EMV
Integrated Circuit Card Specifications for Payment Systems

Book 1

Application Independent ICC to Terminal Interface Requirements

Version 4.2
June 2008

EMV
Integrated Circuit Card Specifications for Payment Systems

Book 1

Application Independent ICC to Terminal Interface Requirements

Version 4.2
June 2008

Revision Log - Version 4.2

The following changes have been made to Book 1 since the publication of Version 4.1. Numbering and cross references in this version have been updated to reflect changes introduced by the published bulletins.

Incorporated changes described in the following General bulletins:

General Bulletin no. 11 Second Edition: Lower Voltage Card Migration Mandatory Implementation Schedule

Incorporated changes described in the following Specification Updates:

Specification Update Bulletin no. 42: Voice Referrals

Specification Update Bulletin no. 43: Correction to the Coding of TA2

Specification Update Bulletin no. 49: Data Errors during List of AIDs Selection

Updated in support of the following Application Notes:

Application Note no. 26: Warm Reset Timing

Application Note no.38: Recommendations Regarding the Use of Historical Bytes

Minor editorial clarifications, including those described in the following Specification Updates:

Specification Update Bulletin no. 37 Fourth Edition: Editorial Errors in Release 4.1 of the EMV Specifications

Contents

Part IV - Annexes

Part V - Common Core Definitions

Tables

Figures

Part I

General

1 Scope

This document, the *Integrated Circuit Card (ICC) Specifications for Payment Systems - Book 1, Application Independent ICC to Terminal Interface Requirements*, describes the minimum functionality required of integrated circuit cards (ICCs) and terminals to ensure correct operation and interoperability independent of the application to be used. Additional proprietary functionality and features may be provided, but these are beyond the scope of this specification and interoperability cannot be guaranteed.

The *Integrated Circuit Card Specifications for Payment Systems* includes the following additional documents, all available on http://www.emvco.com:

- Book 2 - Security and Key Management
- Book 3 - Application Specification
- Book 4 - Cardholder, Attendant, and Acquirer Interface Requirements

1.1 Changes in Version 4.2

This release incorporates all relevant Specification Update Bulletins, Application Notes, amendments, etc. published up to the date of this release.

The Revision Log at the beginning of the Book provides additional detail about changes to this Book.

1.2 Structure

Book 1 consists of the following parts:

Part I	-	**General**
Part II	-	**Electromechanical Characteristics, Logical Interface, and Transmission Protocols**
Part III	-	**Files, Commands, and Application Selection**
Part IV	-	**Annexes**
Part V	-	**Common Core Definitions**

Part I includes this introduction, as well as data applicable to all Books: normative references, definitions, abbreviations, notations, data element format convention, and terminology.

Part II defines electromechanical characteristics, logical interface, and transmission protocols as they apply to the exchange of information between an ICC and a terminal. In particular it covers:

- Mechanical characteristics, voltage levels, and signal parameters as they apply to both ICCs and terminals.
- An overview of the card session.
- Establishment of communication between the ICC and the terminal by means of the answer to reset.
- Character- and block-oriented asynchronous transmission protocols.

Part III defines data elements, files, and commands as they apply to the exchange of information between an ICC and a terminal. In particular it covers:

- Data elements and their mapping onto data objects.
- Structure and referencing of files.
- Structure and coding of messages between the ICC and the terminal to achieve application selection.

Part III also defines the application selection process from the standpoint of both the card and the terminal. The logical structure of data and files within the card that is required for the process is specified, as is the terminal logic using the card structure.

Part IV includes examples of exchanges using T=0, a data elements table specific to application selection, and example directory structures.

Part V defines an optional extension to be used when implementing the Common Core Definitions (CCD).

The Book also includes a revision log and an index.

1.3 Underlying Standards

This specification is based on the ISO/IEC 7816 series of standards and should be read in conjunction with those standards. However, if any of the provisions or definitions in this specification differ from those standards, the provisions herein shall take precedence.

1.4 Audience

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

2 Normative References

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

FIPS 180-2	Secure Hash Standard
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/changes to ISO 639-1:1988: Codes for the Representation of Names of Languages 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/IEC 7811-1	Identification cards – Recording technique – Part 1: Embossing
ISO/IEC 7811-3	Identification cards – Recording technique – Part 3: Location of embossed characters on ID-1 cards
ISO/IEC 7813	Identification cards – Financial transaction cards
ISO/IEC 7816-1	Identification cards – Integrated circuit(s) cards with contacts – Part 1: Physical characteristics
ISO/IEC 7816-2	Information technology – Identification cards – Integrated circuit(s) cards with contacts – Part 2: Dimensions and location of contacts
ISO/IEC 7816-3	Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols

ISO/IEC 7816-4	Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange
ISO/IEC 7816-5	Identification cards — Integrated circuit cards — Part 5: Registration of application providers
ISO/IEC 7816-6	Identification cards – Integrated circuit cards – Part 6: Interindustry data elements for interchange
ISO 8583:1987	Bank card originated messages – Interchange message specifications – Content for financial transactions
ISO 8583:1993	Financial transaction card originated messages – Interchange message specifications
ISO/IEC 8825-1	Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
ISO/IEC 8859	Information processing – 8-bit single-byte coded graphic character sets
ISO 9362	Banking – Banking telecommunication messages – Bank identifier codes
ISO 9564-1	Banking – PIN management and security – Part 1: Basic principles and requirements for online PIN handling in ATM and POS systems
ISO 9564-3	Banking – PIN management and security – Part 3: Requirements for offline PIN handling in ATM and POS systems
ISO/IEC 9796-2:2002	Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms
ISO/IEC 9797-1	Information technology – Security techniques – Message Authentication Codes – Part 1: Mechanisms using a block cipher

ISO/IEC 10116	Information technology – Security techniques – Modes of operation for an n-bit block cipher
ISO/IEC 10118-3	Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions
ISO/IEC 10373	Identification cards – Test methods
ISO 13491-1	Banking – Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods
ISO 13616	Banking and related financial services – International bank account number (IBAN)
ISO 16609	Banking – Requirements for message authentication using symmetric techniques
ISO/IEC 18031	Information technology - Security techniques - Random bit generation
ISO/IEC 18033-3	Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers

3 Definitions

The following terms are used in one or more books of these specifications.

Accelerated Revocation	A key revocation performed on a date sooner than the published key expiry date.
Application	The application protocol between the card and the terminal and its related set of data.
Application Authentication Cryptogram	An Application Cryptogram generated by the card when declining a transaction
Application Cryptogram	A cryptogram generated by the card in response to a GENERATE AC command. See also: • Application Authentication Cryptogram • Authorisation Request Cryptogram • Transaction Certificate
Authorisation Request Cryptogram	An Application Cryptogram generated by the card when requesting online authorisation
Authorisation Response Cryptogram	A cryptogram generated by the issuer in response to an Authorisation Request Cryptogram.
Asymmetric Cryptographic Technique	A cryptographic technique that uses two related transformations, a public transformation (defined by the public key) and a private transformation (defined by the private key). The two transformations have the property that, given the public transformation, it is computationally infeasible to derive the private transformation.
Authentication	The provision of assurance of the claimed identity of an entity or of data origin.
Block	A succession of characters comprising two or three fields defined as prologue field, information field, and epilogue field.

Byte	8 bits.
Card	A payment card as defined by a payment system.
Certificate	The public key and identity of an entity together with some other information, rendered unforgeable by signing with the private key of the certification authority which issued that certificate.
Certification Authority	Trusted third party that establishes a proof that links a public key and other relevant information to its owner.
Ciphertext	Enciphered information.
Cold Reset	The reset of the ICC that occurs when the supply voltage (VCC) and other signals to the ICC are raised from the inactive state and the reset (RST) signal is applied.
Combined DDA/Application Cryptogram Generation	A form of offline dynamic data authentication.
Command	A message sent by the terminal to the ICC that initiates an action and solicits a response from the ICC.
Compromise	The breaching of secrecy or security.
Concatenation	Two elements are concatenated by appending the bytes from the second element to the end of the first. Bytes from each element are represented in the resulting string in the same sequence in which they were presented to the terminal by the ICC, that is, most significant byte first. Within each byte bits are ordered from most significant bit to least significant. A list of elements or objects may be concatenated by concatenating the first pair to form a new element, using that as the first element to concatenate with the next in the list, and so on.
Contact	A conducting element ensuring galvanic continuity between integrated circuit(s) and external interfacing equipment.

Cryptogram	Result of a cryptographic operation.
Cryptographic Algorithm	An algorithm that transforms data in order to hide or reveal its information content.
Data Integrity	The property that data has not been altered or destroyed in an unauthorised manner.
Deactivation Sequence	The deactivation sequence defined in section 6.1.5.
Decipherment	The reversal of a corresponding encipherment.
Digital Signature	An asymmetric cryptographic transformation of data that allows the recipient of the data to prove the origin and integrity of the data, and protect the sender and the recipient of the data against forgery by third parties, and the sender against forgery by the recipient.
Dynamic Data Authentication	A form of offline dynamic data authentication
Embossing	Characters raised in relief from the front surface of a card.
Encipherment	The reversible transformation of data by a cryptographic algorithm to produce ciphertext.
Epilogue Field	The final field of a block. It contains the error detection code (EDC) byte(s).
Exclusive-OR	Binary addition with no carry, giving the following values: 0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 1 + 1 = 0
Financial Transaction	The act between a cardholder and a merchant or acquirer that results in the exchange of goods or services against payment.
Function	A process accomplished by one or more commands and resultant actions that are used to perform all or part of a transaction.

Guardtime — The minimum time between the trailing edge of the parity bit of a character and the leading edge of the start bit of the following character sent in the same direction.

Hash Function — A function that maps strings of bits to fixed–length strings of bits, satisfying the following two properties:

- It is computationally infeasible to find for a given output an input which maps to this output.
- It is computationally infeasible to find for a given input a second input that maps to the same output.

Additionally, if the hash function is required to be collision–resistant, it must also satisfy the following property:

- It is computationally infeasible to find any two distinct inputs that map to the same output.

Hash Result — The string of bits that is the output of a hash function.

Inactive — The supply voltage (VCC) and other signals to the ICC are in the inactive state when they are at a potential of 0.4 V or less with respect to ground (GND).

Integrated Circuit Module — The sub-assembly embedded into the ICC comprising the IC, the IC carrier, bonding wires, and contacts.

Integrated Circuit(s) — Electronic component(s) designed to perform processing and/or memory functions.

Integrated Circuit(s) Card — A card into which one or more integrated circuits are inserted to perform processing and memory functions.

Interface Device — That part of a terminal into which the ICC is inserted, including such mechanical and electrical devices as may be considered part of it.

Issuer Action Code — Any of the following, which reflect the issuer-selected action to be taken upon analysis of the TVR:

- Issuer Action Code - Default
- Issuer Action Code - Denial
- Issuer Action Code - Online

Kernel	The set of functions required to be present on every terminal implementing a specific interpreter. The kernel contains device drivers, interface routines, security and control functions, and the software for translating from the virtual machine language to the language used by the real machine. In other words, the kernel is the implementation of the virtual machine on a specific real machine.
Key	A sequence of symbols that controls the operation of a cryptographic transformation.
Key Expiry Date	The date after which a signature made with a particular key is no longer valid. Issuer certificates signed by the key must expire on or before this date. Keys may be removed from terminals after this date has passed.
Key Introduction	The process of generating, distributing, and beginning use of a key pair.
Key Life Cycle	All phases of key management, from planning and generation, through revocation, destruction, and archiving.
Key Replacement	The simultaneous revocation of a key and introduction of a key to replaced the revoked one.
Key Revocation	The key management process of withdrawing a key from service and dealing with the legacy of its use. Key revocation can be as scheduled or accelerated.
Key Revocation Date	The date after which no legitimate cards still in use should contain certificates signed by this key, and therefore the date after which this key can be deleted from terminals. For a planned revocation the Key Revocation Date is the same as the key expiry date.
Key Withdrawal	The process of removing a key from service as part of its revocation.
Keypad	Arrangement of numeric, command, and, where required, function and/or alphanumeric keys laid out in a specific manner.

Library	A set of high-level software functions with a published interface, providing general support for terminal programs and/or applications.
Logical Compromise	The compromise of a key through application of improved cryptanalytic techniques, increases in computing power, or combination of the two.
Magnetic Stripe	The stripe containing magnetically encoded information.
Message	A string of bytes sent by the terminal to the card or vice versa, excluding transmission-control characters.
Message Authentication Code	A symmetric cryptographic transformation of data that protects the sender and the recipient of the data against forgery by third parties.
Nibble	The four most significant or least significant bits of a byte.
Padding	Appending extra bits to either side of a data string.
Path	Concatenation of file identifiers without delimitation.
Payment System Environment	The set of logical conditions established within the ICC when a payment system application conforming to this specification has been selected, or when a Directory Definition File (DDF) used for payment system application purposes has been selected.
Physical Compromise	The compromise of a key resulting from the fact that it has not been securely guarded, or a hardware security module has been stolen or accessed by unauthorised persons.
PIN Pad	Arrangement of numeric and command keys to be used for personal identification number (PIN) entry.
Plaintext	Unenciphered information.
Planned Revocation	A key revocation performed as scheduled by the published key expiry date.

Potential Compromise	A condition where cryptanalytic techniques and/or computing power has advanced to the point that compromise of a key of a certain length is feasible or even likely.
Private Key	That key of an entity's asymmetric key pair that should only be used by that entity. In the case of a digital signature scheme, the private key defines the signature function.
Prologue Field	The first field of a block. It contains subfields for node address (NAD), protocol control byte (PCB), and length (LEN).
Public Key	That key of an entity's asymmetric key pair that can be made public. In the case of a digital signature scheme, the public key defines the verification function.
Public Key Certificate	The public key information of an entity signed by the certification authority and thereby rendered unforgeable.
Response	A message returned by the ICC to the terminal after the processing of a command message received by the ICC.
Script	A command or a string of commands transmitted by the issuer to the terminal for the purpose of being sent serially to the ICC as commands.
Secret Key	A key used with symmetric cryptographic techniques and usable only by a set of specified entities.
Signal Amplitude	The difference between the high and low voltages of a signal.
Signal Perturbations	Abnormalities occurring on a signal during normal operation such as undershoot/overshoot, electrical noise, ripple, spikes, crosstalk, etc. Random perturbations introduced from external sources are beyond the scope of this specification.
Socket	An execution vector defined at a particular point in an application and assigned a unique number for reference.

State H	Voltage high on a signal line. May indicate a logic one or logic zero depending on the logic convention used with the ICC.
State L	Voltage low on a signal line. May indicate a logic one or logic zero depending on the logic convention used with the ICC.
Static Data Authentication	Offline static data authentication
Symmetric Cryptographic Technique	A cryptographic technique that uses the same secret key for both the originator's and recipient's transformation. Without knowledge of the secret key, it is computationally infeasible to compute either the originator's or the recipient's transformation.
T=0	Character-oriented asynchronous half duplex transmission protocol.
T=1	Block-oriented asynchronous half duplex transmission protocol.
Template	Value field of a constructed data object, defined to give a logical grouping of data objects.
Terminal	The device used in conjunction with the ICC at the point of transaction to perform a financial transaction. The terminal incorporates the interface device and may also include other components and interfaces such as host communications.
Terminal Action Code	Any of the following, which reflect the acquirer-selected action to be taken upon analysis of the TVR: • Terminal Action Code - Default • Terminal Action Code - Denial • Terminal Action Code - Online
Terminate Card Session	End the card session by deactivating the IFD contacts according to section 6.1.5, and displaying a message indicating that the ICC cannot be used to complete the transaction
Terminate Transaction	Stop the current application and deactivate the card.

Transaction	An action taken by a terminal at the user's request. For a POS terminal, a transaction might be payment for goods, etc. A transaction selects among one or more applications as part of its processing flow.
Transaction Certificate	An Application Cryptogram generated by the card when accepting a transaction
Virtual Machine	A theoretical microprocessor architecture that forms the basis for writing application programs in a specific interpreter software implementation.
Warm Reset	The reset that occurs when the reset (RST) signal is applied to the ICC while the clock (CLK) and supply voltage (VCC) lines are maintained in their active state.

4 Abbreviations, Notations, Conventions, and Terminology

4.1 Abbreviations

μA	Microampere
μm	Micrometre
μs	Microsecond
a	Alphabetic (see section 4.3, Data Element Format Conventions)
AAC	Application Authentication Cryptogram
AC	Application Cryptogram
ACK	Acknowledgment
ADF	Application Definition File
AEF	Application Elementary File
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
an	Alphanumeric (see section 4.3)
ans	Alphanumeric Special (see section 4.3)
APDU	Application Protocol Data Unit
API	Application Program Interface
ARC	Authorisation Response Code
ARPC	Authorisation Response Cryptogram
ARQC	Authorisation Request Cryptogram
ASI	Application Selection Indicator
ASN	Abstract Syntax Notation
ATC	Application Transaction Counter

ATM	Automated Teller Machine
ATR	Answer to Reset
AUC	Application Usage Control
b	Binary (see section 4.3)
BCD	Binary Coded Decimal
BER	Basic Encoding Rules (defined in ISO/IEC 8825–1)
BIC	Bank Identifier Code
BGT	Block Guardtime
BWI	Block Waiting Time Integer
BWT	Block Waiting Time
C	Celsius or Centigrade
CAD	Card Accepting Device
C-APDU	Command APDU
CBC	Cipher Block Chaining
CCD	Common Core Definitions
CCI	Common Core Identifier
CDA	Combined DDA/Application Cryptogram Generation
CDOL	Card Risk Management Data Object List
CID	Cryptogram Information Data
C_{IN}	Input Capacitance
CLA	Class Byte of the Command Message
CLK	Clock
cn	Compressed Numeric (see section 4.3)
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSU	Card Status Update
C-TPDU	Command TPDU
CV	Cryptogram Version

CVM	Cardholder Verification Method
CVR	Card Verification Results
CV Rule	Cardholder Verification Rule
CWI	Character Waiting Time Integer
CWT	Character Waiting Time
D	Bit Rate Adjustment Factor
DAD	Destination Node Address
DC	Direct Current
DDA	Dynamic Data Authentication
DDF	Directory Definition File
DDOL	Dynamic Data Authentication Data Object List
DES	Data Encryption Standard
DF	Dedicated File
DIR	Directory
DOL	Data Object List
ECB	Electronic Code Book
EDC	Error Detection Code
EF	Elementary File
EN	European Norm
etu	Elementary Time Unit
f	Frequency
FC	Format Code
FCI	File Control Information
GND	Ground
GP	Grandparent key for session key generation
Hex	Hexadecimal
HHMMSS	Hours, Minutes, Seconds
I/O	Input/Output

IAC	Issuer Action Code (Denial, Default, Online)
IAD	Issuer Application Data
IBAN	International Bank Account Number
I-block	Information Block
IC	Integrated Circuit
ICC	Integrated Circuit(s) Card
I_{CC}	Current drawn from VCC
IEC	International Electrotechnical Commission
IFD	Interface Device
IFS	Information Field Size
IFSC	Information Field Size for the ICC
IFSD	Information Field Size for the Terminal
IFSI	Information Field Size Integer
IIN	Issuer Identification Number
IK	Intermediate Key for session key generation
INF	Information Field
INS	Instruction Byte of Command Message
I_{OH}	High Level Output Current
I_{OL}	Low Level Output Current
ISO	International Organization for Standardization
IV	Initial Vector for session key generation
K_M	Master Key
K_S	Session Key
L	Length
l.s.	Least Significant
Lc	Exact Length of Data Sent by the TAL in a Case 3 or 4 Command
LCOL	Lower Consecutive Offline Limit

L_{DD}	Length of the ICC Dynamic Data
Le	Maximum Length of Data Expected by the TAL in Response to a Case 2 or 4 Command
LEN	Length
Licc	Exact Length of Data Available or Remaining in the ICC (as Determined by the ICC) to be Returned in Response to the Case 2 or 4 Command Received by the ICC
Lr	Length of Response Data Field
LRC	Longitudinal Redundancy Check
M	Mandatory
mΩ	Milliohm
MΩ	Megohm
m.s.	Most Significant
m/s	Meters per Second
mA	Milliampere
MAC	Message Authentication Code
max.	Maximum
MF	Master File
MHz	Megahertz
min.	Minimum
MK	ICC Master Key for session key generation
mm	Millimetre
MMDD	Month, Day
MMYY	Month, Year
N	Newton
n	Numeric (see section 4.3)
NAD	Node Address
NAK	Negative Acknowledgment
nAs	Nanoampere-second

N_{CA}	Length of the Certification Authority Public Key Modulus
NF	Norme Française
N_I	Length of the Issuer Public Key Modulus
N_{IC}	Length of the ICC Public Key Modulus
NIST	National Institute for Standards and Technology
N_{PE}	Length of the ICC PIN Encipherment Public Key Modulus
ns	Nanosecond
O	Optional
O/S	Operating System
P	Parent key for session key generation
P1	Parameter 1
P2	Parameter 2
P3	Parameter 3
PAN	Primary Account Number
PC	Personal Computer
P_{CA}	Certification Authority Public Key
PCB	Protocol Control Byte
PDOL	Processing Options Data Object List
pF	Picofarad
P_I	Issuer Public Key
P_{IC}	ICC Public Key
PIN	Personal Identification Number
PIX	Proprietary Application Identifier Extension
POS	Point of Service
pos.	Position
PSE	Payment System Environment
PTS	Protocol Type Selection

R-APDU	Response APDU
R-block	Receive Ready Block
RFU	Reserved for Future Use
RID	Registered Application Provider Identifier
RSA	Rivest, Shamir, Adleman Algorithm
RST	Reset
SAD	Source Node Address
S-block	Supervisory Block
S_{CA}	Certification Authority Private Key
SDA	Static Data Authentication
SFI	Short File Identifier
SHA-1	Secure Hash Algorithm 1
S_I	Issuer Private Key
S_{IC}	ICC Private Key
SK	Session Key for session key generation
SW1	Status Byte One
SW2	Status Byte Two
TAC	Terminal Action Code(s) (Default, Denial, Online)
TAL	Terminal Application Layer
TC	Transaction Certificate
TCK	Check Character
TDOL	Transaction Certificate Data Object List
t_F	Fall Time Between 90% and 10% of Signal Amplitude
TLV	Tag Length Value
TPDU	Transport Protocol Data Unit
t_R	Rise Time Between 10% and 90% of Signal Amplitude
TS	Initial Character

TSI	Transaction Status Information
TTL	Terminal Transport Layer
TVR	Terminal Verification Results
UCOL	Upper Consecutive Offline Limit
UL	Underwriters Laboratories Incorporated
V	Volt
var.	Variable (see section 4.3)
V_{CC}	Voltage Measured on VCC Contact
VCC	Supply Voltage
V_{IH}	High Level Input Voltage
V_{IL}	Low Level Input Voltage
V_{OH}	High Level Output Voltage
V_{OL}	Low Level Output Voltage
VPP	Programming Voltage
V_{PP}	Voltage Measured on VPP contact
WI	Waiting Time Integer
WTX	Waiting Time Extension
WWT	Work Waiting Time
YYMM	Year, Month
YYMMDD	Year, Month, Day

4.2 Notations

'0' to '9' and 'A' to 'F'	16 hexadecimal characters
xx	Any value
A := B	A is assigned the value of B
A = B	Value of A is equal to the value of B
$A \equiv B \bmod n$	Integers A and B are congruent modulo the integer n, that is, there exists an integer d such that $(A – B) = dn$
A mod n	The reduction of the integer A modulo the integer n, that is, the unique integer r, $0 \le r < n$, for which there exists an integer d such that $A = dn + r$
A / n	The integer division of A by n, that is, the unique integer d for which there exists an integer r, $0 \le r < n$, such that $A = dn + r$
Y := ALG(K)[X]	Encipherment of a data block X with a block cipher as specified in Annex A1 of Book 2, using a secret key K
$X = ALG^{-1}(K)[Y]$	Decipherment of a data block Y with a block cipher as specified in Annex A1 of Book 2, using a secret key K
Y := Sign (S_K)[X]	The signing of a data block X with an asymmetric reversible algorithm as specified in Annex A2 of Book 2, using the private key S_K
X = Recover(P_K)[Y]	The recovery of the data block X with an asymmetric reversible algorithm as specified in Annex A2 of Book 2, using the public key P_K
C := (A || B)	The concatenation of an n-bit number A and an m-bit number B, which is defined as $C = 2^m A + B$.

Leftmost	Applies to a sequence of bits, bytes, or digits and used interchangeably with the term “most significant”. If C = (A || B) as above, then A is the leftmost n bits of C.
Rightmost	Applies to a sequence of bits, bytes, or digits and used interchangeably with the term “least significant”. If C = (A || B) as above, then B is the rightmost m bits of C.
H := Hash[MSG]	Hashing of a message MSG of arbitrary length using a 160-bit hash function
$X \oplus Y$	The symbol '$\oplus$' denotes bit-wise exclusive-OR and is defined as follows:

$X \oplus Y$	The bit-wise exclusive-OR of the data blocks X and Y. If one data block is shorter than the other, then it is first padded to the left with sufficient binary zeros to make it the same length as the other.

