For Publication on the ICAO Website

TECHNICAL REPORT

Logical Data Structure (LDS) for Storage of Data in the Contactless IC

Doc 9303-10 LDS 2 – New Applications

Version 21

November 2018

File: 2018_11_10_Logical Data Structure
Author: ISO/IEC JTC1 SC17 WG3/TF5

Release Control

Release	Date	Description
12.8	20 Sept 15	Added AIDs, incorporated into TR format
12.9	09 Oct 15	Visa records simple TLV tags changed to BER TLV, multiple editorial changes
14.0	09 Nov 15	
15.0	25July 16	Incorporate agreed changes from April 2016 Wellington meeting
16.0	25 Oct 16	Incorporate agreed changes from October 2016 Berlin meeting
16.1		From Berlin meeting added Proprietary search criteria (SearchRecord-Proposal-GD-07-12-2015.pdf). removed annex D example Option 1 SEARCH RECORD
17.0	1 April 2017	Incorporate changes from WG3 March 2017 Milpitas Meeting Add initial Biometrics Application sections
18.0	11 Aug 2017	Incorporate changes from 8 April 2017 telco, NTWG meeting 26 April, and additional posted comments
19.0	11 Jan 2018	Incorporate changes from Oct 2017 Paris meeting Added proprietary WRITE BINARY section 6.4.1 Added SEARCH RECORD method according to ISO/IEC 7816-4:2013/DAmd 1
20.6	14 Mar 2018	Incorporate changes from March 2018 Tokyo meeting. Additional Biometrics transparent files handling part updated. Added proprietary File and Memory Management command Proprietary SEARCH RECORD command option removed
20.7	03 April 2018	Second round reviews from Tokyo meeting. Finalized EF.Biometrics and introduction of proprietary FMM command for memory management
21	10 Nov 2018	Minor changes from WG3. Reduced max certificate size from 1000Bytes to 900bytes in EF.CERTIFICATES

Table of contents

1. INTRODUCTION

This document specifies the file structures required to support the ICAO LDS2 project consisting of three additional and optional applications:

- Travel records (stamps);
- Visa records; and
- Additional biometrics.

The eMRTD may support one, several or all of these applications.

1.1 Scope and Purpose

The Doc 9303-10 LDS2 describes the file structures and application framework aspects of LDS2 and is intended to be integrated into a subsequent Technical Report on LDS2 along with other material covering the PKI and access protocols for the additional applications.

1.2 Assumptions

It has been assumed that the reader is familiar with ICAO Doc 9303 seventh edition and that this document is used in conjunction with the companion LDS2 documents;

- TR LDS2 – Protocols v0.8, April 2017;
- TR LDS2 – PKI Draft 0.8 October 2016.

1.3 Terminology

This document uses the terminology from Doc 9303 7th edition.

1.4 Data Elements Encoding Rules

The context of eMRTD uses two different tag allocation schemes for application class tag, such as defined in Doc9303-10 (LDS tag) and ISO/IEC 7816-6 (Interindustry tag).

- EF.ATR/INFO and EF.DIR use interindustry tag allocation scheme
- DFs with their containing EFs use LDS tag allocation scheme.

Interindustry tags specified in this document are used in LDS context, so coexistent tag allocation scheme is not required.

2. ELEMENTARY FILES

2.1 EF.ATR/INFO (MANDATORY)

EF.ATR/INFO must be located in the MF. The short EF identifier at MF level is ’01’.

Table 1: EF.ATR/INFO

File Name	EF.ATR/INFO
File ID	‘2F01’
Short EF Identifier	‘01’
Select / FMM Access	ALWAYS
Read Access	ALWAYS
Write/Update/Erase Access	NEVER
File structure	Transparent
Size	Variable

The contents of the EF.ATR/INFO can be retrieved by using a SELECT command followed by READ BINARY command. The READ BINARY command response data field contains the content of the EF.ATR/INFO.

Table 2: Data elements of EF.ATR/INFO

<table>
<tr><th>Tag</th><th>Length</th><th colspan="3">Value</th><th>Notes</th></tr>
<tr><td rowspan="4">‘47’</td><td rowspan="4">‘03’</td><td colspan="4">Card capabilities</td></tr>
<tr><td colspan="3">byte 1 - first software function</td><td>b8=1: DF selection by full DF name
b7 to b4 and b1 are out of scope of Doc 9303
b3=1: short EF identifier supported
b2=1: record number supported</td></tr>
<tr><td colspan="3">byte 2 - second software function</td><td>b8, b7, b6 and b5 are out of scope of Doc 9303
b4 to b1=0001: one byte data unit size</td></tr>
<tr><td colspan="3">byte 3 - third software function</td><td>b8=1: command chaining supported
b7=1: Extended Lc and Le fields supported
b6=1: Extended length information in EF.ATR/INFO
b5 to b1 are out of scope of Doc 9303</td></tr>
<tr><td rowspan="3">‘7F66’</td><td rowspan="3">‘xx’</td><td colspan="4">Extended length information</td></tr>
<tr><td>Tag</td><td>Length</td><td>Value</td><td>Notes</td></tr>
<tr><td>‘02’</td><td>‘02’</td><td>Positive integer - the maximum number of bytes in the command data field</td><td>MUST be at least 1000</td></tr>
<tr><td></td><td></td><td>‘02’</td><td>‘02’ or ‘03’</td><td>Positive integer - the maximum number of bytes expected in the response APDU</td><td>MUST be at least 1000</td></tr>
</table>

Note: Further data objects MAY be present in EF.ATR/INFO.
Note: EF.ATR/INFO uses standard tag allocation scheme as defined in ISO/IEC 7816-4

2.2 EF.CardSecurity (CONDITIONAL)

EF.CardSecurity is a transparent EF contained in the MF. The short EF identifier at MF level is ‘1D’.

Table 3: EF.CardSecurity

File Name	EF.CardSecurity
File ID	’011D’

Short EF Identifier	'1D'
Select Access	PACE
Read Access	PACE
Write/Update/Erase Access	NEVER
File structure	Transparent
Size	Variable

EF.CardSecurity contained in the MF is REQUIRED if

- PACE with Chip Authentication Mapping is supported by the IC;
- Terminal Authentication in the MF is supported by the IC; or
- Chip Authentication in the MF is supported by the IC.

and MUST contain

- ChipAuthenticationInfo as required by Chip Authentication;
- ChipAuthenticationPublicKeyInfo as required by PACE-CAM/Chip Authentication;
- TerminalAuthenticationInfo as required by Terminal Authentication;
- the SecurityInfos contained in EF.CardAccess.

2.3 EF.DIR (REQUIRED)

EF.DIR MUST be located in the MF. The short EF identifier at MF level is '1E'.

