GLOBALPLATFORM
THE STANDARD FOR SMART CARD INFRASTRUCTURE

Card Specification 2.1.1 and Amendment A Errata & Precisions List 1.3

December 2004

Table of Contents

Table of Errata and Precisions

This 1.2 release of the Errata and Precisions List applies to both versions 2.1.1 and 2.1 of GlobalPlatform Card Specification. As version 2.1.1 of the specification includes all the Errata and Precisions prior to the Errata and Precisions List release 1.1 (see GlobalPlatform Card Specification 2.1.1 Release Notes for an exhaustive list), this Errata and Precisions List only reflects those issued since the publication of version 2.1.1 in March 2003.

The following table classifies all the Errata and Precisions of the different Errata and Precisions Lists release 1.x into a sequential order that reflects the Card Specification index. **The latest Errata and Precisions are in blue characters**.

Errata / Precision number	Card Specification section number	Description
P.13.1	section 6.3.1.1.2	Selection attempt of an Application in the LOCKED state
P.13.2	section 6.4.1.1	INSTALL [for load] without Security Domain AID
P.13.3	section 6.4.2.1	Security Domain deletion rules
P.13.4	section 6.7.3	Terminating the card
E.13.2	section 6.9 & appendix A.2	GlobalPlatform API: CVM State with setTryLimit() & update() methods and default value for Retry Limit
P.13.5	section 7.3	Security Domain personalization support rules
P.11.1	section 7.6.3	Extradition to the Issuer Security Domain
P.11.2	section 9.1.5	Data objects length indicators
P.12.3	section 9.1.5	Meaning of Le byte
P.11.3	section 9.3.2.2	GET DATA [Sequence Counter] for Secure Channel Protocol '01'
P.11.4	section 9.6.2.3	Load File length fields coding
E.13.1	section 9.7.2	Le for MANAGE CHANNEL command
P.13.6	sections 9.7.2.2 & 3.2.1	MANAGE CHANNEL command: assignment by the terminal
P.13.7	amendment A.1.2 (re: section 9.11.2.3)	DGI for Issuer Security Domain data
P.13.8	appendix A.2	GlobalPlatform on a Java Card: select() method error
P.11.5	appendix A.2	GlobalPlatform API: getSecureChannel()
E.12.1	appendix A.2	GlobalPlatform API: unwrap() behavior in case of C-MAC error
P.13.9	appendix A.2	GlobalPlatform API: unwrap() method returned data
P.13.10	appendix A.2	GlobalPlatform API: wrap() method processing rules
P.13.11	appendix A.2	GlobalPlatform API: encryptData() method processing rules
P.12.1	appendices A.2, D.1 & E.1	Secure Channel Protocol rules and GlobalPlatform API
E.13.3	appendix B.1.2.2	Reference for Retail MAC
P.11.6	appendix C.1.1	Absence of Token or Receipt keys
P.13.12	appendix E.1.1 & amendment A.2.2	R-MAC support indicator
P.11.7	appendix E.4.5	R-MAC for case 1 & case 3 commands
P.12.2	appendix E.4.5	R-MAC in case of error status words

E. ERRATA LISTS

E.11 ERRATA LIST 1.1

(None)

E.12 ERRATA LIST 1.2

E.12.1 GlobalPlatform API: unwrap() behavior in case of C-MAC error

The unwrap() method of the SecureChannel interface of the GlobalPlatform API for Java Card is expected to throw the exception: ISO7816.SW_SECURITY_STATUS_NOT_SATISFIED if secure messaging verification failed, e.g. for a C-MAC verification error. According to the secure communication rules (see *section 8.2.3 – Secure Channel Termination*), the Secure Channel Session shall be terminated and the Security Level reset to 'NO_SECURITY_LEVEL'. The wording of the 3rd bullet of the method is confusing as to the expected behavior in processing a subsequent incoming command, which must be rejected with a security error for both Secure Channel Protocols '01' and '02'.

Note this erratum is **already tested in GlobalPlatform Card Test Kit version 1.1**, which provides test scripts for version 2.1 of GlobalPlatform Card Specification.

In *appendix A.2 – GlobalPlatform on a Java Card*, add a new bullet and complement the 3rd bullet of the processing notes of the unwrap() method of the SecureChannel interface as follows:

- **"Processing this method shall comply to the rules of the Secure Channel Protocol implemented by the Security Domain**.
- If NO_SECURITY_LEVEL or AUTHENTICATED only is indicated, **when complying to the Secure Channel Protocol rules,** this method does not attempt any secure messaging processing on the incoming command, the incoming command remains as is within the incoming APDU object **and** the returned length of the "unwrapped" data is set to the value of the sLength parameter, **otherwise a security error is returned**."