4.3 Data Element Format Conventions

The EMV specifications use the following data element formats:

a	Alphabetic data elements contain a single character per byte. The permitted characters are alphabetic only (a to z and A to Z, upper and lower case).
an	Alphanumeric data elements contain a single character per byte. The permitted characters are alphabetic (a to z and A to Z, upper and lower case) and numeric (0 to 9).
ans	Alphanumeric Special data elements contain a single character per byte. The permitted characters and their coding are shown in the Common Character Set table in Annex B of Book 4. There is one exception: 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.
b	These data elements consist of either unsigned binary numbers or bit combinations that are defined elsewhere in the specification. Binary example: The Application Transaction Counter (ATC) is defined as “b” with a length of two bytes. An ATC value of 19 is stored as Hex '00 13'. Bit combination example: Processing Options Data Object List (PDOL) is defined as “b” with the format shown in Book 3, section 5.4.
cn	Compressed numeric data elements consist of two numeric digits (having values in the range Hex '0'–'9') per byte. These data elements are left justified and padded with trailing hexadecimal 'F's. Example: The Application Primary Account Number (PAN) is defined as “cn” with a length of up to ten bytes. A value of 1234567890123 may be stored in the Application PAN as Hex '12 34 56 78 90 12 3F FF' with a length of 8.
n	Numeric data elements consist of two numeric digits (having values in the range Hex '0'–'9') per byte. These digits are right justified and padded with leading hexadecimal zeroes. Other specifications sometimes refer to this data format as Binary Coded Decimal (“BCD”) or unsigned packed. Example: Amount, Authorised (Numeric) is defined as “n 12” with a length of six bytes. A value of 12345 is stored in Amount, Authorised (Numeric) as Hex '00 00 00 01 23 45'.

var. Variable data elements are variable length and may contain any bit combination. Additional information on the formats of specific variable data elements is available elsewhere.

4.4 Terminology

proprietary	Not defined in this specification and/or outside the scope of this specification
shall	Denotes a mandatory requirement
should	Denotes a recommendation

Part II

Electromechanical Characteristics, Logical Interface, and Transmission Protocols

5 Electromechanical Interface

This section covers the electrical and mechanical characteristics of the ICC and the terminal. ICC and terminal specifications differ to allow a safety margin to prevent damage to the ICC.

The ICC characteristics defined herein are based on the ISO/IEC 7816 series of standards with some small variations.

5.1 Lower Voltage ICC Migration

A phased migration to lower voltage cards is underway. Cards that support class A only are being phased out and shall be replaced by class AB or class ABC cards by end December 2013. When all cards in use support class AB or class ABC, it will be possible to deploy terminals that support class B only in addition to class A only terminals. Refer to General Bulletin 11 on the EMVCo website at http://www.emvco.com for details of the migration schedule.

Section 5 describes the requirements for cards and terminals as the transition occurs. Differences are indicated using the notations shown in Table 1:

Notation	Information applies:	Values:
class A cards until end December 2013	to class A cards	are permitted for cards in circulation until end December 2013. From January 2014, all cards in circulation shall be either class AB or class ABC.
new card values from January 2014	to the following cards:[1] • class A (until end December 2013) • class AB • class ABC	are permitted immediately and until further notice. No class A cards shall be in circulation from January 2014; only class AB or class ABC cards shall be in circulation from January 2014.
class A terminals until end December 2013	to class A terminals (or the class A component of multi-class terminals)	shall be used for class A terminals until end December 2013. From January 2014, there is no requirement to update terminals already in the field built using these values.
new terminal values from January 2014	to class A, class B, and class C terminals	shall not be used before end December 2013. From January 2014, shall be used for new class A or class B terminals. Class C terminals shall not be deployed until stated by EMVCo (except for proprietary purposes outside the scope of EMV).

Table 1: Lower Voltage Card Migration

[1] Class B, class C, class AC, and class BC cards are not allowed.

5.2 Mechanical Characteristics of the ICC

This section describes the physical characteristics, contact assignment, and mechanical strength of the ICC.

5.2.1 Physical Characteristics

Except as otherwise specified herein, the ICC shall comply with the physical characteristics for ICCs as defined in ISO/IEC 7816-1. The ICC shall also comply with the additional characteristics defined in ISO/IEC 7816-1 as related to ultraviolet light, X-rays, surface profile of the contacts, mechanical strength, electromagnetic characteristics, and static electricity and shall continue to function correctly electrically under the conditions defined therein.

5.2.1.1 Module Height

The highest point on the IC module surface shall not be greater than 0.10mm above the plane of the card surface.

The lowest point on the IC module surface shall not be greater than 0.10mm below the plane of the card surface.

5.2.2 Dimensions and Location of Contacts

The dimensions and location of the contacts shall be as shown in Figure 1:

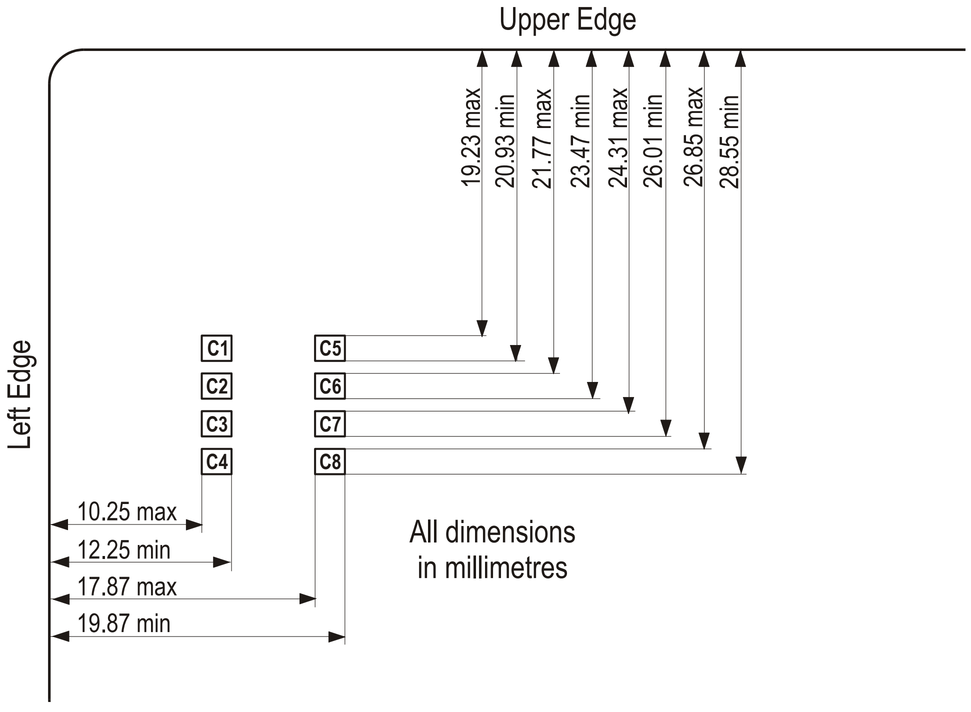


Figure 1: ICC Contact Location and Dimensions

Areas C1, C2, C3, C5, and C7 shall be fully covered by conductive surfaces forming the minimum ICC contacts. Areas C4, C6, C8, and areas Z1 to Z8 as defined in ISO/IEC 7816-2 Annex B may optionally have conductive surfaces, but it is strongly recommended that no conductive surfaces exist in areas Z1 to Z8. If conductive surfaces exist in areas C6, and Z1 to Z8, they shall be electrically isolated from the integrated circuit (IC), from one another, and from any other contact area. (Electrically isolated means that the resistance measured between the conductive surface and any other conductive surface shall be ≥10MΩ with an applied voltage of 5V DC.) In addition, there shall be no connection between the conductive surface of any area and the conductive surface of any other area, other than via the IC. The minimum ICC contacts shall be connected to the IC contacts as shown in Table 2.

The layout of the contacts relative to embossing and/or magnetic stripe shall be as shown in Figure 2:

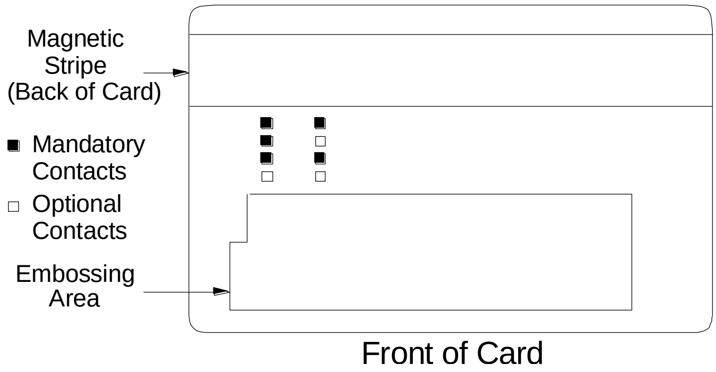


Figure 2: Layout of Contacts

Note: Care should be taken that card embossing does not damage the IC. Further, positioning of the signature panel behind the IC may lead to damage due to heavy pressure being applied during signature.

5.2.3 Contact Assignment

The assignment of the ICC contacts shall be as defined in ISO/IEC 7816-2 and is shown in Table 2:

C1	Supply voltage (VCC)	C5	Ground (GND)
C2	Reset (RST)	C6	RFU [2]
C3	Clock (CLK)	C7	Input/output (I/O)
C4	Not used; need not be physically present	C8	Not used; need not be physically present

Table 2: ICC Contact Assignment

[2] Defined in ISO/IEC 7816-3:1997 as programming voltage (VPP) for class A.

5.3 Electrical Characteristics of the ICC

This section describes the electrical characteristics of the signals as measured at the ICC contacts.

5.3.1 Measurement Conventions

All measurements are made at the point of contact between the ICC and the interface device (IFD) contacts and are defined with respect to the GND contact over an ambient temperature range 0° C to 50° C. ICCs shall be capable of correct operation over an ambient temperature range of at minimum 0° C to 50° C.

All currents flowing into the ICC are considered positive.

Note: The temperature range limits are dictated primarily by the thermal characteristics of polyvinyl chloride (which is used for the majority of cards that are embossed) rather than by constraints imposed by the characteristics of the IC.

5.3.2 Input/Output (I/O)

This contact is used as an input (reception mode) to receive data from the terminal or as an output (transmission mode) to transmit data to the terminal. During operation, the ICC and the terminal shall not both be in transmission mode. In the event that this condition occurs, the state (voltage level) of the I/O contact is indeterminate and no damage shall occur to the ICC.

5.3.2.1 Reception Mode

When in reception mode, and with the supply voltage (VCC) for the applicable class in the range specified in section 5.3.6, the ICC shall correctly interpret signals from the terminal having the characteristics shown in Table 3:

Symbol	Conditions	Minimum	Maximum	Unit
V_{IH}		0.7 x V_{CC}	V_{CC}	V
V_{IL}		0	0.8	V
t_R and t_F		—	1.0	µs
The ICC shall not be damaged by signal perturbations on the I/O line in the range –0.3 V to V_{CC} + 0.3 V.				

class A cards until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{IH}		0.7 x V_{CC}	V_{CC}	V
V_{IL}		0	0.2 x V_{CC}	V
t_R and t_F		—	1.0	µs
The ICC shall not be damaged by signal perturbations on the I/O line in the range –0.3 V to V_{CC} + 0.3 V.				

new card values from January 2014; see Table 1

Table 3: Electrical Characteristics of I/O for ICC Reception

5.3.2.2 Transmission Mode

When in transmission mode, the ICC shall send data to the terminal with the characteristics shown in Table 4:

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$-20\ \mu A < I_{OH} < 0$, V_{CC} = min.	0.7 x V_{CC}	V_{CC}	V
V_{OL}	$0 < I_{OL} < 1$ mA, V_{CC} = min.	0	0.4	V
t_R and t_F	$C_{IN\ (terminal)}$ = 30 pF max.	—	1.0	µs

class A cards until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$-20\ \mu A < I_{OH} < 0$	0.7 x V_{CC}	V_{CC}	V
V_{OL}	Class A: $0 < I_{OL} < 1$ mA	0	0.08 x V_{CC}	V
	Classes B and C: $0 < I_{OL} < 0.5$ mA	0	0.15 x V_{CC}	
t_R and t_F	$C_{IN\ (terminal)}$ = 30 pF max.	—	1.0	µs

new card values from January 2014; see Table 1

Table 4: Electrical Characteristics of I/O for ICC Transmission

Unless transmitting, the ICC shall set its I/O line driver to reception mode. There is no requirement for the ICC to have any current source capability to I/O.

5.3.3 Programming Voltage (VPP)

The ICC shall not require VPP (see note in section 5.4.3).

5.3.4 Clock (CLK)

With VCC in the range specified for the applicable class in section 5.3.6, the ICC shall operate correctly with a CLK signal having the characteristics shown in Table 5:

Symbol	Conditions	Minimum	Maximum	Unit
V_{IH}		V_{CC} – 0.7	V_{CC}	V
V_{IL}		0	0.5	V
t_R and t_F	V_{CC} = min. to max.	—	9% of clock period	
The ICC shall not be damaged by signal perturbations on the CLK line in the range –0.3 V to V_{CC} + 0.3 V.				

class A cards until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{IH}		0.7 x V_{CC}	V_{CC}	V
V_{IL}		0	0.2 x V_{CC}	V
t_R and t_F		—	9% of clock period	
The ICC shall not be damaged by signal perturbations on the CLK line in the range –0.3 V to V_{CC} + 0.3 V.				

new card values from January 2014; see Table 1

Table 5: Electrical Characteristics of CLK to ICC

The ICC shall operate correctly with a CLK duty cycle of between 44% and 56% of the period during stable operation.

The ICC shall operate correctly with a CLK frequency in the range 1 MHz to 5 MHz.

Note: Frequency shall be maintained by the terminal to within ± 1% of that used during the answer to reset throughout the card session.

5.3.5 Reset (RST)

With VCC in the range specified for the applicable class in section 5.3.6, the ICC shall correctly interpret a RST signal having the characteristics shown in Table 6:

Symbol	Conditions	Minimum	Maximum	Unit
V_{IH}		V_{CC} – 0.7	V_{CC}	V
V_{IL}		0	0.6	V
t_R and t_F	V_{CC} = min. to max.	—	1.0	µs
The ICC shall not be damaged by signal perturbations on the RST line in the range –0.3 V to V_{CC} + 0.3 V.				

class A cards until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{IH}		0.7 x V_{CC}	V_{CC}	V
V_{IL}		0	0.2 x V_{CC}	V
t_R and t_F	V_{CC} = min. to max.	—	1.0	µs
The ICC shall not be damaged by signal perturbations on the RST line in the range –0.3 V to V_{CC} + 0.3 V.				

new card values from January 2014; see Table 1

Table 6: Electrical Characteristics of RST to ICC

The ICC shall answer to reset asynchronously using active low reset.

5.3.6 Supply Voltage (VCC)

The ICC shall operate correctly with a supply voltage V_{CC} of 5 V ± 0.5 V DC and have a maximum current requirement of 50 mA when operating at any frequency within the range specified in section 5.3.4.

class A cards until end December 2013; see Table 1

Three classes of operation are defined based on the nominal supply voltage applied to the ICC. These are defined in Table 7. The ICC shall support class A and may optionally support one or more additional consecutive classes. The ICC shall operate correctly on any supply voltage lying within the range(s) specified for the class(es) it supports.

Symbol	Conditions	Minimum	Maximum	Unit
V_{CC}	Class A Class B Class C	4.50 2.70 1.62	5.50 3.30 1.98	V
I_{CC}	Class A Class B Class C		50 50 30	mA
The maximum current consumptions shown apply when operating at any frequency within the range specified in section 5.3.4.				

new card values from January 2014; see Table 1

Table 7: Classes of Operation

The ICC shall not be damaged if it is operated under classes that it does not support (the ICC is considered to be damaged if it no longer operates as specified, or if it contains corrupt data).

If the ICC supports more than one class, it may optionally operate correctly on any supply voltage lying between the ranges specified for the supported classes (see Table 8 below).

new card values from January 2014; see Table 1

Supported Classes	ICC Shall Operate	ICC May Operate	Unit
A and B	4.50–5.50 2.70–3.30	3.30–4.50	V
A, B, and C	4.50–5.50 2.70–3.30 1.62–1.98	3.30–4.50 1.98–2.70	V

Table 8: Mandatory and Optional Operating Voltage Ranges

For proprietary reasons terminals may support the capability to negotiate with the ICC the voltage class to be used, but this is outside the scope of EMV, and there is no requirement for ICCs conforming to this specification to support such negotiation. If the ICC returns a class indicator in the ATR as defined in ISO/IEC 7816-3, the ATR may be rejected in an EMV compliant terminal. To avoid interoperability problems, any class indicator used should be returned in the cold ATR; to guarantee that the ICC will be accepted in the event that the cold ATR is rejected, the warm ATR should be one of the basic ATRs defined in section 8.

Note: It is strongly recommended that the current consumption of ICCs is maintained at as low a value as possible, since the maximum current consumption allowable for the ICC may be reduced in future versions of this specification. Issuers of ICCs bearing multisector applications should ensure that the IC used has a current requirement compatible with all terminals (from all sectors) in which the ICC might be used.

5.3.7 Contact Resistance

The contact resistance as measured across a pair of clean ICC and clean nominal IFD contacts shall be less than 500 mΩ throughout the design life of an ICC (see ISO/IEC 10373 for test method).

Note: A nominal IFD contact may be taken as a minimum of 1.25 μm of gold over 5.00 μm of nickel.

5.4 Mechanical Characteristics of the Terminal

This section describes the mechanical characteristics of the terminal interface device.

5.4.1 Interface Device

The IFD into which the ICC is inserted shall be capable of accepting ICCs having the following characteristics:

- Physical characteristics compliant with ISO/IEC 7816-1
- Contacts on the front, in the position compliant with Figure 2 of ISO/IEC 7816-2
- Embossing compliant with ISO/IEC 7811-1 and ISO/IEC 7811-3

The IFD contacts shall be located such that if an ICC having contacts with the dimensions and locations specified in Figure 3 is inserted into the IFD, correct connection of all contacts shall be made. The IFD should have no contacts present other than those needed to connect to ICC contacts C1 to C8.

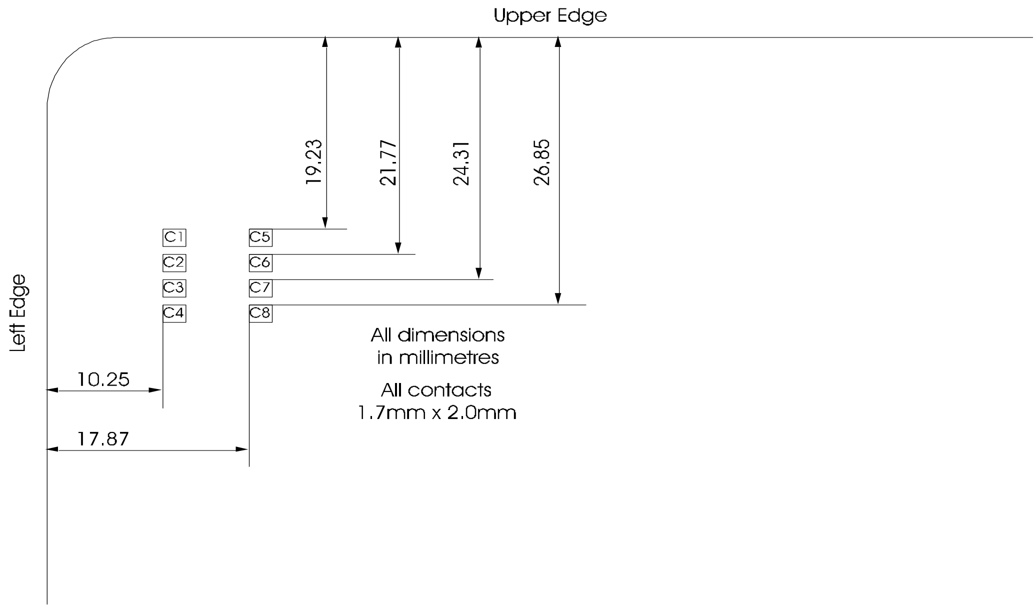


Figure 3: Terminal Contact Location and Dimensions

Location guides and clamps (if used) should cause no damage to ICCs, particularly in the areas of the magnetic stripe, signature panel, embossing, and hologram.

Note: As a general principle, an ICC should be accessible to the cardholder at all times. Where the ICC is drawn into the IFD, a mechanism should exist to return the ICC to the cardholder in the event of a failure (for example, loss of power).

5.4.2 Contact Forces

The force exerted by any one IFD contact on the corresponding ICC contact shall be in the range 0.2 N to 0.6 N.

5.4.3 Contact Assignment

The assignment of the IFD contacts shall be as shown in Table 9:

C1	VCC	C5	GND
C2	RST	C6	Not used for class A [3] RFU for classes B and C
C3	CLK	C7	I/O
C4	Not used; need not be physically present	C8	Not used; need not be physically present

Table 9: IFD Contact Assignment

5.5 Electrical Characteristics of the Terminal

This section describes the electrical characteristics of the signals as measured at the IFD contacts.

5.5.1 Measurement Conventions

All measurements are made at the point of contact between the ICC and the IFD contacts and are defined with respect to GND contact over an ambient temperature range 5° C to 40° C unless otherwise specified by the manufacturer. The internal temperature of the terminal should be limited to avoid damage to ICCs.

All currents flowing out of the terminal are considered positive.

[3] Defined in ISO/IEC 7816-3:1997 as programming voltage (VPP) for class A.

5.5.2 Input/Output (I/O)

This contact is used as an output (transmission mode) to transmit data to the ICC or as an input (reception mode) to receive data from the ICC. During operation, the terminal and the ICC should not both be in transmission mode. In the event that this condition occurs, the state (voltage level) of the contact is indeterminate and no damage shall occur to the terminal.

When both the terminal and the ICC are in reception mode, the contact shall be in the high state. The terminal shall not pull I/O high unless VCC is powered and stable within the tolerances specified in section 5.5.6. See the contact activation sequence specified in section 6.1.2.

The terminal shall limit the current flowing into or out of the I/O contact to ±15 mA at all times.

5.5.2.1 Transmission Mode

When in transmission mode, the terminal shall send data to the ICC with the characteristics shown in Table 10:

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$0 < I_{OH} < 20$ µA, V_{CC} = min.	0.8 x V_{CC}	V_{CC}	V
V_{OL}	– 0.5 mA $< I_{OL} < 0$, V_{CC} = min.	0	0.4	V
t_R and t_F	$C_{IN(ICC)}$ = 30 pF max.	—	0.8	µs
Signal perturbations	Signal low	– 0.25	0.4	V
	Signal high	0.8 x V_{CC}	V_{CC} + 0.25	V

class A terminals until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$0 < I_{OH} < 20$ µA	0.8 x V_{CC}	V_{CC}	V
V_{OL}	– 0.5 mA $< I_{OL} < 0$	0	0.15 x V_{CC}	V
t_R and t_F	$C_{IN(ICC)}$ = 30 pF max.	—	0.8	µs
Signal perturbations	Signal low	– 0.25	0.15 x V_{CC}	V
	Signal high	0.8 x V_{CC}	V_{CC} + 0.25	V

new terminal values from January 2014; see Table 1

Table 10: Electrical Characteristics of I/O for Terminal Transmission

Unless transmitting, the terminal shall set its I/O line driver to reception mode.

5.5.2.2 Reception Mode

When in reception mode, the terminal shall correctly interpret signals from the ICC having the characteristics shown in Table 11:

Symbol	Conditions	Minimum	Maximum	Unit	
V_{IH}		0.6 x V_{CC}	V_{CC}	V	class A terminals until end December 2013; see Table 1
V_{IL}		0	0.5	V	
t_R and t_F		—	1.2	µs	

Symbol	Conditions	Minimum	Maximum	Unit	
V_{IH}		0.6 x V_{CC}	V_{CC}	V	new terminal values from January 2014; see Table 1
V_{IL}		0	0.20 x V_{CC}	V	
t_R and t_F		—	1.2	µs	

Table 11: Electrical Characteristics of I/O for Terminal Reception

5.5.3 Programming Voltage (VPP)

C6 shall be electrically isolated. Electrically isolated means that the resistance measured between C6 and any other contact shall be ≥10MΩ with an applied voltage of 5V DC. If connected in existing class A terminals, C6 shall be maintained at a potential between GND and 1.05 x V_{CC} throughout the card session.

Note: Keeping C6 isolated in new class A terminals facilitates its use for other purposes if so defined in future versions of this specification.