Table 4: EF.DIR

File Name	EF.DIR
File ID	'2F00'
Short EF Identifier	'1E'
Select Access	ALWAYS
Read Access	ALWAYS
Write/Update/Erase Access	NEVER
File structure	Transparent
Size	Variable

EF.DIR MUST indicate a list of applications supported by the eMRTD. It MUST contain a set of application templates and / or application identifier Dos in any order.

Table 5: EF.DIR Format

Tag	L	Value			Description
'61'	'09'				eMRTD Application Template
		Tag	**L**	**Value**	eMRTD Application 'International AID': 'A0 00 00 02 47 10 01'
		'4F'	'07'	'A0 00 00 02 47 10 01'	
'61'	'09'				Travel Records 'Application Template'
		Tag	**L**	**Value**	Travel Records 'International AID': 'A0 00 00 02 47 20 01'
		'4F'	'07'	'A0 00 00 02 47 20 01'	
Tag	**L**	**Value**			
'61'	'09'				Visa Records 'Application Template'
		Tag	**L**	**Value**	Visa Records 'International AID': 'A0 00 00 02 47 20 02'
		'4F'	'07'	'A0 00 00 02 47 20 02'	
Tag	**L**	**Value**			
'61'	'09'				Additional Biometrics 'Application Template'
		Tag	**L**	**Value**	Additional Biometrics 'International AID': 'A0 00 00 02 47 20 03'
		'4F'	'07'	'A0 00 00 02 47 20 03'	

Note: EF.DIR uses standard tag allocation scheme as defined in ISO/IEC 7816-4

The Application present on the eMRTDs MUST be referenced in the EF.DIR.

2.4 EF.CardAccess (CONDITIONAL)

EF.CardAccess is a transparent EF contained in the MF. The short EF identifier at MF level is '1C'.

Table 6: EF.CardAccess

File Name	EF.CardAccess
File ID	'011C'
Short EF Identifier	'1C'
Select Access	ALWAYS
Read Access	ALWAYS
Write/Update/Erase Access	NEVER
File structure	Transparent
Size	Variable

EF.CardAccess contained in the MF is REQUIRED if PACE is supported by the chip and MUST contain the following SecurityInfos for the PACE protocol:

- PACEInfo
- PACEDomainParameterInfo

3. TRAVEL RECORDS APPLICATION (CONDITIONAL)

The Travel Records application MAY be implemented by an issuing State or organization. The following is conditionally REQUIRED if the optional Travel Records application has been invoked.

3.1 File Structure Summary

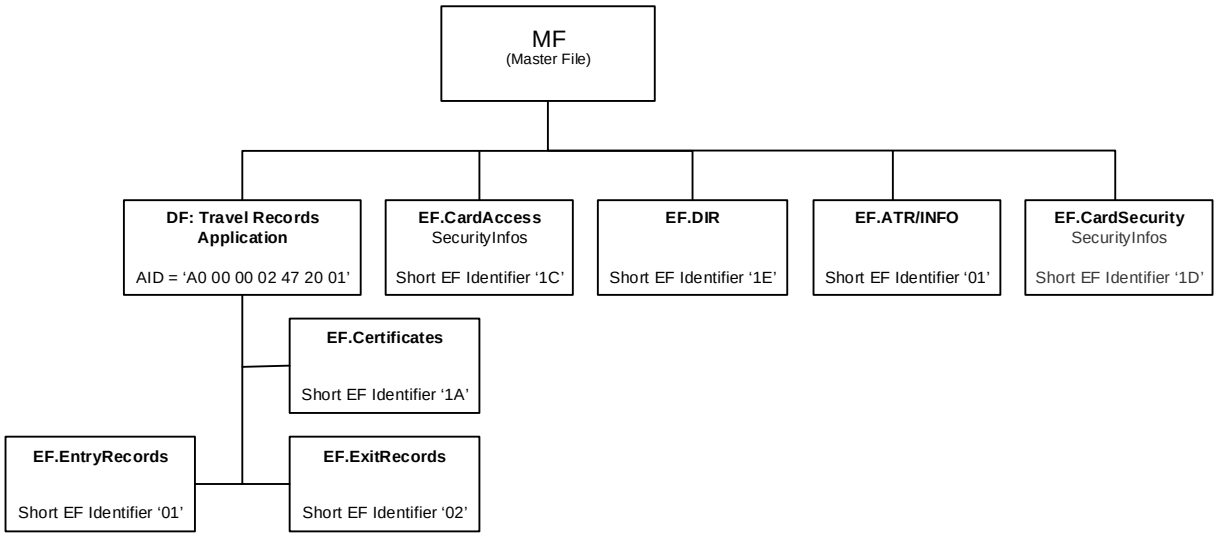


Figure 1: Travel Records Structure

3.2 EF.Certificates (REQUIRED)

The Travel Records Signer certificates are stored in an EF inside the application DF and having Linear Structure with Records of Variable Size. These certificates are intended to be used by the IS to further offline validation of the digital signatures for each record in both the EF.ExitRecords and EF.EntryRecords files.

Table 7: EF.Certificates

File Name	EF.Certificates
File ID	'011A'
Short EF Identifier	'1A'
Select / FMM Access	PACE+TA (Travel record authorization bit b3 according to table 21)
Read record / Search Record Access	PACE+TA (Travel record authorization bit b3 according to table 21)
Append Record Access	PACE+TA (Travel record authorization bit b4 according to table 21)
Write / Update Record Access	NEVER
Erase Record Access	NEVER
File structure	Linear structure with records of variable size
Size	Variable

Certificate record contains a single LDS2-TS Signer X.509 certificate data object. A Certificate record MAY be referenced by one or more Entry or Exit Travel Records.

Table 8: EF.Certificates Record Format

Tag	Content	Mandatory /Optional	Format	Example
5F3A	Certificate serial number	M	V(22)B	'5F3A' 'Len' {Country code || SerialNumber }
72	X.509 certificate	M	V (900) B	'72' Len { X.509 Certificate }

Note: Interindustry tags specified in this table are used in LDS context, so coexistent tag allocation scheme is not required.

DO'5F3A MUST contain a 2 letter country code according to ISO-3166-1 (same encoding and value as X.509 cerificate's subject's countryName) followed by the certificate serial number.

Each X.509 certificate contains a set of ASN.1 encoded data elements illustrated in the table below. Detailed requirements for the X.509 Certificate can be found in Doc 9303-12 Certificate Profile Specification.

Table 9: X.509 Certificate structure example

Field			Description	Example value
Certificate				
	version		Must be ver.3	2
	serialNumber		unique positive integer	20 bytes max
	signature		Signature algorithm	ecdsa-with-SHA256
	issuer			
		countryName	Issuing country name	"US"
		commonName	Issuer name (9 chars max)	"DHSCA0001"
	validity			
		notBefore	Cert. effective date	"131225000000Z"
		notAfter	Cert. expiration date	"230824235959Z"
	subject			
		countryName	IS country name	"US"
		commonName	IS name (9 chars max)	"SFO000001"
	subjectPublicKeyInfo			
		Public Key Algorithm	ecPublicKey	
		Subject Public Key	IS public key	ECC256 Public Key
	extensions			
		AuthorityKeyIdentifier		
		ExtKeyUsage		
Signature Algorithm			ecdsa-with-SHA256	
Signature			Issuer's Signature	ECDSA256 signature

Note: This table is an example for illustration only. Certificate records are written to EF.Certificates located under the Travel Records application DF using the APPEND RECORD command. Certificate records can be read from EF.Certificates using READ RECORD command. Certificate records MUST NOT be updated or erased. The maximum number of records in EF.Certificates under the Travel Records application DF MUST be 254.

