EMV®
Contactless Specifications for Payment Systems

Book C-3

Kernel 3 Specification

Version 2.10
March 2021

Legal Notice

Unless the user has an applicable separate agreement with EMVCo or with the applicable payment system, any and all uses of these Specifications is subject to the terms and conditions of the EMVCo Terms of Use agreement available at www.emvco.com and the following supplemental terms and conditions.

Except as otherwise may be expressly provided in a separate agreement with EMVCo, the license granted in the EMVCo Terms of Use specifically excludes (a) the right to disclose, distribute or publicly display these Specifications or otherwise make these Specifications available to any third party, and (b) the right to make, use, sell, offer for sale, or import any software or hardware that practices, in whole or in part, these Specifications. Further, EMVCo does not grant any right to use the Kernel Specifications to develop contactless payment applications designed for use on a Card (or components of such applications). As used in these supplemental terms and conditions, the term “Card” means a proximity integrated circuit card or other device containing an integrated circuit chip designed to facilitate contactless payment transactions. Additionally, a Card may include a contact interface and/or magnetic stripe used to facilitate payment transactions. To use the Specifications to develop contactless payment applications designed for use on a Card (or components of such applications), please contact the applicable payment system. To use the Specifications to develop or manufacture products, or in any other manner not provided in the EMVCo Terms of Use, please contact EMVCo.

These Specifications are provided "AS IS" without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of these Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of these Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party's infringement of any intellectual property rights in connection with these Specifications.

Revision Log

This section outlines the notable updates that have been made to this specification since the publication of the *EMV Contactless Specifications for Payment Systems Book C-3, version 2.9*.

- Removed Dynamic Reader Limits (DRL) functionality.
- Upon receipt of GPO response with SW1 SW2 = '6984', changed the Try Another Interface outcome parameters so that no specific alternate interface is preferred.
- Removed kernel requirement to support GPO responses in EMV Format 1.
- Clarified completion processing when online is required but online processing is not supported.
- For an Offline Completion outcome of Declined, changed "Data Record Present: No" to "Data Record Present: Yes".
- Added the IIN (tag '42') and IINE (tag '9F0C') as recognized kernel data objects.
- Clarified that the CED (tag '9F7C') and FFI (tag '9F6E') must be included in the data record when returned by the card.

Change bars are used in the specification to denote the sections that have been updated.

Contents

Figures

Tables

Requirements

1 General

This chapter contains information that helps the reader understand and use this specification.

1.1 Scope

This document, the *EMV Contactless Specifications for Payment Systems, Kernel 3 Specification*, describes one of several kernels defined for use with Entry Point.

1.2 Audience

This specification is intended for use by system designers in payment systems and financial institution staff responsible for implementing financial applications.

1.3 Volumes of the Contactless Specifications

This specification is part of a nine-volume set:

Book A: Architecture and General Requirements

Book B: Entry Point Specification

Book C-2: Kernel 2 Specification

Book C-3: Kernel 3 Specification

Book C-4: Kernel 4 Specification

Book C-5: Kernel 5 Specification

Book C-6: Kernel 6 Specification

Book C-7: Kernel 7 Specification

Level 1 Specifications for Payment Systems, EMV Contactless Interface Specification

1.4 Reference Materials

The following specifications and standards contain provisions that are referenced in this specification. The latest version shall apply unless a publication date is explicitly stated.

If any provision or definition in this specification differs from those in the listed specifications and standards, the provision or definition herein shall take precedence.

[EMV 4.3]	EMV Integrated Circuit Card Specifications for Payment Systems, Version 4.3, November 2011, including:
[EMV 4.3 Book 1]	EMV Integrated Circuit Card Specifications for Payment Systems*, Book 1, Application Independent ICC to Terminal Interface Requirements*
[EMV 4.3 Book 2]	EMV Integrated Circuit Card Specifications for Payment Systems*, Book 2, Security and Key Management*
[EMV 4.3 Book 3]	EMV Integrated Circuit Card Specifications for Payment Systems*, Book 3, Application Specification*
[EMV 4.3 Book 4]	EMV Integrated Circuit Card Specifications for Payment Systems*, Book 4, Cardholder, Attendant, and Acquirer Interface Requirements*
[EMV ASRPD]	EMV Specification Update Bulletin No. 175. Application Selection Registered Proprietary Data.
[EMV L1 Contactless]	EMV Level 1 Specifications for Payment Systems, *EMV Contactless Interface Specification*, Version 3.0
[EMV Tokenisation]	EMV Payment Tokenisation Specification; latest version available on EMVCo.com
[ISO 639-1]	Codes for the representation of names of languages – Part 1: Alpha-2 Code **Note:** This standard is updated continuously by ISO. Additions and changes are available on: http://www.loc.gov/standards/iso639-2/php/code_changes.php
[ISO 3166]	Codes for the representation of names of countries and their subdivisions
[ISO 4217]	Codes for the representation of currencies and funds

[ISO 7813]	Identification cards – Financial transaction cards
[ISO 7816-4]	Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange
[ISO 7816-5]	Identification cards – Integrated circuit cards – Part 5: Registration of application providers
[ISO 8583:1987]	Bank card originated messages – Interchange message specifications – Content for financial transactions
[ISO 8859]	Information processing – 8-bit single-byte coded graphic characters sets
[ISO 14443]	Identification cards – Contactless integrated circuit cards – Proximity cards

1.5 Terminology

This specification uses the following terminology.

1.5.1 Implementation and Support Terminology

This specification uses the following terminology to describe the implementation and configuration requirements for kernel functionality:

Implementation-Mandatory *Kernel shall implement this functionality.*

Implementation-Conditional *Kernel shall implement this functionality if the defined conditions are met. Conditions vary based on the functionality in question.*

Implementation-Optional *Kernel may implement this functionality, at the discretion of the implementer.*

Acquirer-Merchant-Conditional *Acquirer-merchant shall enable this functionality if the defined conditions are met. Conditions vary based on the functionality in question.*

Acquirer-Merchant-Optional *Acquirer-merchant may enable this functionality, at their discretion.*

The reader functionality defined in this specification is *Implementation-Mandatory*, except where explicitly stated otherwise. This specification may also explicitly reiterate that specific functionality is *Implementation-Mandatory* to avoid potential ambiguity.

When used to describe requirements and conditions for kernel functionality, the following terminology may be used:

Implement(ed) *Kernel is capable of performing the functionality. The phrase "implement support for" may also be used.*

Enable(d) *Kernel functionality has been activated (i.e. turned on).*

Disable(d) *Kernel functionality has been deactivated (i.e. turned off).*

Support(ed) *Kernel functionality is both implemented and acquirer-merchant enabled.*

1.5.2 “Else” with Parenthesized Text

Parenthesized text following the word “Else” is provided as supplemental information, and does not denote a condition that must be satisfied.

Ex. If Condition, then the kernel shall…

Else (Text), the kernel shall…

In the example above, the parenthesized “(Text)” is merely supplemental information, and does not constitute a condition that must be satisfied.

1.5.3 Legacy References

Within requirements, italic numbers in square brackets are legacy references to an earlier version of this specification.

1.6 Flowcharts

Flowcharts are used to provide a high-level illustration of the processing. The flowcharts may be simplified to illustrate a concept, and may not include all the steps that are performed. Implementations are not required to strictly follow the flowcharts, and are instead required to comply with the requirements in the related text. In the case of a discrepancy between the flowchart and the related text, the text shall take precedence.

1.7 Format of Indicators

Although the indicators used in this specification are explicitly assigned the values of 0 or 1, the format of these internal indicators within an implementation is at the discretion of the implementer.

Performance and security considerations should both be weighed when deciding on the implementation of internal indicators.

1.8 Outcomes

Kernel 3 concludes its processing via the Outcomes as defined in *Book A*. When an Outcome is provided – specified with the words “…provide a ***xxx*** Outcome with the following parameters:” – Kernel 3 executes no further processing except to hand over control to Entry Point, passing the Outcome and its parameters.

1.9 Overview

This volume includes the following chapters and annexes:

Chapter 1 contains general information that helps the reader understand and use this specification.

Chapter 2 provides an overview of the Kernel 3 approach, including configuration for EMV mode, Kernel 3 processing overview, new transaction processing sequence, Integrated Data Storage processing sequence, and Issuer Update Processing.

Chapter 3 defines general functionality and requirements in support of Kernel 3, including messaging requirements, reader configuration requirements, and processing requirements.

Chapter 4 defines overall functionality and requirements for Kernel 3, culminating in the decision of whether to perform new transaction processing or Issuer Update Processing.

Chapter 5 describes requirements for new transaction processing.

Chapter 6 describes requirements for Issuer Update Processing.

Chapter 7 describes requirements for Integrated Data Storage.

Annex A defines the data elements used in this specification for financial transaction interchange.

Annex B details the data elements required for online messages and clearing records.

Annex C describes fast Dynamic Data Authentication (fDDA).

Annex D lists supported commands.

Annex F provides a description of an Integrated Data Storage implementation

Annex G is a glossary of terms and abbreviations used in this specification.

2 Overview of the Kernel 3 Approach

The Kernel 3 approach supports a single configuration: EMV mode.

- The EMV mode configuration supports the EMV minimum set of data elements for clearing messages and for authorisation messages, when used. See section B.1.1. This configuration is for both offline and online transactions with cardholder verification support.

EMV mode readers may optionally support:

- Integrated Data Storage (IDS). This provides a means of reading data from and writing data to cards that also support Integrated Data Storage. The Integrated Data Storage mechanism described in this specification is unique to Kernel 3 and is not interoperable with data storage solutions in other kernels.
- Issuer Update Processing. This provides a means to deliver Issuer Authentication Data and/or Issuer Scripts to a card.

2.1 EMV Mode Configuration

The EMV mode configuration uses EMV commands and constructs with usage adapted for speed and to allow cryptographic operations to be done at the heart of the transaction, rather than when the card might be leaving the field. It features:

- An offline transaction flow for below floor limit transactions, with offline data authentication and a clearing cryptogram.
- An online transaction flow with online card authentication and an option for a second re-presentment of the card for issuer updating of card parameters.
- Cardholder verification support by means of online PIN or signature, when required by the transaction conditions.
- Support for consumer device form factors (e.g. mobile phones) acting as card products, including support for the Consumer Device CVM. The Consumer Device CVM is a CVM performed on, and validated by, the consumer’s payment device, independent of the kernel.

The EMV mode configuration uses the output of Entry Point Pre-Processing and Combination Selection, resulting in the response to the SELECT command and the following data elements and indicators being available at the start of Kernel 3 processing:

- Amount, Authorised
- Unpredictable Number

- Terminal Transaction Qualifiers (TTQ)
- Copy of TTQ

An accelerated offline data authentication method, fast Dynamic Data Authentication (fDDA), is provided through use of the GPO command to initiate the dynamic signature. A DDOL is not used and the results of the authentication are not provided online to the issuer within the TVR or protected by the online authorisation or clearing cryptograms.

The EMV mode configuration uses the data elements defined in section A.2 and provides data for online messages and clearing records as defined in section B.1.1.

2.2 <This section has been deleted>

2.3 Kernel 3 Processing Overview

A new transaction requires the contactless card to be presented once to complete a transaction that is authorised either offline or online. A new transaction can be of the following types:

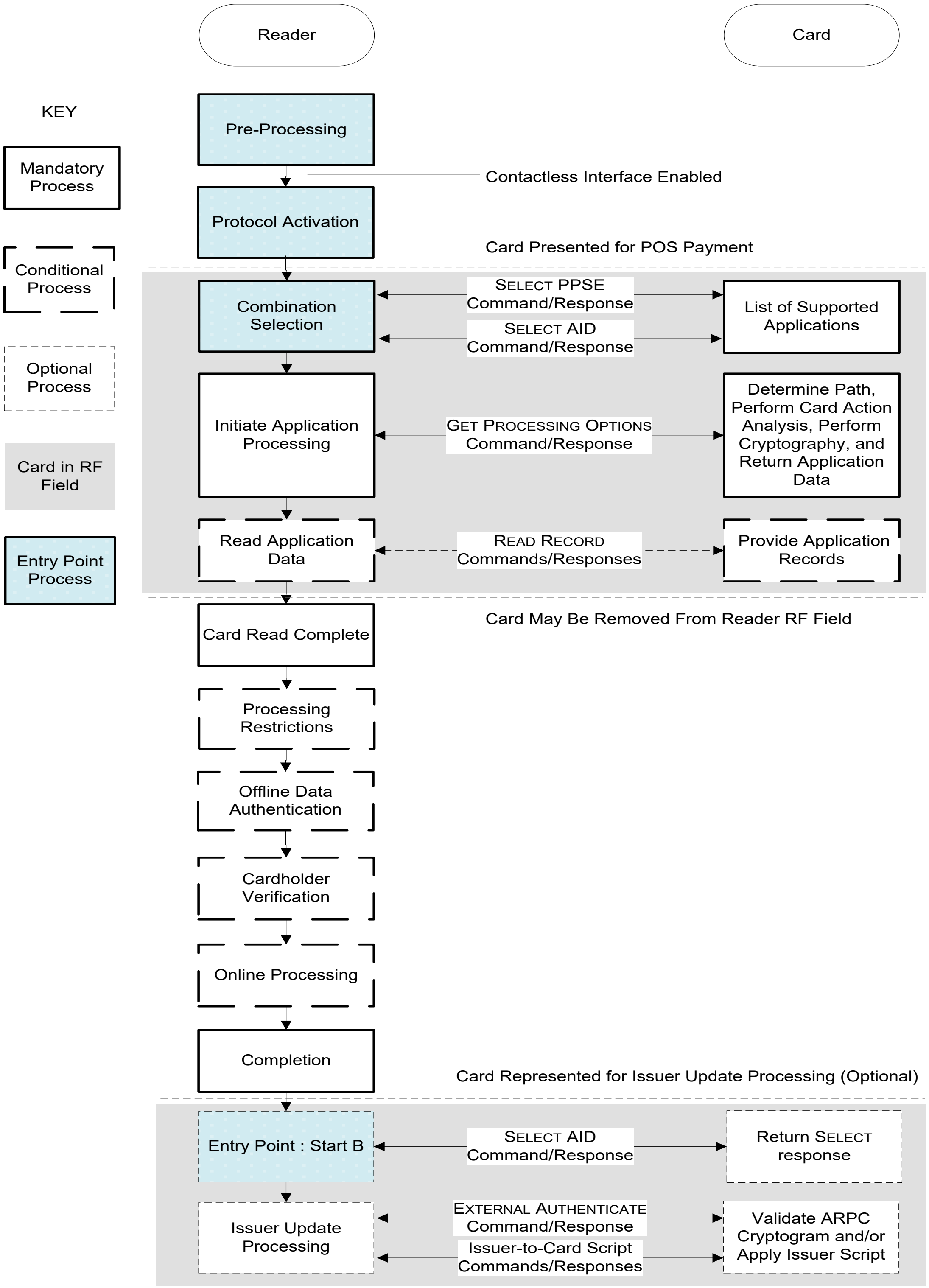
- purchase transaction,
- cash withdrawal, or
- refund.

The new transactions will constitute the majority of transactions and will require no further card interaction. The new transaction processing flow is further outlined in section 2.4

However in some situations a secondary presentment may be beneficial in support of card parameter updates. This function – named issuer update processing – is applicable only for an EMV mode reader and it is optional for both card and reader. The issuer update processing is further outlined in section 2.5.

A full overview of the processing flow is outlined in Figure 2-1.

Figure 2-1: Sample Transaction Flow


Reader
Card
KEY
Mandatory Process
Conditional Process
Optional Process
Card in RF Field
Entry Point Process
Pre-Processing
Contactless Interface Enabled
Protocol Activation
Card Presented for POS Payment
Combination Selection
SELECT PPSE Command/Response
SELECT AID Command/Response
List of Supported Applications
Initiate Application Processing
GET PROCESSING OPTIONS Command/Response
Determine Path, Perform Card Action Analysis, Perform Cryptography, and Return Application Data
Read Application Data
READ RECORD Commands/Responses
Provide Application Records
Card May Be Removed From Reader RF Field
Card Read Complete
Processing Restrictions
Offline Data Authentication
Cardholder Verification
Online Processing
Completion
Card Represented for Issuer Update Processing (Optional)
Entry Point : Start B
SELECT AID Command/Response
Return SELECT response
Issuer Update Processing
EXTERNAL AUTHENTICATE Command/Response
Issuer-to-Card Script Commands/Responses
Validate ARPC Cryptogram and/or Apply Issuer Script

Note: Processing as defined by the shaded shapes is performed by Entry Point.

2.3.1 First or Second Presentment

The cycling of a transaction for both a first and a second presentment is handled by Entry Point. If the first presentment results in an ***Online Request*** Outcome, the card indicates via the Outcome parameters whether a second presentment is supported.

During a first presentment, either a new transaction or Integrated Data Storage processing will occur.

During a second presentment, Issuer Update Processing will occur.

When activated by Entry Point, Kernel 3 checks for the availability of an Integrated Data Storage Directory or Issuer Update Data:

- If neither Issuer Update Data nor Integrated Data Storage Directory are present, Kernel 3 performs a standard new transaction (section 2.4).
- If Issuer Update Data is present, Kernel 3 performs Issuer Update Processing (section 2.6).

  The Issuer Update Processing commands need to be delivered to the same card that was presented for the new transaction. Thus the expectation is that once a card is discovered, Entry Point will directly select the AID used in the previous new transaction in order to deliver the Issuer Update Processing commands to the correct card. If a card with a different AID is presented, then the selection will fail. If a different card is presented using the same AID, then the selection will succeed, but the updates will fail. In practice it is not expected that either situation will occur on a regular basis.

  Note: If Issuer Update Data is present, Kernel 3 performs Issuer Update Processing regardless of the presence of the Integrated Data Storage Directory.
- If an Integrated Data Storage Directory is present, and Issuer Update Data is not present, Kernel 3 performs Integrated Data Storage processing (section 2.5).

For new transactions, Integrated Data Storage and Issuer Update Processing, the reader in combination with the POS system has a time-out that abandons the transaction if no card is presented.

2.4 New Transaction Processing Sequence

This section provides an overview of a Kernel 3 transaction, for which Integrated Data Storage processing is not required, showing the order in which functions are performed and the commands sent by the reader to the card. The transaction starts at the handover from Entry Point, which has performed: Pre-Processing, Protocol Activation, Combination Selection, and Kernel Activation.

2.4.1 Initiate Application Processing

The status word from the response to the SELECT AID command has been evaluated by Entry Point, thus only a successful SELECT AID response including the Processing Options Data Object List (PDOL) be passed to Kernel 3.

Kernel 3 processing for a transaction starts with sending a GET PROCESSING OPTIONS (GPO) command to the card. This includes the data elements requested by the card in the PDOL returned in the response to the SELECT command, which will include the Terminal Transaction Qualifiers.

In the GPO response, the kernel is expected to receive data elements from the card that are appropriate to the conditions indicated in the Terminal Transaction Qualifiers:

- a cryptogram with supporting/additional data, and for offline approved transactions, an Application File Locator (AFL) which points to additional data. Signatures and other data that would cause the response to exceed its size limit are not included, but are instead provided in a record which is indicated in the AFL.

Note that the reader is expected to handle situations where the data elements – or some of them – are received during the Read Application Data function.

If the conditions for usage of the card application have not been fulfilled, the reader must retry or end the transaction.

2.4.2 Read Application Data

If an Application File Locator (AFL) was returned during Initiate Application Processing, the kernel reads card application data from the records indicated in the AFL. This may include the signature if placed in a record due to size limitations as above.

After Read Application Data, the kernel is expected to be in possession of all the data elements necessary to complete the transaction – some data is obtained from the GPO response and if necessary (that is, if the AFL is present) some is obtained from records.

2.4.3 Card Read Complete

During Card Read Complete, the kernel indicates to the cardholder that exchange of data between the reader and the card is complete, and the card may be removed from the field.

The kernel makes the following checks on the available data and terminates the transaction if either check fails:

- That all mandatory data elements were returned by the card.
- That no primitive data element was returned more than once during Initiate Application Processing and Read Application Data.