E.13 ERRATA LIST 1.3

E.13.1 Le for MANAGE CHANNEL command

In *section 9.7.2 – MANAGE CHANNEL Command Message*, in accordance with ISO/IEC 7816-4, the parameter Le shall be **set to '01'** (not '00') when opening the next available Supplementary Logical Channel (P1 = '00').

Table 9-43 - MANAGE CHANNEL Command Message shall read as follows:

Code	Value	Meaning
CLA	'00'	ISO/IEC 7816-4 command
INS	'70'	MANAGE CHANNEL
P1	'xx'	Reference control parameter P1
P2	'xx'	Reference control parameter P2
Lc		Not present
Le		**'01'** if P1 = '00' and not present if P1 = '80'

E.13.2 GlobalPlatform API: CVM State with setTryLimit() & update() methods and default value for Retry Limit

According to section 6.9.1.1, to be in the CVM State ACTIVE, both the CVM value and Retry Limit must be set. The notes for the update() and setTryLimit() methods of the CVM interface of the GlobalPlatform API for Java Card are incomplete in the case of a CVM not yet fully operational. If the card implements the Deprecated Open Platform Java Card API, the Retry Limit should be set by default.

Section 6.9 – CVM Management shall be completed as follows:
"**Depending on the Issuer policy, a value for the Retry Limit may be set by default**."

The first sentence of *section 6.9.1.1 – CVM State ACTIVE* shall be revised to read as follows:
"The CVM state shall first become ACTIVE when **both** the CVM value **and** the Retry Limit **are set**."

In *Appendix A.2 – GlobalPlatform on a Java Card*, the 4th bullet in the notes for the update() method of the CVM interface shall be completed as follows:

- "The CVM state is reset to ACTIVE when changing the CVM value. **When setting the CVM value before the CVM state is ACTIVE, the CVM state transitions to ACTIVE only if the Retry Limit is already set**."

In *Appendix A.2 – GlobalPlatform on a Java Card*, the 3rd bullet in the notes for the setTryLimit() method of the CVM interface shall be completed as follows:

- "The CVM state is reset to ACTIVE when **changing** the maximum number of tries. **When setting the maximum number of tries before the CVM state is ACTIVE, the CVM state transitions to ACTIVE only if the CVM value is already set**."

The isActive() method of the CVM interface of the GlobalPlatform API for Java Card shall return false not only if the CVM is not present but also if it is not yet fully operational (i.e. both the CVM value and Retry Limit have not been set).

In *Appendix A.2 – GlobalPlatform on a Java Card*, the words "(**i.e. does not exist**)" shall be **deleted** from the description of the returned information for the isActive() method of the CVM interface so that to read as follows:
"Returns:
true if the CVM state is **either** ACTIVE, **INVALID_SUBMISSION, VALIDATED or BLOCKED**, false otherwise_."

In *Appendix A.2 – GlobalPlatform on a Java Card*, the description of the returned information for the isSubmitted() method of the CVM interface shall be clarified as follows:
"Returns:
true if the CVM state is **either** INVALID_SUBMISSION **or VALIDATED**, false otherwise."

E.13.3 Reference for Retail MAC

The reference in *appendix B.1.2.2 – Single DES plus Final Triple DES MAC* is incorrect and shall be: "Algorithm 3 (and not Algorithm 1) with output transformation 3".

Appendix B.1.2.2 – Single DES plus Final Triple DES MAC shall read as follows:
"This is also known as the Retail MAC. It is as defined in [ISO 9797-1] as MAC **Algorithm 3** with output transformation 3, without truncation, and with DES taking the place of the block cipher."

P. PRECISIONS LISTS

P.11 PRECISIONS LIST 1.1

P.11.1 Extradition to the Issuer Security Domain

According to *section 6.4.3 – Content Extradition*, there is no restriction for extraditing Applications to the Issuer Security Domain like any other Application Provider Security Domain, other than the Issuer Security Domain accepting the Application extradition.

The first paragraph of *section 7.6.3 – Delegated Extradition* shall be completed with the following sentence:
"**Extradition may apply for any Security Domain (in the PERSONALIZED state) or the Issuer Security Domain (in any Card Life Cycle State other than CARD_LOCKED and TERMINATED), that accepts the extradited Application**."

P.11.2 Data objects length indicators

Section 9.1.5 – APDU command and response data shall be completed with the following sentence:
"**All length fields of GlobalPlatform messages and data objects provide the unsigned number of bytes used for the message or data value field**."