3.3 Application Selection

The Travel Records application MUST be selected by use of the Application Identifier (AID) as a reserved DF name. The AID MUST consist of the Registered Application Identifier assigned by ISO according to [ISO/IEC 7816-5] followed by the Proprietary Application Identifier Extension (PIX) of the Travel Records application:

- The Registered Application Identifier is ‘A0 00 00 02 47'
- The Travel Records application MUST use PIX = '20 01';
- The full AID of the Travel Records application MUST be A0 00 00 02 47 20 01.

The IC MUST reject the selection of an application if the extension for this application is absent.

3.4 Entry and Exit Records

Entry and Exit Travel Records are stored in two separate Elementary Files EF.EntryRecords and EF.ExitRecords under the Travel Records application DF with both having Linear Structure with Records of Variable Size as per [ISO/IEC 7816-4].

3.5 EF.ExitRecords (REQUIRED)

It is REQUIRED to APPEND Exit Records upon embarkation at the IS.

Table 10: EF.ExitRecords

File Name	EF.ExitRecords
File ID	’0102’
Short EF Identifier	’02’
Select / FMM Access	PACE+TA (Travel record authorization bit b1 according to table 21)
Read Record / Search Record Access	PACE+TA (Travel record authorization bit b1 according to table 21)
Append Record Access	PACE+TA (Travel record authorization bit b2 according to table 21)
Write / Update Record Access	NEVER
Erase Record Access	NEVER
File structure	Linear structure with records of variable size
Size	Variable

The content of an Exit Record is shown in the table below.

Note: Interindustry tags specified in the table below are used in LDS context, so coexistent tag allocation scheme is not required.

Table 11: Entry / Exit Record Format

Tag	Tag	Content	Mandatory /OPTIONAL	Format	Example
5F44		Embarkation/Debarkation State (copy for SEARCH RECORD)	M	F (3) A	USA
73		Entry / Exit Travel Record (signed info)			
	5F44	Embarkation/Debarkation State	M	F (3) A	USA
	5F4C	Visa approvals, refusals, and revocations	O	V (50) A,N,S	Free-form text
	5F45	Travel date (Date of entry/exit)	M	F (8) N	20120814 (yyyymmdd)
	5F4B	Inspection authority	M	V (10) A,N,S	CBP
	5F46	Inspection location (Port of Entry/Exit)	M	V (10) A,N,S	SFO
	5F4A	Inspector reference	M	V (20) A,N,S	SFO00001234
	5F4D	Result of inspection	O	V (50) A,N,S	Free-form text
	5F49	Mode of travel	O	F (1) A	A (Air), S (Sea), L (Land)
	5F48	Duration of stay (days)	O	V (2) B	'00FF' (255 days)
	5F4E	Conditions holder is required to observe whilst in issuing State	O	V(50) A,N,S	Free-form text
5F37		Authenticity token (Signature)	M	V (140) B	'5F' '37' Len {Signature}
5F38		Reference (record number) to LDS2-TS Signer certificate in Certificates Store	M	F (1) B	'01' …'FE'

Note 1: A = Alpha character [a..z, A..Z], N = Numeric character [0..9], S = Special character ['<' ' '], B = 8-bit Binary data (any other than A, N or S), F = fixed-length field, V = variable-length field.

The order of the data objects in a record is fixed. The IS MUST build up the record content using the data objects in the order specified in the table.

Each Record MUST contain a digital signature (Authenticity Token) calculated over the DO'73, including Tag 73 and Length. Signature is generated by the LDS2-TS Signer.

LDS2-TS Signer certificates required to verify Travel Record's signature MUST be stored in the EF.Certificates under the Travel Records application DF if not already available in the same file.

Note 2: Since LDS2-TS Signer certificates are likely to be the same in multiple Travel Records (ex., when entering and exiting a country through the same airport having only one LDS2-TS Signer), before writing/appending a new certificate to the EF.Certificates, the IS should look up the EF.Certificates for a copy of the same certificate, and reference the existing one. This will reduce the size of EF.Certificates and enable faster lookups.

Travel Records are written (appended) to EF using APPEND RECORD. Travel Records MUST NOT be altered (updated) or deleted. The maximum number of records in each EF allowed MUST be 254.

Note 3: The eMRTD does not enforce that an IS writes Entry Records only to the EF.EntryRecords, but not to the EF.ExitRecords, and vice versa.

Note 4: Embarkation/Debarkation State 3-letter code according to Doc9303-3

3.6 EF.EntryRecords (REQUIRED)

It is REQUIRED to use Entry records during debarkation at the IS.

Table 12: EF.EntryRecords

File Name	EF.EntryRecords
File ID	’0101’
Short EF Identifier	’01’
Select / FMM Access	PACE+TA (Travel record authorization bit b1 according to table 21)
Read Record / Search Record Access	PACE+TA (Travel record authorization bit b1 according to table 21)
Append Record Access	PACE+TA (Travel record authorization bit b2 according to table 21)
Write / Update Record Access	NEVER
Erase Record Access	NEVER
File structure	Linear structure with records of variable size
Size	Variable

The structure of the entry record is identical to the structure of the exit record specified in clause 3.5.

4. VISA RECORDS APPLICATION (CONDITIONAL)

The Visa Records application MAY be implemented by an issuing State or organization. The following is conditionally REQUIRED if the optional Visa Records application has been invoked.