5.5.4 Clock (CLK)

The terminal shall generate a CLK signal having the characteristics shown in Table 12:

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$0 < I_{OH} < 50\ \mu A$, V_{CC} = min.	$V_{CC} - 0.5$	V_{CC}	V
V_{OL}	$-50\ \mu A < I_{OL} < 0$, V_{CC} = min.	0	0.4	V
t_R and t_F	$C_{IN(ICC)}$ = 30 pF max.	—	8% of clock period	
Signal perturba-tions	Signal low	– 0.25	0.4	V
	Signal high	$V_{CC} - 0.5$	$V_{CC} + 0.25$	V

class A terminals until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$0 < I_{OH} < 50\ \mu A$	$0.8 \times V_{CC}$	V_{CC}	V
V_{OL}	$-50\ \mu A < I_{OL} < 0$	0	$0.15 \times V_{CC}$	V
t_R and t_F	$C_{IN(ICC)}$ = 30 pF max.	—	8% of clock period	
Signal perturba-tions	Signal low	– 0.25	$0.15 \times V_{CC}$	V
	Signal high	$0.8 \times V_{CC}$	$V_{CC} + 0.25$	V

new terminal values from January 2014; see Table 1

Table 12: Electrical Characteristics of CLK from Terminal

Duty cycle shall be between 45% and 55% of the period during stable operation.

Frequency shall be in the range 1 MHz to 5 MHz and shall not change by more than ± 1% throughout answer to reset and the following stages of a card session (see section 6) unless changed following the answer to reset by means of a proprietary negotiation technique.

5.5.5 Reset (RST)

The terminal shall generate a RST signal having the characteristics shown in Table 13:

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$0 < I_{OH} < 50$ µA, V_{CC} = min.	$V_{CC} – 0.5$	V_{CC}	V
V_{OL}	$– 50$ µA $< I_{OL} < 0$, V_{CC} = min.	0	0.4	V
t_R and t_F	$C_{IN(ICC)}$ = 30 pF max.	—	0.8	µs
Signal perturba-tions	Signal low	– 0.25	0.4	V
	Signal high	$V_{CC} – 0.5$	V_{CC} + 0.25	V

class A terminals until end December 2013; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{OH}	$0 < I_{OH} < 50$ µA	0.8 x V_{CC}	V_{CC}	V
V_{OL}	$– 50$ µA $< I_{OL} < 0$	0	0.15 x V_{CC}	V
t_R and t_F	$C_{IN(ICC)}$ = 30 pF max.	—	0.8	µs
Signal perturba-tions	Signal low	– 0.25	0.15 x V_{CC}	V
	Signal high	0.8 x V_{CC}	V_{CC} + 0.25	V

new terminal values from January 2014; see Table 1

Table 13: Electrical Characteristics of RST from Terminal

5.5.6 Supply Voltage (VCC)

The terminal shall generate a V_{CC} of 5 V ± 0.4 V DC and shall be capable of delivering steady state output current in the range 0 to 55 mA whilst maintaining V_{CC} within these tolerances. The supply shall be protected from transients and surges caused by internal operation of the terminal and from external interference introduced via power leads, communications links, etc. V_{CC} shall never be less than –0.25V with respect to ground.

class A terminals until end December 2013; see Table 1

During normal operation of an ICC, current pulses cause voltage transients on VCC as measured at the ICC contacts. The power supply shall be able to counteract transients in the current consumption of the ICC having a charge ≤30 nAs, a duration ≤400 ns, an amplitude ≤100 mA, and a rate of change of current ≤1 mA/ns, ensuring that VCC remains within the range specified. See Figure 4 for the maximum envelope of the pulse.

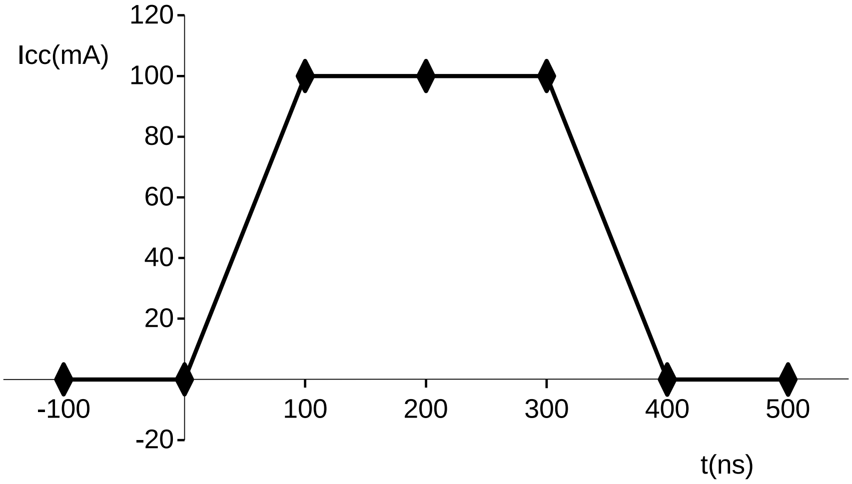


Figure 4: Maximum Current Pulse Envelope

The terminal shall generate a V_{CC} within one of the range(s) specified in Table 14 below for the class(es) supported, and shall be capable of delivering the corresponding steady state output current whilst maintaining V_{CC} within that range. If the terminal supports more than one class, it shall always generate a V_{CC} from the class containing the highest voltage range available.

For proprietary reasons terminals may support the capability to negotiate with the ICC the voltage class to be used, but this is outside the scope of EMV, and is not supported by ICCs conforming to this specification. Attempting class negotiation with such an ICC may result in the ICC being rejected.

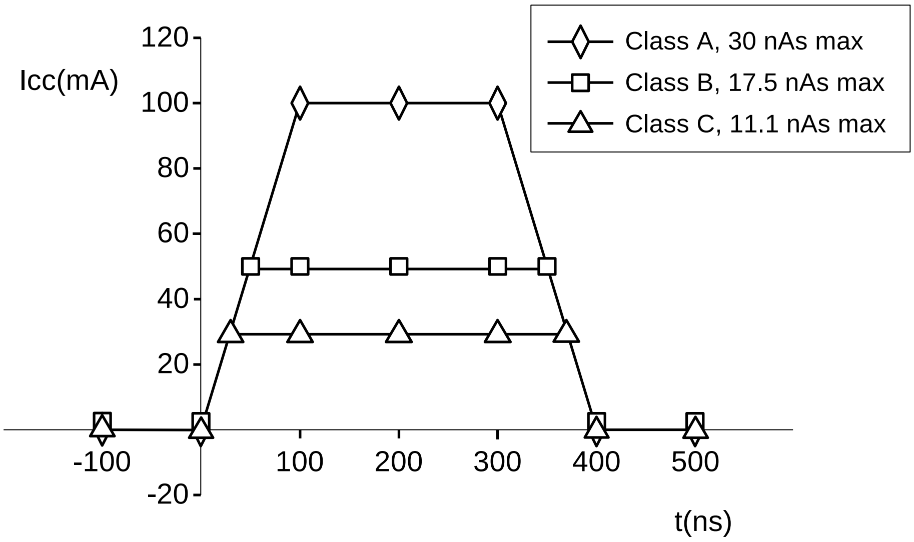
The supply shall be protected from transients and surges caused by internal operation of the terminal and from external interference introduced via power leads, communications links, etc. V_{CC} shall never be less than –0.25V with respect to ground.

new terminal values from January 2014; see Table 1

Symbol	Conditions	Minimum	Maximum	Unit
V_{CC}	Class A Class B Class C	4.60 2.76 1.66	5.40 3.24 1.94	V
I_{CC}	Class A Class B Class C	55 55 35		mA

Table 14: Terminal Supply Voltage and Current

During normal operation of an ICC, current pulses cause voltage transients on VCC as measured at the ICC contacts. The power supply shall be able to counteract transients in the current consumption of the ICC having characteristics within the maximum charge envelope applicable to the class of operation as shown in Figure 5, ensuring that VCC remains within the range specified.



new terminal values from January 2014; see Table 1

Figure 5: Maximum Current Pulse Envelopes

Note: Terminals may be designed to be capable of delivering more than required current, but it is recommended that terminals limit the steady state current that can be delivered to a maximum of 200 mA.

5.5.7 Contact Resistance

The contact resistance as measured across a pair of clean IFD and clean nominal ICC contacts shall be less than 500 mΩ throughout the design life of a terminal (see ISO/IEC 7816-1 for test method).

Note: A nominal ICC contact may be taken as 1.25 µm of gold over 5.00 µm of nickel.

5.5.8 Short Circuit Resilience

The terminal shall not be damaged in the event of fault conditions such as a short circuit between any combinations of contacts. The terminal shall be capable of sustaining a short circuit of any duration between any or all contacts without suffering damage or malfunction, for example, if a metal plate is inserted.

5.5.9 Powering and Depowering of Terminal with ICC in Place

If the terminal is powered on or off with an ICC in place, all signal voltages shall remain within the limits specified in section 5.5, and contact activation and deactivation sequences and timings, as described in sections 6.1.2 and 6.1.5 respectively, shall be respected.

6 Card Session

This section describes all stages involved in a card session from insertion of the ICC into the IFD through the execution of the transaction to the removal of the ICC from the IFD.

6.1 Normal Card Session

This section describes the processes involved in the execution of a normal transaction.

6.1.1 Stages of a Card Session

A card session is comprised of the following stages:

1. Insertion of the ICC into the IFD and connection and activation of the contacts.
2. Reset of the ICC and establishment of communication between the terminal and the ICC.
3. Execution of the transaction(s).
4. Deactivation of the contacts and removal of the ICC.

6.1.2 ICC Insertion and Contact Activation Sequence

On insertion of the ICC into the IFD, the terminal shall ensure that all signal contacts are in state L with values of V_{OL} as defined in section 5.5 and that V_{CC} is 0.4 V or less at the instant galvanic contact is made. When the ICC is correctly seated within the IFD, the contacts shall be activated as follows (see Figure 6):

- RST shall be maintained by the terminal in state L throughout the activation sequence.
- Following establishment of galvanic contact but prior to activation of I/O or CLK, VCC shall be powered.
- Following verification by the terminal that V_{CC} is stable and within the limits defined in section 5.5.6, the terminal shall set its I/O line driver to reception mode and shall provide CLK with a suitable and stable clock as defined in section 5.5.4. The I/O line driver in the terminal may be set to reception mode prior to application of the clock but shall be set to reception mode no later than 200 clock cycles after application of the clock.

Note: The terminal may verify the state of V_{CC} by measurement, by waiting sufficient time for it to stabilise according to the design of the terminal, or otherwise. The state of the I/O line after the terminal has set its I/O line driver to reception mode is dependent upon the state of the I/O line driver in the ICC (see section 6.1.3.1).

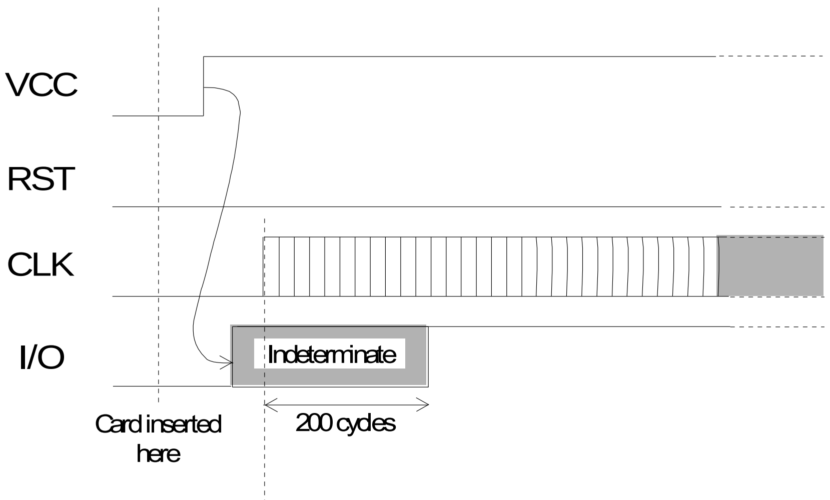


Figure 6: Contact Activation Sequence

6.1.3 ICC Reset

The ICC shall answer to reset asynchronously using active low reset.

The means of transportation of the answer to reset (ATR) are described in section 7 and its contents are described in sections 8.2 and 8.3.

6.1.3.1 Cold Reset

Following activation of the contacts according to section 6.1.2, the terminal shall initiate a cold reset and obtain an ATR from the ICC as follows (see Figure 7):

- The terminal shall apply CLK at a notional time T0.
- Within a maximum of 200 clock cycles following T0, the ICC shall set its I/O line driver to reception mode. Since the terminal shall also have set its I/O line driver to reception mode within this period, the I/O line is guaranteed to be in state H no later than 200 clock cycles following time T0.
- The terminal shall maintain RST in state L through time T0 and for a period of between 40,000 and 45,000 clock cycles following time T0 to time T1, when it shall set RST to state H.
- The answer to reset on I/O from the ICC shall begin between 400 and 40,000 clock cycles after time T1 (time *t1* in Figure 7).
- The terminal shall have a reception window which is opened no later than 380 clock cycles after time T1 and closed no earlier than 42,000 clock cycles after time T1 (time *t1* in Figure 7). If no answer to reset is received from the ICC, the terminal shall initiate the deactivation sequence no earlier than 42,001 clock cycles after time T1, and no later than 42,000 clock cycles plus 50ms after time T1.

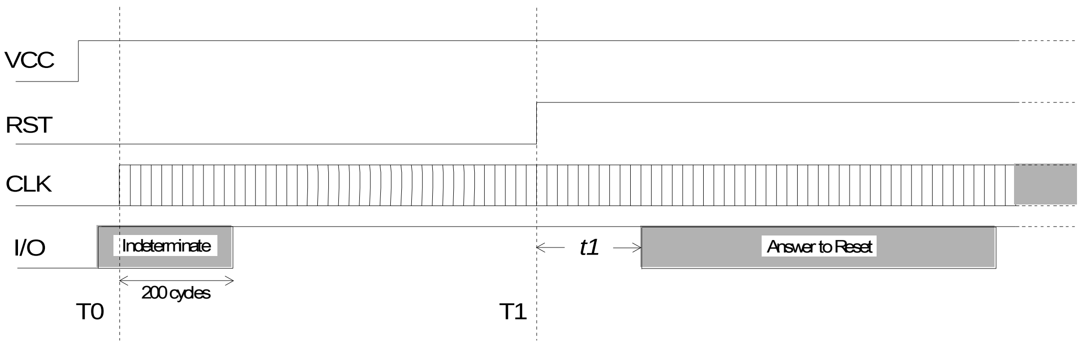


Figure 7: Cold Reset Sequence

6.1.3.2 Warm Reset

If the ATR received following a cold reset as described in section 6.1.3.1 does not conform to the specification in section 8, the terminal shall initiate a warm reset and obtain an ATR from the ICC as follows (see Figure 8):

- A warm reset shall start at a notional time T0', at which time the terminal shall set RST to state L.
- The terminal shall maintain VCC and CLK stable and within the limits defined in sections 5.5.4 and 5.5.6 throughout the warm reset sequence.
- Within a maximum of 200 clock cycles following T0', the ICC and terminal shall set their I/O line drivers to reception mode. The I/O line therefore is guaranteed to be in state H no later than 200 clock cycles following time T0'.
- The terminal shall maintain RST in state L from time T0' for a period of between 40,000 and 45,000 clock cycles following time T0' to time T1', when it shall set RST to state H.
- The answer to reset on I/O from the ICC shall begin between 400 and 40,000 clock cycles after time T1' (time *t1'* in Figure 8).
- The terminal shall have a reception window which is opened no later than 380 clock cycles after time T1' and closed no earlier than 42,000 clock cycles after time T1' (time *t1'* in Figure 8). If no answer to reset is received from the ICC, the terminal shall initiate the deactivation sequence no earlier than 42,001 clock cycles after time T1', and no later than 42,000 clock cycles plus 50ms after time T1'.

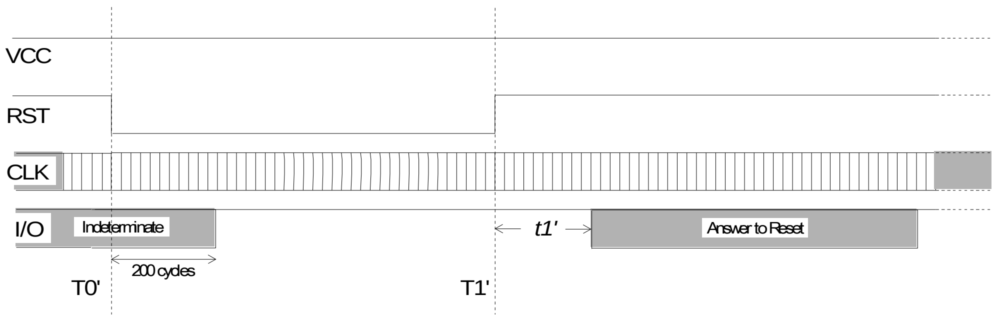


Figure 8: Warm Reset Sequence

Note: Figure 8 indicates that the terminal may initiate the warm reset sequence during the time that the card is still transmitting the cold ATR, and in the event that it does, the card shall be able to respond correctly with the warm ATR.

6.1.4 Execution of a Transaction

Selection of the application in the ICC and the subsequent exchange of information between the ICC and the terminal necessary to perform a transaction are described in section 12 and in Book 3.

6.1.5 Contact Deactivation Sequence

As the final step in the card session, upon normal or abnormal termination of the transaction (including withdrawal of the ICC from the IFD during a card session), the terminal shall deactivate the IFD contacts as follows (see Figure 9):

- The terminal shall initiate the deactivation sequence by setting RST to state L.
- Following the setting of RST to state L but prior to depowering VCC, the terminal shall set CLK and I/O to state L.
- Following the setting of RST, CLK, and I/O to state L but prior to galvanic disconnection of the IFD contacts, the terminal shall depower VCC. V_{CC} shall be 0.4 V or less prior to galvanic disconnection of the IFD contacts.
- The deactivation sequence shall be completed within 100 ms. This period is measured from the time that RST is set to state L to the time that V_{CC} reaches 0.4 V or less.

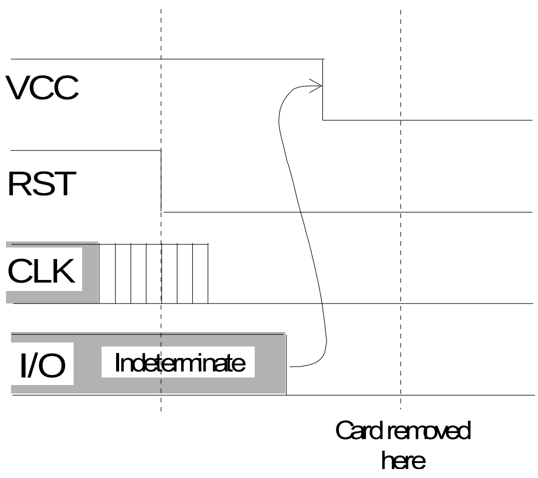


Figure 9: Contact Deactivation Sequence

6.2 Abnormal Termination of Transaction Process

If an ICC is prematurely removed from a terminal during execution of a transaction at speeds of up to 1 m/s, the terminal shall be capable of sensing the movement of the ICC relative to the IFD contacts, and of deactivating all IFD contacts in the manner described in section 6.1.5 before the relative movement exceeds 1 mm. No electrical or mechanical damage shall be caused to the ICC under these conditions.

Note: For 'sliding carriage' type IFDs, it may be possible for the terminal to sense the movement of the ICC/IFD contact sub-assembly relative to the main body of the IFD. In this event, it is not mandatory to be able to sense the movement of the ICC relative to the IFD contacts, but deactivation of the contacts shall be complete before any electrical contact is broken between the ICC and IFD.

7 Physical Transportation of Characters

During the transaction process, data is passed bi-directionally between the terminal and the ICC over the I/O line in an asynchronous half duplex manner. A clock signal is provided to the ICC by the terminal, and this shall be used to control the timing of this exchange. The mechanism of exchanging bits and characters is described below. It applies during the answer to reset and is also used by both transmission protocols as described in section 9.

7.1 Bit Duration

The bit duration used on the I/O line is defined as an elementary time unit (etu). A linear relationship exists between the etu on the I/O line and CLK frequency (*f*).

During the answer to reset, the bit duration is known as the initial etu, and is given by the following equation:

$$\text{initial etu} = \frac{372}{f} \text{ seconds, where } f \text{ is in Hertz}$$

Following the answer to reset (and establishment of the global parameters F and D, as described in section 8), the bit duration is known as the current etu, and is given by the following equation:

$$\text{current etu} = \frac{\mathrm{F}}{\mathrm{D}f} \text{ seconds, where } f \text{ is in Hertz}$$

Note: For the basic answer(s) to reset described in this specification, only values of F = 372 and D = 1 are supported. In the following sections, “etu” indicates current etu unless otherwise specified.

7.2 Character Frame

Data is passed over the I/O line in a character frame as described below. The convention used is specified in the initial character (TS) transmitted by the ICC in the ATR (see section 8.3.1).

Prior to transmission of a character, the I/O line shall be in state H.

A character consists of 10 consecutive bits (see Figure 10):

- 1 start bit in state L
- 8 bits, which comprise the data byte
- 1 even parity checking bit

The start bit is detected by the receiving end by periodically sampling the I/O line. The sampling time should be less than or equal to 0.2 etu.

The number of logic ones in a character shall be even. The 8 bits of data and the parity bit itself are included in this check but the start bit is not.

The time origin is fixed as midway between the last observation of state H and the first observation of state L. The existence of a start bit should be verified within 0.7 etu. Subsequent bits should be received at intervals of (n + 0.5 ± 0.2) etu (n being the rank of the bit). The start bit is bit 1.

Within a character, the time from the leading edge of the start bit to the trailing edge of the nth bit is (n ± 0.2) etu.

The interval between the leading edges of the start bits of two consecutive characters is comprised of the character duration (10 ± 0.2) etu, plus a guardtime. Under error free transmission, during the guardtime both the ICC and the terminal shall be in reception mode (I/O line in state H). For T=0 only, if the ICC or terminal as receiver detects a parity error in a character just received, it shall set I/O to state L to indicate the error to the sender (see section 9.2.3).

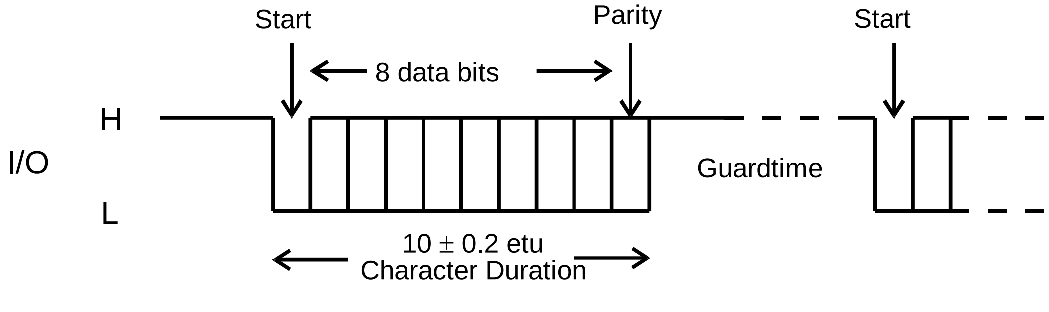


Figure 10: Character Frame

At the terminal transport layer (TTL), data shall always be passed over the I/O line most significant (m.s.) byte first. The order of bits within a byte (that is, whether the least significant (l.s.) or m.s. bit is transferred first) is specified in character TS returned in the answer to reset (see section 8.3).

8 Answer to Reset

After being reset by the terminal as described in section 6.1.3, the ICC answers with a string of bytes known as the ATR. These bytes convey information to the terminal that defines certain characteristics of the communication to be established between the ICC and the terminal. The method of transporting these bytes, and their meaning, is described below.

Note: In sections 8 and 9, the m.s. bit of a character refers to bit b8 and the l.s. bit refers to bit b1. A value in straight single quotes is coded in hexadecimal notation; for example, 'A' or '3F'.

8.1 Physical Transportation of Characters Returned at Answer to Reset

This section describes the structure and timing of the characters returned at the answer to reset.

The bit duration is defined in section 7.1, and the character frame is defined in section 7.2.

During the answer to reset, the minimum interval between the leading edges of the start bits of two consecutive characters shall be 12 initial etus, and the maximum interval between the leading edges of the start bits of two consecutive characters shall be 9600 initial etus.

The ICC shall transmit all the characters to be returned during an answer to reset (warm or cold) within 19,200 initial etus.[4] This time is measured between the leading edge of the start bit of the first character (TS) and 12 initial etus after the leading edge of the start bit of the last character.

[4] The maximum time allowed for the answer to reset varies according to clock frequency, since the period represented by an etu is frequency dependent (see section 7.1).