P.11.3 GET DATA [Sequence Counter] for Secure Channel Protocol '01'

Since the Secure Channel Sequence Counter is only defined for Secure Channel Protocol '02' per *appendix E.1.2 – Secure Channel Protocol '02' Entity Authentication*, the GET DATA [Sequence Counter] command must be supported only when Secure Channel Protocol '02' is supported. For a (Issuer) Security Domain supporting only Secure Channel Protocol '01', a GET DATA command on the Sequence Counter for the default Key Version Number (tag 'C1') shall fail.

Section 9.3.2.2 – GET DATA command parameters P1 and P2 shall be completed as follows:
"**For a Security Domain or Issuer Security Domain supporting only Secure Channel Protocol '01', the attempt to retrieve the Sequence Counter for the default Key Version Number (tag 'C1') shall be rejected.**"

P.11.4 Load File length fields coding

According to *section 9.6.2.3 – Data field sent in the LOAD command*, the Load File contains BER-TLV coded data elements: no, one or more DAP Blocks (no, one or more tags 'E2') and one Load File Data Block (one tag 'C4'). The length fields or these data elements shall follow the BER-TLV coding rules, that is: one byte for length values up to 127, two bytes for length values between 128 and 255, etc.

The third sentence of *section 9.6.2.3 – Data field sent in the LOAD command* shall be completed as follows:
"The data objects defined in Table 9-40 shall be coded according to ASN.1 Basic Encoding Rules, **and in particular, the length fields (e.g. one byte for length values up to 127)**."

P.11.5 GlobalPlatform API: getSecureChannel()

The getSecureChannel() method of the GPSystem class of the GlobalPlatform API for Java Card always returns the handle of the associated Security Domain, regardless of the Life Cycle State of the associated Security Domain. According to *section 5.3.2 – Security Domain Life Cycle States*, only the services of the Security Domain are not available when the Security Domain Life Cycle State is different from PERSONALIZED.

In *appendix A.2 – GlobalPlatform on a Java Card*, complement the last bullet of the processing notes of the getSecureChannel()method of the org.globalplatform.GPSystem class:

- "The OPEN locates the entry of the current applet context in the GlobalPlatform Registry to determine the Application's associated Security Domain, **regardless of the Security Domain Life Cycle State**."

P.11.6 Absence of Token or Receipt keys

When Delegated Management is supported, delegated loading, delegated installation and delegated extradition shall fail if the Token verification key is not present on-card (Issuer Security Domain). When Receipt generation is supported, delegated loading, delegated installation, delegated extradition and delegated deletion shall fail if the Receipt generation key is not present on-card (Issuer Security Domain).

Appendix C.1.1.1 – Token key shall be completed as follows:
"**Token verification and the corresponding delegated management operation shall fail if the Token verification key is not present**."

Appendix C.1.1.2 – Receipt key shall be completed as follows:
"**Receipt generation and the corresponding delegated management operation shall fail if the Receipt generation key is not present**."

P.11.7 R-MAC for case 1 & case 3 commands

When a R-MAC session is started requiring a R-MAC to be returned for every response to each command (P1 set to '1x' in the EXTERNAL AUTHENTICATE command or P1 set to '10' in the BEGIN R-MAC SESSION command of Secure Channel Protocol '02', see *appendix E.1.2 – Secure Channel Protocol '02' APDU commands*), the off-card system is expected to be aware of it and to provide Le in the command message: case 1 and case 2 commands become respectively case 2 and case 4 commands. For such a R-MAC session, regardless of the off-card system actual behavior and regardless of the case of the command – i.e. including case 1 and case 3 – according to *appendix E.4.5 – APDU Response R-MAC Generation and Verification*, the card shall compute a R-MAC and return it in every response message.

In other words, the C-APDU message of case 1 and 3 commands shall be processed by the card as a case 2 or case 4 command message with Le set to '00'.
Note: the R-MAC computation does not include Le, but the actual length (Li) of the response data field.

The third paragraph of *appendix E.4.5 - APDU Response R-MAC Generation and Verification* shall be completed with the following sentence:
"**When a R-MAC is returned in response to every command, in the event of case 1 and case 3 commands received by the card, these commands shall be processed respectively as case 2 and case 4 commands with Le set to zero**."

P.12 PRECISIONS LIST 1.2

P.12.1 Secure Channel Protocol rules and GlobalPlatform API

In *appendix A.2 – GlobalPlatform on a Java Card*, add a new bullet to the processing notes of the **processSecurity(), wrap(), encrypt(),** and **decrypt()** methods of the SecureChannel interface as follows:

- **"Processing this method shall comply to the rules of the Secure Channel Protocol implemented by the Security Domain**."