4.1 File Structure Summary

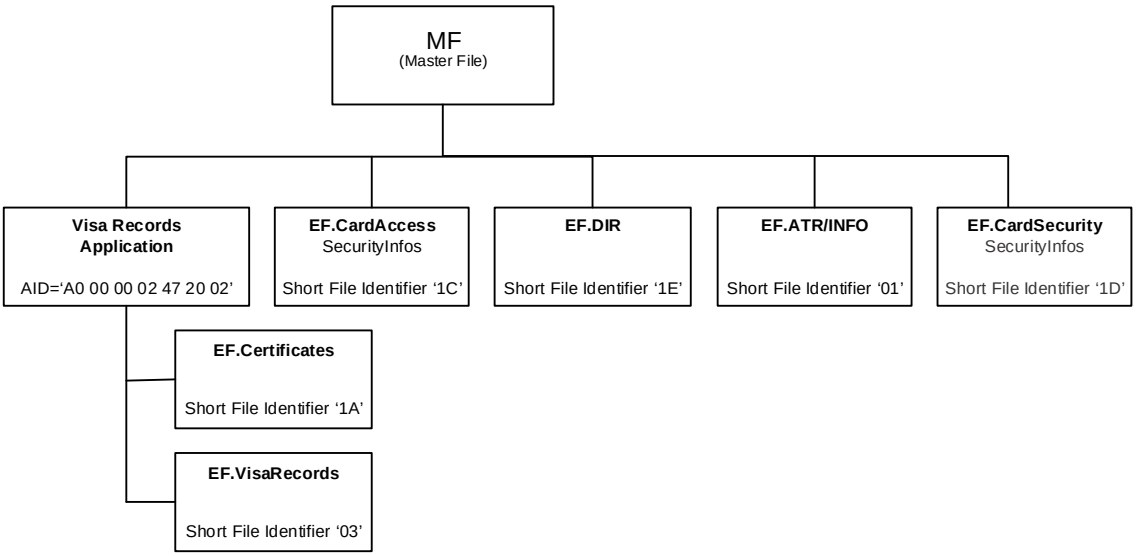


Figure 2: Visa Records Structure

4.2 EF.Certificates (REQUIRED)

The Visa Records Signer certificates are stored in EF.Certificates inside the application DF and having linear structure with records of variable size. These certificates are intended to be used by the IS to further offline validation of the digital signature for each record in the EF.VisaRecords.

Table 13: EF.Certificates

File Name	EF.Certificates
File ID	'011A'
Short EF Identifier	'1A'
Select / FMM Access	PACE+TA (Visa record authorization bit b3 according to table 22)
Read Record / Search Record Access	PACE+TA (Visa record authorization bit b3 according to table 22)
Append Record Access	PACE+TA (Visa record authorization bit b4 according to table 22)
Write / Update Record Access	NEVER
Erase Record Access	NEVER
File structure	Linear structure with records of variable size
Size	Variable

Certificate record contains a single LDS2-V Signer X.509 certificate data object. A Certificate Record MAY be referenced by one or more Visa Records.

The structure of the Certificate record in Visa Application is identical to the structure of the Certificate record in Travel Record Application specified in clause 3.2 Table 8.

Certificate records are written to EF.Certificates located under the Visa Records application DF using APPEND RECORD command. Certificate records can be read from EF.Certificates

using READ RECORD command. Certificate records MUST NOT be updated or erased. The maximum number of records in EF.Certificates under the Visa Records application DF MUST be 254.

4.3 Application Selection

The Visa Records application MUST be selected by use of the Application Identifier (AID) as a reserved DF name. The AID MUST consist of the Registered Application Identifier assigned by ISO according to [ISO/IEC 7816-5] followed by the Proprietary Application Identifier Extension (PIX) of the Visa Records application:

- The Registered Application Identifier is 'A0 00 00 02 47'
- The Visa Records application MUST use PIX = '20 02';
- The full AID of the Visa Records application is ‘A0 00 00 02 47 20 02’.

The IC MUST reject the selection of an application if the extension for this application is absent.

4.4 EF.VisaRecords (REQUIRED)

Visa Records MUST be stored in a single EF.VisaRecords having Linear Structure with Records of Variable Size.

Table 14: EF.VisaRecords

File Name	EF.VisaRecords
File ID	’0103’
Short EF Identifier	’03’
Select / FMM Access	PACE+TA (Visa record authorization bit b1 according to table 22)
Read Record / Search Record Access	PACE+TA (Visa record authorization bit b1 according to table 22)
Append Record Access	PACE+TA (Visa record authorization bit b2 according to table 22)
Write / Update Record Access	NEVER
Erase Record Access	NEVER
File structure	Linear structure with records of variable size
Size	Variable

Each Visa Record MUST contain a sequence of BER-TLV data objects (DO’s 5F28 and 71), followed by the Authenticity Token (Signature) DO and DO containing reference to LDS2-V Signer certificate in EF.Certificates. Tag 71 contains a set of DO’s (fields) listed in the table below.

Note: Interindustry tags specified in the table below are used in LDS context, so coexistent tag allocation scheme is not required.

Table 15: EF.VisaRecords Format

Tag	Tag	Content	MANDATORY/ OPTIONAL/ CONDITIONAL	Format	Example
5F28		Issuing State or organization (Copy for SEARCH RECORD)	M	F (3) A	NLD
71	Visa Record (signed info)				
	5F28	Issuing State or organization	M	F (3) A	NLD
	43	Document Type	M	F (2) A,N,S	VS
	5F71	Machine-Readable Visa of Type A	O	F (48) A,N,S	
	5F72	Machine-Readable Visa of Type B	O	F (44) A,N,S	VCD<<DENT<<ARTHUR< PHILIP<<<<<<<<<<<
	5F73	Number of entries	O	V (1) B	'01' – 'FF'
	5F74	Duration of stay (days, months, years)	O	F (3) B	'010000' – 'FFFFFF'
	5F75	Passport number	O	F (9) A,N,S	XI85935F8
	5F76	Visa Type/class/category	O	V (4) B	
	5F77	Territory Information	O	V (8) B	
	49	Place of issuance (Issuing authority)	M	V (50) A, Sp	NEW YORK
	5F25	Effective Date (Date of issuance)	M	F (8) N	20120826 (yyyymmdd)
	5F24	Expiration Date	M	F (8) N	20130826 (yyyymmdd)
	5A	Document number	M	F (9) A,N,S	XI85935F8
	5F32	Additional information (endorsements: duration, limitations, and fees paid)	O	V (50) A,N,S	Free-form text
	5B	Name of holder (full name)	M	V (50) A, Sp	VAN DER STEEN MARIANNE LOUISE
	5F33	Primary Identifier (surname)	M	V (50) A, Sp	VAN DER STEEN
	5F34	Secondary Identifier (given name)	M	V (50) A, Sp	MARIANNE LOUISE
	5F35	Sex	M	F (1) A,S	F, M, or <
	5F2B	Date of birth	M	F (8) N,S	19870814 (yyyymmdd)
	5F2C	Nationality	M	F (3) A	NLD
	5F1F	MRZ	M	V (50) A,N,S	VAN<DER<STEEN<< MARIANNE<LOUISE
	5F40	Reference to Additional Biometrics EF	O	F (2) B	'0201'
5F37		Authenticity token (Signature)	M	V (140), B	'5F' '37' Len {Signature}
5F38		Reference (record number) to LDS2-V Signer certificate in Certificates Store	M	F (1) B	'01' …'FE'

Note 1: A = Alpha character [a..z, A..Z], N = Numeric character [0..9], S = Special character ['<' ' '], B = 8-bit Binary data (any other than A, N or S), F = fixed-length field, V = variable-length field, Sp=Space.

The order of the data objects in a record is fixed. The IS MUST build up the record content using the data objects in the order specified in the table.