8.2 Characters Returned by ICC at Answer to Reset

The number and coding of the characters returned by the ICC at the answer to reset varies depending upon the transmission protocol(s) and the values of the transmission control parameters supported. This section describes two basic answers to reset, one for ICCs supporting T=0 only and the other for ICCs supporting T=1 only. It defines the characters to be returned and the allowable ranges of values for the transmission control parameters. ICCs returning one of the two answers to reset described here are assured correct operation and interoperability in terminals conforming to this specification.

For proprietary reasons ICCs may optionally support more than one transmission protocol, but one of the supported protocols shall be T=0 or T=1. The first offered protocol shall be T=0 or T=1, and the terminal shall continue the card session using the first offered protocol unless for proprietary reasons it supports a mechanism for selecting an alternative protocol offered by the ICC. Support for such a mechanism is not required by, and is beyond the scope of these specifications.

Note: This specification does not support ICCs having both T=0 and T=1 protocols present at the same time. This can only be achieved by proprietary means beyond the scope of this specification.

Also for proprietary reasons ICCs may optionally support other values of the transmission control parameters at the issuer's discretion. However, such support is considered outside the scope of this specification and such ICCs may be rejected at terminals conforming to this specification, which need not have the corresponding additional proprietary functionality required to support the ICC.

The characters returned by the ICC at the answer to reset for the two basic answers to reset are shown in Table 15 and Table 16. The characters are shown in the order in which they are sent by the ICC, that is, TS first.

If protocol type T=0 only is supported (character-oriented asynchronous transmission protocol), the characters returned shall be as shown in Table 15:

Character	Value	Remarks
TS	'3B' or '3F'	Indicates direct or inverse convention
T0	'6x'	TB1 and TC1 present; x indicates the number of historical bytes present
TB1	'00'	VPP not required
TC1	'00' to 'FF'	Indicates the amount of extra guardtime required. Value 'FF' has a special meaning (see section 8.3.3.3)

Table 15: Basic ATR for T=0 Only

If protocol type T=1 only is supported (block-oriented asynchronous transmission protocol), the characters returned shall be as shown in Table 16:

Character	Value	Remarks
TS	'3B' or '3F'	Indicates direct or inverse convention
T0	'Ex'	TB1 to TD1 present; x indicates the number of historical bytes present
TB1	'00'	VPP not required
TC1	'00' to 'FF'	Indicates amount of extra guardtime required. Value 'FF' has special meaning (see section 8.3.3.3)
TD1	'81'	TA2, TB2, and TC2 absent; TD2 present; T=1 to be used
TD2	'31'	TA3 and TB3 present; TC3 and TD3 absent; T=1 to be used
TA3	'10' to 'FE'	Returns IFSI, which indicates initial value for information field size for the ICC and IFSC of 16–254 bytes
TB3	m.s. nibble '0' to '4' l.s. nibble '0' to '5'	BWI = 0 to 4 CWI = 0 to 5
TCK	See section 8.3.4	Check character

Table 16: Basic ATR for T=1 Only

8.3 Character Definitions

This section provides detailed descriptions of the characters that may be returned at the answer to reset.

Each character description includes the following information:

- title
- explanation of usage as described in ISO/IEC 7816-3
- basic response (This response should always be used in a warm ATR to ensure interoperability.)
- required terminal behaviour in the event that a terminal receives characters outside the range allowed by EMV

The 'basic response' indicates the presence or absence of the character, and the allowable range of values it may take (if present) if it is to conform to one of the basic ATRs. The description of a basic response (even though indicated by 'shall') is not intended to preclude the use of other values of the characters, nor the omission/inclusion of a character at the issuer's discretion. For example, the ICC may return additional characters if it supports more than one transmission protocol (see section 9). However, only ICCs returning a basic ATR, or an ATR supported by the minimum required terminal functionality described below, are guaranteed to be supported correctly in interchange.

Terminals conforming to this specification are only required (as a minimum) to support the basic ATRs described here together with any additional requirements specified in 'terminal behaviour'. Terminals may thus reject an ATR containing interface bytes not described in, or having values not specified in, this specification. However, terminals may correctly interpret such an ATR if it is returned by an ICC for proprietary (for example, national) use. Such terminal functionality is not mandatory and is beyond the scope of this specification. As a general principle, a terminal should accept a non-basic ATR if it is able to function correctly with it.

Terminals shall be capable of checking the parity of characters returned in the answer to reset, but not necessarily as they are received. If the terminal detects a parity error, it shall reject the ICC.

Table 17 describes the action indicated by several terms in the following character descriptions:

If it is indicated that a terminal shall:	then:
reject an ATR	• If the terminal is rejecting a cold ATR, the terminal shall issue a warm reset. • If the terminal is rejecting a warm ATR, the terminal shall terminate the card session by deactivating the ICC contacts.
reject an ICC	The terminal shall terminate the card session by deactivating the ICC contacts.
accept an ATR	The terminal shall accept the ATR, but *only* if the requirements specified in this section for all other characters are also met.

Table 17: Terminal Behaviour

8.3.1 TS - Initial Character

TS performs two functions. It provides a known bit pattern to the terminal to facilitate bit synchronisation, and it indicates the logic convention to be used for the interpretation of the subsequent characters.

Using inverse logic convention, a low state L on the I/O line is equivalent to a logic one, and the m.s. bit of the data byte is the first bit sent after the start bit. Using direct logic convention, a high state H on the I/O line is equivalent to a logic one, and the l.s. bit of the data byte is the first bit sent after the start bit. The first four bits LHHL are used for bit synchronisation.

Basic responses: The ICC shall return an ATR containing TS having one of two values:

- `(H)LHHLLLLLLH` inverse convention value '3F'
- `(H)LHHLHHHLLH` direct convention value '3B'

The convention used may differ between cold and warm resets.

Terminal behaviour: The terminal shall be capable of supporting both inverse and direct convention and shall accept an ATR containing TS having a value of either '3B' or '3F'. An ICC returning an ATR containing TS having any other value shall be rejected.

Note: It is strongly recommended that a value of '3B' is returned by the ICC since a value of '3F' may not be supported in future versions of this specification.

8.3.2 T0 - Format Character

T0 is comprised of two parts. The m.s. nibble (b5–b8) is used to indicate whether the subsequent characters TA1 to TD1 are present. Bits b5–b8 are set to the logic one state to indicate the presence of TA1 to TD1, respectively. The l.s. nibble (b1–b4) indicates the number of optional historical bytes present (0 to 15). (See Table 18 for the basic response coding of character T0.)

Basic responses: If T=0 only is to be used, the ATR shall contain T0 = '6x', indicating that characters TB1 and TC1 are present. If T=1 only is to be used, the ATR shall contain T0 = 'Ex', indicating that characters TB1 to TD1 are present. The value of 'x' represents the number of optional historical bytes to be returned.[5]

Terminal behaviour: The terminal shall accept an ATR containing T0 of any value provided that the value returned correctly indicates and is consistent with the interface characters TA1 to TD1 and historical bytes actually returned.

	b8	b7	b6	b5	b4	b3	b2	b1
T=0 only	0	1	1	0	x	x	x	x
T=1 only	1	1	1	0	x	x	x	x

Table 18: Basic Response Coding of Character T0

8.3.3 TA1 to TC3 - Interface Characters

TA1 to TC3 convey information that shall be used during exchanges between the terminal and the ICC subsequent to the answer to reset. They indicate the values of the transmission control parameters F, D, I, P, and N, and the IFSC, block waiting time integer (BWI), and character waiting time integer (CWI) applicable to T=1 as defined in ISO/IEC 7816-3. The information contained in TA1, TB1, TC1, TA2, and TB2 shall apply to all subsequent exchanges irrespective of the protocol type to be used.

[5] Although their use is not forbidden, the EMV Specifications do not explicitly support the use of historical bytes, and their usage, structure and meaning are outside the scope of EMV. However, if used, it is strongly recommended that they are encoded to have a structure and meaning according to ISO/IEC 7816-4. EMV-compliant terminals should ignore any historical bytes present in the ATR. However, if an EMV-compliant terminal does support historical bytes, it should never be designed in such a way that non-recognition or misinterpretation of any historical bytes present in the ATR causes termination of the transaction. Since Issuers are free to encode the historical bytes in any way they choose, they should recognise that unintentional conflict of coding between cards may exist, leading to misinterpretation at terminals. Great care should be exercised by the terminal that it is able to unambiguously identify a card before interpreting any historical bytes returned in the ATR.

8.3.3.1 TA1

TA1 conveys the values of FI and DI where:

- the m.s. nibble FI is used to determine the value of F, the clock rate conversion factor, which may be used to modify the frequency of the clock provided by the terminal subsequent to the answer to reset
- the l.s. nibble DI is used to determine the value of D, the bit rate adjustment factor, which may be used to adjust the bit duration used subsequent to the answer to reset

See section 7.1 for calculation of the bit duration subsequent to the answer to reset (current etu).

Default values of FI = 1 and DI = 1 indicating values of F = 372 and D = 1, respectively, shall be used during the answer to reset.

Basic response: The ATR shall not contain TA1 and thus the default values of F = 372 and D = 1 shall continue be used during all subsequent exchanges.

Terminal behaviour: If TA1 is present in the ATR (indicated by b5 of T0 set to 1) and TA2 is returned with b5 = 0 (specific mode, parameters defined by the interface bytes), the terminal shall:

- Accept the ATR if the value of TA1 is in the range '11' to '13',[6] and immediately implement the values of F and D indicated (F=372 and D = 1, 2, or 4).
- Reject the ATR if the value of TA1 is not in the range '11' to '13', unless it is able to support and immediately implement the conditions indicated.

If TA1 is present in the ATR (indicated by b5 of T0 set to 1) and TA2 is not returned (negotiable mode), the terminal shall accept the ATR and shall continue using the default values of D = 1 and F = 372 during all subsequent exchanges, unless it supports a proprietary technique for negotiating the parameters to be used.

If TA1 is absent from the ATR, the default values of D = 1 and F = 372 shall be used during all subsequent exchanges.

[6] Terminals compliant to version 3.1.1 of the EMV Specifications may reject an ATR (not an ICC) if TA1 is present and coded other than '11'. ATRs indicating the higher allowable values of D will include TA1 coded '12' or '13', and thus may be rejected in 3.1.1 compliant terminals. Therefore, to ensure that an ICC supporting higher data transfer rates is always accepted in 3.1.1 compliant terminals (albeit operating at basic data transfer rates), it is essential that any TA1 indicating higher data rates is present in the cold ATR only, and that a warm ATR is always present which either does not contain TA1, or includes a TA1 having the value '11'.

8.3.3.2 TB1

TB1 conveys the values of PI1 and II where:

- PI1 is specified in bits b1 to b5 and is used to determine the value of the programming voltage P required by the ICC. PI1 = 0 indicates that VPP is not connected in the ICC.
- II is specified in bits b6 and b7 and is used to determine the maximum programming current, I_{pp}, required by the ICC. This parameter is not used if PI1 = 0.
- Bit 8 is not used and shall be set to logic zero.

Basic response: The ATR shall contain TB1 = '00', indicating that VPP is not connected in the ICC.

Terminal behaviour: In response to a cold reset, the terminal shall accept only an ATR containing TB1 = '00'. In response to a warm reset the terminal shall accept an ATR containing TB1 of any value (provided that b6 of T0 is set to 1) or not containing TB1 (provided that b6 of T0 is set to 0) and shall continue the card session as though TB1 = '00' had been returned. V_{PP} shall never be generated.

Note: Existing terminals may maintain V_{PP} in the idle state (see section 5.4.3).

The basic response coding of character TB1 is shown in Table 19:

b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	0	0	0	0	0

Table 19: Basic Response Coding of Character TB1

8.3.3.3 TC1

TC1 conveys the value of N, where N is used to indicate the extra guardtime that shall be added to the minimum duration between the leading edges of the start bits of two consecutive characters for subsequent exchanges from the terminal to the ICC. N is binary coded over bits b1–b8 of TC1, and its value represents the number of etus to be added as extra guardtime. TC1='FF' has a special meaning and indicates that the minimum delay between the leading edges of the start bits of two consecutive characters shall be reduced to 12 etus if T=0 is to be used, or 11 etus if T=1 is to be used.

Note: TC1 applies only to the timing between two consecutive characters sent from the terminal to the ICC. It does not apply to the timing between consecutive characters sent from the ICC to the terminal, nor does it apply to the timing between two characters sent in opposite directions (see sections 9.2.2.1 and 9.2.4.2.2).

N may take any value between 0 and 255.

If the value of TC1 is in the range '00' to 'FE', between 0 and 254 etus of extra guardtime shall be added to the minimum character to character duration, which for subsequent transmissions shall be between 12 and 266 etus.

If the value of TC1 = 'FF', then the minimum character to character duration for subsequent transmissions shall be 12 etus if T=0 is to be used, or 11 etus if T=1 is to be used.

Basic response: The ATR shall contain TC1 having a value in the range '00' to 'FF'.

Terminal behaviour: The terminal shall accept an ATR not containing TC1 (provided that b7 of T0 is set to 0), and shall continue the card session as though TC1 = '00' had been returned.

The basic response coding of character TC1 is shown in Table 20:

b8	b7	b6	b5	b4	b3	b2	b1
x	x	x	x	x	x	x	x

Table 20: Basic Response Coding of Character TC1

Note: It is strongly recommended that the value of TC1 be set to the minimum acceptable for the ICC. Large values of TC1 lead to very slow communication between the terminal and the ICC, and thus lengthy transaction times.

8.3.3.4 TD1

TD1 indicates whether any further interface bytes are to be transmitted and information concerning the protocol type(s) where:

- The m.s. nibble is used to indicate whether the characters TA2 to TD2 are present. These bits (b5–b8) are set to the logic one state to indicate the presence of TA2 to TD2 respectively.
- The l.s. nibble provides information concerning the protocol type(s) to be used for subsequent exchanges.

Basic responses: The ATR shall not contain TD1 if T=0 only is to be used, and protocol type T=0 shall be used as a default for all subsequent transmissions. The ATR shall contain TD1 = '81' if T=1 only is to be used, indicating that TD2 is present and that protocol type T=1 shall be used for all subsequent transmissions.

Terminal behaviour: The terminal shall accept an ATR containing TD1 with the m.s. nibble having any value (provided that the value returned correctly indicates and is consistent with the interface characters TA2 to TD2 actually returned), and the l.s. nibble having a value of '0' or '1'. The terminal shall reject an ATR containing other values of TD1.

The basic response coding of character TD1 is shown in Table 21:

	b8	b7	b6	b5	b4	b3	b2	b1
T=1	1	0	0	0	0	0	0	1

Table 21: Basic Response Coding of Character TD1

8.3.3.5 TA2

The presence or absence of TA2 indicates whether the ICC will operate in specific mode or negotiable mode respectively following the answer to reset. When present, TA2 conveys information regarding the specific mode of operation where:

- b8 indicates whether the ICC is capable of changing its mode of operation. It is capable of changing if b8 is set to 0, and unable to change if b8 is set to 1.
- b7–b6 are RFU (set to 00).
- b5 indicates whether the transmission parameters to be used following Answer to Reset are defined in the interface characters or are implicitly known by the terminal. The transmission parameters are defined by the interface characters if b5 is set to 0, or are implicitly known by the terminal if b5 is set to 1.
- l.s. nibble b4-b1 indicates the protocol to be used in specific mode.

Basic response: The ATR shall not contain TA2; the absence of TA2 indicates the negotiable mode of operation.

Terminal behaviour: The terminal shall accept an ATR containing TA2 provided that all the following conditions are met:

- The protocol indicated in the l.s. nibble is also the first indicated protocol in the ATR.
- b5 = 0
- The terminal is able to support the exact conditions indicated in the applicable interface characters and immediately uses those conditions.

Otherwise, the terminal shall reject an ATR containing TA2.

8.3.3.6 TB2

TB2 conveys PI2, which is used to determine the value of programming voltage P required by the ICC. When present it overrides the value indicated by PI1 returned in TB1.

Basic response: The ATR shall not contain TB2.

Terminal behaviour: The terminal shall reject an ATR containing TB2.

Note: Existing terminals may maintain V_{PP} in the idle state (see section 5.4.3).

8.3.3.7 TC2

TC2 is specific to protocol type T=0 and conveys the work waiting time integer (WI) that is used to determine the maximum interval between the leading edge of the start bit of any character sent by the ICC and the leading edge of the start bit of the previous character sent either by the ICC or the terminal (the work waiting time). The work waiting time is given by 960 x D x WI.

Basic response: The ATR shall not contain TC2 and a default value of WI = 10 shall be used during subsequent communication.

Terminal behaviour: The terminal shall:

- reject an ATR containing TC2 = '00'
- accept an ATR containing TC2 = '0A'
- reject an ATR containing TC2 having any other value unless it is able to support it.

8.3.3.8 TD2

TD2 indicates whether any further interface bytes are to be transmitted and the protocol type to be used for subsequent transmissions, where:

- The m.s. nibble is used to indicate whether the characters TA3 to TD3 are present. These bits (b5–b8) are set to the logic one state to indicate the presence of TA3 to TD3, respectively.
- The l.s. nibble indicates the protocol type to be used for subsequent exchanges. It shall take the value '1' as T=1 is to be used.

Basic responses: The ATR shall not contain TD2 if T=0 is to be used, and the protocol type T=0 shall be used as a default for all subsequent transmissions. The ATR shall contain TD2 = '31' if T=1 is to be used, indicating that TA3 and TB3 are present and that protocol type T=1 shall be used for all subsequent transmissions.

Terminal behaviour: The terminal shall accept an ATR containing TD2 with the m.s. nibble having any value (provided that the value returned correctly indicates and is consistent with the interface characters TA3 to TD3 actually returned), and the l.s. nibble having a value of '1' (or 'E' if the l.s. nibble of TD1 is '0'). The terminal shall reject an ATR containing other values of TD2.

The basic response coding of character TD2 is shown in Table 22:

	b8	b7	b6	b5	b4	b3	b2	b1
T=1	0	0	1	1	0	0	0	1

Table 22: Basic Response Coding of Character TD2

8.3.3.9 TA3

TA3 (if T=1 is indicated in TD2) returns the information field size integer for the ICC (IFSI), which determines the IFSC, and specifies the maximum length of the information field (INF) of blocks that can be received by the card. It represents the length of IFSC in bytes and may take any value between '01' and 'FE'. Values of '00' and 'FF' are reserved for future use.

Basic response: If T=1 is to be used, the ATR shall contain TA3 having a value in the range '10' to 'FE' indicating an initial IFSC in the range 16 to 254 bytes.

Terminal behaviour: The terminal shall accept an ATR not containing TA3 (provided that b5 of TD2 is set to 0), and shall continue the card session using a value of '20' for TA3. The terminal shall reject an ATR containing TA3 having a value in the range '00' to '0F' or a value of 'FF'.

The basic response coding of character TA3 is shown in Table 23:

	b8	b7	b6	b5	b4	b3	b2	b1
T=1	x	x	x	x	x	x	x	x
	'00' to '0F' and 'FF' not allowed							

Table 23: Basic Response Coding of Character TA3

8.3.3.10 TB3

TB3 (if T=1 is indicated in TD2) indicates the values of the CWI and the BWI used to compute the CWT and BWT respectively. The l.s. nibble (b1–b4) is used to indicate the value of CWI, whilst the m.s. nibble (b5–b8) is used to indicate the value of BWI.

Basic response: If T=1 is to be used, the ATR shall contain TB3 having the l.s. nibble in the range '0' to '5', and the m.s. nibble in the range '0' to '4', indicating values of 0 to 5 for CWI and 0 to 4 for BWI.

The basic response coding of character TB3 is shown in Table 24:

	b8	b7	b6	b5	b4	b3	b2	b1
T=1	0	x	x	x	0	y	y	y
	xxx is in the range 000 to 100 yyy is in the range 000 to 101							

Table 24: Basic Response Coding of Character TB3

Terminal behaviour: The terminal shall reject an ATR not containing TB3, or containing a TB3 indicating BWI greater than 4 and/or CWI greater than 5, or having a value such that $2^{CWI} \leq (N + 1)$. It shall accept an ATR containing a TB3 having any other value.

Note: N is the extra guardtime indicated in TC1. When using T=1, if TC1='FF', the value of N shall be taken as –1. Since the maximum value for CWI allowed by these specifications is 5, note that when T=1 is used, TC1 shall have a value in the range '00' to '1E' or a value of 'FF' in order to avoid a conflict between TC1 and TB3.

8.3.3.11 TC3

TC3 (if T=1 is indicated in TD2) indicates the type of block error detection code to be used. The type of code to be used is indicated in b1, and b2 to b8 are not used.

Basic response: The ATR shall not contain TC3, thus indicating longitudinal redundancy check (LRC) as the error code to be used.

Terminal behaviour: The terminal shall accept an ATR containing TC3 = '00'. It shall reject an ATR containing TC3 having any other value.

8.3.4 TCK - Check Character

TCK has a value that allows the integrity of the data sent in the ATR to be checked. The value of TCK is such that the exclusive-OR'ing of all bytes from T0 to TCK inclusive is null.

Basic responses: The ATR shall not contain TCK if T=0 only is to be used. In all other cases TCK shall be returned in the ATR.

Terminal behaviour: The terminal shall be able to evaluate TCK when appropriately returned. It shall accept an ICC returning an ATR not containing TCK if T=0 only is indicated. In all other cases, the terminal shall reject an ICC returning an ATR not containing TCK, or containing an incorrect TCK.

8.4 Terminal Behaviour during Answer to Reset

Following activation of the ICC contacts as described in section 6.1.2 the terminal shall initiate a cold reset as described in section 6.1.3.1. Subsequently the following shall apply:

- If the terminal rejects the ICC as described in section 8.3, it shall initiate the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the ATR.
- If the terminal rejects a cold ATR as described in section 8.3, it shall not immediately abort the card session but shall initiate a warm reset within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the cold ATR to the time that RST is set low. The terminal shall not initiate the warm reset until the T0 character has been received.
- If the terminal rejects a warm ATR as described in section 8.3, it shall initiate the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the warm ATR.
- The terminal shall be able to receive an ATR having a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 initial etus.
- The terminal shall be able to receive an ATR having maximum interval between two consecutive characters of 10,080 initial etus (9,600 + 480 initial etus). If a character is not received, the terminal shall abort the card session by initiating the deactivation sequence within 14,400 initial etus (9,600 + 4,800 initial etus) following the leading edge of the start bit of the last received character (the character from which timeout occurred).

- The terminal shall be able to receive an ATR having a duration of less than or equal to 20,160 initial etus. If the ATR (warm or cold) is not completed the terminal shall abort the card session by initiating the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) following the leading edge of the start bit of the TS character.
- If the terminal detects a parity error in a character returned in the ATR, it shall initiate the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the ATR.
- Upon receipt of a valid cold or warm reset complying with the timings described above, the terminal shall proceed with the card session using the returned parameters. It may continue the card session as soon as the last character of the valid ATR (as indicated by the bit map characters T0 and/or TDi) and TCK, if present, has been received. Before transmitting, it shall wait at least the guardtime applicable to the protocol to be used (16 etus for T=0, BGT for T=1) measured from the leading edge of the start bit of the last character of the valid ATR.

8.5 Answer to Reset - Flow at the Terminal

Figure 11 illustrates an example of the process of an ICC returning an ATR to the terminal and the checks performed by the terminal to ensure conformance to section 8.

Note 1: 'Case' is a process variable used to indicate whether a cold or warm reset is operative. Case = 1 for a cold reset, and Case = 2 for a warm reset.

Note 2: If the process aborts at this point, the ICC may be acceptable in this terminal by business agreement. The terminal should be primed to accept the ICC prior to insertion. Any subsequent processing is proprietary and beyond the scope of this specification.

Note 3: If the process aborts at this point, reset may be retried after removing the ICC from the terminal and taking corrective action as required. An appropriate message should be displayed on the terminal.

Note 4: A proprietary card session beyond the scope of this specification may be started at this point.