Subsequent transaction processing includes all the remaining functions in this chapter (section 2.4.4 thru section 2.5).

2.4.4 Processing Restrictions

If supported by the kernel for EMV mode configurations, the kernel checks the application expiration date and application usage, and may need to check whether the card is on the Terminal Exception File.

2.4.5 Offline Data Authentication

Offline Data Authentication is implemented for readers supporting offline transactions and is performed for card requested offline transactions. The kernel verifies the dynamic signature returned by the card and authenticates the data from the card.

The success or failure is indicated by means of the appropriate Outcome during Completion.

Special purpose readers may perform offline data authentication for online transactions.

2.4.6 Cardholder Verification

Cardholder Verification is implemented for readers implementing EMV mode transactions. During Cardholder Verification, the kernel determines the Cardholder Verification Method to be performed (if any) and makes the necessary request by means of the Outcome and parameters.

Kernel 3 is not involved with any CVM processing for ATM devices. This is determined solely by the terminal and acceptance conditions.

2.4.7 Online Processing

Online Processing is implemented for EMV mode readers supporting online transactions. The kernel indicates the need for online processing by means of the Outcome and parameters.

The data provided by the kernel for an online authorisation includes an Application Cryptogram – either an ARQC, or a TC if the card indicates that online processing is preferred if Offline Data Authentication fails.

2.4.8 Completion

Completion concludes transaction processing and the kernel will indicate the Outcome of the transaction and the need for any further action to Entry Point.

The available Outcomes to Entry Point and the expectations for further actions are as below:

- **Select Next** – The kernel has determined that the selected Combination is unsuitable and the next Combination (if any) should be tried.
- **Try Again** – The kernel requires that the device be presented again; this may be a result of an error, such as tearing, that could resolve if the transaction is attempted again.
- **Approved** – The kernel is satisfied that the transaction is acceptable to the selected contactless card application and wants the transaction to be approved. This is the expected Outcome for a successful offline transaction, but could also occur following re-selection of a kernel after an online response.
- **Declined** – The kernel has found that the transaction is not acceptable to the selected contactless card application and wants the transaction to be declined. This Outcome could also occur following re-selection of a kernel after an online response.
- **Online Request** – The transaction requires an online authorisation to determine the approved or declined status.
- **Try Another Interface** – The kernel is unable to complete the transaction with the selected contactless card application, but knows from the configuration data that another interface (e.g. contact or magnetic-stripe) is available. The kernel could indicate a preference for the alternate interface.
- **End Application** – The kernel experienced an application error, such as missing data, that will not resolve if the transaction is attempted again with the same selected contactless card application.

2.5 Integrated Data Storage Processing Sequence

Integrated Data Storage is an implementation and acquirer-merchant option for EMV Mode configurations. Integrated Data Storage Processing is performed during first presentment if the card application has been selected using Extended Selection and has responded with FCI containing the Integrated Data Storage Directory (IDSD, Tag ‘D2’).

Integrated Data Storage processing follows the sequence of New Transaction Processing, as described in section 2.4, with the following exceptions:

- Prior to Initiate Application Processing the kernel may send one or more READ RECORDs to the card and there may be one or more Data Exchanges between the kernel and an IDS Operator Application; the IDS Operator Application implements the proprietary business logic of the IDS Operator.
- If IDS updates on the card are needed then the EXTENDED GPO (EGPO) command is used in place of the standard Kernel 3 GPO command.

2.5.1 IDS Operator Application

The IDS Operator Application implements the proprietary business logic of the IDS Operator.

The IDS Operator Application is out of scope of this specification.

The functions of the IDS Operator Application include:

- Analysing the IDS Directory provided by the card in the SELECT response.
- Determining whether the card contains IDS records for any IDS Operator(s) supported by the terminal. If so, instructing the kernel to read the relevant IDS Records.
- Interpreting the content of any IDS Records read from the card.
- Determining whether the card contains IDS records to be updated. If so, instructing the kernel to update IDS Records (using the EGPO command) and providing appropriate command data.

For simplicity, this specification assumes:

- The IDS Operator Application executes in the terminal.

  In practice the IDS Operator Application could be distributed between the terminal and kernel or located in other system components. For example, the kernel could potentially include proprietary configuration of supported IDS Operators, and ability to read appropriate records without needing to communicate with a separate application.
- Data is communicated between the kernel and the IDS Operator Application using Data Exchange.
- Instructions are communicated from the IDS Operator Application to the kernel using Data Exchange.

The protocol for communicating instructions between an IDS Operator Application and the kernel is kernel implementation proprietary and is out of scope of this specification.

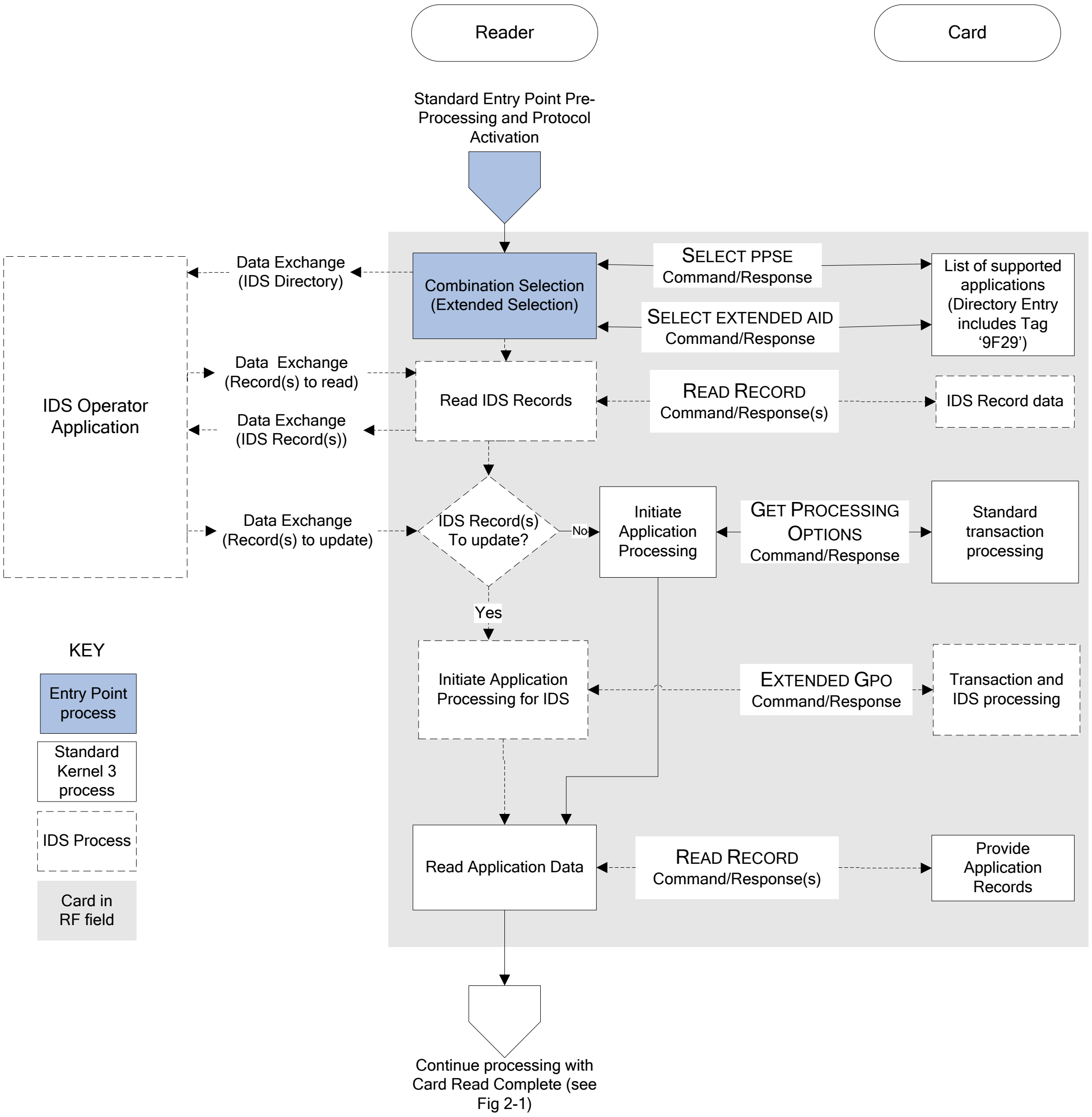
Data Exchange with the IDS Operator Application is described in this specification for cases that directly affect kernel processing and kernel to card communications. Implementers may support additional interactions with IDS Operator Applications that are not described in this specification, for example to communicate:

- Success of the transaction.

- Error conditions, for example to inform the IDS Operator Application of Offline Data Authentication failures, to enable proprietary processing to manage such exception conditions.

Figure 2-2. provides an overview of how Integrated Data Storage processing extends the standard kernel 3 transaction flow. Note that all IDS processing takes place while the card is in the field. Consequently, consideration should be given to the efficiency of the Data Exchange mechanism(s) supported by the kernel implementation in order to minimise the impact on overall performance.

Figure 2-2: Sample Integrated Data Storage Transaction Flow


Reader
Card
Standard Entry Point Pre-Processing and Protocol Activation
Data Exchange (IDS Directory)
Combination Selection (Extended Selection)
SELECT PPSE Command/Response
SELECT EXTENDED AID Command/Response
List of supported applications (Directory Entry includes Tag '9F29')
IDS Operator Application
Data Exchange (Record(s) to read)
Read IDS Records
READ RECORD Command/Response(s)
IDS Record data
Data Exchange (IDS Record(s))
Data Exchange (Record(s) to update)
IDS Record(s) To update?
No
Initiate Application Processing
GET PROCESSING OPTIONS Command/Response
Standard transaction processing
Yes
Initiate Application Processing for IDS
EXTENDED GPO Command/Response
Transaction and IDS processing
Read Application Data
READ RECORD Command/Response(s)
Provide Application Records
KEY
Entry Point process
Standard Kernel 3 process
IDS Process
Card in RF field
Continue processing with Card Read Complete (see Fig 2-1)

2.5.2 Analyze IDS Directory

The kernel sends the IDS Directory (IDSD) to the IDS Operator Application using Data Exchange. The IDSD is analyzed by the IDS Operator Application to determine whether any IDS Records are to be read. If so, the IDS Operator Application sends (via Data Exchange) instructions to the kernel to issue READ RECORD commands, including parameters for READ RECORD.

The analysis of the IDSD and the decision to read any IDS records is IDS Operator proprietary and is outside the scope of this specification.

2.5.3 Read IDS Records

If instructed by the IDS Operator Application, the kernel sends READ RECORD commands to the card, and sends the responses to the IDS Operator Application using Data Exchange.

2.5.4 Initiate Application Processing for IDS

As a result of analysing the Integrated Data Storage Directory and optionally any IDS Records read from the card, the IDS Operator Application determines whether any IDS Records are to be updated.

The analysis of this data and the decision to update any IDS Records is IDS Operator proprietary and is outside the scope of this specification.

If IDS Records are to be updated the IDS Operator Application sends the kernel an instruction to use the EXTENDED GET PROCESSING OPTIONS (EGPO) command, together with the IDS Record Update Template (tag ‘BF60’).

If no IDS Records are to be updated the IDS Operator Application sends to the kernel an instruction to continue with normal Initiate Application Processing, using the GET PROCESSING OPTIONS (GPO) command as described in section 2.4.1 above.

The command data for EGPO contains the PDOL, as for GPO, plus additional data that instructs the card to update one or more IDS records.

After receiving a successful EGPO response, the kernel continues with normal READ APPLICATION DATA processing, as described in 2.4.2 above.

2.6 Issuer Update Processing

Issuer Update Processing is an optional feature within Kernel 3 for situations where the issuer utilizing a specific terminal environment wishes to manage the risk parameters within a card by means of issuer authentication and/or script processing over the contactless interface.

If an online authorisation response following a new transaction includes Issuer Update Data and both reader and card support Issuer Update Processing, then the cardholder can be instructed to present their card for a second time. During the second presentment the reader can deliver the ARPC and/or forward the Issuer Scripts.

This functionality is solely used for updates of card parameters and all new transaction processing will take place during the first presentment.

2.6.1 Issuer Update Processing (Optional for EMV Mode)

Issuer Update Processing is an implementation and acquirer-merchant option for EMV mode configurations. If supported by both card and reader, Issuer Update Processing is performed when the authorisation response message contains Issuer Authentication Data and/or an Issuer Script Template. When the card is re-presented, Entry Point re-activates the kernel and the availability of Issuer Update Data indicates to the kernel that it should branch to this section.

If Issuer Authentication Data is present in the Issuer Update Data, the kernel sends an EXTERNAL AUTHENTICATE command to the card, which performs issuer authentication and updates risk management parameters (e.g. counter reset) as configured in the card by the issuer.

If an Issuer Script Template is present in the Issuer Update Data, the kernel extracts the Script Command(s) and forwards to the card for processing.

3 Requirements in Support of Kernel 3

This section defines general functionality and requirements in support of Kernel 3.

3.1 POS System Assumptions

Requirements – Receipts

3.1.1.1 (Receipts) *[5.4]*

Receipts are not detailed in this specification, however the following applies:

POS systems in support of Kernel 3 shall provide receipts as required by payment system rules (both international and regional).

Information about POS systems supporting transactions over multiple interfaces are defined in *Book A*, Requirements – Multiple Interfaces (8.1.1.11). *[5.5]*

3.2 Messaging Requirements in Support of Kernel 3

3.2.1 EMV Mode Requirements

Requirements – EMV Mode Requirements

3.2.1.1 (EMV mode Message Requirements) *[5.6]*

The data for inclusion in EMV mode online messages and clearing records shall be as specified in section B.1.

3.2.1.2 (EMV mode – FFI and CED) *[5.7]*

If the card application returns the Form Factor Indicator and/or Customer Exclusive Data,
then the kernel shall make the returned data element(s) available for inclusion in messages to the acquirer. See also section B.1.1.

Please check with your payment system representative regarding the required support for these data elements in acquirer messaging.

3.2.1.3 (EMV mode – Token Data)

The kernel shall support output of the following data object (when returned by the card application), for possible use by the merchant and transmission to the acquirer:

- Payment Account Reference (PAR, tag '9F24')

Note: The above data object is not normally included in messages to the acquirer, but readers must be capable of outputting the data object when returned by the card application.

3.2.2 <This section has been deleted>

3.3 Reader Configuration Requirements

This section defines some of the configuration requirements necessary for Kernel 3. Configuration of the reader shall conform to these requirements for AIDs supported by this kernel, but the reader or kernel need not enforce configuration requirements.

3.3.1 General Configuration Requirements

Requirements – General Configuration Requirements	
3.3.1.1	<This requirement has been deleted>
3.3.1.2	<This requirement has been deleted>
3.3.1.3	(Issuer Update Processing and Online Capability) *[5.14]* **If** a reader supports Issuer Update Processing, **then** the reader shall indicate support for Issuer Update Processing by setting TTQ byte 3 bit 8 to 1b and shall indicate online capability by setting TTQ byte 1 bit 4 to 0b.

During Pre-Processing, Entry Point will use the reader risk parameters, configured in the EMV mode-enabled reader, to determine whether an EMV mode transaction with an amount of zero will result in an online transaction (if passed to this kernel) or will be conducted over another interface or passed to another kernel. *[5.15]*

3.3.2 <This section has been deleted>

3.3.3 Configuration Requirements for EMV Mode Readers Supporting Integrated Data Storage

Following are additional requirements for EMV mode readers that support Integrated Data Storage.

Requirements – EMV Mode and Integrated Data Storage

3.3.3.1 (EMV Mode Configuration: Integrated Data Storage)

If an EMV mode reader supports Integrated Data Storage,

then support for Integrated Data Storage shall be Acquirer-Merchant configurable.

3.3.4 Configuration Requirements for Special Purpose EMV Mode Readers

Special purpose readers, such as transit system entry gates, may support the functionality in this section.

Implementation-Optional: Features for Special Purpose EMV Mode readers are an implementation option.

Acquirer-Merchant-Conditional: If implemented in the kernel, support for Special Purpose EMV Mode Reader features is acquirer-merchant-conditional. Refer to your payment system for conditions under which these features are enabled or disabled.

Requirements – Configuration for Special Purpose EMV Mode Readers	
3.3.4.1	(EMV Mode configuration: fDDA for Online) **If** an EMV mode reader supports fDDA for Online transactions, **then** support for fDDA for Online shall be Acquirer-Merchant configurable.
3.3.4.2	<This requirement has been deleted>
3.3.4.3	(EMV Mode configuration: Offline Data Authentication for Online) **If** an EMV mode reader supports Offline Data Authentication for Online transactions, **and** Offline Data Authentication for Online Transactions has been enabled by the Acquirer-Merchant, **then** the kernel shall indicate that Offline Data Authentication for Online is supported by setting TTQ byte 1 bit 1 to 1b.

3.4 Processing Requirements in Support of Kernel 3

3.4.1 Requirements for Different Types of Transaction

Readers may support different type of transactions. Depending on the transaction type, certain parameters shall be set up accordingly as outlined in this section.

Requirements – Transaction Type Requirements

3.4.1.1 (Purchase Transactions) *[5.18]*

Implementation-Conditional: With the exception of ATMs, support for the purchase of goods and services is *Implementation-Mandatory*. However, cashback functionality described below is *Implementation-Conditional* on reader support for cashback.

For transactions to purchase goods or services, with or without cashback, the EMV mode-enabled kernel shall use:

- Transaction Type ‘00’.
- Amount, Authorised shall be the sum of the purchase amount and the cashback amount (if present).
- Amount, Other shall be the cashback amount (if present).

Note: Acquirers-merchants may require a different Terminal Transaction Qualifiers (TTQ) value and different reader risk parameters for purchase transactions with cashback than is used for purchase transactions without cashback. The reader should allow the acquirer-merchant to configure the TTQ and reader risk parameters for purchase transactions with cashback independently from purchase transactions without cashback.

Requirements – Transaction Type Requirements

3.4.1.2 (Manual Cash Transactions) *[5.19]*

Implementation-Conditional:
If the reader supports manual cash transactions,
then this functionality shall be implemented.

For manual cash transactions, the EMV mode-enabled kernel shall use:

- Transaction Type '01'.
- Amount, Authorised shall be the transaction amount.

Note: Acquirers-merchants may require a different Terminal Transaction Qualifiers (TTQ) value and different reader risk parameters for manual cash transactions than is used for purchase transactions (with or without cashback). The reader should allow the acquirer-merchant to configure the TTQ and reader risk parameters for manual cash transactions independently from purchase transactions.

EMV mode transactions directly result in the purchase of goods or services and/or in the disbursement of cash. Refunds and credits are commonly employed to support the retail or cash disbursement environment, but do not directly result in the purchase of goods or services, nor in the disbursement of cash. Kernel 3 functionality can be used to support refunds and credits, and shall comply with the following requirement for the Transaction Type and Amount, Authorised values used. However, all other reader processing for refunds and credits is outside the scope of this specification.

Note: An offline decline (CID indicates an AAC) returned by the card application for a refund or credit simply indicates completion of card action analysis, and should not be treated as a "decline" of the refund.

Requirements – Transaction Type Requirements

3.4.1.3 (Refund Transactions) *[5.20]*

Implementation-Conditional:
If the reader supports refunds/credits,
then this functionality shall be implemented.

For refunds and credits, the EMV mode-enabled kernel shall use:

- Transaction Type '20'
- Amount, Authorised shall be the refunded/credited amount.

4 Overall Processing Requirements

This section defines overall functionality and requirements for Kernel 3.

4.1 General Requirements

Requirements – Overall Requirements

4.1.1.1 (Form Factor Indicator) *[5.21]*

If EMV mode is enabled in the kernel,
and the card application returns the Form Factor Indicator (FFI),
then the kernel shall replace FFI byte 4 bits 4-1 with the value 0000b (indicating that the transaction was conducted using [ISO 14443]) prior to making the FFI available for inclusion in messages to the acquirer.

Requirements – Overall Requirements

4.1.1.2 (Unrecoverable [EMV CL] Error) *[5.23]*

If a Transmission, Protocol, or Timeout error as defined in [EMV L1 Contactless] is reported to the kernel,
then the kernel shall discard the current transaction data and provide a ***Try Again*** Outcome with the following parameters:

Try Again:

- **Start:** B
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** No
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will cause Entry Point to return to Protocol Activation.