Each Visa Record MUST contain a digital signature (Authenticity Token) calculated over the DO'71, including Tag 71 and Length. Signature is generated by the LDS2-V Signer.

LDS2-V Signer certificates required to verify Visa Record's signature are stored in a separate EF.Certificates store located under the Visa Records application DF.

Each Visa Record MUST be appended to EF.VisaRecords using APPEND RECORD. Visa Records and MUST NOT be altered (updated) or erased. The maximum number of records allowed in EF.VisaRecords MUST be 254.

Note 2: Issuing state 3-letter code according to Doc9303-3

Note 3: Optional DO'5F40, if present, MUST contain the 2 bytes identifier of the EF within the Additional Biometrics application containing biometric data. This DO may only be used provided the Additional Biometrics application is present on the eMRTD.

5. ADDITIONAL BIOMETRICS APPLICATION (CONDITIONAL)

The Additional Biometrics application MAY be implemented by an issuing State or organization. The following is conditionally REQUIRED if the optional Additional Biometrics application has been invoked or any visa record has referenced it.

5.1 File Structure Summary

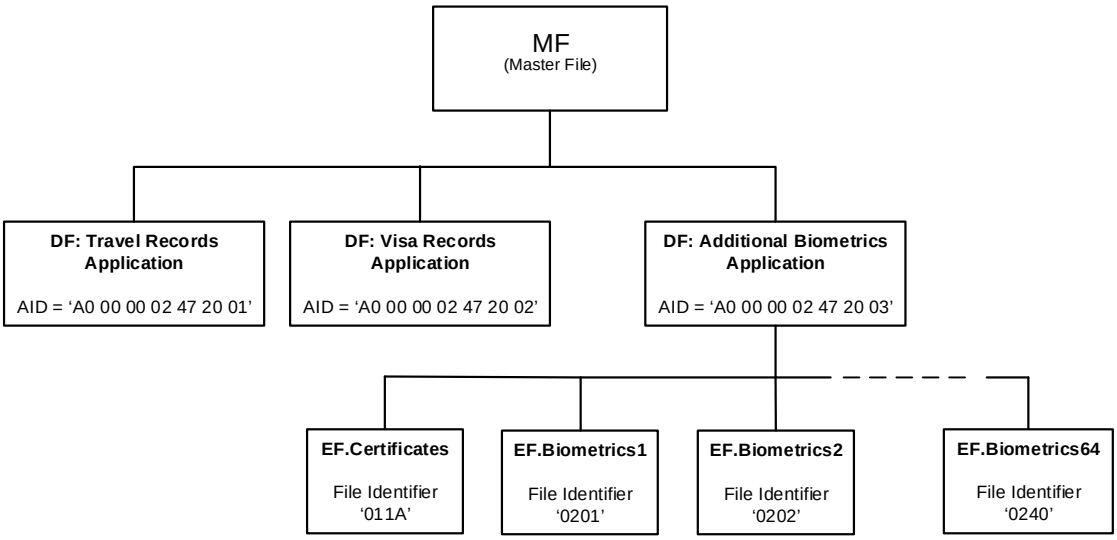


Figure 3: Additional Biometrics Application Structure

5.2 EF.Certificates (REQUIRED)

The Additional Biometrics Signer certificates are stored in EF.Certificates inside the application DF and having linear structure with records of variable size. These certificates are intended to be used by the IS to further offline validation of the digital signature in the EF.Biometrics.

Table 16: EF.Certificates

File Name	EF.Certificates
File ID	'011A'
Short EF Identifier	'1A'
Select / FMM Access	PACE+TA (Additional Biometrics authorization byte 1 bit b1 (see table 23)
Read Record/Search Record access	PACE+TA (Additional Biometrics authorization byte 1 bit b1 (see table 23)
Append Record Access	PACE+TA (Additional Biometrics authorization byte 1 bit b2 (see table 23)
Write / Update Record Access	NEVER
Erase Record Access	NEVER
File structure	Linear structure with records of variable size
Size	Variable

Certificate record contains a single Additional Biometrics Signer X.509 certificate data object. A Certificate Record MAY be referenced by one or more Additional Biometrics EF.

The structure of the Certificate record in Additional Biometrics Application is identical to the structure of the Certificate record in Travel Record Application specified in clause 3.2 Table 8.

Certificate records are written to EF.Certificates located under the Additional Biometrics application DF using APPEND RECORD command. Certificate records can be read from EF.Certificates using READ RECORD command. Certificate records MUST NOT be updated or erased. The maximum number of records in EF.Certificates under the Additional Biometrics application DF MUST be 64.

5.3 Application Selection

The Additional Biometrics application MUST be selected by use of the Application Identifier (AID) as a reserved DF name. The AID MUST consist of the Registered Application Identifier assigned by ISO according to [ISO/IEC 7816-5] followed by the Proprietary Application Identifier Extension (PIX) of the Additional Biometrics application:

- The Registered Application Identifier is 'A0 00 00 02 47'
- The Additional Biometrics application MUST use PIX = '20 03';
- The full AID of the Additional Biometrics application is ‘A0 00 00 02 47 20 03’.

The IC MUST reject the selection of an application if the extension for this application is absent.

5.4 EF.Biometrics

Additional Biometric MUST be stored under Additional Biometrics Application in Elementary Files having Transparent Structure as per [ISO/IEC 7816-4].

Each Additional Biometrics EF MAY be linked to one or more records in EF.VisaRecords in Visa Records Application (or other EFs and applications) using Additional Biometrics Elementary File Identifier.

Table 17: EF.Biometrics

File Name	EF.Biometrics
File ID	’0201’ ... ‘0240’
Short EF Identifier	N/A
Select / FMM / Read Access in LCS Deactivated	PACE+TA (AdditionalBiometrics authorization according to table 23, bits b2, b4, b6, b8 of byte 2 - 17)
Write Access in LCS Deactivated	PACE+TA (AdditionalBiometrics authorization according to table 23, bits b2, b4, b6, b8 of byte 2 - 17)
Activate Access in LCS Deactivated	PACE+TA (AdditionalBiometrics authorization according to table 23, bits b2, b4, b6, b8 of byte 2 - 17)
Select / FMM / Read Access in LCS Activated	PACE+TA (AdditionalBiometrics authorization according to table 23, bits b1, b3, b5, b7 of byte 2 - 17)
Write Access in LCS Activated	NEVER
Activate Access in LCS Activated	NEVER
Erase Access	NEVER
File structure	Transparent structure
Size	Variable

Each Additional Biometric EF MUST contain a BER-TLV data object DO’7F2E encapsulating 3 data objects - the Biometric data DO’5F2E followed by the Authenticity Token (Signature) DO’5F37’ and DO'5F38' containing the reference to an Additional Biometrics Signer certificate in EF.Certificates as shown in the table below.

The content of DO’5F2E is up to the Additional Biometrics issuer and out of scope of this specification.