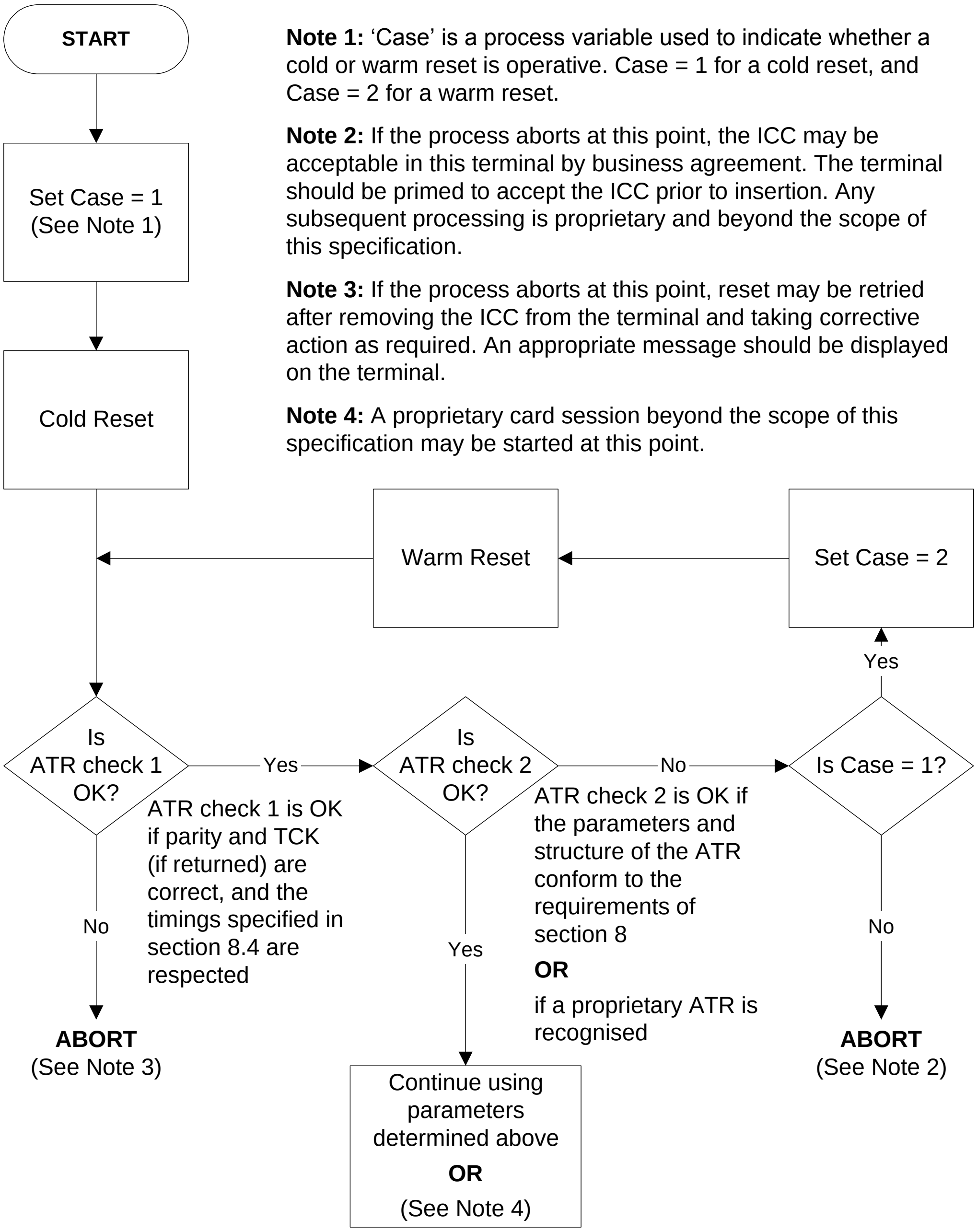


Figure 11: ATR - Example Flow at the Terminal

9 Transmission Protocols

This section defines the structure and processing of commands initiated by the terminal for transmission control and for specific control in asynchronous half duplex transmission protocols.

Two types of protocol are defined, character protocol (T=0) and block protocol (T=1). ICCs shall support either protocol T=0 or protocol T=1. Terminals shall support both protocol T=0 and T=1. The protocol to be used for subsequent communication between the ICC and terminal is indicated in TD1, and shall be T=0 or T=1. If TD1 is absent in the ATR, T=0 is assumed. The protocol indicated by the ICC applies immediately after the answer to reset, as there is no PTS procedure. Other parameters returned in the ATR and relevant to a specific protocol are defined in sections 9.2 through 9.4.

The protocols are defined according to the following layering model:

- Physical layer, which describes the exchanges of bits and is common to both protocols.
- Data link layer, which includes the following sub-definitions:
    - Character frame, defining the exchanges of characters common to both protocols.
    - Character protocol T=0, defining the exchange of characters specific to T=0.
    - Error detection and correction specific to T=0.
    - Block protocol T=1, defining the exchanges of blocks specific to T=1.
    - Error detection and correction specific to T=1.
- Transport layer, which defines the transmission of application-oriented messages specific to each protocol.
- Application layer, which defines the exchange of messages according to an application protocol that is common to both transmission protocols.

9.1 Physical Layer

Both protocols T=0 and T=1 use the physical layer and character frame as defined in section 7.

9.2 Data Link Layer

This section describes the timing, specific options, and error handling for protocols T=0 and T=1.

9.2.1 Character Frame

The character frame as defined in section 7.2 applies to all messages exchanged between the ICC and the terminal.

9.2.2 Character Protocol T=0

9.2.2.1 Specific Options - Character Timing for T=0

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the terminal to the ICC shall be between 12 and 266 etus as determined by the value of TC1 returned at the answer to reset (see sections 8.2 and 8.3). This interval may be less than the minimum interval of 16 etus allowed between two characters sent in opposite directions. If the value returned in TC1 is N, the ICC shall be able to correctly interpret characters sent by the terminal with a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 + N etus.

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the ICC to the terminal shall be 12 etus. The terminal shall be able to correctly interpret characters sent by the ICC with a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 etus.

The maximum interval between the leading edge of the start bit of any character sent by the ICC and the leading edge of the start bit of the previous character sent either by the ICC or the terminal (the Work Waiting Time, or WWT) shall not exceed 960 x D x WI etus (D and WI are returned in TA1 and TC2, respectively).

The terminal shall be able to correctly interpret a character sent by the ICC with a maximum interval between the leading edge of the start bit of the character and the leading edge of the start bit of the previous character sent either by the ICC or the terminal of {WWT + (D x 480)} etus. If no character is received, the terminal shall initiate the deactivation sequence within {WWT + (D x 9600)} etus following the leading edge of the start bit of the character from which the timeout occurred.

For the ICC or terminal, the minimum interval between the leading edges of the start bits of the last character received and the first character sent in the opposite direction shall be 16 etus. The ICC or terminal shall be able to correctly interpret a character received within 15 etus timed from the leading edge of the start bit of the last character sent to the leading edge of the start bit of the received character. These timings do not apply during character repetition.

9.2.2.2 Command Header

A command is always initiated by the terminal application layer (TAL) which sends an instruction via the TTL to the ICC in the form of a five byte header called the command header. The command header is comprised of five consecutive bytes, CLA, INS, P1, P2, and P3, where:

- CLA is the command class.
- INS is the instruction code.
- P1 and P2 contain additional instruction-specific parameters.
- P3 indicates either the length of data to be sent with the command to the ICC, or the maximum length of data expected in the response from the ICC, depending on the coding of INS.

These bytes together with any data to be sent with the command constitute the command transport protocol data unit (C-TPDU) for T=0. The mapping of the command application protocol data unit (C-APDU) onto the C-TPDU is described in section 9.3.

The TTL transmits the five-byte header to the ICC and waits for a procedure byte.

9.2.2.3 Command Processing

Following reception of a command header by the ICC, the ICC shall return a procedure byte or status bytes SW1 SW2 (hereafter referred to as 'status') to the TTL. Both the TTL and ICC shall know implicitly at any point during exchange of commands and data between the TTL and the ICC what the direction of data flow is and whether it is the TTL or the ICC that is driving the I/O line.

9.2.2.3.1 Procedure Byte

The procedure byte indicates to the TTL what action it shall take next. The coding of the byte and the action that shall be taken by the TTL is shown in Table 25.

Procedure Byte Value	Action
Equal to INS byte	All remaining data bytes shall be transferred by the TTL, or the TTL shall be ready to receive all remaining data bytes from the ICC
Equal to complement of INS byte ($\overline{\text{INS}}$)	The next data byte shall be transferred by the TTL, or the TTL shall be ready to receive the next data byte from the ICC
'60'	The TTL shall provide additional work waiting time as defined in this section
'61'	The TTL shall wait for a second procedure byte then send a GET RESPONSE command header to the ICC with a maximum length of 'xx', where 'xx' is the value of the second procedure byte
'6C'	The TTL shall wait for a second procedure byte then immediately resend the previous command header to the ICC using a length of 'xx', where 'xx' is the value of the second procedure byte

Table 25: Terminal Response to Procedure Byte

In all cases, after the action has taken place the TTL shall wait for a further procedure byte or status.

9.2.2.3.2 Status Bytes

The status bytes indicate to the TTL that command processing by the ICC is complete. The meaning of the status bytes is related to the command being processed and is defined in section 11 and in Book 3. The coding of the first status byte and the action that shall be taken by the TTL are shown in Table 26.

First Status Byte Value	Action
'6x' or '9x' (except '60', '61' and '6C') - status byte SW1	TTL shall wait for a further status byte (status byte SW2)

Table 26: Status Byte Coding

Following receipt of the second status byte, the TTL shall return the status bytes (together with any appropriate data - see section 9.3.1) to the TAL in the response APDU (R-APDU) and await a further C-APDU.

9.2.2.4 Transportation of C-APDUs

A C-APDU containing only command data to be sent to the ICC, or only expecting data in response from the ICC (cases 2 and 3 in section 9.4), is mapped without change onto a T=0 C-TPDU. A C-APDU that contains and expects no data, or a C-APDU that requires data transmission to and from the ICC (cases 1 and 4 in section 9.4) is translated according to the rules defined in section 9.3 for transportation by a C-TPDU for T=0.

9.2.3 Error Detection and Correction for T=0

This procedure is mandatory for T=0 but does not apply during the answer to reset.

If a character is received with a parity error, the receiver shall indicate an error by setting the I/O line to state L at time (10.5 ± 0.2) etus following the leading edge of the start bit of the character for a minimum of 1 etu and a maximum of 2 etus.

The transmitter shall test the I/O line (11 ± 0.2) etus after the leading edge of the start bit of a character was sent, and assumes that the character was correctly received if the I/O line is in state H.

If the transmitter detects an error, it shall repeat the disputed character after a delay of at least 2 etus following detection of the error. The transmitter shall repeat the same disputed character a maximum of three more times, and shall therefore send a character up to a maximum of five times in total (the original transmission followed by the first repeat and then three further repeats) in an attempt to achieve error free transmission.

If the last repetition is unsuccessful, the terminal shall initiate the deactivation sequence within (D x 960) etus following reception of the leading edge of the start bit of the invalid character (if it is the receiver), or within (D x 960) etus following detection of the signalling of the parity error by the ICC (if it is the transmitter).

Character repetition timing is illustrated in Figure 12.

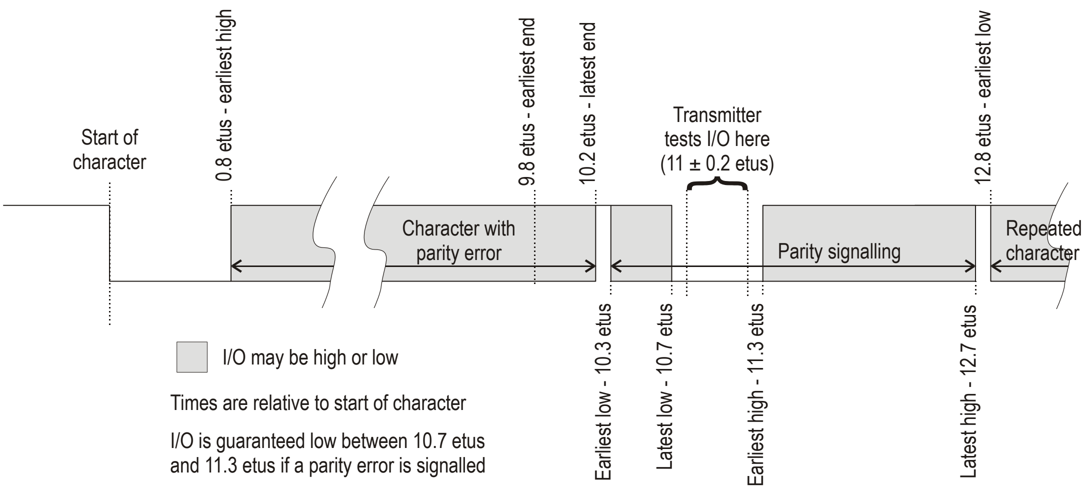


Figure 12: Character Repetition Timing

When awaiting a procedure byte or status byte, if the byte returned by the ICC has a value other than specified in sections 9.2.2.3.1 and 9.2.2.3.2, the terminal shall initiate the deactivation sequence within 9,600 etus following the leading edge of the start bit of the (invalid) byte received.

9.2.4 Block Protocol T=1

The protocol consists of blocks transmitted between the TAL and the ICC to convey command and R-APDUs and transmission control information (for example, acknowledgment).

The data link layer block frame structure, specific options of the protocol, and protocol operations (including error handling) are defined below.

9.2.4.1 Block Frame Structure

The character frame as defined in section 7.2 applies.

The block is structured as illustrated in Table 27:

Prologue Field - Mandatory -			Information Field - Optional -	Epilogue Field - Mandatory -
Node Address (NAD)	**Protocol Control Byte (PCB)**	**Length (LEN)**	**APDU or Control Information (INF)**	**Error Detection Code (EDC)**
1 byte	1 byte	1 byte	0–254 bytes	1 byte

Table 27: Structure of a Block

9.2.4.1.1 Prologue Field

The Prologue Field is mandatory and consists of three mandatory bytes:

- Node address (NAD) to identify source and intended destination of the block and to provide VPP state control
- Protocol control byte (PCB) to control data transmission
- Length (LEN) of the optional information field

Node Address

NAD is mandatory. Bits b1–b3 of NAD indicate the source node address (SAD) of the block, whilst bits b5–b7 indicate the intended destination node address (DAD) of the block. Bits b4 and b8 [7] are unused and shall be set to 0.

These specifications do not support node addressing. The first block sent by the terminal following the ATR and all following blocks transmitted by either the terminal or ICC shall have the NAD = '00'.

If during the card session the terminal or ICC receives a block with a NAD ≠ '00', it may treat the block as invalid. In this event, it shall apply the error detection and correction techniques described in section 9.2.5.

Protocol Control Byte

The PCB is mandatory, and indicates the type of block. There are three types of blocks, as defined in Table 28:

Type of Block	Short Name	Purpose
Information block	I-block	to convey APDUs
Receive-ready block	R-block	to convey acknowledgments (ACK or NAK)
Supervisory block	S-block	to exchange control information

Table 28: Types of Blocks

[7] Defined in ISO/IEC 7816-3:1997 as VPP control for class A. A value of 0 indicates that VPP shall be maintained in the idle state.

The coding of the PCB depends on its type and is defined in Table 29, Table 30, and Table 31.

b8	0
b7	Sequence number
b6	Chaining (more data)
b5–b1	Reserved for future use (RFU)

Table 29: Coding of the PCB of an I-block

b8	1
b7	0
b6	0
b5	Sequence number
b4–b1	0 = Error free 1 = EDC and/or parity error 2 = Other error(s) Other values RFU

Table 30: Coding of the PCB of a R-block

b8	1
b7	1
b6	0 = Request 1 = Response
b5–b1	0 = Resynchronisation request 1 = Information field size request 2 = Abort request 3 = Extension of BWT request 4 = VPP error [8] Other values RFU

Table 31: Coding of the PCB of a S-block

[8] Not used by ICCs and terminals conforming to this specification.

Length

The Length (LEN) is mandatory, and indicates the length of the INF part of the block; it may range from 0 to 254 depending on the type of block.

Note: This specification does not support I-blocks with LEN = 0.

9.2.4.1.2 Information Field

The Information Field (INF) is conditional.

- When present in an I-block, it conveys application data.
- When present in a S-block, it conveys control information.
- A R-block shall not contain an INF.

9.2.4.1.3 Epilogue Field

The Epilogue Field is mandatory, and contains the EDC of the transmitted block. A block is invalid when a parity error and/or an EDC error occurs. This specification supports only the LRC as EDC. The LRC is one byte in length and is calculated as the exclusive-OR of all the bytes starting with the NAD and including the last byte of INF, if present.

Note: TCi (i > 2), which indicates the type of error detection code to be used, is not returned by the ICC in the ATR. The normal default of the LRC is thus used for the EDC.

9.2.4.1.4 Block Numbering

I-blocks are numbered using a modulo-2 number coded on one bit. The numbering system is maintained independently at the ICC and the terminal as senders. The value of the number starts with zero for the first I-block sent after the answer to reset by a sender and is incremented by one after sending each I-block. The number is reset to zero by the sender after resynchronisation.

R-blocks are numbered using a modulo-2 number coded on one bit. A R-block is used to acknowledge a chained I-block or to request retransmission of an invalid block. In either case, b5 of the PCB of the R-block carries the sequence number of the next I-block its sender expects to receive.

A S-block carries no number.

9.2.4.2 Specific Options

This section defines the information field sizes and timings to be used with protocol type T=1.

9.2.4.2.1 Information Field Sizes

The IFSC is the maximum length of the information field of blocks that can be received by the ICC, and is defined as follows. At the answer to reset the IFSI is returned by the ICC in TA3 indicating the size of the IFSC that can be accommodated by the ICC. IFSI may take values in the range '10' to 'FE' that indicate values for IFSC in the range 16 to 254 bytes. The maximum block size that can be received by the ICC is therefore (IFSC + 3 + 1) bytes including the prologue and epilogue fields. The size established during the answer to reset shall be used throughout the rest of the card session or until a new value is negotiated by the ICC by sending a S(IFS request) block to the terminal.

The information field size for the terminal (IFSD) is the maximum length of the information field of blocks that can be received by the terminal. The initial size immediately following the answer to reset shall be 254 bytes, and this size shall be used throughout the rest of the card session.

9.2.4.2.2 Timing for T=1

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the terminal to the ICC shall be between 11 and 42 etus as indicated by the value of TC1 returned at the answer to reset (see sections 8.2 and 8.3). If the value returned in TC1 is N, the ICC shall be able to correctly interpret characters sent by the terminal with a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 + N etus.

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the ICC to the terminal shall be 11 etus. The terminal shall be able to correctly interpret characters sent by the ICC with a minimum interval between the leading edges of the start bits of two consecutive characters of 10.8 etus.

The maximum interval between the leading edges of the start bits of two consecutive characters sent in the same block (the character waiting time, CWT) shall not exceed (2^{CWI} + 11) etus. The character waiting time integer, CWI shall have a value of 0 to 5 as described in section 8.3.3.10, and thus CWT lies in the range 12 to 43 etus. The receiver shall be able to correctly interpret a character having a maximum interval between the leading edge of the start bit of the character and the leading edge of the start bit of the previous character of (CWT + 4) etus.

The maximum interval between the leading edge of the start bit of the last character that gave the right to send to the ICC and the leading edge of the start bit of the first character sent by the ICC (the block waiting time, BWT) shall not exceed {(2^{BWI} x 960) + 11} etus. The block waiting time integer, BWI shall have a value in the range 0 to 4 as described in section 8.3.3.10, and thus BWT lies in the range 971 to 15,371 etus for a D of 1.

The terminal shall be able to correctly interpret the first character of a block sent by the ICC following a time BWT + (D x 960) etus.

For the ICC or terminal, the minimum interval between the leading edges of the start bits of the last received character and the first character sent in the opposite direction (the block guard time, BGT) shall be 22 etus. The ICC or terminal shall be able to correctly interpret a character received within 21 etus timed from the leading edge of the start bit of the last character that it sent to the leading edge of the start bit of the received character.

Note: In general, for values of FI and DI other than 1, BWT is calculated using the formula:

$$\mathrm{BWT} = \left(\left(2^{\mathrm{BWI}} \times 960 \times \frac{372\mathrm{D}}{\mathrm{F}}\right) + 11\right) \mathrm{etu}$$

9.2.4.3 Error Free Operation

The protocol rules for error free operation are as follows:

1. The first block transmitted after the answer to reset shall be sent by the terminal to the ICC and shall be a S(IFS request) block with PCB = 'C1' and with IFSD = 254 (value indicated in the single byte INF field). No further S(IFS request) blocks shall be sent by the terminal during the card session.
2. The ICC shall return a S(IFS response) block to the terminal acknowledging the change to the size of the IFSD. The PCB of the S(IFS response) block sent in response shall have the value 'E1', and the INF field shall have the same value as the INF field of the block requesting the change.
3. If the ICC wishes to change the size of the IFSC from the initial value indicated at the answer to reset, it shall send a S(IFS request) block to the terminal. The PCB of the S(IFS request) block shall have the value 'C1' indicating a request to change the IFSC. The INF field shall contain a byte the value of which indicates the size in bytes of the requested new IFSC. This byte shall have a value in the range '10' to 'FE'. The terminal shall return a S(IFS response) block to the ICC acknowledging the change to the size of the IFSC. The PCB of the S(IFS response) block sent in response shall have the value 'E1', and the INF field shall have the same value as the INF field of the block requesting the change.
4. During the card session, only blocks as defined in this section shall be exchanged. The half duplex block protocol consists of blocks transmitted alternately by the terminal and the ICC. When the sender has transmitted a complete block, the sender switches to the receiving state.
5. When the receiver has received the number of characters in accordance with the value of LEN and the EDC, the receiver gains the right to send.
6. The ICC shall acknowledge an I-block transmitted by the terminal. The acknowledgment is indicated in the sequence number of the I-block, or the R-block if chaining is in use (except the last block of the chain), that the ICC returns to the terminal.
7. A non-chained I-block or the last I-block of a chain is considered by the sender to be acknowledged when the sequence number of the I-block received in response differs from the sequence number of the previously received I-block. If no I-block was previously received, the sequence number of the I-block sent in response shall be 0.
8. When a R-block is received, b5 shall be evaluated. The receiver is not required to evaluate bits b4–b1 of the PCB. Optional evaluation of bits b4–b1 shall not result in any action which contradicts the protocol rules defined in this specification

9. During chaining, a chained I-block (except the last I-block of a chain) is considered by the sender to be acknowledged when the sequence number of the R-block sent in response differs from the sequence number of the I-block being acknowledged.
10. If the ICC requires more than the BWT to process the previously received I-block, it shall send a waiting time extension request S(WTX request) block, where the INF contains the one-byte binary integer multiplier of the BWT value requested. The terminal shall acknowledge by sending a waiting time extension response S(WTX response) block with the same value in the INF. The time allocated (which is the time requested in the S(WTX request) block, and which replaces BWT for this instance only) starts at the leading edge of the last character of the S(WTX response) block. After the ICC responds, BWT is again used as the time allowed for the ICC to process the I-block.
11. S-blocks are used only in pairs. A S(request) block is always followed by a S(response) block.

When synchronisation as outlined above is lost, the procedure described in section 9.2.5 shall apply.

9.2.4.4 Chaining

When the sender has to transmit data of length greater than IFSC or IFSD bytes, it shall divide it into several consecutive I-blocks. The transmission of these multiple I-blocks is achieved using the chaining function described below.

The chaining of I-blocks is controlled by b6 of the PCB. The coding of b6 is as follows:

- b6 = 0 Last block of the chain
- b6 = 1 Subsequent block follows

Any I-block with b6 = 1 shall be acknowledged by a R-block according to section 9.2.4.1.

The last block of a chain sent by the terminal shall be acknowledged by either an I-block if correctly received, or a R-block if incorrectly received. The last block of a chain sent by the ICC shall be acknowledged by a R-block if incorrectly received; if correctly received, the terminal will only transmit further I-blocks if another command is to be processed.

9.2.4.4.1 Rules for Chaining

The TTL shall support chaining for both transmitted and received blocks. The ICC may optionally chain blocks sent to the terminal. Chaining is only possible in one direction at a time. The following rules for chaining apply:

- When the terminal is the receiver, the terminal shall accept a sequence of chained I-blocks sent from the ICC of length ≤ IFSD bytes per block.
- When the ICC is the receiver, the ICC shall accept a sequence of chained I-blocks sent from the terminal all having length LEN = IFSC except the last block, whose length may be in the range 1 to IFSC bytes inclusive.
- When the ICC is the receiver, if an I-block sent by the terminal has length > IFSC, the ICC shall reject it using a R-block.
- If the ICC as sender chains blocks sent to the terminal it shall send I-blocks of length ≤IFSD bytes per block.
- When the terminal is the sender, all I-blocks of a chain sent to the ICC shall have LEN = IFSC bytes except the last, which shall have a length in the range 1 to IFSC bytes inclusive.
- During chaining, the ICC shall not attempt to negotiate a new IFSC by sending a S(IFSC request) block to the terminal.

9.2.4.4.2 Construction of Chained Blocks

C-APDUs are transported from the TTL to the ICC in the INF field of I-blocks (see section 9.3.2). If a C-APDU is too large to fit in one block, it is chained over several as illustrated in Figure 13.

Block(1)	CLA INS P1 P2	Lc	Data Data
Block(2)	Data Data Data		
Block(n)	Data Data	Le	

Figure 13: Chaining C-APDU

The data and status returned by the ICC may optionally be chained over several I-blocks as illustrated in Figure 14.

Block(1)	Data Data Data	
Block(2)	Data Data Data	
Block(n)	Data Data	SW1 SW2

Figure 14: Chaining I-Blocks

Note: The above examples are for a case 4 command and show only the INF fields of the chained blocks. Each block also has a prologue and epilogue field. All chained blocks shall contain an INF field having a length in the range 1 to IFSD bytes if the ICC is the sender, or IFSC bytes during chaining and 1 to IFSC bytes in the last block of the chain if the terminal is the sender.