Requirements – Overall Requirements

4.1.1.3 (Cardholder Messaging) *[5.24]*

For contactless transactions, the reader (or terminal) shall clearly communicate to the cardholder and merchant:

- Present the card
- The progress of the transaction
- The outcome of the transaction – approve, decline, or terminate

Recommended cardholder messages and indications for various transaction status are defined in *Book A*.

If the card provides the Available Offline Spending Amount,
then the EMV mode-enabled reader may display this when it indicates a successful card read and may provide it on the transaction receipt.

Note: This specification indicates when communication with the cardholder and merchant occurs, but does not specify the means of communication. User interface requirements are specified on a payment system, regional, or country basis.

4.1.1.4 (Erroneous Data) *[5.25]*

It is the responsibility of the issuer to ensure that data in the card is formatted correctly, and no format checking other than that specifically defined is required on the part of the kernel.

However, **if** in the course of normal processing the kernel recognizes that data is incorrectly formatted,
then the kernel shall, unless otherwise specified, terminate the transaction by providing an ***End Application*** Outcome as defined in requirement 4.2.1.1.

Incorrectly formatted data includes, but is not limited to, the bulleted list provided in [EMV 4.3 Book 3], section 7.5.

4.2 End Application

If the kernel wishes to terminate the transaction with an ***End Application*** Outcome, the following Outcome and parameters is returned to Entry Point.

Requirements – End Application

4.2.1.1 (End Application)

The kernel shall provide an ***End Application*** Outcome with the following parameters:

End Application:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '1C' ("Insert, swipe or try another card")
  - Status: Processing Error
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

4.3 Display Available Offline Spending Amount

If, during a successful Outcome, the kernel wishes to have the Available Offline Spending Amount (AOSA) displayed or provided on a receipt, the UI Request parameters will be passed as part of the Outcome.

Requirements – Display AOSA

4.3.1.1 (Display AOSA)

If the card returns Available Offline Spending Amount (AOSA),
and the reader supports displaying or providing the AOSA on a receipt,
then:

- The reader shall display or, if a receipt is provided, include the AOSA.
- The kernel shall include the following UI Request parameters in the Outcome:
  - Value Qualifier: “Balance”
  - Value: AOSA (Tag ‘9F5D’ from card)
  - Currency Code: Transaction Currency Code
- The kernel shall set the Outcome parameter Discretionary Data Present to Yes and include AOSA in the kernel Discretionary Data.

Else the kernel shall not include the UI Request parameters Value Qualifier, Value, and Currency Code in the Outcome

4.4 New Transaction, Integrated Data Storage or Issuer Update Processing

When activated, the kernel analyses the data provided by Entry Point to determine whether to perform:

- New Transaction processing only.

  During a first presentment, Entry Point has provided the FCI and Status Word SW1 SW2 that were received from the card in the SELECT (AID) response and the FCI contains the Processing Options Data Object List (PDOL, Tag '9F38') and does not contain either IDSD nor Issuer Update Data.

- New Transaction and Integrated Data Storage processing

  During a first presentment, Entry Point has provided the FCI and Status Word SW1 SW2 that were received from the card in the SELECT (AID) response, the kernel supports Integrated Data Storage, the FCI contains both the PDOL and the Integrated Data Storage Directory (IDSD, Tag 'D2').

- Issuer Update processing

  During a second presentment, Entry Point has provided Issuer Update Data to the kernel, in addition to the SELECT (AID) response data.

Requirements – Distinguish between New Transaction, Integrated Data Storage and Issuer Update Processing

4.4.1.1 (New Transaction, Integrated Data Storage or Issuer Update Processing)

If Issuer Update Data is present when the kernel is activated,
then the kernel shall process the transaction as defined in Chapter 6 (Issuer Update Processing Reader Requirements).

Else (IDS Directory is present)

- **If** the kernel supports Integrated Data Storage

  and the IDSD (Tag 'D2') is present when the kernel is activated

  then the kernel shall process the transaction as defined in Chapter 7 (Data Storage Requirements).

Else (neither Issuer Update Data nor IDS Directory are present)

then the kernel shall process the transaction as defined in Chapter 5 (New Transaction Kernel Requirements).

5 New Transaction Kernel Requirements

The typical contactless experience is one presentment of the contactless card to the reader during which the capture of card data for a standard new transaction takes place. After capturing the card data, the reader can then evaluate the transaction disposition and no further presentment of the card is necessary. This chapter defines the requirements for this standard new transaction flow. A new transaction can be a purchase transaction, a cash withdrawal, or a refund transaction.

The processing assumes that the card in question has been selected and has returned SW1 SW2 = ‘9000’ in response to the SELECT command issued by Entry Point. The kernel will have available the FCI from Entry Point as well as the necessary data relevant for the transaction (e.g. Amount, Authorised).

As outlined in requirement 4.4.1.1:

- If the Integrated Data Storage Directory is present when Entry Point hands over control to Kernel 3, additional processing is performed and, if IDS Records on the card are to be updated, the EXTENDED GPO (EGPO) command is used instead of the standard GPO (see Chapter 7).

  Processing of the EGPO response SW1 SW2, and all subsequent Kernel 3 processing is the same as for GPO.

- New transaction processing is not performed if Issuer Update Data is present when Entry Point hands over control to Kernel 3.

5.1 <This section has been deleted>

5.2 Initiate Application Processing

During Initiate Application Processing, the reader issues the GET PROCESSING OPTIONS (GPO) command to the card, and includes any data that the card has requested in the PDOL during Final Combination Selection by Entry Point. Application data necessary to process the transaction is returned by the card application during Initiate Application Processing, and may also be returned during Read Application Data.

5.2.1 GET PROCESSING OPTIONS (GPO) Command

Requirements – GPO Command

5.2.1.1 (GPO Command) *[5.56]*

The kernel shall support the GET PROCESSING OPTIONS (GPO) command, as defined in Annex D of this specification.

Data Object List (DOL) coding is performed according to [EMV 4.3 Book 3], section 5.4. The kernel shall be able to provide the value of reader data elements (defined in Annex A) requested by the card application.

5.2.1.2 (Format 2) *[5.57]*

The kernel shall support GPO responses in EMV Format 2 as defined in Annex D of this specification.

5.2.1.3 (Recognized and Unrecognized Data) *[5.58]*

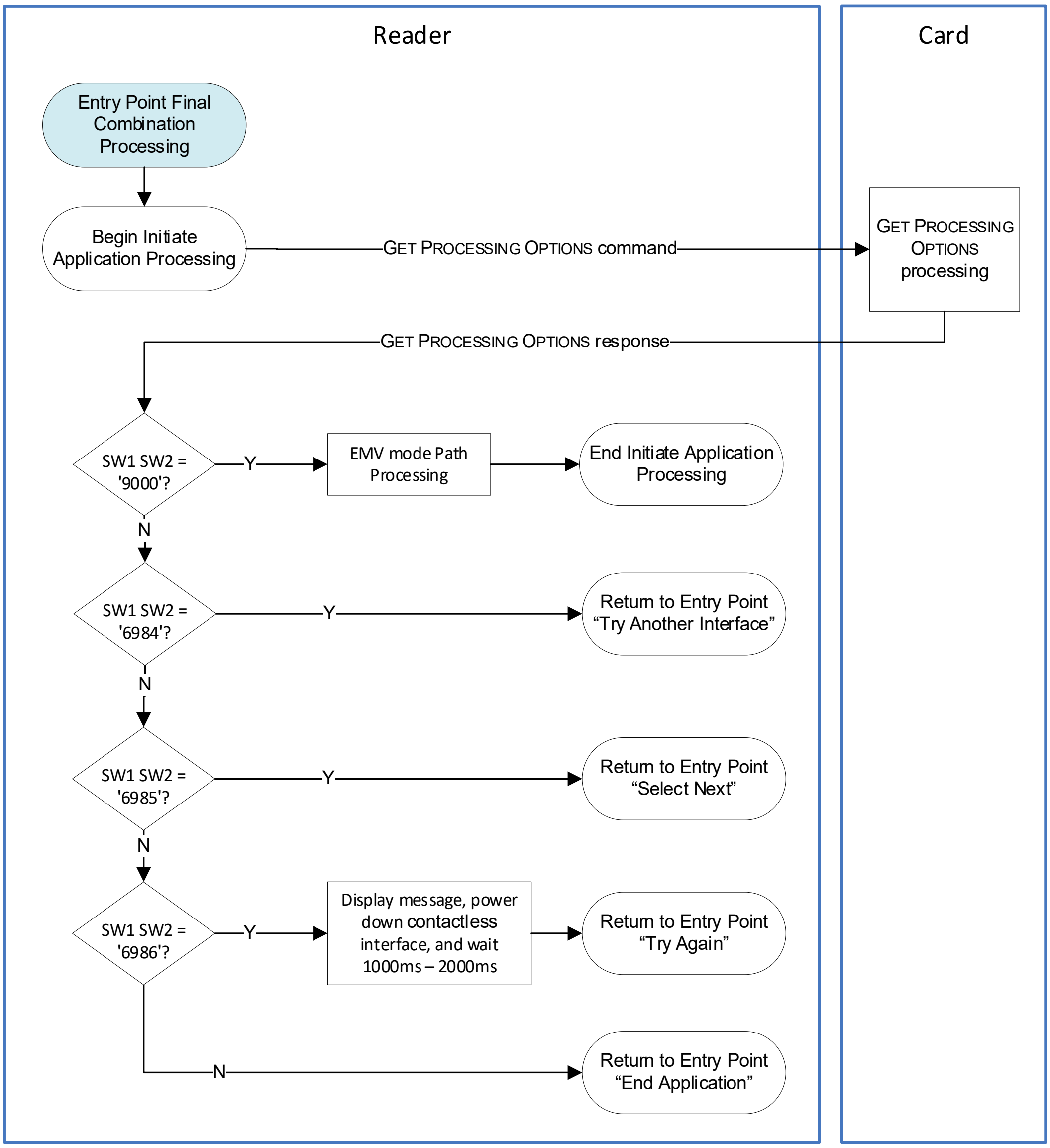
The kernel shall store all recognized data elements read, whether mandatory or optional, for later use in transaction processing. Data elements that are not recognized by the kernel (that is, their tags are unknown by the kernel) may be ignored and do not need to be stored.

Note: As the card application may return application data in both the GPO response and in records, the kernel shall not check for the presence of mandatory data elements until Card Read Complete processing. For example, although the card is required to return the Issuer Application Data in the GPO response (amongst other data elements), the kernel is required to be able to handle receiving these data elements regardless of whether they are returned in the GPO response or in a READ RECORD response.

5.2.2 Initiate Application Processing

Figure 5-1 outlines Initiate Application Processing.

Figure 5-1: Initiate Application Processing (Reader)


Reader
Card
Entry Point Final Combination Processing
Begin Initiate Application Processing
GET PROCESSING OPTIONS command
GET PROCESSING OPTIONS processing
GET PROCESSING OPTIONS response
SW1 SW2 = '9000'?
Y
EMV mode Path Processing
End Initiate Application Processing
N
SW1 SW2 = '6984'?
Y
Return to Entry Point "Try Another Interface"
N
SW1 SW2 = '6985'?
Y
Return to Entry Point "Select Next"
N
SW1 SW2 = '6986'?
Y
Display message, power down contactless interface, and wait 1000ms – 2000ms
Return to Entry Point "Try Again"
N
Return to Entry Point "End Application"

Prior to initiating the transaction with the card, Entry Point checked the SELECT response for the presence of the Processing Options Data Object List (PDOL, Tag '9F38'), and for the presence of the Terminal Transaction Qualifiers (TTQ, Tag '9F66') in the PDOL.

The kernel shall perform the following procedure to initiate the transaction with the card.

Requirements – Initiate Application Processing

5.2.2.1 (Issue GPO Command) *[5.60]*

The kernel shall issue the GET PROCESSING OPTIONS (GPO) command. The command data field is a data object coded according to the PDOL provided by the card, preceded by the Command Template tag (Tag ‘83’) and length.

Requirements – Initiate Application Processing

5.2.2.2 (GPO Response SW1 SW2) *[5.61]*

If the kernel receives SW1 SW2 = '9000' in response to the GPO (or EGPO) command,
then the kernel shall complete Initiate Application Processing and continue processing the EMV mode transaction.

Else:

- **If** the kernel receives SW1 SW2 = '6984' in response to the GPO (or EGPO) command,
  then the kernel shall provide a ***Try Another Interface*** Outcome with the following parameters:

  Try Another Interface:

  - **Start:** N/A
  - **Online Response Data:** N/A
  - **CVM:** N/A
  - **UI Request on Outcome Present:** Yes
    - Message Identifier: '18' ("Please insert or swipe card")
    - Status: Processing Error
  - **UI Request on Restart Present:** No
  - **Data Record Present:** No
  - **Discretionary Data Present:** No
  - **Alternate Interface Preference:** N/A
  - **Receipt:** N/A
  - **Field Off Request:** N/A
  - **Removal Timeout:** Zero

  This will cause the reader to switch interfaces.

- **Else, if** the kernel receives SW1 SW2 = '6985' in response to the GPO (or EGPO) command,
  then the kernel shall provide a ***Select Next*** Outcome with the following parameters:

Requirements – Initiate Application Processing

Select Next:

- **Start:** C
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** No
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will cause Entry Point to remove the current combination from the candidate list and return to Entry Point Start C, in Combination Selection.

- **Else, if** the kernel receives SW1 SW2 = ‘6986’ in response to the GPO (or EGPO) command,
  then the kernel shall provide a ***Try Again*** Outcome with the following parameters:

Requirements – Initiate Application Processing

Try Again:

- **Start:** B
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: ‘20’ (“See Phone for Instructions”)
  - Status: Processing Error
  - Hold time: 13[2]
- **UI Request on Restart Present:** Yes
  - Status: Ready to Read
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** 13[2]
- **Removal Timeout:** Zero

This will instruct the cardholder to refer to their payment device for further instructions and will interrupt the power for approximately 1300ms before Entry Point returns to Protocol Activation with the same message displayed to the cardholder.

- **Else** (SW1 SW2 does not match any of the above), the reader shall provide an ***End Application*** Outcome as defined in requirement 4.2.1.1. This will terminate the transaction.

5.2.2.3 <This requirement has been deleted>

[2] Kernel 3 testing will accept a range of 1 to 2 seconds.

5.3 Read Application Data

The kernel performs Read Application Data if an Application File Locator (AFL) was returned during Initiate Application Processing, and proceeds to Card Read Complete processing if an AFL is not returned.

During Read Application Data, the kernel uses the READ RECORD command to retrieve the card data necessary to process the transaction.

5.3.1 READ RECORD Command

Requirements – READ RECORD Command

5.3.1.1 (READ RECORD Command) *[5.63]*

The kernel shall support the READ RECORD command, as defined in Annex D of this specification.

5.3.2 Read Application Data

The kernel shall perform the following procedure to read the card application data.

Requirements – Read Application Data

5.3.2.1 (Application File Locator) *[5.64]*

If the Application File Locator (AFL, Tag ‘94’) is returned during Initiate Application Processing (section 5.2.2),
then the kernel shall perform Read Application Data as specified in [EMV 4.3 Book 3], section 10.2.

Else (the AFL was not returned) the kernel shall proceed to Card Read Complete processing (section 5.4).

5.4 Card Read Complete

After Read Application Data has been completed, the reader indicates to the cardholder that card read is complete, and the cardholder should remove their card from the reader's RF field. The kernel checks the application data returned by the card to ensure that all mandatory data for the transaction has been returned, and that redundant primitive data was not returned. Primitive data elements are redundant if more than one occurrence of the same primitive data element was returned by the card during Initiate Application Processing and Read Application Data.

The kernel shall perform the procedure described in this section for Card Read Complete.

5.4.1 Cardholder Messaging

Requirements – Cardholder Messaging

5.4.1.1 (Card Read Complete Cardholder Messaging) *[5.65]*

The kernel shall send a User Interface Request with the following parameters:

- Message Identifier: '17' ("Card Read OK")
- Status: Card Read Successfully

Depending on the configuration of the POS system, the reader may in conjunction with this message or the kernel Outcome turn the field off and/or perform a card removal procedure as defined in [EMV L1 Contactless]. This is managed by the POS system using the Autorun parameter as well as other configuration parameters. *[5.66]*

5.4.2 Mandatory and Redundant Data

The kernel examines application data returned by the card during Initiate Application Processing and Read Application Data to determine whether all mandatory data elements were returned, and whether redundant primitive data elements were returned.

Requirements – Mandatory and Redundant Data

5.4.2.1 (Mandatory Data) *[5.67]*

The kernel shall ensure that all mandatory data elements – as defined in section A.2 – are returned by the card.

If any mandatory data element for the applicable path is not present, **then** the kernel shall provide an ***End Application*** Outcome as defined in requirement 4.2.1.1. This will terminate the transaction.

5.4.2.2 (Redundant Data) *[5.68]*

Redundant primitive data elements are not permitted.

If the kernel encountered more than one occurrence of a single primitive data element while reading data from the card during Initiate Application Processing (section 5.2) and Read Application Data (section 5.3), **then** the kernel shall provide an ***End Application*** Outcome as defined in requirement 4.2.1.1. This will terminate the transaction.

5.4.3 EMV Mode Path Processing – Determine Card Transaction Disposition

The kernel shall perform the procedure defined in this section for EMV mode transactions.

Requirements – EMV Mode – Determine Card Transaction Disposition

5.4.3.1 (Cryptogram Information Data) *[5.69]*

If the card does not return the Cryptogram Information Data (CID), **then** the kernel shall:

- Construct the CID and initialize it with a value of ‘00’.
- Set CID bits 8-7 to the value of Issuer Application Data byte 5 bits 6-5[3] using identical bit settings.

[3] See section A.2 for the format of Issuer Application Data.

Requirements – EMV Mode – Determine Card Transaction Disposition

5.4.3.2 (Cryptogram Type Transaction Disposition) *[5.70]*

The kernel examines the Cryptogram Information Data (CID) to determine the cryptogram type (TC, ARQC, or AAC).

If the card returns an Application Authentication Cryptogram (AAC),
then the kernel shall set the Decline Required by Reader Indicator to 1.

If the card returns an Authorisation Request Cryptogram (ARQC),
or 'Online Cryptogram Required' by the reader (TTQ byte 2 bit 8 is 1b),
then the kernel shall set the Online Required by Reader Indicator to 1.

If the cryptogram type cannot be determined (that is, if the cryptogram is not identifiably TC, ARQC, or AAC),
then the kernel shall set the Decline Required by Reader Indicator to 1.

5.4.4 <This section has been deleted>

5.5 Processing Restrictions

The kernel performs Processing Restrictions to determine whether there are restrictions for the transaction. The kernel checks the Application Expiration date and Application Usage Control, and may check whether the card application is on the Terminal Exception File.

5.5.1 EMV Mode Path Processing

The kernel shall perform the procedure defined in this section for EMV mode transactions.

Requirements – Processing Restrictions – EMV Mode Path Processing

5.5.1.1 (Application Expired Check) *[5.74]*

Implementation-Conditional: The Application Expired Check shall be implemented for readers supporting offline transactions.

If the card application returns a Transaction Certificate (TC),
and the Terminal Transaction Date (local to the reader) is greater than the Application Expiration Date (or the Application Expiration Date is not returned by the card application),
then the application has expired and the kernel shall examine the Card Transaction Qualifiers (CTQ) to determine further processing:

- **If** 'Go online if application expired' indicated by card (CTQ byte 1 bit 4 is 1b),
  then the kernel shall set the Online Required by Reader Indicator to 1.

Else, the kernel shall set the Decline Required by Reader Indicator to 1.

Requirements – Processing Restrictions – EMV Mode Path Processing

5.5.1.2 (Terminal Exception File Check) *[5.75]*

Implementation-Optional: Terminal Exception File checking is an implementation option.

Acquirer-Merchant-Optional:
If the Terminal Exception File Check is implemented,
then it shall be acquirer-merchant configurable to be enabled or disabled.