The Additional Biometrics EF creation mechanism is out of scope of this specification. Issuer SHOULD pre-create a number of Additional Biometrics EFs.

Note: Interindustry tags specified in the table below are used in LDS context, so coexistent tag allocation scheme is not required.

Table 18: EF.Biometrics Format

Tag	Tag	Content	MANDATORY/ OPTIONAL/ CONDITIONAL	Format	Example
7F2E		Biometric Data Template	M		'7F' '2E' Len {DO'5F2E' || DO'5F37' || DO'5F38'}
	5F2E	Additional Biometric data	M	V, B	'5F' '2E' Len {Biometric data}
	5F37	Authenticity token (Signature)	M	V (140), B	'5F' '37' Len {Signature}
	5F38	Reference (record number) to Additional Biometrics Signer certificate in Certificates Store	M	F (1) B	'01' …'40'

Note 1: B = 8-bit Binary data (any other than A, N or S), F = fixed-length field, V = variable-length field.

The order of the data objects in EF is fixed.

Each Additional Biometrics EF MUST contain a digital signature (Authenticity Token) calculated over the DO'5F2E, including Tag and Length. Signature is generated by the Additional Biometrics Signer.

Additional Biometrics Signer certificate required to verify Additional Biometric's signature is stored in a separate EF.Certificates store located under the Additional Biometrics application DF.

Each Additional Biometrics EF MUST be written using UPDATE BINARY command (see 6.4.1).

Additional Biometrics EF MUST NOT be altered (updated) or erased. The maximum number of Additional Biometrics EFs is 64.

All possible Additional Biometrics EF names, identifiers and short identifiers are listed in Table 19.

Table 19: EF.Biometrics identifiers

EF name	EF identifier	Short EF identifier	EF name	EF identifier	Short EF identifier
EF.Biometrics1	'0201'	N/A	EF.Biometrics33	'0221'	N/A
EF.Biometrics2	'0202'	N/A	EF.Biometrics34	'0222'	N/A
EF.Biometrics3	'0203'	N/A	EF.Biometrics35	'0223'	N/A
EF.Biometrics4	'0204'	N/A	EF.Biometrics36	'0224'	N/A
EF.Biometrics5	'0205'	N/A	EF.Biometrics37	'0225'	N/A
EF.Biometrics6	'0206'	N/A	EF.Biometrics38	'0226'	N/A
EF.Biometrics7	'0207'	N/A	EF.Biometrics39	'0227'	N/A
EF.Biometrics8	'0208'	N/A	EF.Biometrics40	'0228'	N/A
EF.Biometrics9	'0209'	N/A	EF.Biometrics41	'0229'	N/A
EF.Biometrics10	'020A'	N/A	EF.Biometrics42	'022A'	N/A
EF.Biometrics11	'020B'	N/A	EF.Biometrics43	'022B'	N/A
EF.Biometrics12	'020C'	N/A	EF.Biometrics44	'022C'	N/A
EF.Biometrics13	'020D'	N/A	EF.Biometrics45	'022D'	N/A
EF.Biometrics14	'020E'	N/A	EF.Biometrics46	'022E'	N/A
EF.Biometrics15	'020F'	N/A	EF.Biometrics47	'022F'	N/A
EF.Biometrics16	'0210'	N/A	EF.Biometrics48	'0230'	N/A
EF.Biometrics17	'0211'	N/A	EF.Biometrics49	'0231'	N/A
EF.Biometrics18	'0212'	N/A	EF.Biometrics50	'0232'	N/A
EF.Biometrics19	'0213'	N/A	EF.Biometrics51	'0233'	N/A
EF.Biometrics20	'0214'	N/A	EF.Biometrics52	'0234'	N/A
EF.Biometrics21	'0215'	N/A	EF.Biometrics53	'0235'	N/A
EF.Biometrics22	'0216'	N/A	EF.Biometrics54	'0236'	N/A
EF.Biometrics23	'0217'	N/A	EF.Biometrics55	'0237'	N/A
EF.Biometrics24	'0218'	N/A	EF.Biometrics56	'0238'	N/A
EF.Biometrics25	'0219'	N/A	EF.Biometrics57	'0239'	N/A
EF.Biometrics26	'021A'	N/A	EF.Biometrics58	'023A'	N/A
EF.Biometrics27	'021B'	N/A	EF.Biometrics59	'023B'	N/A
EF.Biometrics28	'021C'	N/A	EF.Biometrics60	'023C'	N/A
EF.Biometrics29	'021D'	N/A	EF.Biometrics61	'023D'	N/A
EF.Biometrics30	'021E'	N/A	EF.Biometrics62	'023E'	N/A
EF.Biometrics31	'021F'	N/A	EF.Biometrics63	'023F'	N/A
EF.Biometrics32	'0220'	N/A	EF.Biometrics64	'0240'	N/A

6. FILE ACCESS CONDITIONS

6.1 Roles and Default Authorization Levels (REQUIRED)

Each CV certificate contains a Certificate Holder Authorization Template (CHAT) that identifies the certificate holder role (IS, DV, CVCA) and contains access rights to DG3/DG4 of the REQUIRED eMRTD Application (for legacy reasons or other national uses).

CHAT comprises a sequence of 2 objects:

1. An object identifier specifying the terminal type and the format of the template [TR-03110]:
   id-roles OBJECT IDENTIFIER ::= {bsi-de applications(3) mrtd(1) 2}
   id-IS OBJECT IDENTIFIER ::= {id-roles 1}
2. A discretionary data object (tag '53') containing bit-encoded role and read-only access rights of the certificate holder according to the following table:

Table 20: Default CHAT Authorization

	Description	Byte 1							
		b8	b7	b6	b5	b4	b3	b2	b1
Role	CVCA	1	1						
	DV (domestic)	1	0						
	DV (foreign)	0	1						
	IS	0	0						
Read Access	RFU								
	RFU								
	RFU								
	RFU								
	DG4 (Iris)							1	
	DG3 (Finger)								1

Note: The eMRTD MUST ignore the value of RFU bits in the Certificate Holder Authorization.

6.2 Application Authorization Levels (REQUIRED)

Certificate holder authorizations for each LDS2 application are encoded in CV-certificate-extensions (one extension per application). Certificate extension is a discretionary template (‘73’) comprising 2 data objects - an Authorization Object Identifier (tag ‘06’) for a specific application and a discretionary data object (tag ‘53’) containing bit-encoded access rights of the certificate holder to specified application.

To determine the effective authorization of a certificate holder, the eMRTD chip calculates a bitwise Boolean ’and’ of the access rights contained in the certificate extensions of the IS Certificate and referenced DV and CVCA Certificates.