9.2.5 Error Detection and Correction for T=1

The following errors shall be detected by the TTL:

- Transmission error including parity error, EDC error, and BWT time-out.
- Loss of synchronisation assumed when the actual block size is inconsistent with the size indicated by the value in LEN.
- Protocol error (infringement of the rules of the protocol).
- Abort request for a chain of blocks.

If a parity error is detected, character repetition shall not be implemented when using T=1.

Error recovery is attempted in the following manner.

The TTL shall attempt error recovery by trying the following techniques in the order shown.

1. Retransmission of blocks

Deactivation of the ICC contacts

The ICC shall attempt error recovery by trying retransmission of blocks.

If a block is retransmitted, the retransmitted block shall be identical to the originally transmitted block.

Note: In some terminals the TTL may not be solely responsible for error handling. Where 'TTL' is used it includes any functionality present in the terminal as applicable.

The following types of block are considered invalid:

- Blocks containing transmission errors, i.e. parity/EDC incorrect
- Blocks that have formatting errors, i.e. blocks constructed incorrectly by the sender (syntax error)
- Blocks that are unexpected according to the rules of the protocol at any particular point in an exchange, for example, a S(Response) block received in response to an I-block.

A R-block received indicating an error condition is not an invalid block.

9.2.5.1 Protocol Rules for Error Handling

The following rules apply for error handling and correction. In each case where a R-block is sent, the error coding bits b4–b1 may optionally be evaluated, but shall not result in any action which contradicts the protocol rules defined in this specification.

1. If the first block received by the ICC after the answer to reset is invalid, it shall return a R-block to the TTL with b5 = 0 and NAD = 0.
2. If there is no response from the ICC to a block sent by the TTL, the terminal shall:

   (a) initiate the deactivation sequence

   OR

   (b) if the block not responded to was an I-block, R-block, or S(Response) block, transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4

   OR

   (c) if the block not responded to was a S(Request) block, retransmit the S(Request) block

   between {BWT + (D x 960)} and {BWT + (D x 4,800)} etus (or between {WTX + (n x D x 960)} and {WTX + (n x D x 4,800)} etus if a waiting time extension has been negotiated) from the leading edge of the start bit of the last character of the block to which there was no response.
3. If during reception of a block by the terminal an expected character is not received, the terminal shall:

   (a) initiate the deactivation sequence

   OR

   (b) if the block not responded to was an I-block, R-block, or S(Response) block, transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4

   OR

   (c) if the block not responded to was a S(Request) block, retransmit the S(Request) block

   within (CWT + 4) and (CWT + 4,800) etus from the leading edge of the start bit of the last character received.

4. If an invalid block is received in response to an I-block, the sender shall transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4.
5. If an invalid block is received in response to a R-block, the sender shall retransmit the R-block.
6. If a correct S(... response) block is not received in response to a S(... request) block, the sender shall retransmit the S(... request) block.
7. If an invalid block is received in response to a S(... response) block, the sender shall transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4.
8. If the TTL has sent three consecutive blocks of any type without obtaining a valid response, it shall initiate the deactivation sequence within {BWT + (D x 14,400)} etus following the leading edge of the start bit of the last character of the block requesting retransmission.
9. Note: Resynchronisation is not required by this specification. If for proprietary reasons the terminal supports resynchronisation, it may attempt this by sending a S(RESYNCH request) block, then behave as specified in ISO/IEC 7816-3.
10. If the ICC has sent a block a maximum of twice in succession (the original transmission followed by one repeat) without obtaining a valid response, it shall remain in reception mode.
11. A S(ABORT request) shall not be sent by the terminal. If the terminal receives a S(ABORT request) from the ICC, it shall terminate the card session by initiating the deactivation sequence within (D x 9,600) etus following reception of the leading edge of the start bit of the last character of the S(ABORT request) block.

    Note: Transaction abortion is not required by this specification. If an ICC or terminal supports abortion for proprietary reasons, it may issue a S(ABORT request), but note that it will receive an invalid response if the receiver does not support abortion. In this event, the card session will be terminated according to the rules above. If a terminal optionally supporting abortion receives a S(ABORT request) from an ICC, it may return a S(ABORT response) rather than terminating the card session.

9.3 Terminal Transport Layer (TTL)

This section describes the mechanism by which command and response APDUs are transported between the terminal and the ICC. APDUs are command or response messages, and since both command and response messages may contain data, the TTL shall be capable of managing the four cases defined in section 9.4. The construction of C-APDUs and R-APDUs are described in sections 9.4.1 and 9.4.2, respectively.

The C-APDU is passed from the TAL to the TTL where it is mapped in a manner appropriate to the transmission protocol to be used before being sent to the ICC. Following processing of the command by the ICC, data (if present) and status are returned by the ICC to the TTL, which maps it onto the R-APDU.

9.3.1 Transport of APDUs by T=0

This section describes the mapping of C-APDUs and R-APDUs, the mechanism for exchange of data between the TTL and the ICC, and the use of the GET RESPONSE command for retrieval of data from the ICC when case 2 or 4 commands are used.

9.3.1.1 Mapping of C-APDUs and R-APDUs and Data Exchange

The mapping of the C-APDU onto the T=0 command header is dependent upon the case of the command. The mapping of the data (if present) and status returned by the ICC onto the R-APDU is dependent upon the length of the data returned and the meaning of the status bytes.

Procedure bytes '61xx' and '6Cxx' are returned by the ICC to control exchanges between the TTL and the ICC, and should never be returned to the TAL. Command processing in the ICC is not complete if it has returned procedure bytes '61xx' or '6Cxx'.

Note: For proprietary reasons, the TTL may in addition be capable of accepting data from the ICC without using the '61' and '6C' procedure bytes. Such functionality is not required and is beyond the scope of these specifications.

Normal status on completion of processing a command is indicated if the ICC returns status bytes SW1 SW2 = '9000' to the TTL. The TTL shall discontinue processing of a command (i.e. pass the R-APDU to the TAL and wait for a further C-APDU from the TAL) on receipt of any other status (but not on receipt of procedure bytes '61xx' and '6Cxx') from the ICC. (For case 4 commands only, immediately following successful transmission of command data to the ICC, the TTL shall continue processing the command if warning status bytes ('62xx' or '63xx') or application related status bytes ('9xxx' except '9000') are received.)

The following descriptions of the mapping of data and status returned by the ICC onto the R-APDU are for information, and apply only after the ICC has completed processing of the command, successfully or otherwise, and all data (if present) has been returned by the ICC under the control of '61xx' and '6Cxx' procedure bytes. Detailed use of the INS, $\overline{\text{INS}}$, and '60' procedure bytes is not described.

The status returned by the ICC shall relate to the most recently received command; where a GET RESPONSE command is used to complete the processing of a case 2 or case 4 command, any status returned by the ICC after receipt of the GET RESPONSE command shall relate to the GET RESPONSE command, not to the case 2 or case 4 command which it completes.

9.3.1.1.1 Case 1

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and P3 of the T=0 command header is set to '00'.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header the ICC, under normal or abnormal processing, shall return status to the TTL.

   (The ICC shall analyse the T=0 command header to determine whether it is processing a case 1 command or a case 2 command requesting all data up to the maximum length available.)
3. On receipt of status from the ICC, the TTL shall discontinue processing of the command.

See Annex A1 for details of the exchanges between the TTL and the ICC.

The status returned to the TTL from the ICC after completion of processing of the command is mapped onto the mandatory trailer of the R-APDU without change.

9.3.1.1.2 Case 2

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and length byte 'Le' from the conditional body of the C-APDU is mapped onto P3 of the T=0 command header. READ RECORD commands issued during application selection and all case 2 commands issued according to Book 3 shall have Le = '00'.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header:
   (a) under normal processing, the ICC shall return data and status to the TTL, using procedure bytes '6Cxx' (and if required, procedure bytes '61xx') to control the return of data

   OR

   (b) under abnormal processing, the ICC shall return status only to the TTL.
3. On receipt of the data (if present) and status from the ICC, the TTL shall discontinue processing the command.

See Annex A2 and Annex A5, for details of the exchanges between the TTL and the ICC, including use of the '61xx' and '6Cxx' procedure bytes.

The data (if present) and status returned to the TTL from the ICC after completion of processing of the command, or the status returned by the ICC that caused the TTL to discontinue processing of the command, are mapped onto the R-APDU as follows:

- The data returned (if present) is mapped onto the conditional body of the R-APDU. If no data is returned, the conditional body of the R-APDU is left empty.
- The status returned is mapped onto the mandatory trailer of the R-APDU without change.

9.3.1.1.3 Case 3

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and length byte 'Lc' from the conditional body of the C-APDU is mapped onto P3 of the T=0 command header.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header:
    (a) If the ICC returns a procedure byte, the TTL shall send the data portion of the conditional body of the C-APDU to the ICC under the control of procedure bytes returned by the ICC.

    OR

    (b) If the ICC returns status, the TTL shall discontinue processing of the command.
3. If processing was not discontinued in step 2(b), the ICC shall return status following receipt of the conditional body of the C-APDU and completion of processing the command.
4. On receipt of status from the ICC, the TTL shall discontinue processing the command.

See Annex A3, for details of the exchanges between the TTL and the ICC.

The status returned to the TTL from the ICC after completion of processing of the command, or the status returned by the ICC that caused the TTL to discontinue processing of the command, is mapped onto the R-APDU without change.

9.3.1.1.4 Case 4

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and length byte 'Lc' from the conditional body of the C-APDU is mapped onto P3 of the T=0 command header. SELECT commands issued during application selection and all case 4 commands issued according to Book 3 shall have Le = '00'.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header:
   (a) If the ICC returns a procedure byte, the TTL shall send the data portion of the conditional body of the C-APDU to the ICC under the control of procedure bytes returned by the ICC.

   OR

   (b) If the ICC returns status, the TTL shall discontinue processing of the command.
3. If processing was not discontinued in step 2(b), following receipt of the conditional body of the C-APDU:
   (a) under normal processing, the ICC shall return procedure bytes '61xx' to the TTL requesting the TTL to issue a GET RESPONSE command to retrieve the data from the ICC.

   OR

   (b) under abnormal processing, the ICC shall return status only to the TTL.
4. On receipt of the procedure bytes or status returned in step 3:
   (a) If the ICC returned '61xx' procedure bytes as in step 3(a), the TTL shall send a GET RESPONSE command header to the ICC with P3 set to a value less than or equal to the value contained in the 'xx' byte of '61xx' procedure bytes.

   OR

   (b) If the ICC returned status as in step 3(b) that indicates a warning ('62xx' or '63xx'), or which is application related ('9xxx' but not '9000'), the TTL shall send a GET RESPONSE command with Le='00'.

   OR

   (c) If the ICC returned status as in step 3(b) other than that described in step 4(b), the TTL shall discontinue processing of the command.
5. If processing was not discontinued in step 4(c), the GET RESPONSE command shall be processed according to the rules for case 2 commands in section 9.3.1.1.2.

See Annex A4 and Annex A6, for details of the exchanges between the TTL and the ICC, including use of the '61xx' and '6Cxx' procedure bytes.

The data (if present) and status returned to the TTL from the ICC after completion of processing of the command, or the status returned by the ICC that caused the TTL to discontinue processing of the command, are mapped onto the R-APDU as follows:

- The data returned (if present) is mapped onto the conditional body of the R-APDU. If no data is returned, the conditional body of the R-APDU is left empty.
- The first status returned during processing of the entire case 4 command, including the GET RESPONSE command if used, is mapped onto the mandatory trailer of the R-APDU without change.

9.3.1.2 Use of Procedure Bytes '61xx' and '6Cxx'

The ICC returns procedure bytes '61xx' and '6Cxx' to the TTL to indicate to it the manner in which it should retrieve the data requested by the command currently being processed. These procedure bytes are only used when processing case 2 and 4 commands.

Procedure bytes '61xx' instruct the TTL to issue a GET RESPONSE command to the ICC. P3 of the GET RESPONSE command header is set to ≤ 'xx'.

Procedure bytes '6Cxx' instruct the TTL to immediately resend the previous command header setting P3 = 'xx'.

Usage of these procedure bytes during error free processing with case 2 and 4 commands is as follows. In the case of an error, the ICC may return status indicating error or warning conditions instead of the '61xx' or '6Cxx' response.

9.3.1.2.1 Case 2 Commands

1. If the ICC receives a case 2 command header and Le = '00' or Le > Licc, it shall return:

   (a) procedure bytes '6C Licc' instructing the TTL to immediately resend the command header with P3 = Licc

   OR

   (b) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

   Note: If Le = '00' and the ICC has 256 bytes of data to return, it should proceed as defined in the following rule for Le = Licc.

2. If the ICC receives a case 2 command header and Le = Licc, it shall return:

   (a) data of length Le (= Licc) under the control of the INS, $\overline{\text{INS}}$, or '60' procedure bytes followed by the associated status

   OR

   (b) procedure bytes '61xx' instructing the TTL to issue a GET RESPONSE command with a maximum length of 'xx'

   OR

   (c) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

3. If the ICC receives a case 2 command header and Le < Licc, it shall return:

   (a) data of length Le under the control of the INS, $\overline{\text{INS}}$, or '60' procedure bytes followed by procedure bytes '61xx' instructing the TTL to issue a GET RESPONSE command with a maximum length of 'xx'

   OR

   (b) procedure bytes '6C Licc' instructing the TTL to immediately resend the command header with P3 = Licc

   OR

   (c) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

3(b) above is not a valid response by the ICC to a GET RESPONSE command.

9.3.1.2.2 Case 4 Commands

1. If the ICC receives a case 4 command, after processing the data sent with the C-APDU, it shall return:

   (a) procedure bytes '61 xx' instructing the TTL to issue a GET RESPONSE command with a maximum length of 'xx'

   OR

   (b) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

The GET RESPONSE command so issued is then treated as described in section 9.3.1.2.1 for case 2 commands.

9.3.1.3 GET RESPONSE Command

The GET RESPONSE command is issued by the TTL to obtain available data from the ICC when processing case 2 and 4 commands. It is employed only when the T=0 protocol type is in use.

The structure of the command message is shown in Table 32:

CLA	'00'
INS	'C0'
P1	'00'
P2	'00'
Le	Maximum length of data expected

Table 32: Structure of Command Message

Following normal processing, the ICC returns status bytes SW1 SW2 = '9000' and Licc bytes of data.

In the event that an error condition occurs, the coding of the error status bytes (SW1 SW2) is shown in Table 33:

SW1	SW2	Meaning
'62'	'81'	Part of returned data may be corrupted
'67'	'00'	Length field incorrect
'6A'	'86'	P1 P2 ≠ '00'
'6F'	'00'	No precise diagnosis

Table 33: GET RESPONSE Error Conditions

9.3.2 Transportation of APDUs by T=1

The C-APDU is sent from the TAL to the TTL. The TTL maps the C-APDU onto the INF field of an I-block without change, and sends the I-block to the ICC.

Response data (if present) and status are returned from the ICC to the TTL in the INF field of an I-block. If The ICC returns status indicating normal processing ('61xx'), a warning ('62xx' or '63xx'), which is application related ('9xxx'), or is '9000', it shall also return data (if available) associated with processing of the command.

No data shall be returned with any other status. The contents of the INF field of the I-block are mapped onto the R-APDU without change and returned to the TAL.

Note: C-APDUs and response data/status may be chained over the INF fields of multiple blocks if required.

9.4 Application Layer

The application protocol consists of an ordered set of exchanges between the TAL and the TTL. Each step in an application layer exchange consists of a command-response pair, where the TAL sends a command to the ICC via the TTL, and the ICC processes it and sends a response via the TTL to the TAL. Each specific command has a specific response. An APDU is defined as a command message or a response message.

Both command and response messages may contain data. Thus, four cases shall be managed by the transmission protocols via the TTL, as shown in Table 34:

Case	Command Data	Response Data
1	Absent	Absent
2	Absent	Present
3	Present	Absent
4	Present	Present

Table 34: Definition of Cases for Data in APDUs

Note: When secure messaging is used only case 3 and case 4 commands exist since data (as a minimum, the MAC) is always sent to the ICC. When using secure messaging, case 1 commands will become case 3, and case 2 commands will become case 4.

9.4.1 C-APDU

The C-APDU consists of a mandatory header of four consecutive bytes denoted CLA, INS, P1, and P2, followed by a conditional body of variable length.

These mandatory header bytes are defined as follows:

- CLA: Instruction class; may take any value except 'FF'.
- INS: Instruction code within the instruction class. The INS is only valid if the l.s. bit is 0, and the m.s. nibble is neither '6' nor '9'.
- P1, P2: Reference bytes completing the INS.

Note: The full coding of the headers for each command is covered in section 11.

The conditional body consists of a string of bytes defined as follows:

- 1 byte, denoted by Lc, defining the number of data bytes to be sent in the C-APDU. The value of Lc may range from 1 to 255.
- String of bytes sent as the data field of the C-APDU, the number of bytes sent being as defined by Lc.
- 1 byte, denoted by Le, indicating the maximum number of data bytes expected in the R-APDU. The value of Le may range from 0 to 255; if Le = 0, the maximum number of bytes expected in the response is 256.

Note: The full coding of the data field of the conditional body for each command is covered in section 11.

The four possible C-APDU structures are defined in Table 35:

Case	Structure
1	CLA INS P1 P2
2	CLA INS P1 P2 Le
3	CLA INS P1 P2 Lc Data
4	CLA INS P1 P2 Lc Data Le

Table 35: C-APDU Structures

9.4.2 R-APDU

The R-APDU is a string of bytes consisting of a conditional body followed by a mandatory trailer of two bytes denoted SW1 SW2.

The conditional body is a string of data bytes with a maximum length as defined by Le in the C-APDU.

The mandatory trailer indicates the status of the ICC after processing the command.

The coding of SW1 SW2 is defined in section 11.

Part III

Files, Commands, and Application Selection

10 Files

An application in the ICC includes a set of items of information, often contained within files. These items of information may be accessible to the terminal after a successful application selection.

An item of information is called a data element. A data element is the smallest piece of information that may be identified by a name, a description of logical content, a format, and a coding.

It is up to the issuer to ensure that data in the card is of the correct format.

The data element directory defined in Annex B includes those data elements that may be used for application selection. Data elements used during application selection that are not specified in Annex B are outside the scope of these specifications.

10.1 File Structure

The file organisation applying to this specification is deduced from and complies with the basic organisations as defined in ISO/IEC 7816-4.

This section describes the file structure of applications conforming to this specification.

The files within the ICC are seen from the terminal as a tree structure. Every branch of the tree is an Application Definition File (ADF) or a Directory Definition File (DDF). An ADF is the entry point to one or more Application Elementary Files (AEFs). An ADF and its related data files are seen as being on the same branch of the tree. A DDF is an entry point to other ADFs or DDFs.

10.1.1 Application Definition Files

The tree structure of ADFs:

- Enables the attachment of data files to an application.
- Ensures the separation between applications.
- Allows access to the logical structure of an application by its selection.

An ADF is seen from the terminal as a file containing only data objects encapsulated in its file control information (FCI) as shown in Table 45.

10.1.2 Application Elementary Files

The structure and use of AEFs is application dependent. For the EMV Debit/Credit application, the files are described in Book 3.

10.1.3 Mapping of Files Onto ISO/IEC 7816-4 File Structure

The following mapping onto ISO/IEC 7816-4 applies:

- A dedicated file (DF) as defined in ISO/IEC 7816-4 and containing a FCI is mapped onto an ADF or a DDF. It may give access to elementary files and DFs. The DF at the highest level of the card is the master file (MF).
- An elementary file (EF) as defined in ISO/IEC 7816-4 is mapped onto the AEF. An EF is never used as an entry point to another file.

If DFs are embedded, retrieval of the attached EF is transparent to this specification.

10.1.4 Directory Structure

When the Payment System Environment (PSE) as described in section 12.2.2 is present, the ICC shall maintain a directory structure for the list of applications within the PSE that the issuer wants to be selected by a directory. In that case, the directory structure consists of a Payment System Directory file (DIR file) and optional additional directories introduced by a DDF as described in this section.

The directory structure allows for the retrieval of an application using its Application Identifier (AID) or the retrieval of a group of applications using the first n bytes of their AID as DDF name.

The presence of the DIR file shall be coded in the response message to the selection of the PSE (see the SELECT command).

The DIR file is an AEF (in other words, an EF) with a record structure according to this specification including the following data objects according to ISO/IEC 7816-4:

- One or more Application Templates (tag '61') as described in section 12.
- Optionally, other data objects present within a Directory Discretionary Template (tag '73'). The data objects contained in this template are outside the scope of this specification.

Directories are optional within an ICC, and when present there is no defined limit to the number of such directories that may exist. Each such directory is located by a directory SFI data object contained in the FCI of each DDF.

10.2 File Referencing

A file may be referred to by a name or a SFI depending on its type.

10.2.1 Referencing by Name

Any ADF or DDF in the card is referenced by its DF name. A DF name for an ADF corresponds to the AID or contains the AID as the beginning of the DF name. Each DF name shall be unique within a given card.

10.2.2 Referencing by SFI

SFIs are used for the selection of AEFs. Any AEF within a given application is referenced by a SFI coded on 5 bits in the range 1 to 30. The coding of the SFI is described in every command that uses it. A SFI shall be unique within an application.

11 Commands

11.1 Message Structure

Messages are transported between the terminal and the card according to the transmission protocol selected at the ATR (see Part II). The terminal and the card shall also implement the physical, data link, and transport layers as defined in Part II.

To run an application, an additional layer called application protocol is implemented in the terminal. It includes steps consisting of sending a command to the card, processing it in the card, and sending back the response to the terminal. All commands and responses referred to in the remainder of this Book are defined at the application layer.

The command message sent from the application layer and the response message returned by the card to the application layer are called Application Protocol Data Units (APDU). A specific response corresponds to a specific command. These are referred to as APDU command-response pairs. In an APDU command-response pair, the command message and the response message may contain data.

This section describes the structure of the APDU command-response pairs necessary to the application protocols defined in this specification. This Book describes only those commands necessary to the functioning of application selection. All other commands shall be implemented as required by specific applications, but shall conform to the APDU structures (formats) defined in Book 3, Part II.

11.1.1 Command APDU Format

The command APDU consists of a mandatory header of four bytes followed by a conditional body of variable length, as shown in Figure 15:

CLA	INS	P1	P2	Lc	Data	Le
← Mandatory Header →				← Conditional Body →		

Figure 15: Command APDU Structure

The number of data bytes sent in the command APDU is denoted by Lc (length of command data field).

The maximum number of data bytes expected in the response APDU is denoted by Le (length of expected data). When Le is present and contains the value zero, the maximum number of data bytes available (≤ 256) is requested. READ RECORD and SELECT commands issued during application selection and all case 2 and case 4 commands issued according to Book 3 shall have Le = '00'.

The content of a command APDU message is as shown in Table 36:

Code	Description	Length
CLA	Class of instruction	1
INS	Instruction code	1
P1	Instruction parameter 1	1
P2	Instruction parameter 2	1
Lc	Number of bytes present in command data field	0 or 1
Data	String of data bytes sent in command (= Lc)	var.
Le	Maximum number of data bytes expected in data field of response	0 or 1

Table 36: Command APDU Content

The different cases of command APDU structure are described in Table 35.

11.1.2 Response APDU Format

The response APDU format consists of a conditional body of variable length followed by a mandatory trailer of two bytes, as shown in Figure 16:

Data	SW1	SW2
← Body →	← Trailer →	

Figure 16: Response APDU Structure

The number of data bytes received in the response APDU is denoted by Lr (length of response data field). Lr is not returned by the transport layer. The application layer may rely on the object oriented structure of the response message data field to calculate Lr if needed.

The trailer indicates in two bytes the processing state of the command as returned by the transport layer.

The content of a response APDU message is as shown in Table 37:

Code	Description	Length
Data	String of data bytes received in response	var(= Lr)
SW1	Command processing status	1
SW2	Command processing qualifier	1

Table 37: Response APDU Content

11.2 READ RECORD Command-Response APDUs

11.2.1 Definition and Scope

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

The response from the ICC consists of returning the record.

11.2.2 Command Message

The READ RECORD command message is coded according to Table 38:

Code	Value
CLA	'00'
INS	'B2'
P1	Record number
P2	Reference control parameter (see Table 39)
Lc	Not present
Data	Not present
Le	'00'

Table 38: READ RECORD Command Message

Table 39 defines the reference control parameter of the command message:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x				SFI
					1	0	0	P1 is a record number

Table 39: READ RECORD Command Reference Control Parameter

11.2.3 Data Field Sent in the Command Message

The data field of the command message is not present.

11.2.4 Data Field Returned in the Response Message

The data field of the response message of any successful READ RECORD command contains the record read. Records read during application selection are directory records which are formatted as in section 12.2.3. The format of records read during application processing is application dependent.