If the card application returns a Transaction Certificate (TC),
and the Application PAN is present on the Terminal Exception File[4],
then the kernel shall set the Decline Required by Reader Indicator to 1.

5.5.1.3 (Application Usage Control – Manual Cash Transactions) *[5.76]*

Implementation-Conditional:

If the reader supports manual cash transactions,
then this functionality shall be implemented.

Acquirer-Merchant-Optional:
If the AUC-Manual Cash check is implemented,
then it shall be acquirer-merchant configurable to be enabled or disabled.

If either of the following is true:

- Issuer Country Code matches the Terminal Country Code **and** the card application is 'Valid for domestic cash transactions' (AUC byte 1 bit 8 is 1b)
- **or** Issuer Country Code does not match the Terminal Country Code **and** the card application is 'Valid for international cash transactions' (AUC byte 1 bit 7 is 1b)

then the manual cash transaction is allowed and the kernel continues processing the transaction.

Else (application is not valid for the manual cash transaction, or Issuer Country Code or AUC is not returned by the card application), the kernel shall examine the Card Transaction Qualifiers to determine further processing:

[4] This check may be achieved using the Processing Data Exchange as referenced in *Book A*, requirement 8.1.1.12.

Requirements – Processing Restrictions – EMV Mode Path Processing

- **If** 'Switch interface for cash transactions' supported by card (CTQ byte 1 bit 3 is 1b),
  then the kernel shall provide a ***Try Another Interface*** Outcome with the following parameters.

Try Another Interface:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '18' ("Please insert or swipe card")
  - Status: Processing Error
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will cause the reader to switch interface.

Else ('Switch interface for cash transactions' not supported by card or CTQ not returned), the kernel shall set the Decline Required by Reader Indicator to 1.

Requirements – Processing Restrictions – EMV Mode Path Processing

5.5.1.4 (Application Usage Control – Cashback Transactions) *[5.77]*

Implementation-Conditional:
If the reader supports transactions with cashback,
then this functionality shall be implemented.

Acquirer-Merchant-Optional:
If the AUC-Cashback check is implemented,
then it shall be acquirer-merchant configurable to be enabled or disabled.

If either of the following is true:

- Issuer Country Code matches the Terminal Country Code **and** 'Domestic cashback allowed' by card application (AUC byte 2 bit 8 is 1b)
- **or** Issuer Country Code does not match the Terminal Country Code **and** 'International cashback allowed' by card application (AUC byte 2 bit 7 is 1b)

then the transaction with cashback is allowed and the kernel continues processing the transaction.

Else (application is not valid for the transaction with cashback, or Issuer Country Code or AUC is not returned by the card application), the kernel shall examine the Card Transaction Qualifiers to determine further processing:

Requirements – Processing Restrictions – EMV Mode Path Processing

- **If** 'Switch interface for cashback transactions' supported by card (CTQ byte 1 bit 2 is 1b),
  then the kernel shall provide a ***Try Another Interface*** Outcome with the following parameters.

Try Another Interface:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '18' ("Please insert or swipe card")
  - Status: Processing Error
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will cause the reader to switch interface.

Else ('Switch interface for cashback transactions' not supported by card or CTQ not returned), the kernel shall set the Decline Required by Reader Indicator to 1.

5.5.1.5 <This requirement has been deleted>

5.6 Offline Data Authentication

Offline Data Authentication is performed by the kernel to verify the dynamic signature and authenticate the data from the card.

5.6.1 EMV Mode Path Processing

Implementation-Conditional: Offline Data Authentication shall be implemented for readers supporting offline transactions. Offline Data Authentication is performed for card requested offline transactions.

The kernel shall perform the procedure defined in this section for EMV mode transactions if the Online Required by Reader Indicator is 0 and the Decline Required by Reader Indicator is 0.

Requirements – Offline Data Authentication – EMV Mode Path Processing

5.6.1.1 (Offline capable readers) *[5.3]*

Offline capable readers shall support fDDA as defined in Annex C.

5.6.1.2 (fDDA Verification) *[5.78]*

The kernel shall verify the DDA dynamic signature according to [EMV 4.3 Book 2] and the definition of fDDA in Annex C.

If fDDA fails
or the kernel is unable to perform fDDA,
then the kernel shall examine the Card Transaction Qualifiers (CTQ) to determine further processing:

- **If** 'Go online if ODA fails' indicated by card (CTQ byte 1 bit 6 is 1b) **and** Online supported by reader, **then** the kernel shall set the Online Required by Reader Indicator to 1 and continue processing the transaction.

Else:

- **If** 'Switch interface if ODA fails' indicated by card (CTQ byte 1 bit 5 is 1b) **and** EMV contact chip supported by reader, **then** the kernel shall provide a ***Try Another Interface*** Outcome with the following parameters:

Requirements – Offline Data Authentication – EMV Mode Path Processing

Try Another Interface:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '1D' ("Please insert card")
  - Status: Processing Error
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** Contact Chip
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will switch interface with explicit indication that the contact chip interface is to be used.

Else (neither of the above or CTQ not returned), the kernel shall set the Decline Required by Reader Indicator to 1 and continue processing the transaction.

5.6.2 EMV Mode Path Processing – Special Purpose Readers

Implementation-Optional: Special purpose readers, such as transit system entry gates, may support Offline Data Authentication for Online Transactions (see requirement 3.3.4.3).

Acquirer-Merchant-Conditional: Refer to your payment system for conditions under which Offline Data Authentication for Online Transactions is to be enabled or disabled.

A special purpose reader kernel shall perform the procedure defined in this section for EMV mode transactions if the Online Required by Reader indicator is 1 and Decline Required by Reader is 0.

Special purpose readers may perform additional non-EMV processing based upon the success or failure of ODA for online processing. This is out of scope of this specification.

Requirements – Offline Data Authentication – Special Purpose Readers

5.6.2.1 (fDDA Verification for Online)

If the payment application returns Signed Dynamic Application Data (SDAD),

and the Cryptogram Information Data (CID) indicates an Authorization Request Cryptogram (ARQC),

and fDDA for Online is enabled in the kernel,

then the kernel shall verify the DDA dynamic signature according to [EMV 4.3 Book 2] section 6 and the definition of fDDA in Annex C, with the following exception:

- the reader shall check that the Signed Data Format in the recovered data (obtained after applying the recovery function to Signed Dynamic Application Data) is the value ‘95’ (instead of ‘05’)

5.6.2.2 <This requirement has been deleted>

5.7 Cardholder Verification

The kernel determines if a Cardholder Verification Method (CVM) is to be performed. The CVMs that may be supported for Kernel 3 are Online PIN, Consumer Device CVM, and Signature.

Note: A Consumer Device CVM is a CVM performed on, and validated by, the consumer's payment device, independent of the reader.

5.7.1 Cardholder Verification – EMV Mode Path Processing

Implementation-Conditional: With the exception of ATMs, Cardholder Verification shall be implemented for EMV mode-enabled readers.

Note: ATMs may need to support an appropriate minimum level of cardholder verification, as determined by the payment system or local law, regardless of the CVMs supported by the card. As a consequence, ATMs are not subject to the Cardholder Verification processing requirements of this specification.

Acquirer-Merchant-Optional: The acquirer-merchant shall be able to enable and disable the supported CVMs. However, support for the Consumer Device CVM shall be enabled (TTQ byte 3 bit 7 is 1b).

The kernel shall perform the procedure defined in this section for EMV mode transactions if the Decline Required by Reader Indicator is 0.

Requirements – Cardholder Verification – EMV Mode Path Processing

5.7.1.1 (CTQ Not Returned by Card) *[5.79]*

If the kernel requires a CVM,
and the payment application does not return the Card Transaction Qualifiers (CTQ, Tag '9F6C'),
then:

- **If** the reader supports Signature,
  then the kernel shall request a signature in the Outcome.
- **If** the reader supports only the Consumer Device CVM and Online PIN,
  then the kernel shall request Online PIN in the Outcome and shall set the Online Required by Reader Indicator to 1.
- **If** the reader supports only the Consumer Device CVM,
  then the kernel shall set the Decline Required by Reader Indicator to 1.

Note: Reader support for the Consumer Device CVM is mandatory, as is indication of its support in the Terminal Transaction Qualifiers. In addition to supporting the Consumer Device CVM, the reader may optionally be configured to support Online PIN and/or Signature.

Requirements – Cardholder Verification – EMV Mode Path Processing

5.7.1.2 (CTQ Returned by Card) *[5.80]*

If the payment application returns the CTQ (Tag '9F6C'),
then the kernel shall examine the CTQ (in the order specified below) to determine the CVM to be performed:

- **If** Online PIN Required by card (CTQ byte 1 bit 8 is 1b)
  and Online PIN supported by reader,
  then the kernel shall:

  - set the CVM parameter in the Outcome to Online PIN,
  - not examine the remaining CTQ bits for CVM processing, and
  - set the Online Required by Reader Indicator to 1, and
  - conclude CVM processing.

  Else (Online PIN not required or not supported),
  if Consumer Device CVM Performed by card (CTQ byte 2 bit 8 is 1b),
  then:

  - **If** the Card Authentication Related Data was returned during the transaction,
    then:

    - **If** Card Authentication Related Data bytes 6-7 match CTQ bytes 1-2 (respectively),
      then the kernel shall:

      - not examine the remaining CTQ bits for CVM processing,
      - set the CVM parameter in the Outcome to Confirmation Code Verified, and
      - conclude CVM processing.

Requirements – Cardholder Verification – EMV Mode Path Processing

Else (Card Authentication Related Data bytes do not match CTQ bytes[5]), the kernel shall:

- set the Decline Required by Reader Indicator to 1,
- not examine the remaining CTQ bits for CVM processing, and
- conclude CVM processing,

Else (Card Authentication Related Data was not returned during the transaction):

If the cryptogram type is an ARQC,
then the kernel shall:

- not examine the remaining CTQ bits for CVM processing,
- set the CVM parameter in the Outcome to Confirmation Code Verified, and
- conclude CVM processing.

Else (cryptogram type is not an ARQC), the kernel shall:

- set the Decline Required by Reader Indicator to 1,
- not examine the remaining CTQ bits for CVM processing, and
- conclude CVM processing.

Else (neither Online PIN required nor Consumer Device CVM performed):

o **If** Signature Required (CTQ byte 1 bit 7 is 1b),
and the reader supports Signature (TTQ byte 1 bit 2 is 1b),
then the kernel shall set the CVM parameter in the Outcome to Obtain Signature.

[5] *Note:* If the Card Authentication Related Data returned by the card is less than 7 bytes in length, then the Card Authentication Related Data bytes 6-7 cannot possibly match CTQ bytes 1-2.

Requirements – Cardholder Verification – EMV Mode Path Processing	
	Else (None of the above), a common CVM was not indicated in the CTQ and a CVM is not performed. *Note:* Cardholder Verification processing to determine the CVM to perform for the transaction, if any, is performed in this section. However, actual performance of the CVM does not take place during this point in the transaction (e.g. acquiring a signature for the transaction).
5.7.1.3	(CVM Required and CVM Not Performed) *[5.81]* **If** the reader requires a CVM and a CVM will not be performed, **then** the kernel shall set the Decline Required by Reader Indicator to 1.

5.8 Online Processing

The reader sends an authorisation request to the issuer host. Online Processing allows the issuer host to review and authorise or decline transactions using the issuer's host based risk management parameters.

5.8.1 Online Processing – EMV Mode Path Processing

Implementation-Conditional: Online Processing shall be implemented for EMV mode readers supporting online transactions (TTQ byte 1 bit 4 is 0b).

Requirements – Online Processing – EMV Mode Path Processing

5.8.1.1 (EMV Mode Online Authorisation) *[5.82]*

If the Online Required by Reader Indicator is 1,
and the Decline Required by Reader Indicator is 0,
then the kernel shall provide an ***Online Request*** Outcome with the following parameters:

Online Request:

- **Start:** as defined in requirement 5.8.1.2
- **Online Response Data:** as defined in requirement 5.8.1.2
- **CVM:** as defined in section 5.7.1
- **UI Request on Outcome Present:** Yes
  - depending on implementation, Message Identifier is either not present or set to '1B' ("Authorising, Please Wait")
  - Value Qualifier, Value, and Currency Code: see requirement 4.3.1.1
- **UI Request on Restart Present:** as defined in requirement 5.8.1.2
- **Data Record Present:** Yes

The minimum data requirements for EMV mode online messages are specified in section B.1.1.

- **Discretionary Data Present:** see requirement 4.3.1.1

As defined in requirement 4.3.1.1, AOSA may be present in Discretionary Data.

- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** as defined in requirement 5.8.1.2
- **Removal Timeout:** Zero

This will request an online transaction.

Requirements – Online Processing – EMV Mode Path Processing

5.8.1.2 (EMV Mode Online Processing Restart) *[5.83]*

If all of the following are true:

- an ***Online Request*** Outcome is being provided in response to requirement 5.8.1.1,
- **and** Issuer Update Processing is supported by the reader,
- **and** Issuer Update Processing is supported by the card (CTQ returned by card and CTQ byte 2 bit 7 is 1b),

Then the following parameters of the Outcome shall be:

- **Start:** B
- **Online Response Data:** EMV Data
- **UI Request on Restart Present:** Yes
  - Message Identifier: ‘21’ (“Present Card Again”)
- **Field Off Request:** Zero

Else the following parameters of the Outcome shall be:

- **Start:** N/A
- **Online Response Data:** N/A
- **UI Request on Restart Present:** No
- **Field Off Request:** N/A

5.9 Offline Completion

5.9.1 Offline Completion – EMV Mode Path Processing

Requirements – Offline Completion – EMV Mode Path Processing

5.9.1.1 (Cardholder Messaging for Approved) *[5.85]*

If the Online Required by Reader Indicator is 0,
and the Decline Required by Reader Indicator is 0,
then the kernel shall provide an ***Approved*** Outcome with the following parameters:

Approved:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** as defined in section 5.7.1
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '03' ("Approved")
  - Value Qualifier, Value, and Currency Code: see requirement 4.3.1.1
- **UI Request on Restart Present:** No
- **Data Record Present:** Yes

  The minimum data requirements for EMV mode clearing records are specified in section B.1.1.

- **Discretionary Data Present:** see requirement 4.3.1.1

  As defined in requirement 4.3.1.1, AOSA may be present in Discretionary Data

- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will indicate to the cardholder and merchant that the transaction has been approved.

Requirements – Offline Completion – EMV Mode Path Processing

5.9.1.2 (Cardholder Messaging for Declined) *[5.84][5.86]*

If the Decline Required by Reader Indicator is 1 **or** Online Required by Reader Indicator is 1 but online processing is not supported,
then the kernel shall provide a ***Declined*** Outcome with the following parameters:

Declined:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** No CVM
- **UI Request on Outcome Present:** Yes[6]
  - Message Identifier: '07' ("Not Authorised")
  - Value Qualifier, Value, and Currency Code: see requirement 4.3.1.1
- **UI Request on Restart Present:** No
- **Data Record Present:** Yes[6]
- **Discretionary Data Present:** see requirement 4.3.1.1

  As defined in requirement 4.3.1.1, AOSA may be present in Discretionary Data.
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will decline the transaction and indicate to the cardholder and merchant that the transaction has been declined.

5.9.1.3 (Do Not Reattempt Transaction for Declined) *[5.87]*

If Decline Required by Reader Indicator is 1,
then the reader shall not attempt to perform the transaction over another interface.

[6] Note that for non-purchase type of transactions, for instance a refund as defined in requirement 3.4.1.3, special transaction handling needs to take place.

6 Issuer Update Processing Reader Requirements

Whilst the standard new transaction processing only utilizes one card presentment, it is possible to request a second presentment in support of issuer update processing. This requires both card and reader to support a second presentment of a card after an issuer response to an online request. If so, and Issuer Update Data has been returned in an online authorisation response, the reader will request the cardholder to re-present the card. Once the card is re-presented, the Issuer Authentication and/or Issuer Script process will be performed. If the same card that was used for the new transaction is presented, then the Issuer Update Processing will complete. If a different card is presented, then either it will not be successfully selected by the reader using the same Combination as for the new transaction, or the Issuer Update Processing will fail.

The processing assumes that the same combination as used for the previous new transaction is used to select the card and that SW1 SW2 = ‘9000’ has been returned in the response to the SELECT command issued by Entry Point. The kernel will have the FCI available as well as the necessary data relevant for the transaction (e.g. Issuer Update Data).

It is also assumed that the final transaction disposition presented to the cardholder as the result of the online authorisation is deferred until after the Outcome from the issuer update processing.

Issuer update processing is performed if Issuer Update Data is present when Entry Point hands over control to Kernel 3 – as outlined in requirement 4.4.1.1.

Implementation-Optional: Issuer Update Processing is an implementation option for readers implementing EMV mode, and if supported, shall meet the requirements of both section 6.1 and section 6.2.

Acquirer-Merchant-Optional: If implemented, support for Issuer Update Processing is acquirer-merchant optional. Reader support for Issuer Update Processing is indicated to the card by setting ‘Issuer Update Processing supported’ by reader (TTQ byte 3 bit 8 to 1b).

If supported, Issuer Update Processing is conditionally performed for EMV Mode Path Processing

6.1 Issuer Update Processing Commands

To facilitate Issuer Update Processing, support for the EXTERNAL AUTHENTICATE command and issuer script processing is required.

Requirements – Issuer Update Processing Commands

6.1.1.1 (EXTERNAL AUTHENTICATE Command) *[5.88]*

The kernel shall support the EXTERNAL AUTHENTICATE command, as defined in section D.3.

6.1.1.2 (Issuer Scripts) *[5.89]*

The kernel shall support Issuer Scripts according to [EMV 4.3 Book 3], section 10.10, and [EMV 4.3 Book 4], section 6.3.9, except for the following:

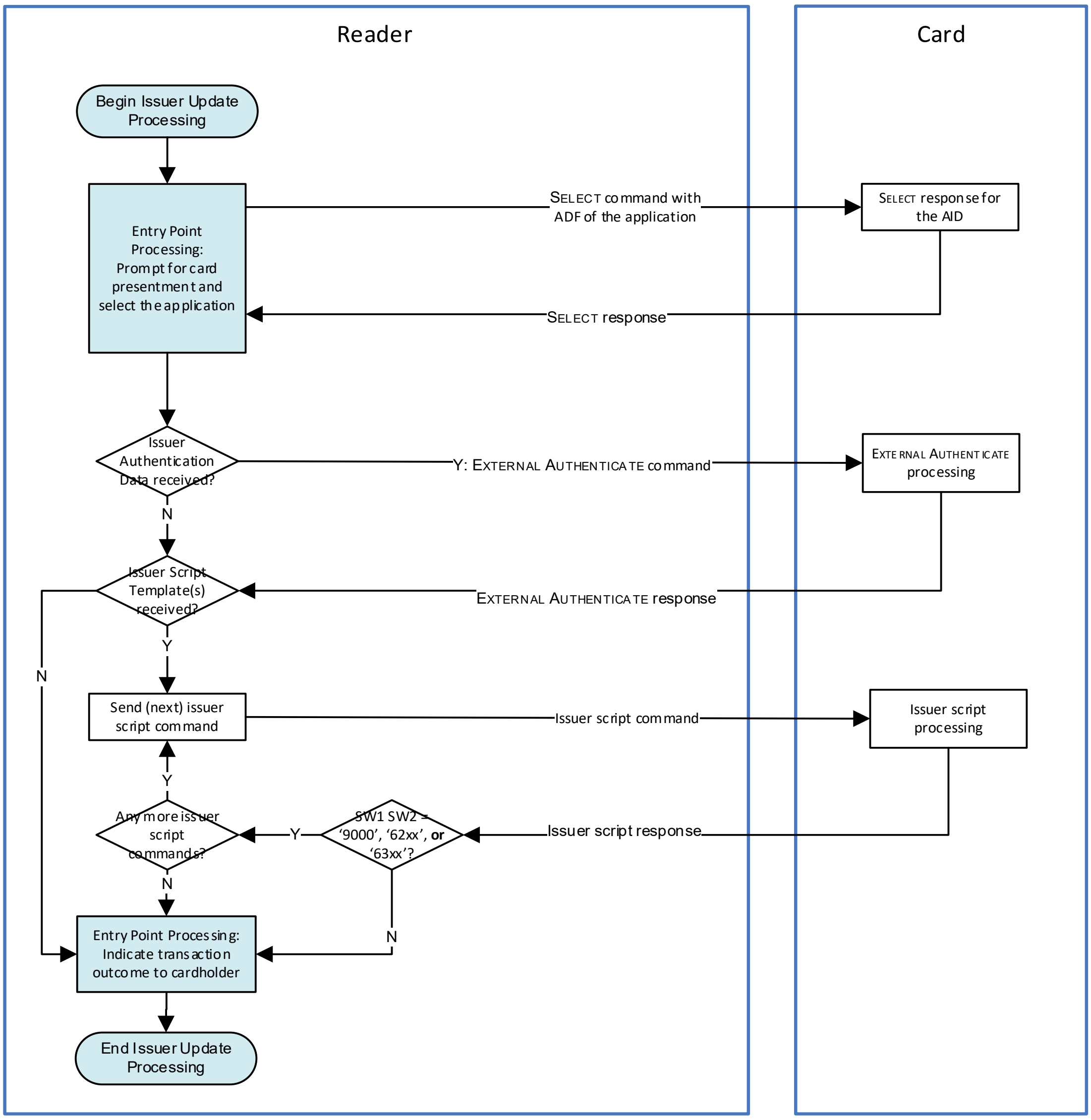
- Unlike [EMV 4.3], Issuer Script Templates with Tag '71' or Tag '72' are processed and issued by the kernel in the same manner. There is no GENERATE AC command, and Issuer Script Templates with Tag '71' and '72' are both processed during Issuer Update Processing.
- No processing is performed on the Transaction Status Information (TSI) or Terminal Verification Results (TVR).