To clarify the secure communication rules of Secure Channel Protocol '01', add the following new *appendix D.1.x – Protocol Rules*:

"In accordance with the general rules described in *section 8 – Secure Communication*, the following protocol rules apply to Secure Channel Protocol '01':

- **The successful initiation of a Secure Channel Session shall set the Security Level to the requested security level indicated in the EXTERNAL AUTHENTICATE command: it is at least set to AUTHENTICATED.**
- **The requested security level shall apply to the entire Secure Channel Session.**
- **When the current Security Level is equal to NO_SECURITY_LEVEL:**
    - **If the requested security level is not reset (i.e. abort of a previous Secure Channel Session during the same Application Session), the incoming command shall be rejected with a security error.**
    - **Otherwise (i.e. no Secure Channel Session is currently active and no requested security level is set), no security verification of the incoming command shall be performed. The Application processing the command is responsible to apply its own security rules.**
- **If a Secure Channel Session is currently active (i.e. current Security Level at least set to AUTHENTICATED), the security of the incoming command shall be checked according to the current Security Level regardless of the command secure messaging indicator:**
    - **When the security of the command does not match (nor exceeds) the Security Level, the command shall be rejected with a security error, the Secure Channel Session aborted and the Security Level reset to NO_SECURITY_LEVEL. The requested security level shall not be reset.**
    - **If a security error is found, the command shall be rejected with a security error, the Secure Channel Session aborted and the Security Level reset to NO_SECURITY_LEVEL. The requested security level shall not be reset.**
    - **In all other cases, the Secure Channel Session shall remain active and the Security Level unmodified. The Application is responsible for further processing the command.**
- **The current Secure Channel Session shall be closed (if still active), the Security Level reset to NO_SECURITY_LEVEL and the requested security level reset on either:**
    - **Attempt to initiate a new Secure Channel Session (new INITIALIZE UPDATE command),**
    - **Termination of the Application Session (e.g. new Application selection),**
    - **Termination of the associated logical channel,**
    - **Termination of the Card Session (card reset or power off),**
    - **Explicit termination by the Application (e.g. invoking GlobalPlatform API)."**

To clarify the secure communication rules of Secure Channel Protocol '02', add the following new *appendix E.1.x – Protocol Rules*:

"In accordance with the general rules described in *section 8 – Secure Communication*, the following protocol rules apply to Secure Channel Protocol '02' with explicit initiation mode:

- **The successful explicit initiation of a Secure Channel Session shall set the Security Level to the requested security level indicated in the EXTERNAL AUTHENTICATE command: it is at least set to AUTHENTICATED.**
- **The requested security level shall apply to the entire Secure Channel Session.**
- **When the current Security Level is not set to AUTHENTICATED (i.e. equal to NO_SECURITY_LEVEL or R-MAC only – case of a R-MAC session in progress originally initiated with a BEGIN R-MAC SESSION command):**
    - **If the requested security level is not reset (i.e. abort of a previous Secure Channel Session during the same Application Session), the incoming command shall be rejected with a security error.**
    - **Otherwise (i.e. for incoming commands, no Secure Channel Session is currently active and no requested security level is set), no security verification of the incoming command shall be performed. The Application processing the command is responsible to apply its own security rules.**
- **If a Secure Channel Session is currently active for incoming commands (i.e. current Security Level at least set to AUTHENTICATED), the security of the incoming command shall be checked according to the current Security Level regardless of the command secure messaging indicator:**
    - **When the security of the command does not match (nor exceeds) the Security Level, the command shall be rejected with a security error, the Secure Channel Session aborted and the Security Level reset to NO_SECURITY_LEVEL. The requested security level shall not be reset.**
    - **If a security error is found, the command shall be rejected with a security error, the Secure Channel Session aborted and the Security Level reset to NO_SECURITY_LEVEL. The requested security level shall not be reset.**
    - **In all other cases, the Secure Channel Session shall remain active and the Security Level unmodified. The Application is responsible for further processing the command.**
- **The current Secure Channel Session shall be closed (if still active), the Security Level reset to NO_SECURITY_LEVEL and the requested security level reset on either:**
    - **Attempt to initiate a new Secure Channel Session (new INITIALIZE UPDATE command),**
    - **Termination of the Application Session (e.g. new Application selection),**
    - **Termination of the associated logical channel,**
    - **Termination of the Card Session (card reset or power off),**
    - **Explicit termination by the Application (e.g. invoking GlobalPlatform API).**

In accordance with the general rules described in *section 8 – Secure Communication*, the following protocol rules apply to Secure Channel Protocol '02' with implicit initiation mode:

- **The successful implicit initiation of a Secure Channel Session for incoming commands shall set the Security Level to AUTHENTICATED and C-MAC (the R-MAC indicator remaining as is).**
- **The requested security level for incoming commands is implicit and set to AUTHENTICATED for the entire Secure Channel Session. It is reset on the normal termination or abort of a Secure Channel Session for incoming commands.**
- **When the current Security Level is not set to AUTHENTICATED (i.e. equal to NO_SECURITY_LEVEL or R-MAC only), no Secure Channel Session is currently active for incoming commands:**
  - **If no secure messaging is indicated in the incoming command, no security verification of the command shall be performed. The Application processing the command is responsible to apply its own security rules.**
  - **If secure messaging is indicated, a new Secure Channel Session initiation for incoming commands shall be attempted and the security of the incoming command verified.**
- **If a Secure Channel Session is currently active for incoming commands (i.e. current Security Level at least set to AUTHENTICATED), the security of the incoming command shall be checked according to the current Security Level:**
  - **If no secure messaging is indicated in the command, the Secure Channel Session shall remain active, the Security Level set to AUTHENTICATED (the R-MAC indicator remaining as is) and no further security verification of the command performed. The Application processing the command is responsible to apply its own security rules.**
  - **If secure messaging is indicated and the current Security Level is set to AUTHENTICATED and C-MAC (ignoring the R-MAC indicator), the security of the incoming command shall be verified:**
    - **If a security error is found, the command shall be rejected with a security error, the Secure Channel Session aborted and Security Level reset to NO_SECURITY_LEVEL or R-MAC only (in case of a R-MAC session in progress).**
    - **Otherwise, the Secure Channel Session shall remain active and the Security Level unmodified. The Application is responsible for further processing the command.**
  - **If secure messaging is indicated and the current Security Level is set to AUTHENTICATED (ignoring the R-MAC indicator), the current Secure Channel Session for incoming commands shall be closed and the Security Level reset to NO_SECURITY_LEVEL or R-MAC only (in case of a R-MAC session in progress). A new Secure Channel Session initiation for incoming commands shall be attempted and the security of the incoming command verified.**
- **The current Secure Channel Session shall be closed (if still active) and the Security Level reset to NO_SECURITY_LEVEL on either:**
  - **Termination of the Application Session (e.g. new Application selection),**
  - **Termination of the associated logical channel,**
  - **Termination of the Card Session (card reset or power off),**
  - **Explicit termination by the Application (e.g. invoking GlobalPlatform API)."**

P.12.2 R-MAC in case of error status words

According to *appendix E.4.5 - APDU Response R-MAC Generation and Verification*, the R-MAC computation includes the status words regardless if the command is successfully processed or not. When a R-MAC session is started requesting a R-MAC to be returned for every response to each command (P1 set to '1x' in the EXTERNAL AUTHENTICATE command or P1 set to '10' in the BEGIN R-MAC SESSION command, see *appendix E.1.2 – Secure Channel Protocol '02' APDU commands*), the R-MAC value is returned as part of the APDU response message data field. In case of an error in processing the command, ISO/IEC 7816-4 prohibits data in an APDU response message with status words indicating an error.

When a R-MAC is requested with every response, the R-MAC shall be computed and returned in APDU response messages according to the following principles:

- The status words indicating a successful execution or a warning shall be included in the R-MAC computation and returned in the response trailer of the APDU response message.
- The status words indicating an error shall be included in the R-MAC computation. The APDU response message shall include only these status words: no R-MAC is returned with that response message. The corresponding R-MAC may be subsequently retrieved with an END R-MAC SESSION command.

The END R-MAC Session command is expanded to allow for retrieving the current R-MAC without ending the R-MAC Session: new value '01' (return R-MAC) of parameter P2.

The 3rd paragraph, 2nd bullet of *appendix E.4.5 - APDU Response R-MAC Generation and Verification* shall be completed as follows:

- "**In case of a successful execution or a warning,** the response data preceded with a byte that codes its length modulo 256,
- **In case of an error, a byte set to '00' indicating no response data**."

The 5th paragraph of *appendix E.4.5 - APDU Response R-MAC Generation and Verification* shall be completed as follows:
"The R-MAC can be retrieved from the card by sending an END R-MAC SESSION command. **The END R-MAC SESSION command allows the off-card entity to instruct the card to either terminate or continue** the R-MAC session. **The END R-MAC SESSION command is not included in the R-MAC computation**."

Table E-19 of *appendix E.5.4.4 - END R-MAC SESSION Reference Control Parameter P2* shall be expanded as follows:

b8	b7	b6	B5	b4	b3	b2	b1	Description
0	0	0	0	0	0	1	1	End R-MAC session **& return R-MAC**
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**Return R-MAC**

Table E-19: END R-MAC SESSION Reference Control Parameter P2

P.12.3 Meaning of Le byte

To clarify the meaning of the Le byte, *section 9.1.5 – APDU Command and Response Data* shall be completed with the following sentence:
"All GlobalPlatform commands that expect response data have their Le byte set to zero, indicating that all available response data shall be returned. According to ISO/IEC 7816-4, all GlobalPlatform responses returned in APDU response messages shall have a maximum length of 256 bytes."

