.11.2 the data stored within the CRS application in order to interact with the user.

The list of User Interaction Parameters that shall be provided by the Service provider in case of Delegated Mode Applet loading and installation is the following:

- Application AID;
- Uniform Resource Locator;
- Display Message (The encoding format shall be UTF-8);
- Application Family;

Annex C Document Management

C.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	02 July 2012	Submitted to PSMC for approval, final approval date 30th July 2012	NFC/PSMC	Davide Pratone, Telecom Italia
2.0	03 October 2013	Submitted to PSMC#116 for approval	NFC/PSMC	Davide Pratone, Telecom Italia
2.0	31 March 2014	Transferred from NFC Fast Track project to SIM Group	SIM Group	Davide Pratone, Telecom Italia
3.0	25 June 2014	Approved by SIM Group and submitted to PSMC for approval	PSMC	Davide Pratone, Telecom Italia
3.1	10 June 2015	Approved by NFCSIM WI1 and submitted to NFCSIM for approval	NFCSIM	Davide Pratone, Telecom Italia
3.2	11 June 2015	Version 3.2 updated by NFCSIM and submitted for 14 days review by delegates	NFCSIM	Davide Pratone, Telecom Italia
3.3	29 June 2015	Version 3.3 updated by NFCSIM for CR creation	NFCSIM	Davide Pratone, Telecom Italia
4.0	30th September 2015	Published at version 4.0	NFCSIM	Davide Pratone, Telecom Italia

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com.

Your comments or suggestions & questions are always welcome.