6.2 Issuer Update Processing

The reader performs Issuer Update Processing as defined in this section.

Figure 6-1 outlines reader processing to perform Issuer Update Processing.

Figure 6-1: Issuer Update Processing (Reader)


Reader
Card
Begin Issuer Update Processing
Entry Point Processing: Prompt for card presentment and select the application
SELECT command with ADF of the application
SELECT response for the AID
SELECT response
Issuer Authentication Data received?
Y: EXTERNAL AUTHENTICATE command
EXTERNAL AUTHENTICATE processing
N
Issuer Script Template(s) received?
EXTERNAL AUTHENTICATE response
Y
N
Send (next) issuer script command
Issuer script command
Issuer script processing
Y
Any more issuer script commands?
Y
SW1 SW2 = '9000', '62xx', or '63xx'?
Issuer script response
N
N
Entry Point Processing: Indicate transaction outcome to cardholder
End Issuer Update Processing

Note: Processing as defined by the shaded shapes is outside the scope of this kernel (e.g. performed by Entry Point).

The kernel shall perform the following procedure to perform Issuer Update Processing.

Requirements – Issuer Update Processing

6.2.1.1 (EXTERNAL AUTHENTICATE Command) *[5.92]*

If Issuer Authentication Data was received in the authorisation response message,
then the kernel shall issue an EXTERNAL AUTHENTICATE command using the Issuer Authentication Data received.

Note: The kernel does not perform processing based on the card response to the EXTERNAL AUTHENTICATE command. The kernel continues Issuer Update Processing regardless of the SW1 SW2 value returned by the card.

6.2.1.2 (Issuer Script Commands) *[5.93]*

If an Issuer Script Template was received in the authorisation response,
then the kernel issues the issuer script command(s) as follows.

- Issuer Script Templates shall follow Issuer Script Template 1 or 2, and an Issuer Script Template may have multiple issuer script commands.
- The kernel shall parse all Issuer Script Templates received.
- The kernel shall parse each Issuer Script Template to retrieve each issuer script command, and shall transmit the commands to the card one by one.

If the response to an issuer script command is not SW1 SW2 = ‘9000’, ‘62xx’, or ‘63xx’,
then the kernel shall not send any further issuer script commands and shall discontinue Issuer Script Command(s) processing.

Requirements – Issuer Update Processing

6.2.1.3 (Second Presentment Completed) *[5.94]*

After completing Issuer Update Processing, the kernel shall provide an ***End Application*** Outcome with the following parameters:

End Application:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** No
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** Zero

This will terminate the transaction, allowing the final disposition (from the online authorisation response) to be presented to the cardholder.

Note: The POS system will indicate to the cardholder the transaction outcome based on the issuer authorisation response, regardless of the results of Issuer Update Processing.

Note: If the Available Offline Spending Amount (AOSA) is returned during Initiate Application Processing (section 5.2) or Read Application Data (section 5.3), then it is not included in the UI Request parameters for the second presentment.

7 Integrated Data Storage Requirements

Implementation-Optional: Integrated Data Storage processing is an implementation option for EMV mode readers. If supported it must meet the requirements of section 7.

Acquirer-Merchant-Optional: If implemented in the kernel, support for Integrated Data Storage is acquirer-merchant optional.

If supported in the kernel implementation, and enabled by the Acquirer-Merchant, Integrated Data Storage processing is conditionally performed during a first presentment if:

- Entry Point Final Combination Selection (see *Book B*) has selected a combination using Extended Selection.
- The card's response to the SELECT (AID) command contains the Integrated Data Storage Directory (IDSD, Tag 'D2'); see requirement 4.4.1.1.

The Integrated Data Storage Processing Sequence is described in section 2.5.

7.1 Read IDS Records

If instructed by the IDS Operator Application (by receiving a kernel implementation proprietary instruction via Data Exchange), the kernel sends one or more READ RECORD commands to the card, and sends the READ RECORD responses to the IDS Operator Application using Data Exchange.

Requirements – Reading IDS records

7.1.1.1 (Read IDS records – READ RECORD Command)

If the kernel is instructed by IDS Operator Application to read one or more IDS Records,

then the kernel shall use the READ RECORD command, as defined in Annex D of this specification, to read IDS Records from the IDS Record AEF.

The READ RECORD response(s) shall be sent to the IDS Operator Application using Data Exchange.

Note: As described in section 2.5, the IDS Operator Application, and the kernel implementation proprietary instructions to Read IDS Records, are out of scope of this specification.

7.2 Initiate Application Processing for IDS

After analysing the IDS Directory, and any IDS Record data provided by the card during Read IDS Records, the IDS Operator Application determines whether to update any IDS Record(s).

If the IDS Operator Application determines that one or more IDS Records are to be updated, then the EXTENDED GPO (EGPO) command is used, as described in this section. In this case the IDS Operator Application sends to the kernel, via Data Exchange, an instruction to use the EGPO command, together with the IDS Record Update Template to be used in EGPO command data.

The kernel is not required to validate the content of the IDS Record Update Template.

If the IDS Operator Application determines that no IDS Records are to be updated then processing reverts to standard Initiate Application Processing, using the GPO command, as defined in section 5.2.2. In this case the IDS Operator Application sends to the kernel, via Data Exchange, an instruction to use the GPO command.

7.2.1 EXTENDED GPO (EGPO) Command

Requirements – EGPO Command

7.2.1.1 (EGPO Command)

The kernel shall support the EXTENDED GET PROCESSING OPTIONS (EGPO) command, as defined in Annex D.4 of this specification.

7.2.2 Initiate Application Processing for IDS

Requirements – Initiate Application Processing for IDS

7.2.2.1 (Issue GPO or EGPO Command)

If the kernel is instructed by IDS Operator Application that no IDS Record updates are required,

then the kernel shall continue with requirement 5.2.2.1 (Issue GPO Command)

Else (IDS Record updates are required),

then the kernel shall issue the EXTENDED GET PROCESSING OPTIONS (EGPO) command.

Note: As described in section 2.5, the IDS Operator Application, and the kernel implementation proprietary instructions to issue the GPO or EGPO commands, are out of scope of this specification.

If IDS Record updates are required the IDS Operator Application will send to the kernel the IDS Record Update Template (required for the EGPO command) via Data Exchange.

7.2.2.2 (EGPO Response SW1 SW2)

When the kernel receives the EGPO response SW1 SW2, the kernel shall continue processing with requirement 5.2.2.2 (GPO Response SW1 SW2).

Annex A Kernel 3 Data Elements

This annex defines the data elements used in this specification for financial transaction interchange and their mapping onto data objects.

Table A-1 on page 81 lists the data elements used in Kernel 3, sorted by name.

Table A-2 on page 112 lists the data elements by tag.

Table A-3 on page 115 lists the data elements that the reader must make available to Kernel 3 as specified in *Book A*, requirement 8.1.1.7.

A.1 Data Element Descriptions

A.1.1 Requirements

Requirements – Data Elements

A.1.1.1 (Data Element Requirements) *[D.1]*

The kernel shall comply with the requirements, where specified and applicable, in Table A-1: Kernel 3 Data Elements.

A.1.2 Name Column

The Name column of Table A-1 lists the name of the data element, and also includes the following:

- Format (F) of the data element. The [EMV 4.3]defined supported formats are as follows:
    - n (numeric)
    - cn (compressed numeric)
    - b (binary or bit string)
    - an (alphanumeric)
    - ans (alphanumeric special)
- Tag (T) of the data element in hexadecimal.

- Length (L) of the data element. The value of the length is shown in decimal. When the length defined for the data object is greater than the length of the actual data, the following rules apply:
  - A data element in format n is right-justified and padded with leading hexadecimal zeros
  - A data element in format cn is left-justified and padded with trailing hexadecimal ‘F’s
  - A data element in format an or format ans is left-justified and padded with trailing hexadecimal zeros

  When data is moved from one entity to another (for example, card to reader), it shall always be passed in order from high order to low order, regardless of how it is internally stored. The same rules apply when concatenating data.
- Source (S) of the data element, indicated as “Card”, “Reader”, “Issuer” or “Data Exchange”.

A.1.3 Requirement Column

The Requirement column of Table A-1 lists the requirements for the data element:

- Mandatory – The data element must always be present and provided to the kernel if the source is the card. If the data element is not received by the kernel, then the kernel terminates the transaction as defined in requirement 5.4.2.1.
- Required – The data element must always be present, but the kernel does not terminate the transaction if the data element is not present.
- Conditional – The data element is necessary under the conditions specified.
- Optional – The data element is optional.

A.1.4 Retrieval Column

The Retrieval column of Table A-1 lists the command(s) typically used to retrieve each data element that has the card as the source.

As defined in requirement 5.4.2.1, the reader shall not evaluate the presence of mandatory data elements per command, and shall only evaluate the presence of mandatory data when the reader/card interaction has been finalized.

The following values are used to indicate support for retrieval of card data elements:

- **N/A** indicates that the data element is not a card data element.
- **GPO** indicates that the data element may be retrieved as part of the data sent in the response to the GET PROCESSING OPTIONS command.

- **READ RECORD** indicates that retrieval of the data element is allowed using the READ RECORD command.
- **SELECT** indicates that the data element may be retrieved as part of the data sent in the response to the SELECT command.

A.2 Data Elements by Name

Table A-1: Kernel 3 Data Elements

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Amount, Authorised (Numeric) F: n 12 T: '9F02' L: 6 S: Reader	Required	Authorised amount of the transaction (including Amount, Other and excluding adjustments).	N/A	
Amount, Other (Numeric) F: n 12 T: '9F03' L: 6 S: Reader	Conditional If cashback supported	Secondary amount associated with the transaction representing a cashback amount.	N/A	
Application Cryptogram (AC) F: b 64 T: '9F26' L: 8 S: Card	Mandatory	Cryptogram returned by the card in response to the GPO command.	GPO	
Application Definition File (ADF) Name		See entry for "Application Identifier" (Tag '4F').		

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Application Expiration Date F: n 6 YYMMDD T: ‘5F24’ L: 3 S: Card	Conditional If fDDA supported	Date after which the card application expires. For transactions where Offline Data Authentication is performed, the Application Expiration Date is returned. For transactions where Offline Data Authentication is not performed, the Application Expiration Date does not need to be returned.	READ RECORD	
Application File Locator (AFL) F: var. T: ‘94’ L: var. up to 252 S: Card	Conditional If returning record data for the transaction	Indicates the location (SFI, range of records) of the AEFs related to a given application.	GPO	For each file to be read, the Application File Locator contains the following four bytes: Byte1 bits 8-4 = SFI bits 3-1 = 000 Byte 2: First (or only) record number to be read for that SFI (never equal to zero) Byte 3: Last record number to be read for that SFI (shall be greater than or equal to byte 2) Byte 4: Number of consecutive records involved in authentication of static data, starting with record number in byte 2 (may range from zero to the value of the third byte minus the value of the second byte + 1)

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Application Identifier (ADF Name) F: b 40-128 T: ‘4F’ L: 5-16 S: Card	Mandatory	The ADF Name identifies the application as described in [ISO 7816-5]. The AID is made up of the Registered Application Provider Identifier (RID) and the Proprietary Identifier Extension (PIX).	SELECT	Applicable AID as defined by the payment system
Application Identifier (AID) F: b 40-128 T: ‘9F06’ L: 5-16 S: Reader	Required	Identifies the application as described in [ISO 7816-5].	N/A	

Name (Format; Tag; Length; Source; Path)	**Requirement**	**Description**	**Retrieval**	**Values**
Application Interchange Profile (AIP) F: b 16 T: ‘82’ L: 2 S: Card	Mandatory	Indicates the capabilities of the card to support specific functions in the application. Kernel 3 shall not act on AIP bit settings that are not supported for Kernel 3 or that are Reserved for Future Use (RFU).	GPO	Byte 1 bits 8-7: RFU (0,0) bit 6: 1 = DDA is supported for EMV mode bit 5: Not used for Kernel 3 bit 4: Not used for Kernel 3 bit 3: Not used for Kernel 3 bit 2: Not used for Kernel 3 bit 1: Not used for Kernel 3 Byte 2 bit 8: 1 = Mag-stripe mode is supported bit 7: 1 = Mobile phone bit 6: 1 = Contactless transaction bits 5-1: RFU (0,0,0,0,0) *Note*: The AIP 'Mag-stripe mode is supported' bit is set to 0b for products using this specification.
Application Label F: ans 1-16 * T: ‘50’ L: 1-16 S: Card * (special characters limited to spaces)	Optional	Mnemonic associated with AID according to [ISO 7816-5]. Used in application selection. Application Label is optional in the File Control Information (FCI) of an Application Definition File (ADF) and optional in an ADF directory entry	SELECT	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Application Preferred Name F: ans 1-16 T: '9F12' L: 1-16 S: Card P: E M	Optional	Preferred mnemonic associated with the AID.	SELECT	The permitted characters for Application Preferred Name are the non-control characters defined in the ISO/IEC 8859 part designated in the Issuer Code Table Index associated with the Application Preferred Name
Application Primary Account Number (PAN) F: var. up to cn 19 T: '5A' L: var. up to 10 S: Card	Conditional If fDDA supported	Cardholder account number. For transactions where Offline Data Authentication is performed, the Application PAN is returned. For transactions where Offline Data Authentication is not performed, the Application PAN does not need to be returned.	READ RECORD	If the value of the Application Primary Account Number does not match the account number in Track 2 Equivalent Data (tag '57'), the reader shall terminate the transaction.
Application Primary Account Number Sequence Number (PSN) F: n 2 T: '5F34' L: 1 S: Card	Optional	Identifies and differentiates card applications with the same PAN.	GPO, READ RECORD	*Note:* Although this field is optional in the card, if it is present in the card it is sent in online messages. If it is not sent in online messages, the value is assumed to be 00 for key derivations.

Name (Format; Tag; Length; Source; Path)	**Requirement**	**Description**	**Retrieval**	**Values**
Application Priority Indicator F: b 8 T: ‘87’ L: 1 S: Card	Conditional If multiple contactless payment applications on card	Indicates the priority of a given application or group of applications in a directory.	SELECT	bit 8: Not used for Kernel 3 bits 7-5: RFU (0,0,0) bits 4-1: 0000 = No priority assigned xxxx = Order in which the application is to be selected, ranging from 1 to 15, with 1 being the highest priority
Application Program Identifier (Program ID) F: b T: ‘9F5A’ L: var. 1-16 S: Card	Optional	Payment system proprietary data element identifying the Application Program ID of the card application. When personalised, the Application Program ID is returned in the FCI Issuer Discretionary Data of the SELECT response (Tag ‘BF0C’).	SELECT	As defined by the payment system
Application Selection Registered Proprietary Data (ASRPD) F: b T: ‘9F0A’ L: var. S: Card	Optional	Market-proprietary data that may be required by local regulatory authority to offer specific services based on this information, as defined in [EMV ASRPD].	SELECT	See [EMV ASRPD].

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Application Transaction Counter (ATC) F: b 16 T: ‘9F36’ L: 2 S: Card	Mandatory	Count of the number of transactions initiated since personalisation. Maintained by the application in the card.	GPO	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Application Usage Control (AUC) F: b 16 T: '9F07' L: 2 S: Card	Conditional If supporting AUC check for cash or cashback	Indicates issuer-specified restrictions on the geographic usage and services allowed for the card application.	READ RECORD	Byte 1 bit 8: 1 = Valid for domestic cash transactions bit 7: 1 = Valid for international cash transactions bits 6-3: Not used for Kernel 3 bit 2: 1 = Valid at ATMs *Note:* The AUC 'Valid at ATMs' bit is used to indicate whether the card application supports *contact chip* ATM transactions. Bit 1: RFU (0) Byte 2 bit 8: 1 = Domestic cashback allowed bit 7: 1 = International cashback allowed bits 6-1: RFU (0,0,0,0,0,0) *Note:* To determine whether Application Usage Control restrictions for "domestic" and "international" transactions have been met, the reader compares the Issuer Country Code to the Terminal Country Code.