P.13 PRECISIONS LIST 1.3

P.13.1 Selection attempt of an Application in the LOCKED state

The OPEN runtime behavior requirements need to be defined in the following eventuality:

- an Application is currently selected on a logical channel,
- the Life Cycle State of that Application is changed to LOCKED (e.g. SET STATUS command received by the Issuer Security Domain currently selected on another logical channel)
- a Select command matching that Application is sent on the same logical channel that the Application is currently selected on.

The 5th bullet of runtime behavior in *section 6.3.1.1.2 – Explicit Selection on the Basic Logical Channel* shall be split in two and completed as follows:

- "If a full or partial match is found and this Application is in the Life Cycle State INSTALLED, continue searching through the GlobalPlatform Registry for a subsequent full or partial match. If no subsequent full or partial match is found, the OPEN shall return the appropriate error to the off-card entity.
- If a full or partial match is found and this Application is in the Life Cycle State LOCKED, continue searching through the GlobalPlatform Registry for a subsequent full or partial match. **In the eventuality that this locked Application is already currently selected on the Basic Logical Channel, the OPEN shall terminate this Application Session.** If no subsequent full or partial match is found, the OPEN shall return the appropriate error to the off-card entity. **In the eventuality that the current Application Session shall be terminated and no subsequent full or partial match is found, the OPEN shall select the Application with the default selected privilege.**"

The 5th bullet of runtime behavior in *section 6.3.2.1.2 – Explicit Selection on Supplementary Logical Channel* shall be split in two and completed as follows:

- "If a full or partial match is found and this Application is in the Life Cycle State INSTALLED, continue searching through the GlobalPlatform Registry for a subsequent full or partial match. If no subsequent full or partial match is found, the OPEN shall return the appropriate error to the off-card entity.
- If a full or partial match is found and this Application is in the Life Cycle State LOCKED, continue searching through the GlobalPlatform Registry for a subsequent full or partial match. **In the eventuality that this locked Application is already currently selected on the same Supplementary Logical Channel, the OPEN shall terminate this Application Session.** If no subsequent full or partial match is found, the OPEN shall return the appropriate error to the off-card entity."

P.13.2 INSTALL [for load] without Security Domain AID

When no Security Domain AID is provided in the INSTALL [for load] command, the currently selected Security Domain performing the load is the Security Domain that will be associated to the Executable Load File upon successful completion of the load process.

The 1st bullet of runtime behavior in *section 6.4.1.1 – Card Content Loading* shall be completed with the following sub-bullet:

- "**If no associated Security Domain AID is indicated, the currently selected Security Domain that is performing the load is by default the associated Security Domain.**"

P.13.3 Security Domain deletion rules

Since both Executable Load Files and Applications are associated to Security Domains, the rules for deleting Security Domains apply to Executable Load Files as well as Applications.

The 5th bullet of runtime behavior in *section 6.4.2.1 – Application Removal* shall be completed as follows:

- "If a Security Domain is being deleted, determine that none of the Applications **or Executable Load Files** present in the card are associated with this Security Domain."

P.13.4 Terminating the card

According to the rules applying to GlobalPlatform commands per Card Life Cycle State described in Tables 9-1, D-3 and E-5, all commands, except the Get Data command, shall be rejected when the Card Life Cycle State is TERMINATED. *Section 6.7.3 – Card Termination* describes the transient state when the Card Life Cycle State is set to TERMINATED but the current Application Sessions can be continued until their end.

The last paragraph of *section 6.7.3 – Card Termination* shall be completed as follows:
"Card Content management requests are denied from the instant the card transitions to the TERMINATED Life Cycle State. **Depending on the security policy of the Issuer, other operations may be allowed to continue during the remaining current Application Sessions.**"

P.13.5 Security Domain personalization support rules

A Security Domain may offer personalization support services to its associated Applications as described in section 7.3 – Personalization Support. To perform this service, the Security Domain needs to access the corresponding interface of the Application. As the implementation of this interface is optional for an Application, a Security Domain offering personalization support services shall be able to handle and accept (or reject) personalization commands destined to Applications according to the Applications' own implementation (or not) of this interface.