11.2.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

11.3 SELECT Command-Response APDUs

11.3.1 Definition and Scope

The SELECT command is used to select the ICC PSE, DDF, or ADF corresponding to the submitted file name or AID. The selection of an application is described in section 12.

A successful execution of the command sets the path to the PSE, DDF, or ADF.

Subsequent commands apply to AEFs associated with the selected PSE, DDF, or ADF using SFIs.

The response from the ICC consists of returning the FCI.

11.3.2 Command Message

The SELECT command message is coded according to Table 40:

Code	Value
CLA	'00'
INS	'A4'
P1	Reference control parameter (see Table 41)
P2	Selection options (see Table 42)
Lc	'05'–'10'
Data	File name
Le	'00'

Table 40: SELECT Command Message

Table 41 defines the reference control parameter of the SELECT command message:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0				
					1			Select by name
						0	0	

Table 41: SELECT Command Reference Control Parameter

Table 42 defines the selection options P2 of the SELECT command message:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
						0	0	First or only occurrence
						1	0	Next occurrence

Table 42: SELECT Command Options Parameter

11.3.3 Data Field Sent in the Command Message

The data field of the command message contains the PSE name or the DF name or the AID to be selected.

11.3.4 Data Field Returned in the Response Message

The data field of the response message contains the FCI specific to the selected PSE, DDF, or ADF. The tags defined in Table 43, Table 44, and Table 45 apply to this specification. No additional data elements shall be present in the FCI template (tag '6F') returned in the response to the SELECT command other than those contained in template 'BF0C'. Data elements present in templates '6F' and/or 'BF0C' that are not expected or understood by the terminal because the terminal does not support any issuer-specific processing shall be ignored.

Table 43 defines the FCI returned by a successful selection of the PSE:

Tag				Value	Presence
'6F'				FCI Template	M
	'84'			DF Name	M
	'A5'			FCI Proprietary Template	M
		'88'		SFI of the Directory Elementary File	M
		'5F2D'		Language Preference	O
		'9F11'		Issuer Code Table Index	O
		'BF0C'		FCI Issuer Discretionary Data	O
			'XXXX' (Tag constructed according to Book 3, Annex B)	1 or more additional proprietary data elements from an application provider, issuer, or IC card supplier, or EMV-defined tags that are specifically allocated to 'BF0C'	O

Table 43: SELECT Response Message Data Field (FCI) of the PSE

Table 44 defines the FCI returned by a successful selection of a DDF:

Tag	Value				Presence
'6F'	FCI Template				M
	'84'	DF Name			M
	'A5'	FCI Proprietary Template			M
		'88'	SFI of the Directory Elementary File		M
		'BF0C'	FCI Issuer Discretionary Data		O
			'XXXX' (Tag constructed according to Book 3, Annex B)	1 or more additional proprietary data elements from an application provider, issuer, or IC card supplier, or EMV-defined tags that are specifically allocated to 'BF0C'	O

Table 44: SELECT Response Message Data Field (FCI) of a DDF

Table 45 defines the FCI returned by a successful selection of an ADF:

Tag	Value				Presence
'6F'	FCI Template				M
	'84'	DF Name			M
	'A5'	FCI Proprietary Template			M
		'50'	Application Label		O
		'87'	Application Priority Indicator		O
		'9F38'	PDOL		O
		'5F2D'	Language Preference		O
		'9F11'	Issuer Code Table Index		O
		'9F12'	Application Preferred Name		O
		'BF0C'	FCI Issuer Discretionary Data		O
			'9F4D'	Log Entry	O
			'XXXX' (Tag constructed according to Book 3, Annex B)	1 or more additional proprietary data elements from an application provider, issuer, or IC card supplier, or EMV-defined tags that are specifically allocated to 'BF0C'	O

Table 45: SELECT Response Message Data Field (FCI) of an ADF

Note: For multi-application ICCs, it is strongly recommended that the Application Label data element be included in the response message in order to facilitate cardholder choice/confirmation of the application to be used when a terminal employs the List of AIDs method for application selection.

11.3.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

ICC support for the selection of a DF file using only a partial DF name is not mandatory. However, if the ICC does support partial name selection, it shall comply with the following:

- If, after a DF file has been successfully selected, the terminal repeats the SELECT command having P2 set to the Next Occurrence option (see Table 42) and with the same partial DF name, the card shall select a different DF file matching the partial name, if such other DF file exists.
- Repeated issuing of the same command with no intervening application level commands shall retrieve all such files, but shall retrieve no file twice.
- After all matching DF files have been selected, repeating the same command again shall result in no file being selected, and the card shall respond with SW1 SW2 = '6A82' (file not found).

12 Application Selection

12.1 Overview of Application Selection

During an EMV card session as defined in section 6.1.1, application selection using the commands and techniques described in sections 11 and 12 shall be the first process performed immediately after contact activation/reset of the card and prior to the first application function. If a proprietary processing session (including any proprietary application selection method) is performed immediately before or after an EMV card session, there is no requirement to remove/reinsert the card between the sessions. However, if proprietary processing occurs before the EMV card session, the card contacts shall be deactivated before starting the EMV card session.

This section describes the application selection process from the standpoint of both the card and the terminal. It specifies the logical structure of data and files within the card that are required for the process, then describes the terminal logic using the card structure.

It is not recommended that the ICC and the terminal use implicit selection as defined in ISO 7816, as it is not useful in an interchange environment. If used, it shall be performed outside the EMV card session as defined in section 6.1.1.

The application selection process described in this section is the process by which the terminal uses data in the ICC according to protocols defined herein to determine the terminal program and the ICC application to be used in processing a transaction. The process is described in two steps:

1. Create a list of ICC applications that are supported by the terminal. (This list is referred to below using the name ‘candidate list.’) This process is described in section 12.3.
2. Select the application to be run from the list generated above. This process is described in section 12.4.

This section of the specification describes the necessary information in the card and two terminal selection algorithms that yield the correct results. Other terminal selection algorithms that yield the same results are permitted in place of the selection algorithms described here.

A payment system application is comprised of the following:

- A set of files in the ICC providing data customised by the issuer
- Data in the terminal provided by the acquirer or the merchant
- An application protocol agreed upon by both the ICC and the terminal

Applications are uniquely identified by AIDs conforming to ISO/IEC 7816-4 (see section 12.2.1).

The techniques chosen by the payment systems and described herein are designed to meet the following key objectives:

- Ability to work with ICCs with a wide range of capabilities.
- Ability for terminals with a wide range of capabilities to work with all ICCs supporting payment system applications according to this specification.
- Conformance with ISO standards.
- Ability of ICCs to support multiple applications, not all of which need to be payment system applications.
- Ability for ICCs to provide multiple sets of applications to be supported by a single terminal program. (For example, a card may contain multiple credit/debit applications, each representing a different type or level of service or a different account).
- As far as possible, provide the capability for applications conforming with this specification to co-reside on cards with presently existing applications.
- Minimum overhead in storage and processing.
- Ability for the issuer to optimise the selection process.

The set of data that the ICC contains in support of a given application is defined by an ADF selected by the terminal using a SELECT command and an Application File Locator (AFL) returned by the ICC in response to a GET PROCESSING OPTIONS command.

12.2 Data in the ICC Used for Application Selection

12.2.1 Coding of Payment System Application Identifier

The structure of the AID is according to ISO/IEC 7816-4 and consists of two parts:

1. A Registered Application Provider Identifier (RID) of 5 bytes, unique to an application provider and assigned according to ISO/IEC 7816-4.
2. An optional field assigned by the application provider of up to 11 bytes. This field is known as a Proprietary Application Identifier Extension (PIX) and may contain any 0–11 byte value specified by the provider. The meaning of this field is defined only for the specific RID and need not be unique across different RIDs.

Additional ADFs defined under the control of other application providers may be present in the ICC but shall avoid duplicating the range of RIDs assigned to payment systems. Compliance with ISO/IEC 7816-4 will assure this avoidance.

12.2.2 Structure of the PSE

The PSE begins with a DDF given the name '1PAY.SYS.DDF01'. The presence of this DDF in the ICC is optional but, if present, it shall comply with this specification. If it is present, this DDF is mapped onto a DF within the card, which may or may not be the MF. As with all DDFs, this DDF shall contain a Payment System Directory. The FCI of this DDF shall contain at least the information defined for all DDFs in section 11 and, optionally, the Language Preference (tag '5F2D') and the Issuer Code Table Index (tag '9F11').

The Language Preference and Issuer Code Table Index are optional data objects that may occur in two places: the FCI of the PSE and the FCI of ADF files. If either of these data elements is present in one location but not the other, the terminal shall use the data element that is present. If either data element is present in both locations but has different values in the two locations, the terminal may use either value.[9]

The directory attached to this initial DDF contains entries for ADFs that are formatted according to this specification, although the applications defined by those ADFs may or may not conform to this specification. The directory may also contain entries for other payment system's DDFs, which shall conform to this specification.

The directory is not required to have entries for all DDFs and ADFs in the card, and following the chain of DDFs may not reveal all applications supported by the card. However, if the PSE exists, only applications that are revealed by following the chain of DDFs beginning with the initial directory can be assured of international interoperability.

See Annex C for examples of the internal logic structure of an ICC containing the PSE.

---

[9] A terminal building a candidate list using the process described in section 12.3.2 will encounter the values specified in the FCI of the PSE and will not see the values specified in the FCI of the ADF until the application to be run has been chosen. A terminal building the candidate list using the process described in section 12.3.3 will encounter the values specified in the FCI of the ADFs. To ensure consistent interface to the cardholder, the values must be the same.

12.2.3 Coding of a Payment System Directory

A Payment System Directory is a linear EF file identified by a SFI in the range 1 to 10. The SFI for the Payment System Directory is contained in the FCI of the DDF to which the directory is attached. The Payment System Directory is read using the READ RECORD command as defined in section 11. A record may have several entries, but a single entry shall always be encapsulated in a single record.

Each record in the Payment System Directory is a constructed data object, and the value field is comprised of one or more directory entries as described below. Each record is formatted as shown in Table 46:

Tag '70'	Data Length (L)	Tag '61'	Length of directory entry 1	Directory entry 1 (ADF or DDF)	...	Tag '61'	Length of directory entry n	Directory entry n (ADF or DDF)

Table 46: Payment System Directory Record Format

Each entry in a Payment System Directory is the value field of an Application Template (tag '61') and contains the information according to Table 47 or Table 48. No additional data elements shall be present in the Payment System Directory Record (tag ‘70’) other than those contained in template ‘73’. Data elements present in the Payment System Directory Record, template '61', or template '73' that are not expected or understood by the terminal because the terminal does not support any issuer-specific processing, shall be ignored.

Tag	Length	Value		Presence
'9D'	5–16	DDF Name		M
'73'	var.	Directory Discretionary Template		O [10]
	'XXXX' (Tag constructed according to Book 3, Annex B)	var.	1 or more additional proprietary data elements from an application provider, issuer, or IC card supplier, or EMV-defined tags that are specifically allocated to template '73'	O

Table 47: DDF Directory Entry Format

[10] Other data objects not relevant to this specification may appear in this constructed data object.

Tag	Length	Value		Presence
'4F'	5–16	ADF Name		M
'50'	1–16	Application Label		M
'9F12'	1–16	Application Preferred Name		O
'87'	1	Application Priority Indicator (see Table 49)		O
'73'	var.	Directory Discretionary Template		O [11]
	'XXXX' (Tag constructed according to Book 3, Annex B)	var.	1 or more additional proprietary data elements from an application provider, issuer, or IC card supplier, or EMV-defined tags that are specifically allocated to template '73'	O

Table 48: ADF Directory Entry Format

b8	b7–b5	b4–b1	Definition
1			Application cannot be selected without confirmation of cardholder
0			Application may be selected without confirmation of cardholder
	xxx		RFU
		0000	No priority assigned
		xxxx (except 0000)	Order in which the application is to be listed or selected, ranging from 1–15, with 1 being highest priority

Table 49: Format of Application Priority Indicator

[11] Other data objects not relevant to this specification may appear in this constructed data object.

12.2.4 Coding of Other Directories

Each directory in an ICC is contained by a separate DDF. DDFs and directories in the card are optional, and when present there is no defined limit to the number that may exist. Each directory is located by a Directory SFI data object which must be contained in the FCI of the DDF (see section 11.3 for a description of the SELECT command). The low order five bits of the Directory SFI contain the SFI to be used in READ RECORD commands for reading the directory. The SFI shall be valid for reading the directory when the DDF containing the directory is the current file selected.

All directories, including the initial directory, have the same format, as described in section 12.2.3.

12.2.5 Error Handling for FCI Response Data

The data elements Application Label, Application Preferred Name, Issuer Code Table Index, and Language Preference are present for the convenience of the cardholder and are not critical to the successful processing of application selection. If these data elements are present in the FCI, the issuer is responsible for their correct encoding.

Terminals shall not enforce the correct formatting of these data elements. If Application Preferred Name or Application Label contains a character that is not valid for the defined format, the terminal shall display the character if it is able to, or if the terminal is unable to display the invalid character, it should omit the character or substitute a space character or any other appropriate character. Otherwise, if the terminal detects format errors in any of these data elements, the terminal shall disregard these errors and act as if the response provided by the card did not contain these data elements. More specifically, the terminal shall not terminate the card session but shall proceed with application selection.

If the terminal does not understand the value in Issuer Code Table Index or Language Preference, it shall treat the data element as not present.

12.3 Building the Candidate List

The terminal shall maintain a list of applications supported by the terminal and their AIDs. This section describes two procedures for determining which of those applications is to be run. If the card contains no PSE, the procedure described in section 12.3.3 must be followed.

The terminal may know other ways that are not described in this section to locate proprietary applications or to eliminate specific applications in the ICC from consideration. This is permitted as long as all interoperable applications can be located in the ICC using the techniques described here.

12.3.1 Matching Terminal Applications to ICC Applications

The terminal determines which applications in the ICC are supported by comparing the AIDs for applications in the terminal with AIDs for applications within the ICC.

In some cases, the terminal supports the ICC application only if the AID in the terminal has the same length and value as the AID in the ICC. This case limits the ICC to at most one matching ADF.

In other cases, the terminal supports the ICC application if the AID in the ICC begins with the entire AID kept within the terminal. This allows the ICC to have multiple ADFs matching the terminal application by adding unique information to the AID used by each of the ADFs. If the card has only one ADF matching the terminal AID, it should identify that ADF with the exact AID known to the terminal. If the ICC has multiple ADFs supported by a single terminal AID, the following requirements must be met by the ICC:

- The ICC must support partial name selection as described in section 11.3.5 (see SELECT command).
- All of the matching AIDs in the ICC must be distinguished by adding unique data to the PIX. None of the ICC AIDs shall be the same length as the AID in the terminal.

For each of the AIDs within the list of applications supported by the terminal, the terminal shall keep an indication of which matching criterion to use.

12.3.2 Using the PSE

If a terminal chooses to support application selection using the PSE method, it shall follow the procedure described in this section to determine the applications supported by the card. Figure 17 is a flow diagram of the logic described here.

The terminal performs the following steps:

1. The terminal begins by selecting the PSE using a SELECT command as described in section 11 and a file name of ‘1PAY.SYS.DDF01’. This establishes the PSE and makes the initial Payment System Directory accessible.

   If the card is blocked or the SELECT command is not supported (both conditions represented by SW1 SW2 = '6A81'), the terminal terminates the session.

   If there is no PSE in the ICC, the ICC shall return '6A82' (‘File not found’) in response to the SELECT command for the PSE. In this case, the terminal shall use the List of AIDs method described in section 12.3.3.

   If the PSE is blocked, the ICC shall return '6283'. In this case, the terminal shall use the List of AIDs method described in section 12.3.3.

   If the ICC returns SW1 SW2 = '9000', the terminal proceeds to step 2.

   If the card returns any other value in SW1 SW2, the terminal shall use the List of AIDs method described in section 12.3.3.

   If any error, including a SW1 SW2 different from '90 00' or '6A 83', occurs in steps 2 through 5, the terminal shall clear the candidate list and restart the application selection process using the List of AIDs method described in section 12.3.3 to find the matching applications.

2. The terminal uses the Directory SFI from the FCI returned and reads all the records in the Payment System Directory beginning with record number 1 and continuing with successive records until the card returns SW1 SW2 = '6A83', which indicates that the record number requested does not exist. (The card shall return '6A83' if the record number in the READ RECORD command is greater than the number of the last record in the file). If the card returns SW1 SW2 = '6A83' in response to a READ RECORD for record number 1 for the Payment System Directory, no directory entries exist, and step 6 (below) applies.

   For each record in the Payment System Directory, the terminal begins with the first directory entry and processes each directory entry in turn as described in steps 3 through 5. If there are no directory entries in the record, the terminal proceeds to the next directory record.

3. If the entry is for an ADF and the ADF name matches one of the applications supported by the terminal as defined in section 12.3.1, the application joins the candidate list for final application selection under control of the Application Selection Indicator (ASI) maintained in the terminal for that AID.

   The ASI indicates whether the AID in the terminal shall match exactly (both in length and name) or need only partially match the associated ADF name in the card (tag '4F').

   The application is added to the candidate list in either of the following cases:

   - the AID entry retrieved is an exact match, or
   - the ASI for the AID in the terminal indicates that a partial match is allowed.

   The application is not added to the candidate list if the ADF entry retrieved is not an exact match and the ASI for the AID in the terminal indicates that an exact match is required.

4. If the entry is for a DDF, the terminal interrupts processing of the current directory record, places resumption information on the stack, and selects the DDF indicated using the DDF name. The new directory is read and processed according to steps 2 through 5, after which the terminal resumes processing the previously interrupted directory at the point of interruption.

5. When the terminal finishes processing all entries in the last record of the Payment System Directory, all ADFs that can be found by this procedure have been determined. The search and the candidate list are complete. If at least one matching AID was found, the terminal continues processing as described in section 12.4.

6. If steps 1 through 5 yield no directory entries that match applications supported by the terminal, the terminal shall use the list of AIDs method described in section 12.3.3 to find a match.

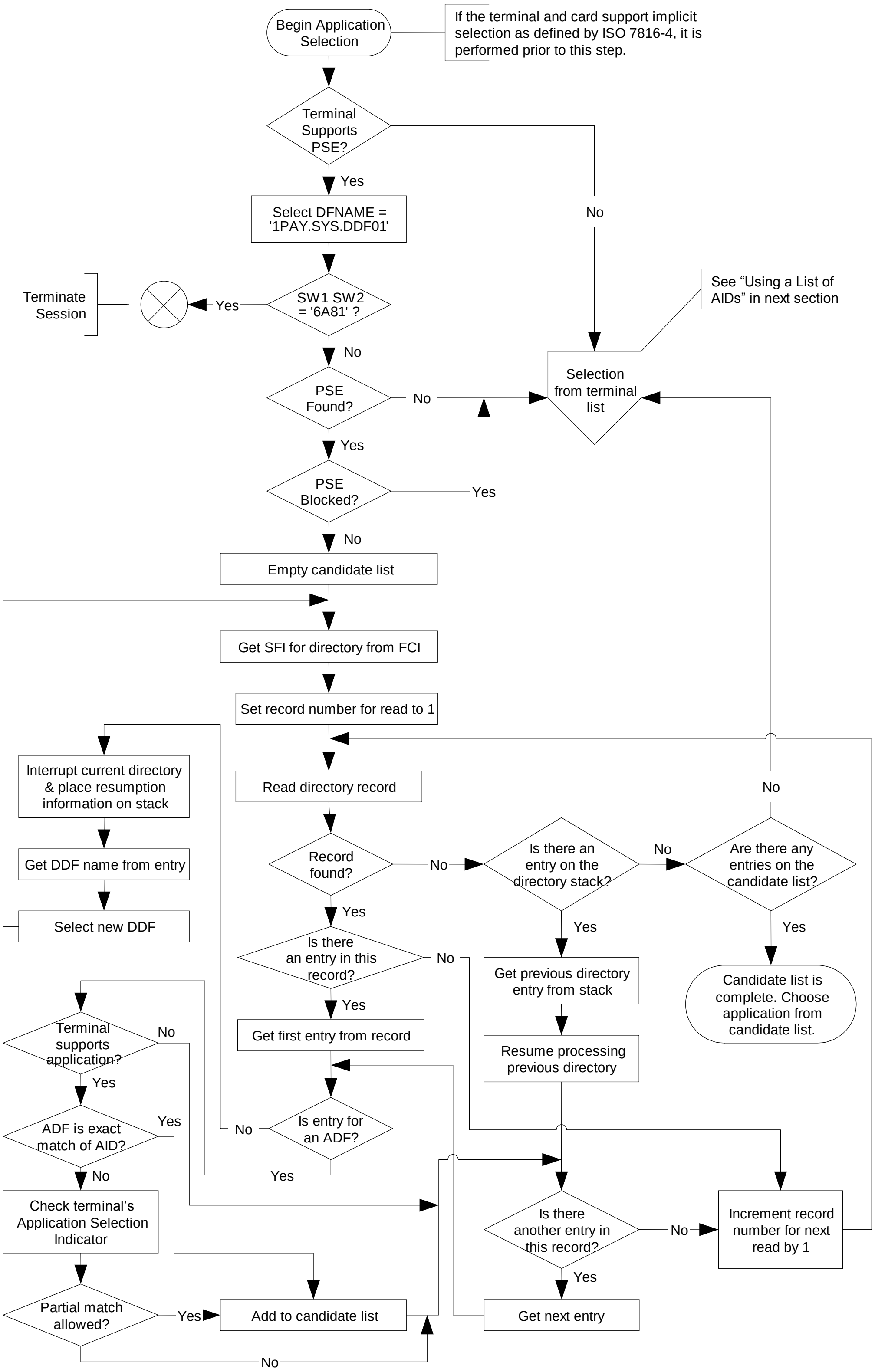


Figure 17: Terminal Logic Using Directories

12.3.3 Using a List of AIDs

If either the card or the terminal does not support the PSE method or if the terminal is unable to find a matching application using the Payment System Directory selection method, the terminal shall use a list of AIDs that it supports to build the candidate list. Figure 18 is a flow diagram of the logic described here.

The terminal performs the following steps:

1. The terminal issues a SELECT command using the first AID [12] in the terminal list as the file name.
2. If the SELECT command fails because the card is blocked or the command is not supported by the ICC (SW1 SW2 = '6A81'), the terminal terminates the card session.
3. If the SELECT command is successful (SW1 SW2 = '9000' or '6283'), the terminal compares the AID with the DF Name field returned in the FCI. The DF Name will either be identical to the AID (including the length), or the DF Name will start with the AID but will be longer. If the names are identical, the terminal proceeds with step 4. If the DF Name is longer, the card processed the command as a partial name selection, and the terminal proceeds to step 6.

   If the card returns any other status or if mandatory data is missing from the SELECT response or if the FCI contains formatting errors not described in Section 12.2.5, the terminal proceeds to Step 5 without adding the DF Name to the candidate list.
4. If the SELECT command is successful (SW1 SW2 = '9000'), the terminal adds the FCI information from the selected file to the candidate list 13 and proceeds to step 5. If the application is blocked (SW1 SW2 = '6283'), the terminal proceeds to step 5 without adding the DF Name to the candidate list.
5. The terminal issues another SELECT command using the next AID in its list and returns to step 3. If there are no more AIDs in the list, the candidate list is complete, and the terminal proceeds as specified in section 12.4.

[12] To assist in a clear understanding of the process described in this section, it is necessary to distinguish between the AID kept in the terminal and the AID kept in the ICC. As can be seen in section 12.3.1, these might not be identical even for matching applications. In this procedure, the term AID is used for the application identifier kept in the terminal, and DF Name is used for the application identifier in the card.

[13] The Application Label and Application Preferred Name must also be saved if the cardholder will be provided a list during final selection. The DF Name and the Application Priority Indicator will be required in any case.

6. Along with the AID list, the terminal keeps an Application Selection Indicator that indicates whether the card may have multiple occurrences of the application within the card. The terminal checks this indicator. If the indicator says that an exact match (in both length and name) is required, the terminal does not add the file to the candidate list, but proceeds to step 5.

   If multiple occurrences are permitted, the partial name match is sufficient. If the application is not blocked (SW1 SW2 = '9000'), the terminal adds the FCI information to the candidate list and proceeds to step 7.

   If multiple occurrences are permitted but the application is blocked (SW1 SW2 ≠ '9000'), the terminal proceeds to step 7 without adding the FCI information to the candidate list.

7. The terminal repeats the SELECT command using the same command data as before, but changes P2 in the command to '02' (Select Next). If the ICC returns SW1 SW2 = '9000', '62xx', or '63xx', the terminal returns to step 3. If it returns a different SW1 SW2, the terminal goes to step 5.