For Travel Records application the Authorization Object Identifiers and access right encoding are:

id-icao-lds2-travelRecords OBJECT IDENTIFIER ::= {id-icao-lds2 1}
id-icao-lds2-travelRecords-access OBJECT IDENTIFIER ::= {id-icao-lds2-travelRecords 3}

Table 21: Authorizations for Travel Records Application

	Description	Byte 1							
		b8	b7	b6	b5	b4	b3	b2	b1
Access rights	RFU								
	RFU								
	RFU								
	RFU								
	Append EF.Certificates					1			
	Read/Search/Select/FMM EF.Certificates						1		
	Append EF.EntryRecords/ExitRecords							1	
	Read/Search/Select/FMM EF.EntryRecords/ExitRecords								1

For Visa Records application the Authorization Object Identifiers and access right encoding are:

```
id-icao-lds2-visaRecords          OBJECT IDENTIFIER     ::= {id-icao-lds2 2}
id-icao-lds2-visaRecords-access   OBJECT IDENTIFIER     ::= {id-icao-lds2-visaRecords 3}
```

Table 22: Authorizations for Visa Records Application

	Description	Byte 1							
		b8	b7	b6	b5	b4	b3	b2	b1
Access rights	RFU								
	RFU								
	RFU								
	RFU								
	Append EF.Certificates					1			
	Read/Search/Select/FMM EF.Certificates						1		
	Append EF.VisaRecords							1	
	Read/Search/Select/FMM EF.VisaRecords								1

For Additional Biometrics application the Authorization Object Identifiers and access right encoding are:

id-icao-lds2-additionalBiometrics OBJECT IDENTIFIER ::= {id-icao-lds2 3}
id-icao-lds2-additionalBiometrics-access OBJECT IDENTIFIER::= {id-icao-lds2-additionalBiometrics 3}

Table 23: Authorizations for Additional Biometrics Application

	Description	EF identifier	Authorizations							
			b8	b7	b6	b5	b4	b3	b2	b1
Byte 1	RFU									
	RFU									
	RFU									
	RFU									
	RFU									
	RFU									
	Append EF.Certificates	011A							1	
	Select/FMM/Read/Search EF.Certificates	011A								1
Byte 2	Select/FMM/Write/Activate/Read EF.Biometrics1 in Deactivated LCS	0201	1							
	Select/FMM/Read EF.Biometrics1 in Activated LCS	0201		1						
	Select/FMM/Write/Activate/Read EF.Biometrics2 in Deactivated LCS	0202			1					
	Select/FMM/Read EF.Biometrics2 in Activated LCS	0202				1				
	Select/FMM/Write/Activate/Read EF.Biometrics3 in Deactivated LCS	0203					1			
	Select/FMM/Read EF.Biometrics3 in Activated LCS	0203						1		
	Select/FMM/Write/Activate/Read EF.Biometrics4 in Deactivated LCS	0204							1	
	Select/FMM/Read EF.Biometrics4 in Activated LCS	0204								1
	...									
Byte 17	Select/FMM/Write/Activate/Read EF.Biometrics61 in Deactivated LCS	023D	1							
	Select/FMM/Read EF.Biometrics61 in Activated LCS	023D		1						
	Select/FMM/Write/Activate/Read EF.Biometrics62 in Deactivated LCS	023E			1					
	Select/FMM/Read EF.Biometrics62 in Activated LCS	023E				1				
	Select/FMM/Write/Activate/Read EF.Biometrics63 in Deactivated LCS	023F					1			
	Select/FMM/Read EF.Biometrics63 in Activated LCS	023F						1		
	Select/FMM/Write/Activate/Read EF.Biometrics64 in Deactivated LCS	0240							1	
	Select/FMM/Read EF.Biometrics64 in Activated LCS	0240								1

Note 1 - See table 17 for a mapping of the authorizations to the EF.Biometrics life cycle state.

Note 2 - The eMRTD MUST ignore the value of RFU bits in the Certificate Holder Authorization.

Note 3 - Issuing State or organization MUST NOT issue terminal certificates with Write/Activate authorizations to the IS that are only supposed to read Additional Biometrics and not supposed to write them.

6.3 Records Handling

Travel Records, Visa Records and Certificates MUST be stored in EF under the respective applications and having Linear Structure with Records of Variable Size.

Records within each EF MUST be referenced by a record number. Each record number MUST be unique and sequential (zero is out of the scope of this specification).

Within each EF supporting a linear structure, the record numbers MUST be sequentially assigned when appending, such as in the order of creation; the first record (number one) is the first created record.

The following [ISO/IEC 7816-4] commands MUST be used for records access:

- APPEND RECORD Writing/appending of Travel Records, Visas, Certificates
- READ RECORD Reading of one or more Travel Records, Visas, Certificates
- SEARCH RECORD Search of one or more Travel Records, Visas, Certificates

Note: Acronyms used in this sub-clause are defined in ISO/IEC 7816-4.

6.3.1 APPEND RECORD command

The command initiates the writing of a new record at the end of a linear structure.

Table 24: APPEND RECORD Command

CLA	'00' / '0C'
INS	'E2'
P1	'00' (any other value is invalid)
P2	See Table 26
L_c field	Length of the command data field
Data field	Record to be appended
L_e field	Absent

Table 25: APPEND RECORD Response

Data field	Absent
SW1-SW2	'9000', Checking or Execution error '6A84' Not enough memory space in the file '67 00' Wrong length (the record to be appended is longer than the specified maximum length)

Table 26: Coding of P2 in the APPEND RECORD Command

b8	***b7***	***b6***	***b5***	***b4***	***b3***	***b2***	***b1***	***Meaning***
x	*x*	*x*	*x*	*x*	-	-	-	*Short EF identifier*
-	-	-	-	-	*0*	*0*	*0*	*Any other value is RFU*

6.3.2 READ RECORD Command

The response data field returns the contents of the addressed specified record(s) [or the beginning part of one record] within the respective EF.

Figure 4 illustrates the response data field. The comparison of N_r with the TLV structure indicates whether the unique record (read one record) or the last record (read all records) is incomplete, complete or padded.