Section 7.3 – Personalization Support shall be completed as follows:

- **"If an Application does not support this second method, the Security Domain shall reject any STORE DATA command destined to this Application.**"

P.13.6 MANAGE CHANNEL command: assignment by the terminal

Logical channel number assignment by the terminal is an optional feature of GlobalPlatform cards supporting logical channels.

Section 9.7.2.2 – MANAGE CHANNEL Reference Control Parameter P2 shall be completed as follows:
"**A card supporting logical channel assignment by off-card entities may accept a reference control parameter P2 indicating the Supplementary Logical Channel number to be opened (i.e. '01', '02' or '03').**"

To comply with ISO/IEC 7816-4, the penultimate paragraph of *section 3.2.1 – GlobalPlatform Environment (OPEN)* shall be completed as follows:
"Support of logical channels is optional. **When supported, the maximum number of available logical channels and assignment mechanism(s) shall be indicated in the Answer-To-Reset (third software functional table of the historical bytes) according to ISO/IEC 7816-4**."

P.13.7 DGI for Issuer Security Domain data

Section A.1 of Amendment A introduces the DGI formatting for the STORE DATA command data field. The Issuer Security Domain Data may be presented in this DGI formatting and not necessarily in a BER-TLV formatting as stated in section 9.11.2.3 of the specification.

In *Amendment A, section A.1.2 – STORE DATA Command Message Data Field* shall amend *section 9.11.2.3 – STORE DATA Command Message Data Field* of the specification as follows:
"The Issuer Security Domain shall support at least the following __ data objects:

- Issuer Identification Number (tag '42')
- Card Image Number (tag '45')
- Issuer Security Domain AID (tag '4F')
- Card Data (tag '66')

When DGI formatting is supported, these data objects may be included in a DGI. In that case, the DGI '0070' shall be used. Otherwise, these data objects shall be presented in their BER-TLV format."

P.13.8 GlobalPlatform on a Java Card: select() method error

The Guidelines for Developing Java Card Applications on GlobalPlatform Cards requests that "proper care must be taken to ensure that the select() method does not fail for any reason whatsoever (i.e. does not throw an exception), and that this method also always returns true. Throwing an exception in the select() method, or returning false from the select() method, will prevent the JCRE from proceeding with the application selection" (see section 6.2 of the Guidelines). Such scenario aborts the selection process under progress and leaves the corresponding logical channel open with no currently selected application.

To cater for the special situation created by Java Card specifications, the introduction of a*ppendix A.2 – GlobalPlatform on a Java Card* shall be completed as follows:
"**Selection**
On GlobalPlatform cards, Java Card applets are requested not to fail processing the select() method (e.g. throwing an exception) nor return false from the select() method. If an error occurs during the processing of the select() method, the selection process under progress is aborted by the JCRE and a Java Card specific error is returned to the off-card entity as specified in the Java Card Runtime Environment Specifications. Such situation leaves the corresponding logical channel open with no currently selected Application. Since no Application is currently selected, any subsequent command, other than a MANAGE CHANNEL or SELECT [by name] command, will be rejected. It is expected that the off-card entity will take appropriate action on such an error, e.g. select another Application, close the corresponding logical channel, reset or power off the card."

P.13.9 GlobalPlatform API: unwrap() method returned data

In case of a successful verification of secure messaging, the unwrap() method of the SecureChannel interface of the GlobalPlatform API for Java Card returns the unwrapped APDU command message with all data relating to the secure messaging removed. In case of no secure messaging verification in accordance with the Secure Channel Protocol rules, the unwrap() method returns the APDU command message without any modification. In both cases, the returned length includes the APDU command header.

To clarify the meaning of the returned command data, in *appendix A.2 – GlobalPlatform on a Java Card*, the description of the unwrap() method of the SecureChannel interface shall be completed as follows:
"<u>Returns</u>:
the length of the unwrapped data, i.e. the length of the command **header and** data **field.**"

P.13.10 GlobalPlatform API: wrap() method processing rules

The wrap() method of the SecureChannel interface of the GlobalPlatform API for Java Card performs no secure messaging on a response message outside of a R-MAC Session and not only when the current Security Level is set to NO_SECURITY_LEVEL.

In *appendix A.2 – GlobalPlatform on a Java Card*, the 3rd bullet in the notes for the wrap() method of the SecureChannel interface shall be completed as follows:

- "If **the current Security Level does not indicate R_MAC and/or R_ENCRYPTION**, this method will do no processing **and the outgoing response message will remain as is in the APDU object. The returned length of the "wrapped" data is set to the value of the sLength parameter**."

P.13.11 GlobalPlatform API: encryptData() method processing rules

The encryptData() method of the SecureChannel interface of the GlobalPlatform API for Java Card performs no encryption of the sensitive data when there is no Secure Channel Session open and there is no data encryption key available (e.g. case of Sceure Channel Protocol '01'). The Application requesting the sensitive data encryption service shall be informed of this unavailability.