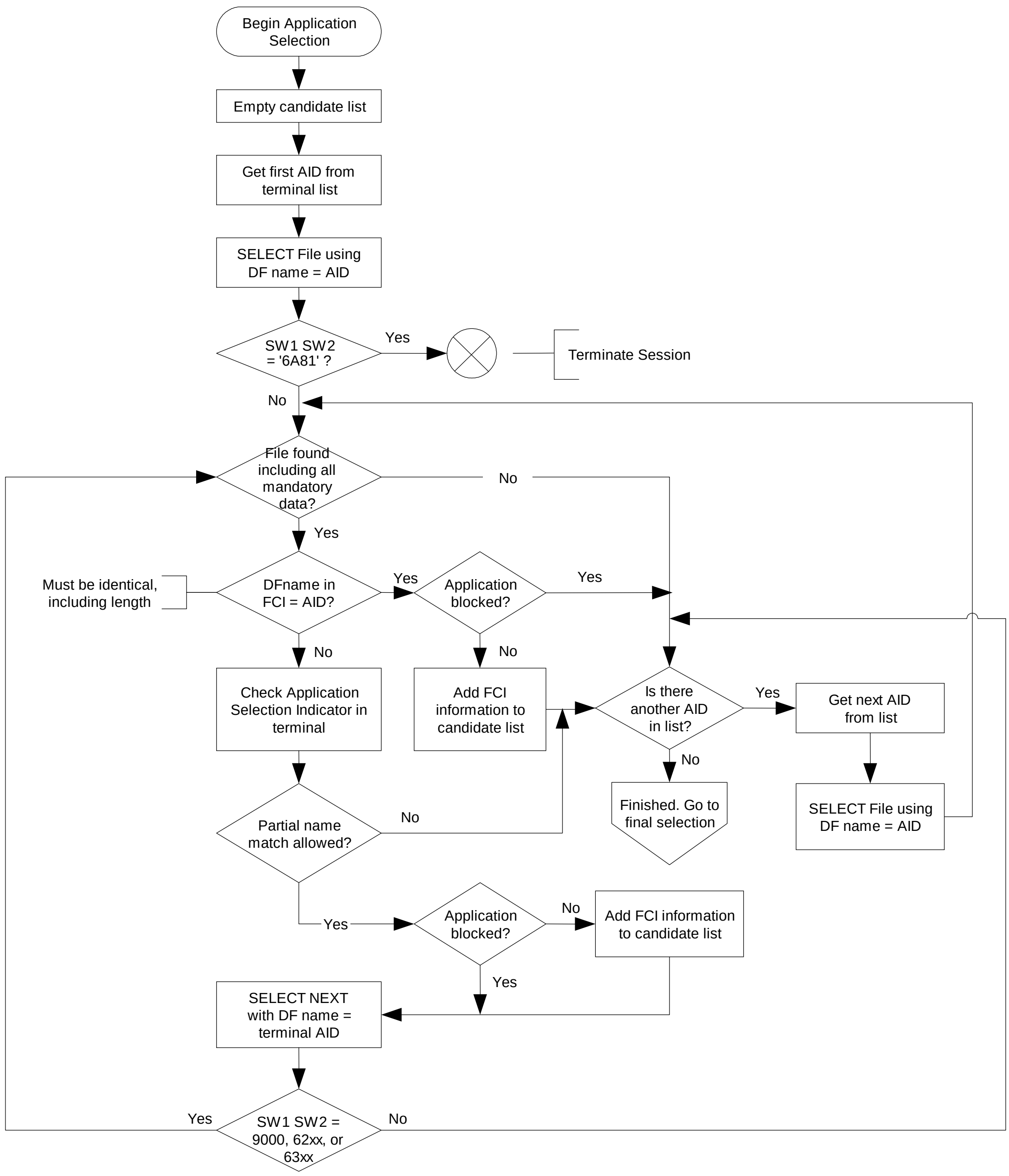


Figure 18: Using the List of AIDs in the Terminal

12.4 Final Selection

Once the terminal determines the list of mutually supported applications, it proceeds as follows:

1. If there are no mutually supported applications, the transaction is terminated.
2. If there is only one mutually supported application, the terminal checks b8 of the Application Priority Indicator for that application if present.
   - If b8 = '0', the terminal selects the application.
   - If b8 = '1' and the terminal provides for confirmation by the cardholder, the terminal requests confirmation and selects the application if the cardholder approves. If the terminal does not provide for confirmation by the cardholder, or if the terminal requests confirmation and the cardholder does not approve, the terminal terminates the session.
3. If multiple applications are supported, the terminal may offer a selection to the cardholder as described in step 4, or make the selection itself as described in step 5. Step 4 is the preferred method.
4. If a list is presented to the cardholder, it shall be in priority sequence, with the highest priority application listed first. If there is no priority sequence specified in the card, the list should be in the order in which the applications were encountered in the card, unless the terminal has its own preferred order. The same applies where duplicate priorities are assigned to multiple applications or individual entries are missing the Application Priority Indicator; that is, in this case, the terminal may use its own preferred order or display the duplicate priority or non-prioritised applications in the order encountered in the card.
5. The terminal may select the application without cardholder assistance. In this case, the terminal shall select the highest priority application from the list of mutually supported applications, except that if the terminal does not provide for confirmation of the selected application, applications prohibiting such selection (b8 = '1' in the Application Priority Indicator) shall be excluded from possible selection.

Once the application to be run is determined by the terminal or by the cardholder, the application shall be selected. A SELECT command coded according to section 11 shall be issued by the terminal for the application using the ADF Name field (if the directories were read) or the DF Name field from the FCI (if the List of AIDs method was used) found during the building of the candidate list. If the command returns other than '9000' in SW1 SW2 or the SELECT response contains format errors other than those described in section 12.2.5, the application shall be removed from the candidate list, and processing shall resume at step 1. If the cardholder selects or confirms the selection of an application that is subsequently removed from the candidate list due to its being blocked or for any other reason, no application is to be selected without cardholder confirmation.

In any case, the terminal shall inform the cardholder of the action taken, if appropriate.

Part IV

Annexes

Annex A Examples of Exchanges Using T=0

The following examples illustrate exchanges of data and procedure bytes between the TTL and ICC.

Note the following:

- The use of procedure bytes '60' and $\overline{\text{INS}}$ is not illustrated.
- [Data(x)] means x bytes of data.
- Case 2 and 4 commands have Le = '00' requesting the return of all data from the ICC up to the maximum available. Le = '00' is used in these examples to illustrate typical exchanges that may be observed when executing the application specified in Book 3. Le may take other values when executing a proprietary application.

Sections A1 to A4 illustrate typical exchanges using case 1 to 4 commands. Sections A5 and A6 illustrate the more extensive use of procedure bytes '61 xx' when used with case 2 and 4 commands. Section A7 illustrates a warning condition with a case 4 command.

A1 Case 1 Command

A C-APDU of {CLA INS P1 P2} is passed from the TAL to the TTL (note that P3 of the C-TPDU is set to '00').

TTL	ICC
[CLA INS P1 P2 00] ⇒	
	⇐ 90 00

A R-APDU of {90 00} is returned from the TTL to the TAL.

A2 Case 2 Command

A C-APDU of {CLA INS P1 P2 00} is passed from the TAL to the TTL.

TTL	ICC
[CLA INS P1 P2 00] ⇒	
	⇐ 6C Licc
[CLA INS P1 P2 Licc] ⇒	
	⇐ INS [Data(Licc)] 90 00

A R-APDU of {[Data(Licc)] 90 00} is returned from the TTL to the TAL.

A3 Case 3 Command

A C-APDU of {CLA INS P1 P2 Lc [Data(Lc)]} is passed from the TAL to the TTL.

TTL	ICC
[CLA INS P1 P2 Lc] ⇒	
	⇐ INS
[Data(Lc)] ⇒	
	⇐ 90 00

A R-APDU of {90 00} is returned from the TTL to the TAL.

A4 Case 4 Command

A C-APDU of {CLA INS P1 P2 Lc [Data (Lc)] 00} is passed from the TAL to the TTL.

TTL	ICC
[CLA INS P1 P2 Lc] ⇒	
	⇐ [INS]
[Data(Lc)] ⇒	
	⇐ 61 Licc
[00 C0 00 00 Licc] ⇒	
	⇐ C0 [Data(Licc)] 90 00

A R-APDU of {[Data(Licc)] 90 00} is returned from the TTL to the TAL.

A5 Case 2 Command Using the '61' and '6C' Procedure Bytes

A C-APDU of {CLA INS P1 P2 00} is passed from the TAL to the TTL.

TTL	ICC
[CLA INS P1 P2 00] ⇒	
	⇐ 6C Licc
[CLA INS P1 P2 Licc] ⇒	
	⇐ 61 xx
[00 C0 00 00 yy] ⇒	
	⇐ C0 [Data(yy)] 61 zz
[00 C0 00 00 zz] ⇒	
	⇐ C0 [Data(zz)] 90 00

Where yy ≤ xx

A R-APDU of {[Data(yy + zz)] 90 00} is returned from the TTL to the TAL.

A6 Case 4 Command Using the '61' Procedure Byte

A C-APDU of {CLA INS P1 P2 Lc [Data Lc] 00} is passed from the TAL to the TTL.

TTL	ICC
[CLA INS P1 P2 Lc] ⇒	
	⇐ [INS]
[Data(Lc)] ⇒	
	⇐ 61 xx
[00 C0 00 00 xx] ⇒	
	⇐ C0 [Data(xx)] 61 yy
[00 C0 00 00 yy] ⇒	
	⇐ C0 [Data(yy)] 90 00

A R-APDU of {[Data(xx + yy)] 90 00} is returned from the TTL to the TAL.

A7 Case 4 Command with Warning Condition

A C-APDU of {CLA INS P1 P2 Lc [Data Lc] 00} is passed from the TAL to the TTL.

TTL	ICC
[CLA INS P1 P2 Lc] ⇒	
	⇐ [INS]
[Data(Lc)] ⇒	
	⇐ 62 xx
[00 C0 00 00 00] ⇒	
	⇐ 6C Licc
[00 C0 00 00 Licc] ⇒	
	⇐ C0 [Data(Licc)] 90 00

A R-APDU of {[Data(Licc)] 62 xx} is returned from the TTL to the TAL containing the data returned together with the warning status bytes.

Annex B Data Elements Table

Table 50 defines those data elements that may be used for application selection and their mapping onto data objects and files.[14] Table 51 lists the data elements in tag sequence.

The characters used in the "Format" column are described in section 4.3, Data Element Format Convention.

B1 Data Elements by Name

Name	Description	Source	Format	Template	Tag	Length
Application Identifier (AID) - card	Identifies the application as described in ISO/IEC 7816-4	ICC	b	'61'	'4F'	5–16
Application Identifier (AID) - terminal	Identifies the application as described in ISO/IEC 7816-4	Terminal	b	—	'9F06'	5–16

Table 50: Data Elements Table

[14] Annex A of Book 3 provides a complete data elements table, defining all data elements that may be used for financial transaction interchange and their mapping onto data objects and files.

Name	Description	Source	Format	Template	Tag	Length
Application Label	Mnemonic associated with the AID according to ISO/IEC 7816-4	ICC	ans with the special character limited to space	'61' or 'A5'	'50'	1–16
Application Preferred Name	Preferred mnemonic associated with the AID	ICC	ans (see section 4.3)	'61' or 'A5'	'9F12'	1–16
Application Priority Indicator	Indicates the priority of a given application or group of applications in a directory	ICC	b	'61' or 'A5'	'87'	1
Application Selection Indicator	For an application in the ICC to be supported by an application in the terminal, the Application Selection Indicator indicates whether the associated AID in the terminal must match the AID in the card exactly, including the length of the AID, or only up to the length of the AID in the terminal There is only one Application Selection Indicator per AID supported by the terminal	Terminal	At the discretion of the terminal. The data is not sent across the interface	—	—	See Format
Application Template	Contains one or more data objects relevant to an application directory entry according to ISO/IEC 7816-4	ICC	b	'70'	'61'	var. up to 252

Table 50: Data Elements Table, continued

Name	Description	Source	Format	Template	Tag	Length
Bank Identifier Code (BIC)	Uniquely identifies a bank as defined in ISO 9362.	ICC	var.	'BF0C' or '73'	'5F54'	8 or 11
Dedicated File (DF) Name	Identifies the name of the DF as described in ISO/IEC 7816-4	ICC	b	'6F'	'84'	5–16
Directory Definition File (DDF) Name	Identifies the name of a DF associated with a directory	ICC	b	'61'	'9D'	5–16
Directory Discretionary Template	Issuer discretionary part of the directory according to ISO/IEC 7816-4	ICC	var.	'61'	'73'	var. up to 252
File Control Information (FCI) Issuer Discretionary Data	Issuer discretionary part of the FCI	ICC	var.	'A5'	'BF0C'	var. up to 222
File Control Information (FCI) Proprietary Template	Identifies the data object proprietary to this specification in the FCI template according to ISO/IEC 7816-4	ICC	var.	'6F'	'A5'	var.
File Control Information (FCI) Template	Identifies the FCI template according to ISO/IEC 7816-4	ICC	var.	—	'6F'	var. up to 252

Table 50: Data Elements Table, continued

Name	Description	Source	Format	Template	Tag	Length
International Bank Account Number (IBAN)	Uniquely identifies the account of a customer at a financial institution as defined in ISO 13616.	ICC	var.	'BF0C' or '73'	'5F53'	Var. up to 34
Issuer Code Table Index	Indicates the code table according to ISO/IEC 8859 for displaying the Application Preferred Name	ICC	n 2	'A5'	'9F11'	1
Issuer Country Code (alpha2 format)	Indicates the country of the issuer as defined in ISO 3166 (using a 2 character alphabetic code)	ICC	a 2	'BF0C' or '73'	'5F55'	2
Issuer Country Code (alpha3 format)	Indicates the country of the issuer as defined in ISO 3166 (using a 3 character alphabetic code)	ICC	a 3	'BF0C' or '73'	'5F56'	3
Industry Identification Number (IIN)	The number that identifies the major industry and the card issuer and that forms the first part of the Primary Account Number (PAN)	ICC	n 6	'BF0C' or '73'	'42'	3
Issuer URL	The URL provides the location of the issuer's Library Server on the Internet	ICC	ans	'BF0C' or '73'	'5F50'	var.

Table 50: Data Elements Table, continued

Name	Description	Source	Format	Template	Tag	Length
Language Preference	1–4 languages stored in order of preference, each represented by 2 alphabetical characters according to ISO 639 **Note:** EMVCo strongly recommends that cards be personalised with data element '5F2D' coded in lowercase, but that terminals accept the data element whether it is coded in upper or lower case.	ICC	an 2	'A5'	'5F2D'	2–8
Log Entry	Provides the SFI of the Transaction Log file and its number of records	ICC	b	'BF0C' or '73'	'9F4D'	2
Processing Options Data Object List (PDOL)	Contains a list of terminal resident data objects (tags and lengths) needed by the ICC in processing the GET PROCESSING OPTIONS command	ICC	b	'A5'	'9F38'	var.
READ RECORD Response Message Template	Contains the contents of the record read. (Mandatory for SFIs 1-10. Response messages for SFIs 11-30 are outside the scope of EMV, but may use template '70'.)	ICC	var.	—	'70'	var. up to 252
Short File Identifier (SFI)	Identifies the AEF referenced in commands related to a given ADF or DDF. It is a binary data object having a value in the range 1 – 30 and with the three high order bits set to zero.	ICC	b	'A5'	'88'	1

Table 50: Data Elements Table, continued

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 zeroes.
- A data element in format a, an, or ans is left justified and padded with trailing hexadecimal zeroes.

When data is moved from one entity to another (for example, card to terminal), it shall always be passed in order from high order to low order, regardless of how it is internally stored. The same rule applies when concatenating data.

B2 Data Elements by Tag

Name	Template	Tag
Industry Identification Number (IIN)	'BF0C' or '73'	'42'
Application Identifier (AID) - card	'61'	'4F'
Application Label	'61' or 'A5'	'50'
Language Preference	'A5'	'5F2D'
Issuer URL	'BF0C' or '73'	'5F50'
International Bank Account Number (IBAN)	'BF0C' or '73'	'5F53'
Bank Identifier Code (BIC)	'BF0C' or '73'	'5F54'
Issuer Country Code (alpha2 format)	'BF0C' or '73'	‘5F55’
Issuer Country Code (alpha3 format)	'BF0C' or '73'	'5F56'
Application Template	'70'	'61'
File Control Information (FCI) Template	—	'6F'
READ RECORD Response Message Template	—	'70'
Directory Discretionary Template	'61'	'73'
Dedicated File (DF) Name	'6F'	'84'
Application Priority Indicator	'61' or 'A5'	'87'
Short File Identifier (SFI)	'A5'	'88'
Directory Definition File (DDF) Name	'61'	'9D'
Application Identifier (AID) - terminal	—	'9F06'
Issuer Code Table Index	'A5'	'9F11'
Application Preferred Name	'61' or 'A5'	'9F12'
Processing Options Data Object List (PDOL)	'A5'	'9F38'
Log Entry	'BF0C' or '73'	'9F4D'
File Control Information (FCI) Proprietary Template	'6F'	'A5'
File Control Information (FCI) Issuer Discretionary Data	'A5'	'BF0C'

Table 51: Data Elements Tags

Annex C Examples of Directory Structures

This annex illustrates some possible logical ICC file structures.

C1 Single Application Card

Figure 19 illustrates a single application card with only a single level directory. In this example, the MF (with file identification of '3F00', as defined by ISO/IEC 7816-4) acts as the only DDF in the card. The MF shall be given the unique payment system's name assigned to the first level DDF as defined in section 12.2, and the FCI of the MF shall contain the SFI data object.

'DIR A' in this example may or may not be the ISO DIR file, but it shall conform to this specification, including the requirement that it has a SFI in the range 1 to 10.

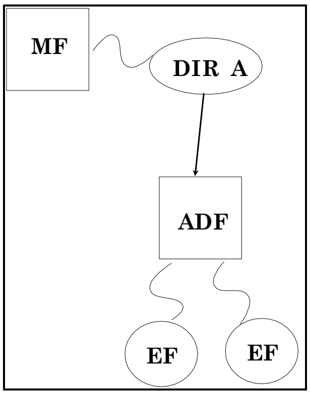


Figure 19: Simplest Card Structure Single Application

C2 Single Level Directory

Figure 20 gives an example of a multi-application card with a single directory. In this example, the root file (MF) does not support an application complying with this specification, and no restrictions are placed on the function of the MF. According to ISO/IEC 7816-4, a DIR file may be present but is not used by the application selection algorithm defined in section 12. Also note that the directory does not have entries for all ADFs (ADF2 to ADF5), as ADF5 is omitted. ADF5 can be selected only by a terminal that 'knows' ADF5 may exist in the card. The manner in which the terminal finds ADF5 is outside the scope of this specification.

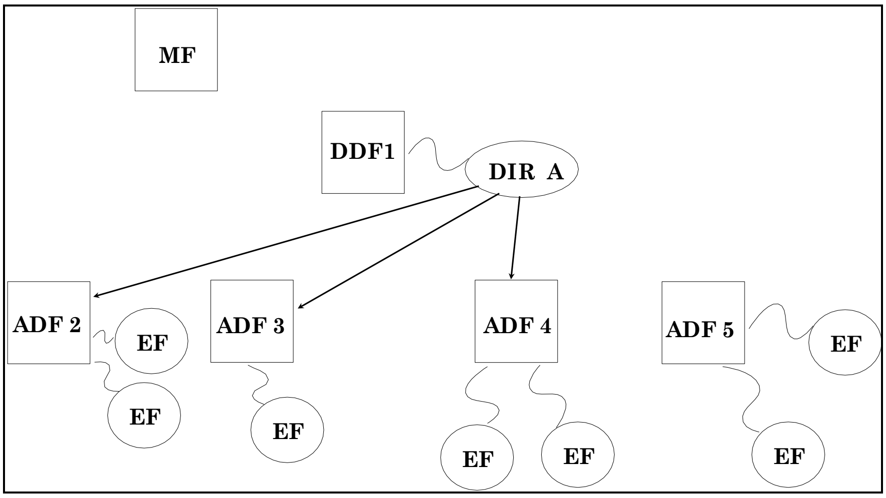


Figure 20: Single Level Directory

C3 Multi-Level Directory

Figure 21 is an example of a multi-application card with an *n* level directory structure. The first level directory ('DIR A') has entries for 2 ADFs – ADF3 and ADF4 – and a single DDF – DDF2. The directory attached to DDF2 ('DIR B') has entries for two ADFs – ADF21 and ADF22 – and a single DDF – DDF6. DDF5 has no entry in the root directory and can be found only by a terminal that 'knows' of the existence of DDF5. The manner in which the terminal finds and selects DDF5 is outside the scope of this specification, but the directory attached to DF5 ('DIR C') may conform to this specification, and, if found by the terminal, may lead the terminal to ADFs such as DF51, DF52, and DF53. DIR D, attached to DDF6, is a third level directory and points to four files (not shown), which may be either ADFs or more DDFs.

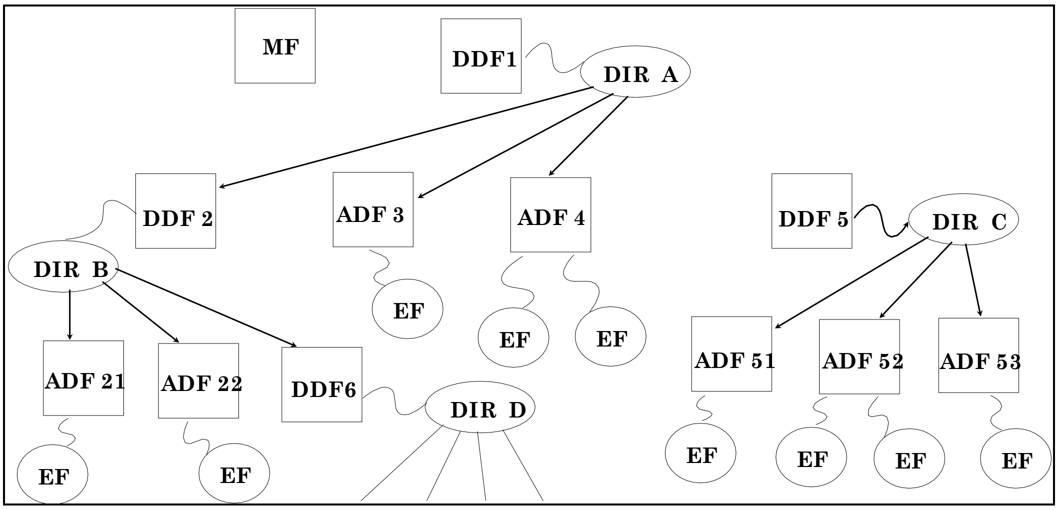


Figure 21: Third Level Directory

Part V

Common Core Definitions

Common Core Definitions

This Part describes an optional extension to this Book, to be used when implementing the Common Core Definitions (CCD). It is to be used in conjunction with Books 2, 3, and 4, including the Common Core Definitions part of Books 2 and 3.

These Common Core Definitions specify a minimum common set of card application implementation options, card application behaviours, and data element definitions sufficient to accomplish an EMV transaction. Terminals certified to be compliant with the existing EMV specifications will, without change, accept cards implemented according to the Common Core Definitions, since the Common Core Definitions are supported within the existing EMV requirements. To be compliant with the Common Core Definitions, an implementation shall implement all the additional requirements in the Common Core Definitions sections of all affected Books.

Changed Sections

Each section heading below refers to the section in this Book to which the additional requirements apply. The text defines requirements for a common core implementation, in addition to the requirements already specified in the referenced section of EMV.

Part III - Files, Commands, and Application Selection

10 Files

10.1 File Structure

10.1.4 Directory Structure

The directory structure within the PSE shall not contain any optional additional directories introduced by a DDF.

11 Commands

11.3 SELECT Command-Response APDUs

11.3.5 Processing State Returned in the Response Message

The ICC shall support partial name selection and shall accept SELECT command messages containing at least the 5 high-order bytes of the DF name (that is, the RID). Select Next functionality shall be supported.

12 Application Selection

12.2 Data in the ICC Used for Application Selection

12.2.2 Structure of the PSE

If the card has a PSE, the PSE shall contain only one DDF, the highest level DDF, '1PAY.SYS.DDF01'. No other DDFs shall be present. A graphic example of the internal logic structure of a CCD-compliant card can be found in Appendix C, Figure 20, where DDF1 is '1PAY.SYS.DDF01'.

12.2.3 Coding of a Payment System Directory

A Payment System Directory Record shall contain only ADF entries; DDF entries are not allowed. Each record in the Payment System Directory shall be formatted as in Table CCD 1:

Tag '70'	Data Length (L)	Tag '61'	Length of directory entry 1	Directory entry 1 (ADF)	...	Tag '61'	Length of directory entry n	Directory entry n (ADF)

Table CCD 1: Payment System Directory Record Format for CCD-Compliant Card

Index

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W