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Authorisation Code F: ans 6 * T: ‘89’ L: 6 S: Issuer * (special characters limited to spaces)	Conditional If Issuer Update Processing supported (From issuer, not passed to card)	Nonzero value generated by the issuer for an approved transaction.	N/A	
Authorisation Response Code (ARC) F: an 2 T: ‘8A’ L: 2 S: Issuer/Reader	Conditional If Issuer Update Processing supported	Indicates the transaction disposition of the transaction received from the issuer for online authorisations.	N/A	Codes generated by the issuer are as indicated in [ISO 8583:1987]. 00, 10, or 11 indicate an issuer approval. 01 or 02 indicates an issuer referral. An ARC other than the ones listed above indicates an issuer decline. The following codes are generated by the reader or terminal for the following conditions: Y1 = Offline approved Z1 = Offline declined Y3 = Unable to go online (offline approved) Z3 = Unable to go online (offline declined)

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Available Offline Spending Amount (AOSA) F: n 12 T: ‘9F5D’ L: 6 S: Card	Optional	Kernel 3 proprietary data element indicating the remaining amount available to be spent offline. The AOSA is a calculated field used to allow the reader to provide on a receipt or display the amount of offline spend that is available on the card	GPO	
Card Authentication Related Data F: b T: ‘9F69’ L: var. 5-16 S: Card	Conditional If fDDA supported	Contains the fDDA Version Number, Card Unpredictable Number, and Card Transaction Qualifiers. For transactions where fDDA is performed, the Card Authentication Related Data is returned in the last record specified by the Application File Locator for that transaction.	READ RECORD	Byte 1: fDDA Version Number (‘01’) Byte 2-5: (Card) Unpredictable Number Byte 6-7: Card Transaction Qualifiers

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Card Transaction Qualifiers (CTQ) F: b 16 T: ‘9F6C’ L: 2 S: Card	Conditional If CVM supported or if issuer CTQ preferences supported or if Issuer Update Processing at the POS supported	In this version of the specification, used to indicate to the device the card CVM requirements, issuer preferences, and card capabilities.	GPO	Byte 1 bit 8: 1 = Online PIN Required bit 7: 1 = Signature Required bit 6: 1 = Go Online if Offline Data Authentication Fails and Reader is online capable. Bit 5: 1 = Switch Interface if Offline Data Authentication fails and Reader supports contact chip. Bit 4: 1 = Go Online if Application Expired bit 3: 1 = Switch Interface for Cash Transactions bit 2: 1 = Switch Interface for Cashback Transactions bit 1: RFU (0) Byte 2 bit 8: 1 = Consumer Device CVM Performed bit 7: 1 = Card supports Issuer Update Processing at the POS bits 6-1: RFU (0,0,0,0,0,0)

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Cardholder Name F: ans 2-26 T: '5F20' L: 2-26 S: Card	Optional	Indicates cardholder name according to [ISO 7813].	GPO, READ RECORD	
Certificate Authority Public Key F: b T: – L: – S: Reader	Conditional If fDDA supported	Payment system public key used for dynamic data authentication.	N/A	Value generated by the payment system CA and loaded to terminal by acquirer. Up to six public keys per payment system must be supported.
Certificate Authority Public Key Check Sum F: b T: – L: 20 S: Reader	Conditional If fDDA supported	A check value calculated on the concatenation of all parts of the Certificate Authority Public Key (RID, Certificate Authority Public Key Index, Certificate Authority Public Key Modulus, Certificate Authority Public Key Exponent) using SHA-1.	N/A	
Certificate Authority Public Key Exponent F: b T: – L: 1 or 3 S: Reader	Conditional If fDDA supported	Value of the exponent part of the Certificate Authority Public Key.	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Certificate Authority Public Key Index (PKI) F: b 8 T: '8F' L: 1 S: Card	Conditional If Offline Data Authentication supported	Identifies the Certificate Authority's public key in conjunction with the RID for use in offline data authentication.	READ RECORD	Values assigned by the payment system.
Certificate Authority Public Key Index (PKI) F: b 8 T: '9F22' L: 1 S: Reader	Conditional If fDDA supported	Identifies the Certificate Authority's public key in conjunction with the RID for use in offline static and dynamic data authentication.	N/A	Values assigned by the payment system.
Certificate Authority Public Key Modulus F: b T: – L: N_{CA} (up to 248) S: Reader	Conditional If fDDA supported	Value of the modulus part of the Certificate Authority Public Key.	N/A	
Command Template F: b T: '83' L: var. S: Reader	Required	Identifies the data field of a command message.	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Cryptogram Information Data (CID) F: b 8 T: ‘9F27’ L: 1 S: Card	Required	Indicates the type of cryptogram (TC, ARQC, or AAC) returned by the card and the actions to be performed by the reader.	GPO	bits 8–7 00 = AAC 01 = TC 10 = ARQC 11 = RFU bits 6-5: RFU (0,0) bits 4-1: Not used for Kernel 3
Customer Exclusive Data (CED) F: b T: ‘9F7C’ L: var. up to 32 S: Card	Optional	Contains data for transmission to the issuer.	GPO, READ RECORD	Customer Exclusive Data, if personalised, consists of one or more proprietary Issuer elements.
Decline Required by Reader Indicator F: – T: – L: – S: Reader	Required	Proprietary internal indicator used during transaction processing to indicate that internal reader processes have indicated that the transaction should be declined.	N/A	This indicator is a transient value, initialized to a value of 0 at the beginning of the transaction. 1 = Offline decline required by reader
Dedicated File (DF) Name F: b 40-128 T: ‘84’ L: 5-16 S: Card	Required	Identifies the name of the DF as described in [ISO 7816-4].	SELECT	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Directory Entry F: var. T: ‘61’ L: var. S: Card	Mandatory	Contains one or more data objects relevant to an application directory entry according to [ISO 7816-5].	SELECT	
fDDA Version Number F: b 8 T: part of ‘9F69’ L: 1 S: Card	Conditional If fDDA supported	Contains the version number for the fDDA version supported by the card	READ RECORD	‘01’ = fDDA Version Number 1
File Control Information (FCI) Issuer Discretionary Data F: var. T: ‘BF0C’ L: var. up to 222 S: Card	PPSE: Mandatory ADF: Conditional If data objects present in this template	Issuer discretionary part of the FCI.	SELECT	
File Control Information (FCI) Proprietary Template F: var. T: ‘A5’ L: var. S: Card	Mandatory	Identifies the data objects proprietary to [EMV 4.3] in the FCI Template according to [ISO 7816-4].	SELECT	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
File Control Information (FCI) Template F: var. T: ‘6F’ L: var. up to 252 S: Card	Mandatory	Identifies the FCI template according to [ISO 7816-4].	SELECT	
Form Factor Indicator (FFI) F: b 32 T: ‘9F6E’ L: 4 S: Card (and reader)	Required	Indicates the form factor of the consumer payment device and the type of contactless interface over which the transaction was conducted. This information is made available to the issuer host.	GPO, READ RECORD	Byte 1 thru 3: Out-of-scope of Kernel 3. Byte 4: Payment Transaction Technology bits 8-5: RFU (0,0,0,0) bits 4-1: Payment Transaction Technology All values not currently defined are RFU. 0000 = Proximity Contactless interface using [ISO 14443] (including NFC)
Integrated Circuit Card (ICC) Public Key Certificate F: b T: ‘9F46’ L: N_I S: Card	Conditional If fDDA supported	ICC Public Key certified by the issuer.	READ RECORD	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Integrated Circuit Card (ICC) Public Key Exponent F: b T: ‘9F47’ L: 1 or 3 S: Card	Conditional If fDDA supported	ICC Public Key Exponent used for the verification of the Signed Dynamic Application Data.	READ RECORD	
Integrated Circuit Card (ICC) Public Key Remainder F: b T: ‘9F48’ L: $N_{IC} – N_{I} + 42$ S: Card	Conditional If fDDA supported and entire public key does not fit into certificate	Digits of the ICC Public Key Modulus which do not fit within the ICC Public Key Certificate.	READ RECORD	
Integrated Data Storage Directory (IDSD) F: b T: ‘D2’ L: var S: Card	Conditional If Integrated Data Storage is supported in card	Directory of Integrated Data Storage records on the card.	SELECT	See Annex F for description of content for an example Integrated Data Storage implementation.
Integrated Data Storage Record Update Template F: b T: ‘BF60’ L: var S: Data Exchange	Conditional If Integrated Data Storage is supported in card	Part of the command data for the EXTENDED GET PROCESSING OPTIONS command. The IDS Record Update Template contains data to be updated in one or more IDS Records.	N/A	See Annex F for description of content for an example Integrated Data Storage implementation.

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Issuer Application Data (IAD) F: b T: '9F10' L: var. up to 32 S: Card	Mandatory	Contains proprietary application data for transmission to the Issuer in an online transaction.	GPO	Byte 1-4: Out of scope Byte 5: bits 8-7: Out of scope bits 6-5: 00 = AAC returned in GPO 01 = TC returned in GPO 10 = ARQC returned in GPO 11 = RFU bit 4-1 = Out of scope Byte 6 - end: Out of scope
Issuer Authentication Data F: b 64-128 T: '91' L: 8-16 S: Issuer	Optional Passed from the issuer through the reader	Issuer data transmitted to card for Issuer Authentication.	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Issuer Code Table Index F: n 2 T: '9F11' L: 1 S: Card	Conditional If Application Preferred Name is personalised	Indicates the code table according to [ISO 8859] for displaying the Application Preferred Name.	SELECT	Values are: 01 = [ISO 8859] Part 1 02 = [ISO 8859] Part 2 03 = [ISO 8859] Part 3 04 = [ISO 8859] Part 4 05 = [ISO 8859] Part 5 06 = [ISO 8859] Part 6 07 = [ISO 8859] Part 7 08 = [ISO 8859] Part 8 09 = [ISO 8859] Part 9 10 = [ISO 8859] Part 10
Issuer Country Code F: n 3 T: '5F28' L: 2 S: Card	Conditional If Application Usage Control is supported	Indicates the country of the issuer, represented according to [ISO 3166].	READ RECORD	
Issuer Identification Number (IIN) F: n 6 T: '42' L: 3 S: Card	Optional	Number that identifies the major industry and the card issuer and that forms part of the Primary Account Number (PAN). In template 'BF0C' of the SELECT response.	SELECT	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Issuer Identification Number Extended (IINE) F: n 6 or 8 T: '9F0C' L: 3 or 4 S: Card	Optional	Number that identifies the major industry and the card issuer and that forms part (6 or 8 digits) of the Primary Account Number (PAN). While the first 6-digits of the IINE (tag '9F0C') and IIN (tag '42') are the same and there is no need to have both data objects on the card, cards may have both the IIN and IINE data objects present. In template 'BF0C' of the SELECT response.	SELECT	
Issuer Public Key Certificate F: b T: '90' L: N_{CA} S: Card	Conditional If Offline Data Authentication supported	Issuer’s public key certified by a certificate authority for use in offline data authentication.	READ RECORD	
Issuer Public Key Exponent F: b T: '9F32' L: 1 or 3 S: Card	Conditional If Offline Data Authentication supported	Issuer public key exponent used for the verification of the ICC Public Key Certificate.	READ RECORD	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Issuer Public Key Remainder F: b T: '92' L: $N_I - N_{CA} + 36$ S: Card	Conditional If Offline Data Authentication supported and entire public key does not fit into certificate	Portion of the Issuer Public Key Modulus which does not fit into the Issuer PK Certificate.	READ RECORD	
Issuer Script Identifier F: b 32 T: '9F18' L: 4 S: Issuer	Optional From issuer to reader. Not passed to card.	May be sent in authorisation response from issuer when response contains Issuer Script. Assigned by the issuer to uniquely identify the Issuer Script.	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Issuer Script Results F: b T: '9F5B' L: var. S: Reader	Conditional If Issuer Update Processing supported	Indicates the results of Issuer Script processing. When the reader/terminal transmits this data element to the acquirer, in this version of Kernel 3, it is acceptable that only byte 1 is transmitted, although it is preferable for all five bytes to be transmitted.	N/A	Byte 1 (Issuer Script Result): bits 8-5: Result of the Issuer Script processing performed by the kernel: '0' = Issuer Script not performed '1' = Issuer Script processing failed '2' = Issuer Script processing successful bits 4-1: Sequence number of the Issuer Script Command: '0' = Not specified '1'–'E' = Sequence number 1-14 'F' = Sequence number 15 or above Bytes 2-5 (Issuer Script Identifier): Issuer Script Identifier received by the reader, if available; zero filled if not available. Mandatory if more than one Issuer Script Template was received by the reader. *– continues –*

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Issuer Script Results *– continued –*				Bytes 1-5 are repeated for each Issuer Script Template processed by the kernel, although in this version of Kernel 3, only one Issuer Script Template may be transmitted in the response message.
Issuer Script Template 1 F: b T: '71' L: var. S: Issuer	Optional Passed from issuer through reader	Contains proprietary issuer data for transmission to the card. The format of the Issuer Script Template is shown in [EMV 4.3 Book 3], section 10.10.	N/A	[EMV 4.3] specifies that terminals and networks must support a total length for all issuer scripts in an online response of up to 128 bytes. Issuers may send longer issuer scripts only when the issuer knows that longer issuer scripts are supported by all entities transporting the script back to the card.
Issuer Script Template 2 F: b T: '72' L: var. S: Issuer	Optional Passed from issuer through reader	Contains proprietary issuer data for transmission to the card. The format of the Issuer Script Template is shown in [EMV 4.3 Book 3], section 10.10.	N/A	[EMV 4.3] specifies that terminals and networks must support a total length for all issuer scripts in an online response of up to 128 bytes. Issuers may send longer issuer scripts only when the issuer knows that longer issuer scripts are supported by all entities transporting the script back to the card.
Language Preference F: an 2 T: '5F2D' L: 2-8 S: Card	Optional	1-4 languages stored in order of preference, each represented by 2 lower case alphabetical characters according to [ISO 639-1].	SELECT	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Log Entry F: b T: '9F4D' L: 2 S: Card	Conditional If transaction logging supported	Data element indicating the location (SFI) and the maximum number of transaction log records.	SELECT	Byte 1: SFI containing the cyclic transaction log file. Byte 2: Maximum number of records in the transaction log file.
Log Format F: b T: '9F4F' L: var. S: Card	Conditional If transaction logging supported	List in tag and length format of data elements that are logged by the transaction	GET DATA	
Merchant Name and Location F: ans T: '9F4E' L: var. S: Reader	Required (at reader)	Indicates the name and location of the merchant. The reader shall return the value of the Merchant Name and Location when requested by the card in a Data Object List.	N/A	
Online Required by Reader Indicator F: – T: – L: – S: Reader	Required	Proprietary internal indicator used during transaction processing to indicate that internal reader processes have indicated that the transaction should be online requested.	N/A	This indicator is a transient value, initialized to a value of 0 at the beginning of the transaction. 1 = Online required by reader

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Payment Account Reference (PAR) F: an T: '9F24' L: 29 S: Card	Optional	A non-financial reference assigned to each unique PAN and used to link a Payment Account represented by that PAN to affiliated Payment Tokens, as defined in [EMV Tokenisation]. The PAR may be assigned in advance of Payment Token issuance.	READ RECORD	See [EMV Tokenisation].
Processing Options Data Object List (PDOL) F: b T: '9F38' L: var. S: Card	Mandatory	List of terminal/reader-related data objects (tags and lengths) requested by the card to be transmitted in the GET PROCESSING OPTIONS command.	SELECT	
Reader Contactless Floor Limit F: n 12 T: – L: 6 S: Reader	Conditional If Entry Point Pre-Processing supported	Indicates the contactless floor limit of the reader for a specific AID. If the transaction amount is greater than the Reader Contactless Floor Limit, then the reader requires online processing for the transaction. As defined in *Book B*.	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Reader Contactless Transaction Limit F: n 12 T: – L: 6 S: Reader	Conditional If Entry Point Pre-Processing supported	Indicates the contactless transaction limit of the reader for a specific AID. If the transaction amount is greater than or equal to the Reader Contactless Transaction Limit, then a contactless transaction is not permitted. Switching the transaction over to another interface is permitted. As defined in *Book B*.	N/A	
Reader CVM Required Limit F: n 12 T: – L: 6 S: Reader	Conditional If Entry Point Pre-Processing supported	Indicates the CVM limit of the reader for a specific AID. If the transaction amount is greater than or equal to the Reader CVM Required Limit, then the reader requires a CVM for the transaction. As defined in *Book B.*	N/A	
Response Message Template Format 2 F: var. T: '77' L: var. S: Card	Required	Contains the data objects (with tags and lengths) returned by the card in response to a command.	GPO	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Short File Identifier F: b 8 T: '88' L: 1 S: Card	Conditional If returning record data for the transaction	Used in the commands related to an application elementary file (AEF) to identify the file. The SFI data object is a binary field with the three high-order bits set to zero.	N/A	Values are: 1–10: Governed by joint payment systems 11–20: Payment system specific 21–30: Issuer specific
Signed Dynamic Application Data (SDAD) F: b T: '9F4B' L: N_{IC} S: Card	Conditional If fDDA supported	Dynamic signature generated by the card and validated by the reader during fDDA processing.	GPO, READ RECORD	
Static Data Authentication (SDA) Tag List F: – T: '9F4A' L: var. S: Card	Optional	Contains list of tags of primitive data objects whose value fields are to be included in the ICC Public Key Certificate hash result.	READ RECORD	The SDA Tag List may not contain tags other than the tag for Application Interchange Profile (AIP).
Terminal Country Code F: n 3 T: '9F1A' L: 2 S: Reader	Required	Indicates the country of the terminal represented according to [ISO 3166].	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Terminal Floor Limit F: b 32 T: '9F1B' L: 4 S: Reader	Conditional If Entry Point Pre-Processing supported and Reader Contactless Floor Limit is not present	Indicates the floor limit in the terminal. As defined in *Book B*.	N/A	
Terminal Transaction Qualifiers (TTQ) F: b 32 T: '9F66' L: 4 S: Reader	Required	Indicates reader capabilities, requirements, and preferences to the card. TTQ byte 2 bits 8-7 are transient values, and reset to zero at the beginning of the transaction. All other TTQ bits are static values, and not modified based on transaction conditions. TTQ byte 3 bit 7 shall be set by the acquirer-merchant to 1b.	N/A	Byte 1 bit 8: 1 = Mag-stripe mode supported bit 7: RFU (0) bit 6: 1 = EMV mode supported bit 5: 1 = EMV contact chip supported bit 4: 1 = Offline-only reader bit 3: 1 = Online PIN supported bit 2: 1 = Signature supported bit 1: 1 = Offline Data Authentication for Online Authorizations supported. *Note*: The TTQ 'Mag-stripe mode supported' bit is set to 0b for products using this specification. *– continues –*

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Terminal Transaction Qualifiers (TTQ) *– continued –*				Byte 2 bit 8: 1 = Online cryptogram required *Note*: A qVSDC online-only reader must have TTQ byte 2 bit 8 set to 1b. It may be coded to 1b or set as a result of device configuration parameters. bit 7: 1 = CVM required bit 6: 1 = (Contact Chip) Offline PIN supported bits 5-1: RFU (0,0,0,0,0) Byte 3 bit 8: 1 = Issuer Update Processing supported bit 7: 1 = Consumer Device CVM supported bits 6-1: RFU (0,0,0,0,0,0) Byte 4 RFU (0,0,0,0,0,0,0,0)
Terminal Verification Results (TVR) F: b 40 T: '95' L: 5 S: Reader	Required	Status of the different functions as seen from the reader/terminal. For EMV mode transactions, all of the TVR bits sent online to the acquirer shall be set to 0b.	N/A	Byte 1: Not used for Kernel 3 ('00') Byte 2: Not used for Kernel 3 ('00') Byte 3: Not used for Kernel 3 ('00') Byte 4: Not used for Kernel 3 ('00') Byte 5: Not used for Kernel 3 ('00')

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Track 2 Equivalent Data F: b T: '57' L: var. up to 19 S: Card	Mandatory	Contains the data elements of the Track 2 according to the [ISO 7813], excluding start sentinel, end sentinel, and LRC	GPO, READ RECORD	
Transaction Currency Code F: n 3 T: '5F2A' L: 2 S: Reader	Required	Indicates the currency code of the transaction according to [ISO 4217]. The implied exponent is indicated by the minor unit of currency associated with the Transaction Currency Code in [ISO 4217].	N/A	
Transaction Date F: n 6 YYMMDD T: '9A' L: 3 S: Reader	Required	Local date that the transaction was authorised.	N/A	
Transaction Type F: n 2 T: '9C' L: 1 S: Reader	Required	Indicates the type of transaction, represented by the values of the first two digits of Processing Code as defined by the payment system.	N/A	

Name (Format; Tag; Length; Source; Path)	Requirement	Description	Retrieval	Values
Unpredictable Number (Card) F: b 32 T: part of '9F69' L: 4 S: Card	Conditional If fDDA supported	Contains the card unpredictable number generated by the card and signed for fDDA. The Unpredictable Number (Card) is generated during GPO processing and returned in the last record as part of the Card Authentication Related Data.	READ RECORD	
Unpredictable Number (Reader-Terminal) F: b 32 T: '9F37' L: 4 S: Reader	Required	Value to provide variability and uniqueness to the generation of the application cryptogram.	N/A	

A.3 Data Elements by Tag

Table A-2: Data Elements Tags

Tag	Data Element	Source
'42'	Issuer Identification Number (IIN)	Card
'4F'	Application Identifier (ADF Name)	Card
'50'	Application Label	Card
'57'	Track 2 Equivalent Data	Card
'5A'	Application Primary Account Number (PAN)	Card
'5F20'	Cardholder Name	Card
'5F24'	Application Expiration Date	Card
'5F28'	Issuer Country Code	Card
'5F2A'	Transaction Currency Code	Reader
'5F2D'	Language Preference	Card
'5F34'	Application Primary Account Number Sequence Number (PSN)	Card
'61'	Directory Entry	Card
'6F'	File Control Information (FCI) Template	Card
'71'	Issuer Script Template 1	Issuer
'72'	Issuer Script Template 2	Issuer
'77'	Response Message Template Format 2	Card
'82'	Application Interchange Profile (AIP)	Card
'83'	Command Template	Reader
'84'	Dedicated File (DF) Name	Card
'87'	Application Priority Indicator	Card
'88'	Short File Identifier	Card
'89'	Authorisation Code	Issuer
'8A'	Authorisation Response Code (ARC)	Issuer/Reader
'8F'	Certificate Authority Public Key Index (PKI)	Card
'90'	Issuer Public Key Certificate	Card
'91'	Issuer Authentication Data	Issuer
'92'	Issuer Public Key Remainder	Card