In *appendix A.2 – GlobalPlatform on a Java Card*, the notes for the encryptData() method of the SecureChannel interface shall include a new bullet as follows:

- "**A Secure Channel Session shall be opened and a sensitive data encryption key shall be available**."

In *appendix A.2 – GlobalPlatform on a Java Card*, the error case list for the encryptData() method of the SecureChannel interface shall be completed as follows:
""Throws:
  - java.lang.ArrayIndexOutOfBoundsException if enciphering would cause access outside array bounds.
  - **ISO7816.SW_SECURITY_STATUS_NOT_SATISFIED if a Secure Channel Session is not open**."

To reflect the ISO7816.SW_SECURITY_STATUS_NOT_SATISFIED error case, in *appendix A.2 – GlobalPlatform on a Java Card*, the notes for the decryptData() method of the SecureChannel interface shall include a new bullet as follows:

- "**A Secure Channel Session shall be opened and a sensitive data decryption key shall be available**."

P.13.12 R-MAC support indicator

To indicate support of R-MAC functionality to an off-card entity, the Secure Channel Protocol '02' options identifier is complemented with new values. A GlobalPlatform card may implement one (or more) of the possible options.

In *appendix E.1.1 – SCP '02' Secure Channel* new option values shall be added as follows:

- **"i" = '24'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 1 Secure Channel base key, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **R-MAC support**,
- **"i" = '25'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 3 Secure Channel Keys, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **R-MAC support**,
- **"i" = '2A'**: Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, no ICV encryption, 1 Secure Channel base key, **R-MAC support**,
- **"i" = '2B'**: Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, no ICV encryption, 3 Secure Channel Keys, **R-MAC support**,

- **"i" = '34'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 1 Secure Channel base key, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **R-MAC support**,
- **"i" = '35'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 3 Secure Channel Keys, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **R-MAC support**,
- **"i" = '3A'**: Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, ICV encryption for C-MAC session, 1 Secure Channel base key, **R-MAC support**,
- **"i" = '3B'**: Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, ICV encryption for C-MAC session, 3 Secure Channel Keys, **R-MAC support**."

In *appendix E.1.1 – SCP '02' Secure Channel* the current option values description shall be completed as follows:

- "i" = '04': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 1 Secure Channel base key, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **no R-MAC**,
- "i" = '05': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 3 Secure Channel Keys, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **no R-MAC**,
- "i" = '0A': Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, no ICV encryption, 1 Secure Channel base key, **no R-MAC**,
- "i" = '0B': Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, no ICV encryption, 3 Secure Channel Keys, **no R-MAC**,
- "i" = '14': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 1 Secure Channel base key, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **no R-MAC**,
- "i" = '15': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 3 Secure Channel Keys, unspecified card challenge generation method (see section A.2.2 of Amendment #A), **no R-MAC**,
- "i" = '1A': Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, ICV encryption for C-MAC session, 1 Secure Channel base key, **no R-MAC**,
- "i" = '1B': Initiation mode implicit, C-MAC on unmodified APDU, ICV set to MAC over AID, ICV encryption for C-MAC session, 3 Secure Channel Keys, **no R-MAC**."

In *Amendment A, section A.2.2 – Secure Channel Protocol '02' Options Identifier* new extended option values shall be added as follows:

- **"i" = '64'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 1 Secure Channel base key, well-known pseudo-random algorithm (card challenge), **R-MAC support**,
- **"i" = '65'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 3 Secure Channel Keys, well-known pseudo-random algorithm (card challenge), **R-MAC support**,
- **"i" = '74'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 1 Secure Channel base key, well-known pseudo-random algorithm (card challenge), **R-MAC support**,
- **"i" = '75'**: Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 3 Secure Channel Keys, well-known pseudo-random algorithm (card challenge), **R-MAC support**.

In *Amendment A, section A.2.2 – Secure Channel Protocol '02' Options Identifier* the extended option values description shall be completed as follows:

- "i" = '44': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 1 Secure Channel base key, well-known pseudo-random algorithm (card challenge), **no R-MAC**,

- "i" = '45': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, no ICV encryption, 3 Secure Channel Keys, well-known pseudo-random algorithm (card challenge), **no R-MAC**,
- "i" = '54': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 1 Secure Channel base key, well-known pseudo-random algorithm (card challenge), **no R-MAC**,
- "i" = '55': Initiation mode explicit, C-MAC on modified APDU, ICV set to zero, ICV encryption for C-MAC session, 3 Secure Channel Keys, well-known pseudo-random algorithm (card challenge), **no R-MAC**.