Tag	Data Element	Source
'94'	Application File Locator (AFL)	Card
'95'	Terminal Verification Results (TVR)	Reader
'9A'	Transaction Date	Reader
'9C'	Transaction Type	Reader
'9F02'	Amount, Authorised (Numeric)	Reader
'9F03'	Amount, Other (Numeric)	Reader
'9F06'	Application Identifier (AID)	Reader
'9F07'	Application Usage Control (AUC)	Card
'9F0C'	Issuer Identification Number Extended (IINE)	Card
'9F10'	Issuer Application Data (IAD)	Card
'9F11'	Issuer Code Table Index	Card
'9F12'	Application Preferred Name	Card
'9F18'	Issuer Script Identifier	Issuer
'9F1A'	Terminal Country Code	Reader
'9F1B'	Terminal Floor Limit	Reader
'9F22'	Certificate Authority Public Key Index (PKI)	Reader
'9F24'	Payment Account Reference (PAR)	Card
'9F26'	Application Cryptogram (AC)	Card
'9F27'	Cryptogram Information Data (CID)	Card
'9F32'	Issuer Public Key Exponent	Card
'9F36'	Application Transaction Counter (ATC)	Card
'9F37'	Unpredictable Number (Reader-Terminal)	Reader
'9F38'	Processing Options Data Object List (PDOL)	Card
'9F46'	Integrated Circuit Card (ICC) Public Key Certificate	Card
'9F47'	Integrated Circuit Card (ICC) Public Key Exponent	Card
'9F48'	Integrated Circuit Card (ICC) Public Key Remainder	Card
'9F4A'	Static Data Authentication (SDA) Tag List	Card
'9F4B'	Signed Dynamic Application Data (SDAD)	Card
'9F4D'	Log Entry	Card
'9F4E'	Merchant Name and Location	Reader
'9F5A'	Application Program Identifier (Program ID)	Card
'9F5B'	Issuer Script Results	Reader

Tag	Data Element	Source
'9F5D'	Available Offline Spending Amount (AOSA)	Card
'9F66'	Terminal Transaction Qualifiers (TTQ)	Reader
'9F69'	Card Authentication Related Data	Card
part of '9F69'	fDDA Version Number	Card
part of '9F69'	Unpredictable Number (Card)	Card
'9F6C'	Card Transaction Qualifiers (CTQ)	Card
'9F6E'	Form Factor Indicator (FFI)	Card
'9F7C'	Customer Exclusive Data (CED)	Card
'A5'	File Control Information (FCI) Proprietary Template	Card
'BF0C'	File Control Information (FCI) Issuer Discretionary Data	Card
‘BF60’	Integrated Data Storage Record Update Template	Data Exchange
‘D2’	Integrated Data Storage Directory (IDSD)	Card

A.4 Data Elements Reader Provides to Kernel 3

Table A-3: Data Elements Reader Provides to Kernel 3

Data Element	Tag
Amount, Authorised (Numeric)	'9F02'
Amount, Other (Numeric)	'9F03'
Certificate Authority Public Key	—
Certificate Authority Public Key Check Sum	—
Certificate Authority Public Key Exponent	—
Certificate Authority Public Key Index (PKI)	'9F22'
Certificate Authority Public Key Modulus	—
Merchant Name and Location	'9F4E'
Terminal Country Code	'9F1A'
Terminal Transaction Qualifiers (TTQ)	'9F66'
Transaction Currency Code	'5F2A'
Transaction Date	'9A'
Transaction Type	'9C'
Unpredictable Number (Reader-Terminal)	'9F37'

Annex B Online Messages and Clearing Records

POS systems will populate online messages and clearing records with the required data elements specified by the payment systems.

Of these data elements, this annex details the data elements specifically required for Kernel 3 contactless transactions. Some are data values to be set when Kernel 3 is the selected kernel and some are provided by Kernel 3 with the Outcome. The data elements are provided to the POS System in the Outcome data record described in *Book A*.

B.1 EMV Mode Acquirers

B.1.1 Data Elements for Acquirers Supporting EMV Mode

This section describes the data elements requirements for messages for acquirers that support EMV mode transactions.

Specific Kernel 3 EMV mode data element necessary for online messages and clearing records are as shown in Table B-1. The presence of a data element in this list – and thereby the data record – implies that it is the responsibility of the kernel to make the data element available for inclusion in the message to the acquirer. Depending on configuration, it may not actually be included in the message itself.

Table B-1: Acquirers Supporting EMV Mode

Data	Tag	Condition
Amount, Authorised	'9F02'	Always
Amount, Other	'9F03'	Conditional (if cashback)
Application Cryptogram (AC)	'9F26'	Always
Application Interchange Profile (AIP)	'82'	Always
Application Transaction Counter (ATC)	'9F36'	Always

Data	Tag	Condition
Application PAN Sequence Number	'5F34'	Conditional If the Application PAN Sequence Number (PSN) is provided to the kernel by the card, this data shall be included in the data record. Else (data not provided to the kernel by the card), the data is not included in the data record.
Customer Exclusive Data (CED)	'9F7C'	Conditional If the Customer Exclusive Data (CED) is provided to the kernel by the card, this data shall be included in the data record. Else (data not provided to the kernel by the card), the data is not included in the data record. See requirement 3.2.1.2.
Form Factor Indicator (FFI)	'9F6E'	Conditional If the Form Factor Indicator (FFI) is provided to the kernel by the card, this data shall be included in the data record. Else (data not provided to the kernel by the card), the data is not included in the data record. See requirement 3.2.1.2.
Issuer Application Data (IAD)	'9F10'	Always
POS Entry Mode	–	Always A value of 07 for EMV mode transactions.
Terminal Country Code	'9F1A'	Always

Data	Tag	Condition
Terminal Entry Capability	–	Always A value of 5 (for POS systems that also support contact chip) or a value of 8 (for POS systems that do not also support contact chip). Check with your payment system regional representative.
Terminal Verification Results (TVR)	'95'	Always For Kernel 3 transactions, the TVR shall have a value of all zeros when included in the data record.
Track 2 Equivalent Data	'57'	Always
Transaction Currency Code	'5F2A'	Always
Transaction Date	'9A'	Always
Transaction Type	'9C'	Always
Unpredictable Number (Reader-Terminal)	'9F37'	Always

B.1.2 EMV Mode Acquirer Requirements

The following additional requirements are applicable to acquirers for EMV mode messages.

Requirements – EMV Mode Acquirer Requirements	
B.1.2.1	(Track 2) *[K.3]* For EMV mode transactions, Track 2 shall be sent online in the authorisation.
B.1.2.2	(Primary Account Number and Expiration Date) *[K.4]* The primary account number and application expiration date used in messaging shall be retrieved from Track 2 Equivalent Data.

B.2 <This section has been deleted>

Annex C Fast Dynamic Data Authentication (fDDA)

In most contactless payment environments, quick transaction speeds are a business requirement. A method of dynamic data authentication, called fDDA (based on DDA as defined in [EMV 4.3]) is therefore defined for offline protection against skimming.

In addition to signing the (Terminal) Unpredictable Number, which is signed in most EMV contact chip applications, fDDA also signs additional transaction dynamic data. The Amount, Authorised; Transaction Currency Code; and (card) Unpredictable Number are all signed using fDDA.

To optimize processing power and reduce transaction times, the fDDA dynamic signature is generated during the GPO command, rather than generating the dynamic signature at the end of the transaction when the card may be moving away from the reader field.

The card uses the PDOL to request data from the reader for fDDA. The card receives the requested data from the reader in the GPO command. The card uses these terminal data elements, along with card data, to create the dynamic signature.

The AFL returned in the GPO points to records containing the RSA certificates and data related to fDDA. Once the last record is read by the reader, the card need no longer remain in the field. The reader then validates the dynamic signature for fDDA. If the validation process fails, the transaction is declined offline, sent online for authorisation, or terminated, dependent on issuer preference (as indicated in the CTQ).

In order to accommodate the possibility of new fDDA algorithms and inputs, the card data element fDDA Version Number (part of Tag '9F69') is defined to identify the fDDA version used by the card. The fDDA Version Number is returned by the card and used by the reader to determine the fDDA version algorithm to perform.

For readers compliant to this version of the specification, only fDDA version '01' is allowed.

For fDDA version '01', the card includes the (Terminal) Unpredictable Number; Amount, Authorised; and Transaction Currency Code received from the reader in the PDOL, combined with the card ATC and Card Authentication Related Data into the calculation of the dynamic signature.

Note: The Static Data Authentication Tag List (Tag '9F4A') is supported as defined in [EMV 4.3 Book 3], section 10.3 and [EMV 4.3 Book 2], section 6 for fDDA processing.

C.1 Dynamic Signature Verification

To verify the fDDA dynamic signature, the kernel shall first retrieve the Certification Authority Public Key Index, as specified in [EMV 4.3 Book 2], section 6.2.

Retrieval of the Issuer Public Key shall then be performed by the kernel in accordance with [EMV 4.3 Book 2], section 6.3.

Retrieval of the ICC Public Key shall be performed by the kernel in accordance with [EMV 4.3 Book 2], section 6.4.

Verification of the dynamic signature shall then be performed by the kernel in accordance with [EMV 4.3 Book 2], section 6.5.2, with the following exception:

- The Terminal Dynamic Data elements input to the hash algorithm shall be as specified in Table C-1 instead of being specified in the DDOL (as the DDOL is not a recognized data element for Kernel 3). The kernel may treat the tags specified in Table C-1 as default DDOLs for fDDA version '01'.

Table C-1: Terminal Dynamic Data for Input to DDA Hash Algorithm

Tag	Data Element	Length	Data Source
'9F37'	Unpredictable Number (UN)	4 bytes	Terminal
'9F02'	Amount, Authorised	6 bytes	Terminal
'5F2A'	Transaction Currency Code	2 bytes	Terminal
'9F69'	Card Authentication Related Data	var.	Card

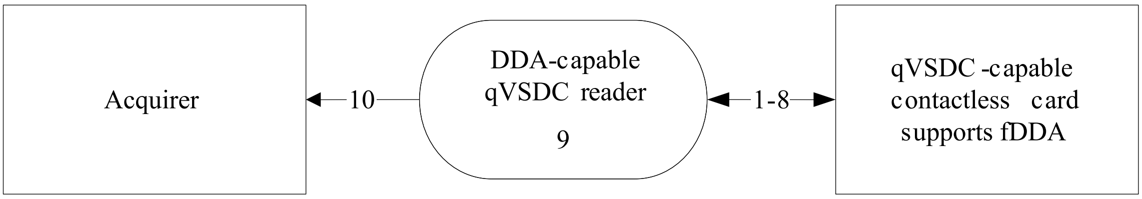
Note: The Card Authentication Related Data is variable length. The kernel shall use the full Card Authentication Related Data returned by the card for dynamic signature verification.

In any of the following cases, fDDA shall fail:

- The result of the verification of the dynamic signature process described above is not successful
- Application Interchange Profile (AIP) indicates that DDA is not supported by the card (AIP byte 1 bit 6 is 0b).
- fDDA is supported and data required to support fDDA is missing.
- The version of fDDA requested by the card is not supported by the reader. fDDA version 01 is the only supported version of fDDA in this specification.
- For offline approval requests, the dynamic signature is using a Signed Data Format that is not '05'.
- For online approval requests at special purpose readers, the dynamic signature is using a Signed Data Format that is not '95'.

fDDA is illustrated in Figure C-1.

Figure C-1: Fast DDA (fDDA) EMV Mode Example



1. Reader SELECTs PPSE.
2. Card responds with single AID.
3. Reader SELECTs AID.
4. Card responds requesting:
   - Terminal Transaction Qualifiers (Tag '9F66')
   - Unpredictable Number (Tag '9F37')
   - Amount, Authorised (Tag '9F02')
   - Transaction Currency Code (Tag '5F2A')
   - Other tags not related to fDDA
5. Reader issues GET PROCESSING OPTIONS providing:
   - Tag '9F66' indicating EMV mode only
   - Tag '9F37' Unpredictable Number
   - Tag '9F02' Amount, Authorised
   - Tag '5F2A' Transaction Currency Code
   - Other card requested data not related to fDDA
6. Card responds with:
   - Dynamic signature
   - AFL listing records related to Offline Data Authentication (fDDA)
   - Other data not related to fDDA
7. Reader reads records indicated in AFL.
8. Card provides RSA certificates and data to validate hash of static data along with Card Authentication Related Data in response to the last READ RECORD.

   Card may leave the field.
9. Reader validates dynamic signature.
10. If fDDA fails, the transaction is declined, switched to another interface, or sent online depending on issuer settings.

Annex D Supported Commands

D.1 GET PROCESSING OPTIONS (GPO)

The GET PROCESSING OPTIONS command initiates the transaction within the card application.

The GET PROCESSING OPTIONS command and response shall be supported as described in [EMV 4.3 Book 3] with the following exception(s):

- Data retrievable using the GET PROCESSING OPTIONS command is not completely as defined in [EMV 4.3 Book 3], section 6.5.8.4 for Format 2 responses. The AFL and AIP are not always included, and the data field may include additional BER-TLV coded data elements. Inclusion of the AFL and AIP is conditional on card application processing.
- Status words (SW1 SW2) may have additional values.

D.2 READ RECORD

The READ RECORD command reads a file record in a linear file.

The READ RECORD command and response shall be supported as described in [EMV 4.3 Book 3].

D.3 EXTERNAL AUTHENTICATE

The EXTERNAL AUTHENTICATE command performs Issuer Authentication (by verifying a cryptogram) and conditionally resets card counters and indicators.

The EXTERNAL AUTHENTICATE command and response shall be supported as described in [EMV 4.3 Book 3].

D.4 EXTENDED GET PROCESSING OPTIONS (EGPO)

The EXTENDED GET PROCESSING OPTIONS (EGPO) command is supported in kernels that support the implementation-optional Integrated Data Storage.

The EGPO command is used to initiate application processing, as for standard GPO, and additionally to update one or more IDS Records.

D.4.1 Command Message

The EGPO command message is coded as shown in Table D-1:

Table D-1: EXTENDED GET PROCESSING OPTIONS (EGPO) Command Message

Code	Value
CLA	'80'
INS	'E0'
P1	'00'
P2	'00'
Lc	Length of the command data field
Data	The data field consists of: • PDOL related data • IDS Record Update Template (TLV coded, tag 'BF60') – See F.2
Le	'00'

D.4.2 Data Field Sent in the Command Message

The command data field consists of:

- PDOL related data in a Command Template (tag ‘83’), as for a standard Kernel 3 GPO command (see section 5.2.1),
- Concatenated with the IDS Record Update Template (tag ‘BF60’).

Figure D-1: EGPO Command Message Data Field

PDOL related data			IDS Record Update Template		
'83'	Length	Value	'BF60'	Length	Value

The kernel is not required to validate the content of the IDS Record Update Template. It is the IDS Operator Application’s responsibility to ensure that the IDS Record Update Data is correctly formatted. The card application will respond with an error if invalid data is provided.

The IDS Record Update Template for an example Integrated Data Storage implementation is provided in section F.2.

D.4.3 Data Field Returned in the Response Message

The data field of the response message is as for the standard Kernel 3 GET PROCESSING OPTIONS (GPO) command as defined in Annex D.1.

Kernel 3 processes the EGPO response data in the same way as standard GPO response data (see requirements 5.2.1.2 and 5.2.1.3).

D.4.4 Processing State Returned in the Response Message

SW1 SW21 = ‘9000’ indicates a successful execution of the command.

Kernel 3 processes EGPO response SW1 SW2 in the same way as standard GPO responses (see requirement 7.2.2.2.

The card may return EGPO response status words that indicate errors related to processing of the IDS Record Update template. Implementations may choose to communicate these errors to the IDS Operator Application via Data Exchange, in order for proprietary IDS Operator processing to determine suitable actions. IDS Operator processing of EGPO error responses is out of scope of this specification.

Annex E <This appendix has been deleted>

Annex F Integrated Data Storage Implementation Overview

(Informative)

This section is informative. It provides a high level overview of an example Integrated Data Storage card application implementation, to assist kernel, terminal and IDS Operator Application developers.

Integrated Data Storage (IDS) provides a means of storing a service provider's data onto payment cards that are processed using kernel 3. An example of a service provider (referred to as an IDS Operator) is a transit system provider that uses IDS to store travel entitlement details on a payment card.

The on-card information primarily consists of the IDS Operator Data written to and returned in an associated record (referred to as an IDS Record), which is supported by data returned in the SELECT response and associated application internal data.

For card applications supporting IDS, IDS Operator terminals may both read from and write to the IDS component of the card application.

F.1 Integrated Data Storage Data Structure

Basic information about the Integrated Data Storage component of the card application is returned as part of the SELECT ADF response, in the IDS Directory data object. The IDS Directory provides information about the IDS component, including entries identifying each IDS Record on the card application. IDS Records are used by IDS Operators to store their proprietary data onto the card application.

Each IDS Record has an entry in the IDS Directory.

An IDS Record consists of two basic components (a publically readable IDS Record, and non-readable IDS Record Internal Data):

- IDS Directory Entry

  An entry in the IDS Directory for a given IDS Record. While the IDS Directory Entry is not strictly part of the IDS Record, it is included in this list as it identifies and provides information about the associated IDS Record. The IDS Directory Entry allows the IDS Operator to identify their IDS Record (or an empty IDS Record that they may use) upon selecting the card application, and to determine whether or not the IDS Record data should be read.

- IDS Record

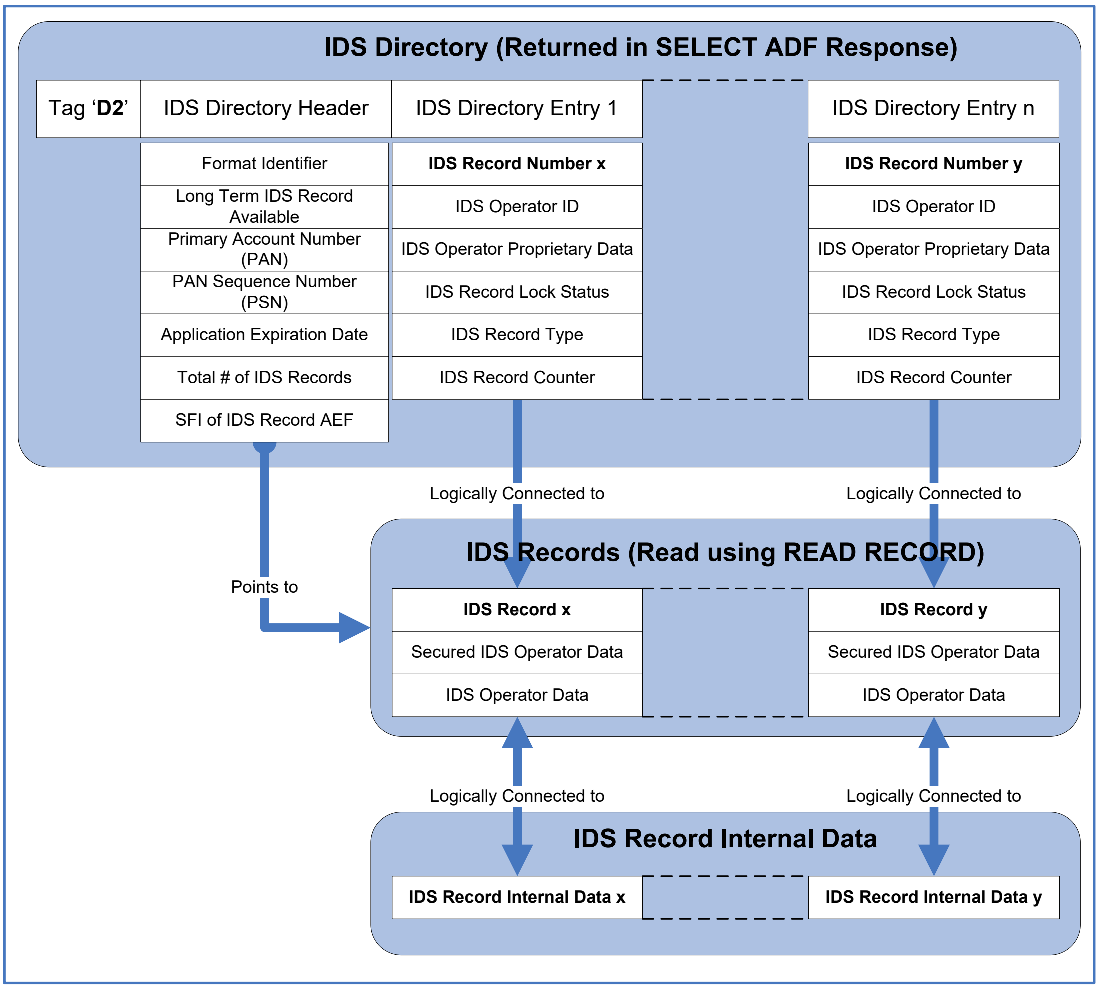
A record containing the IDS Operator’s data stored on the card application.

- IDS Record Internal Data

  Card application internal data associated with, and used to manage, the IDS Record.

Card applications supporting IDS are personalized with one or more IDS Records.

Figure F-1: IDS Data Structure Overview



F.1.1 IDS Directory

The IDS Directory (tag 'D2') is returned in the SELECT ADF response and consists of two components:

- IDS Directory Header

  Contains identifying information about the IDS component of the card application.

- IDS Directory Entries

There are one or more IDS Directory Entries in the IDS Directory. Each IDS Directory Entry is logically connected to an IDS Record, and contains information about the status of that associated IDS Record.

F.1.1.1 IDS Directory Header

The IDS Directory Header consists of the following data:

- Format Identifier

  Identifies the IDS Directory Header format.

- Long Term IDS Record Available

  Indicates whether a 'Long Term' IDS Record is available on the card application (for initialization and use by the terminal). See Section F.1.1.2 for a description of the different types of IDS Records.

- Application Primary Account Number (PAN)

  Identifies the Application Primary Account Number of the card application.

- Application PAN Sequence Number (PSN)

  Identifies the Application PAN Sequence Number of the card application.

- Application Expiration Date

  Identifies the Application Expiration Date (numeric YYMM format) of the card application.

- Total Number of IDS Records

  Indicates the total number of IDS Records supported by the card application.

- SFI of IDS Record AEF

  Identifies the file in which Public IDS Records are stored on the card application.

F.1.1.2 IDS Directory Entry

The IDS Directory Entries contain information about the status of the associated IDS Records. There are one or more IDS Directory Entries present in the IDS Directory, and each IDS Directory Entry is logically connected to an associated IDS Record.

Each IDS Directory Entry consists of the following data:

- IDS Record Number

  Record number within the SFI (identified in the IDS Directory Header) assigned for the associated IDS Record.

- IDS Operator ID

  A unique identifier for the IDS operator.

- IDS Operator Proprietary Data

  IDS Operator proprietary data that is provided in the SELECT ADF response.

- IDS Record Lock Status

  Indicates whether the IDS Record is 'Locked' or 'Unlocked'. The Locking mechanism is defined by the IDS card application.

- IDS Record Type
  - 'Inactive': IDS Record is available for use.
  - 'Short Term': IDS Record expires within 1 day of being written.
  - 'Long Term': IDS Record has an expiration that is greater than 1 day. The card application is personalized by the issuer to limit the number of 'Long Term' IDS Records permitted, to ensure that space is available for 'Short Term' IDS Records.
  - 'Secured': IDS Record is permanent and secured by the card issuer or IDS Operator. 'Secured' IDS Records are typically assigned at personalization time, and designed for use where a relationship exists between the card issuer and the IDS Operator.
  - IDS Record Counter

    A counter that is incremented by the card application when the IDS Operator Data in the associated IDS Record is changed.

F.1.2 IDS Record Data and IDS Record Internal Data

F.1.2.1 IDS Record Data

IDS Record Data is a record in the SFI of the IDS Record AEF that stores IDS Operator data.

The card application personalizer will allocate at least 160 bytes of space for each IDS Record.

The variable length IDS Record Public Data consists of the following data elements:

- Secured IDS Operator Data (Conditional)

  Secured data stored in the IDS Record and present only for IDS Records of type 'Secured'. Secured IDS Operator Data can be written during personalization and post-issuance using issuer scripts. The keys that protect the scripts for a particular secured IDS Record may be held either by the issuer, or by the IDS Operator (if the issuer has chosen to delegate to the IDS Operator the ability to perform post-issuance updates of their secured data).

- IDS Operator Data

  IDS Operator data stored on the card application. IDS Operator Data can be written during personalization, and post-issuance using the EGPO command (see Annex D.4) or issuer scripts.

F.1.2.2 IDS Record Internal Data

IDS Record Internal Data contains card application internal data used to manage an IDS Record. IDS Record Internal Data cannot be accessed externally from the card application.

F.2 Integrated Data Storage Record Update Template

The IDS Record Update Template (Tag ‘BF60) is used in EXTENDED GPO (EGPO) command data to update IDS Operator Data in an IDS Record. See Annex D.4 for details of the EGPO command.

Table F-1: IDS Record Update Template

Tag	Meaning			Len	Presence
'BF60'	IDS Record Update Template			var.	R
	'DFx0’	IDS Record Attributes		var.	R
			IDS Record Number Identifies the IDS Record Number to be updated (IDS Directory Entry byte 1).	1	R
			IDS Operator ID The value of this field for the identified IDS Record.	4	R
			IDS Record Type bits 8-3 = RFU (0,0,0,0,0,0) bits 2-1: (IDS Record Type) 00 = Inactive 01 = Short Term 10 = Long Term 11 = Secured The IDS Record Type for the identified IDS Record.	1	R
			IDS Record Expiry Date (numeric YYMMDD) The updated value of this field for the identified IDS Record. The IDS Record Expiry Date is present when the IDS Record Type (in this tag) is ‘Short Term’ or ‘Long Term’.	3	C
			Existing IDS Record Lock Data All zeros if there is no existing lock data.	8	R
			New IDS Record Lock Data All zeros if a new lock is not required.	8	R

Tag	Meaning		Len	Presence
	'DFx1'	IDS Record Offset Specifies the offset, measured from the beginning of the IDS Operator Data in the IDS Record (offset '00'), at which to begin writing the New IDS Record Data (Tag DFx2 below). The IDS Record Offset is assumed to be '00' when not present. For example: • An IDS Record Offset value of '00' indicates that the New IDS Record Data is to be written beginning at the 1st byte of the existing IDS Operator Data. • An IDS Record Offset value of '01' indicates that the New IDS Record Data is to be written beginning at the 2nd byte of the existing IDS Operator Data If the IDS Record Offset is not included in the EGPO command data, then it is assumed to be '00' by the card application. For a distinct value of 'x', multiple segments of the IDS Record may be updated. Consequently, there may be multiple instances of the IDS Record Offset with the exact same tag value. Each instance of the IDS Record Offset shall be followed by the corresponding New IDS Record Data (tag 'DFx2') to be used at that offset.	1	O
	'DFx2'	New IDS Record Data New IDS Record Data used to update the IDS Record. The New IDS Record Data may be used to replace all or part of the existing IDS Operator Data in the IDS Record. For a distinct value of 'x', multiple segments of the IDS Record may be updated. Consequently, there may be multiple instances of the New IDS Record Data with the exact same tag value.	var.	O
	'DFx3'	IDS Operator Data Length Length of the IDS Operator Data in the IDS Record after it is updated with the New IDS Record Data. There is a single occurrence of the IDS Operator Data Length for a distinct value of 'x', and the value is updated in the corresponding IDS Record Internal Data of the IDS Record identified. The IDS Operator Data Length is present when at least one instance of New IDS Record Data is present.	1	C

Tag	Meaning		Len	Presence
	'DFx4'	IDS Operator Proprietary Data The updated value of this field (in the IDS Directory Entry) for the identified IDS Record. There is a single occurrence of the IDS Operator Proprietary Data for a distinct value of 'x'.	5	O

One or more IDS Records may be updated in a single EGPO command; each IDS Record to be updated is identified by a different value ‘x’ in the tags 'DFx1', 'DFx2', 'DFx3', and 'DFx4'.

Note: The value ‘x’ in tags 'DFx1', 'DFx2', 'DFx3' and 'DFx4' may be any value between '0' and 'F'. The value of 'x' is not linked to the IDS Record Number; it is used only to distinguish between different sets of IDS Record Attributes and IDS Record data.

F.3 Integrated Data Storage Data Elements

This section summarised the data elements used in the example Integrated Data Storage implementation.

Table F-2: Integrated Data Storage Data Elements

Name	Description	Retrieval	Values
Format Identifier	Part of IDS Directory Header. Identifies the IDS Directory Header format.	SELECT	See IDS Directory Header
IDS Directory	IDS data returned in the SELECT response. The IDS Directory is identified by Tag 'D2' and contains: - IDS Directory Header - IDS Directory Entry (one or more)	SELECT	See IDS Directory Header and IDS Directory Entry
IDS Directory Entry	An entry in the IDS Directory that is logically connected to a corresponding IDS Record. The IDS Directory Entry contains IDS Record related data that is returned in the SELECT response. There are one or more IDS Directory Entries in the IDS Directory.	SELECT	Byte 1: IDS Record Number Bytes 2-5: IDS Operator ID Bytes 6-10: IDS Operator Proprietary Data Byte 11: bit 8: Lock Status 0 = Unlocked 1 = Locked bits 7-3: RFU (0,0,0,0,0) bits 2-1:IDS Record Type 00 = Inactive 01 = Short Term 10 = Long Term 11 = Secured Bytes 12-13: IDS Record Transaction Counter

Name	Description	Retrieval	Values
IDS Directory Header	Contains identifying information about the IDS component of the card application.	SELECT	Byte 1: bits 8-5: Format Identifier 0001 = Format ID 1 bits 4-2: RFU (0,0,0) bit 1: 1 = Long Term IDS Record Available Bytes 2-11: Primary Account Number (PAN) Byte 12: PAN Sequence Number Bytes 13-14: Application Expiration Date Byte 15: Total Number of IDS Records Byte 16: bits 8-4: SFI of IDS Record AEF bits 3-1: RFU (0,0,0)
IDS Operator Data	Optional IDS Operator data in an IDS Record.	READ RECORD	Data originates from IDS Operator and may have any value. Note that the total of Secured IDS Operator Data and IDS Operator Data within an IDS Record can not exceed 255 bytes.
IDS Operator ID	Identifier for the IDS Operator, assigned by the Payment System.	SELECT	See IDS Directory Entry
IDS Operator Proprietary Data	Free format field available for IDS Operators to store proprietary data that will be available in the SELECT response. For example, the IDS Operator Proprietary Data may be used to store the timestamp of the last update of the specific IDS Record.	SELECT	See IDS Directory Entry

Name	Description	Retrieval	Values
IDS Record Counter	A counter that is incremented by the card application when the IDS Operator Data in the associated IDS Record is changed.	SELECT	See IDS Directory Entry
IDS Record Lock Status	Indicates whether the IDS Record is ‘Locked’ or ‘Unlocked’.	SELECT	See IDS Directory Entry
IDS Record Number	Part of IDS Directory Entry. The Record Number within the SFI of IDS Record AEF, to which the IDS Directory Entry refers.	SELECT	See IDS Directory Entry
IDS Record Type	The type of IDS Record; Inactive, Short-term, Long Term or Secured.	SELECT	See IDS Directory Entry
IDS Record Update Template	Part of the command data for the Extended GET PROCESING OPTIONS command. The IDS Record Update Template contains data to be updated in one or more IDS Records.	N/A	See Table F-1.
Long Term IDS Record Available	Part of IDS Directory Header.	SELECT	See IDS Directory Header
SFI of IDS Record AEF	Part of IDS Directory Header.	SELECT	See IDS Directory Header
Secured IDS Operator Data	Optional secured data in an IDS Record. Only IDS Records with an IDS Record Type of ‘Secured’ will contain Secured IDS Operator Data.	READ RECORD	Data originates from IDS Operator and may have any value. Note that the total of Secured IDS Operator Data and IDS Operator Data within an IDS Record can not exceed 255 bytes.

Name	Description	Retrieval	Values
Total # of IDS Records	Part of IDS Directory Header.	SELECT	See IDS Directory Header

Annex G Glossary

This is a glossary of terms and abbreviations used in this specification. For descriptions of data elements, see Annex A.

AAC Application Authentication Cryptogram.

AC Application Cryptogram.

Acquirer A financial institution that signs a merchant (or disburses currency to a cardholder in a cash disbursement) and directly or indirectly enters the resulting transaction into interchange.

ADF Application Definition File.

AFL Application File Locator.

AID Application Identifier.

AIP Application Interchange Profile.

an Alphanumeric.

ans Alphanumeric Special, as defined in [EMV 4.3 Book 4], Annex B.

AOSA Available Offline Spending Amount.

Application Cryptogram Cryptogram returned by the card; one of the following cryptogram types:

- TC Transaction Certificate.
- ARQC Authorisation Request Cryptogram.
- AAC Application Authentication Cryptogram.

Approved A Final Outcome.

ARQC Authorisation Request Cryptogram.

ATC Application Transaction Counter.

b Binary or Bit string.

BER Basic Encoding Rules.

C Conditional.

CA Certification Authority.

Candidate List The list of Combinations constructed by Entry Point during the Combination Selection process.

Card As used in these specifications, a consumer device supporting contactless transactions.

Cardholder An individual to whom a card is issued or who is authorised to use that card.

Cardholder Verification Method (CVM) A method used to confirm the identity of a cardholder.

CED Customer Exclusive Data.

CID Cryptogram Information Data.

cn Compressed Numeric.

Combination Any of the following:

For:	**The combination of:**
a card	• an ADF Name. • a Kernel Identifier.
a reader	• an AID. • a Kernel ID.
the Candidate List for final selection	• an ADF Name. • a Kernel ID. • the Application Priority Indicator (if present). • the Extended Selection (if present).

Confirmation Code A code or password entered into a mobile device in order to confirm that a user wishes to perform a contactless mobile payment transaction.

Contactless card See “Card”.

CTQ Card Transaction Qualifiers.

CVM Cardholder Verification Method.

CVN17 A specific Kernel 3 implementation of an Application Cryptogram using a subset of EMV minimum data. Support of CVN17 is supported for both EMV mode transactions and for mag stripe mode transactions.

DDA Dynamic Data Authentication.

DDOL Dynamic Data Authentication Data Object List.

Declined A Final Outcome.

DF Dedicated File.

DOL Data Object List.

E Indicates EMV mode in the “Path” information in Table A-1.

EMV® A global standard for credit and debit payment cards based on chip card technology. The EMV Integrated Circuit Card Specifications for Payment Systems are developed and maintained by EMVCo.

EMV mode An operating mode of the POS System that indicates that this particular acceptance environment and acceptance rules supports chip infrastructure. Typically used in conjunction with the term “transaction” (i.e., EMV mode transaction) to indicate contactless payment utilising a full chip infrastructure carrying EMV minimum data.

EMV Mode Path For transactions conducted over the contactless interface, the EMV Mode Path is an application path taken by the card which results in card behaviour defined for EMV mode. This path is taken for contactless transactions where the card and reader both support EMV mode.

EMV mode reader A reader in an acceptance environment that can handle EMV data for which at least one kernel is configured to accept EMV data.

EMVCo EMVCo LLC is the organisation of payment systems that manages, maintains, and enhances the EMV specifications. EMVCo is currently operated by American Express, JCB, MasterCard, and Visa.

End Application A Final Outcome.

Extended Selection An option in which Entry Point appends the value indicated by the Extended Selection data element (Tag '9F29') to the ADF name in the SELECT command.

F Format.

Fast DDA (fDDA) Leverages DDA as defined in [EMV 4.3] specifications. Used in EMV mode transactions to allow the reader to issue READ RECORD commands to obtain Dynamic Data Authentication (DDA) related data from the card and perform the DDA calculations after the card has left the field.

FCI File Control Information.

fDDA Fast DDA.

FFI Form Factor Indicator.

Final Outcome Result provided to the reader as a result of Entry Point processing the Outcome from the kernel, or provided directly by Entry Point under exception conditions.

GPO GET PROCESSING OPTIONS command.

Hex Hexadecimal.

ICC Integrated Circuit Card.

IEC International Electrotechnical Commission.

IDS Integrated Data Storage.

IDS Operator A service provider that utilises the optional Integrated Data Storage feature.

IDS Operator Application An application that implements the business logic of the IDS Operator.

ISO

International Organization for Standardization.

Issuer

A financial institution that issues contactless cards or contactless payment applications that reside in consumer devices.

Issuer Update Data

Data returned by the issuer in the response to an online request. Can include Issuer Response Data, Issuer Authentication Data, and/or Issuer Scripts Template.

Issuer Update Processing

Optional feature whereby the issuer can update card parameters via a second presentment of a card.

Kernel

The kernel contains interface routines, security and control functions, and logic to manage a set of commands and responses to retrieve the necessary data from a card to complete a transaction. The kernel processing covers the interaction with the card between the Final Combination Selection (excluded) and the Outcome Processing (excluded).

Kernel ID

Identifier to distinguish between different kernels that may be supported by the reader.

Kernel Identifier

Identifier to distinguish between different kernels that may be indicated by the card.

L

Length.

M

Mandatory.

Mag-stripe mode

An operating mode of the POS System that indicates that this particular acceptance environment and acceptance rules supports mag-stripe infrastructure. Typically used in conjunction with the term “transaction” (i.e., mag-stripe mode transaction) to indicate contactless payment based on Track 1 and/or Track 2 Data obtained from the card.

Important: Mag-stripe mode is no longer supported in this specification.

ms

Millisecond.

n

Numeric.

N/A

Not applicable; a possible value for several Outcome and Final Outcome parameters.

N_{CA}	Length of CA Public Key Modulus.
NFC	Near Field Communication.
N_{I}	Length of the Issuer Public Key Modulus.
N_{IC}	Length of the ICC Public Key Modulus.
O	Optional.
ODA	Offline Data Authentication.
Online PIN	A method of PIN verification where the PIN entered by the cardholder into the terminal PIN pad is encrypted and included in the online authorisation request message sent to the issuer.
Online Request	A Final Outcome.
Outcome	Result from the kernel processing, provided to Entry Point, or under exception conditions, result of Entry Point processing. In either case, a primary value with a parameter set.
P	Path.
PAN	Primary Account Number.
path	An application path taken based on reader/terminal interface and capabilities. EMV mode is the only contactless path supported in this version of the specification.
PDOL	Processing Options Data Object List.
PIN	Personal Identification Number.
PIX	Proprietary Identifier Extension.
POS	Point of Sale.
PPSE	Proximity Payment System Environment.
Proximity Payment System Environment (PPSE)	A list of all Combinations supported by the contactless card. PPSE is used in the Entry Point Combination Selection process.
Reader	A component of the POS System; described in detail in *Book A.*

Reader risk parameter	A reader limit or check used to perform reader risk management during the Pre-Processing phase of Entry Point.
RF	Radio Frequency.
RFU	Reserved for Future Use (by EMVCo).
RID	Registered Application Provider Identifier.
S	Source.
Select Next	An Outcome.
SFI	Short File Identifier.
SHA	Secure Hash Algorithm.
Status Check Support	Option within the reader related to the checking of a single unit of currency. The notion of single unit of currency is based on the least significant number of the currency unit as defined in ISO 4217. This option can be used by the reader to determine whether the card is genuine and active.
SW1 SW2	Status Byte One, Status Byte Two.
T	Tag.
TC	Transaction Certificate.
Terminal	A component of the POS System; described in detail in *Book A.*
TLV	Tag Length Value.
Transaction	The reader-card interaction between the first presentment of the card and the decision on whether the transaction is approved or declined. If the transaction is authorised online, this may involve multiple presentments of the card on the reader.
Try Again	An Outcome.
Try Another Interface	A Final Outcome.
TTQ	Terminal Transaction Qualifiers.

TVR	Terminal Verification Results.
var	Variable length.

*** END OF DOCUMENT ***