Table 27: READ RECORD Command

CLA	'00' / '0C'	
INS	'B2'	
P1	Record number ('00' references the current record)	
P2	See Table 29	
L_c field	Absent	
Data field	INS = 'B2'	Absent
L_e field	Maximum number of bytes to be read encoded as extended length field; Le = '00 00 00' (any other value is out of the scope of the specification)	

Table 28: READ RECORD Response

Data field	Data read
SW1-SW2	'9000', Checking or Execution error, '6A83' (Record not found)

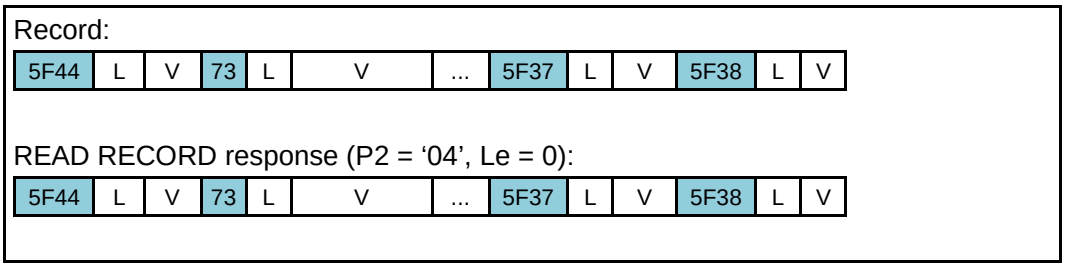
Table 29 - Coding of P2 with the READ RECORD Command

b8	***b7***	***b6***	***b5***	***b4***	***b3***	***b2***	***b1***	***Meaning***
x	*x*	*x*	*x*	*x*	*-*	*-*	*-*	*Short EF identifier*
-	*-*	*-*	*-*	*-*	*1*	*x*	*x*	***Record number in P1***
-	*-*	*-*	*-*	*-*	*1*	*0*	*0*	*— Read record P1*
-	*-*	*-*	*-*	*-*	*1*	*0*	*1*	*— Read all records from P1 up to the last*
					1	*1*	*1*	*RFU*

If the L_e field contains only bytes set to '00', then the command should read completely either the single requested record, or the requested sequence of records, depending on bits 3, 2 and 1 of P2 and within the limit of maximum supported length for extended L_e field.

Note: The READ RECORD command with short length fields is out of the scope of this specification

Case a — Complete read of one record (the Le field contains only bytes set to '00')



Case b — Read several records up to the file end (the L_e field contains only bytes set to '00')

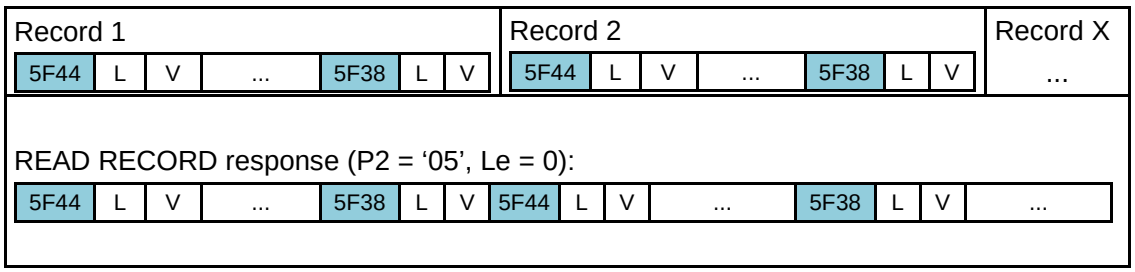


Figure 4: Response data fields

6.3.3 SEARCH RECORD Command

The command initiates an enhanced multiple records search on records stored within the respective EF. In case of the enhanced multiple records search option the data field contains Record handling DO'7F76' defining the search window - offset within the record and the number of bytes to compare. The response data field returns the Record handling DO'7F76' containing one or more DO'02' containing number of record matching the search criteria within the designated EF. In case of the enhanced multiple records search option the command MUST set the first record matching the search criteria as current record.

In an EF supporting records of variable size with linear structure, the search MAY NOT take into account the records with a search window shorter than the search string.

Table 30: SEARCH RECORD Command

CLA	'00' / '0C'
INS	'A2'
P1	'00'
P2	See Table 32
L_c field	Length of command data field
Data field	Record handling DO'7F76' (See Table 33)
L_e field	'00' (short length) or '00 00' (extended length)

Table 31: SEARCH RECORD Response

Data field	Record handling template DO'7F76'containing one file reference DO'51' with one or more integer DO'02' containing record number matching the search criteria
SW1-SW2	'9000', Checking or Execution error or Warning '6282' (Unsuccessful search)

Note 1 - The response data field may be absent if no match is found.

Table 32: Coding of P2 for the SEARCH RECORD Command

b8	*b7*	*b6*	*b5*	*b4*	*b3*	*b2*	*b1*	*Meaning*
1	*1*	*1*	*1*	*1*	*0*	*0*	*0*	*Multiple record handling*
- Any other value is RFU.								

Table 33: Record handling template for enhanced multiple record search

Tag	Value				Notes
'7F76'					Record handling DO
	Tag	**Value**			
	'51'	File identifier or short EF identifier			File reference DO
	'A1'				Search configuration template
		Tag	**Value**		
		'80'	'00' / '30'		Search configuration parameter: - search in record number ascending order - step-width for the search: byte-wise - search termination: '00' – Search all addressed records '30' - Terminate search after first matching
		'B0'			Search window template
			Tag	**Value**	
			'02'	Offset	
			'02'	Number of bytes	
	Tag	**Value**			
	'A3'				Search string template
		Tag	**Value**		
		'B1'			
			Tag	**Value**	
			'81'	Search string	

Note 1: An empty offset DO in the search window template is not supported.

Note 2: If the search window template makes use of the value '00' for the number of bytes, the eMRTD chip MUST search all bytes from the offset in the records.

Note 3: The SEARCH RECORD command supports only the DOs specified in table 33. This implies that the SEARCH RECORD command supports exactly 1 file reference DO in the record handling DO and exactly 1 search string in the search string template. The command MAY ignore additional DOs or answer with an error code if additional DOs are used.

6.4 Transparent Files Handling

The Additional Biometrics transparent EFs are created by the eMRTD issuer in Operational Deactivated state (creation mechanism is out of scope of this specification). In Deactivated state the EF can be selected, written, updated and read with appropriate authorizations. See 6.2, Tables 17 and 23 for details on authorizations.

The following [ISO/IEC 7816-4] commands MUST be used for writing and reading Additional Biometrics transparent EF:

- UPDATE BINARY Writing of Additional Biometrics (Table 34)
- READ BINARY Reading of Additional Biometrics

The following [ISO/IEC 7816-9] command MUST be used for activating the transparent EF after writing and optional reading and verification are successfully finished:

- ACTIVATE Activating of Additional Biometrics EF

In Activated state the EF can be selected and read with appropriate authorizations (related to the Activated state), but can't be written (appended or updated) with any authorization.

The File and Memory Management (FMM) command MUST be used before writing to determine if there is enough available memory space in the EF.

The IS MUST use the following writing sequence for the EF.Biometrics:

1) The first UPDATE BINARY (odd INS) command MUST contain the following DO's in the data field:
   - DO'54 containing the offset '00';
   - DO'53 which MAY contain the first block of the EF. This DO MAY be empty ('53 00');
   - Proprietary DO'C0 indicating the total EF size (memory size to allocate);

   The eMRTD MAY use the EF size information in DO'C0 for the internal memory allocation (e.g. for explicit dynamic memory allocation). If the eMRTD doesn't support the EF size information DO (ex., memory has been allocated statically by the issuer, or eMRTD supports implicit dynamic EF memory reallocation), then the eMRTD MAY ignore the DO'C0, proceed with writing of the first block of the EF and return '9000', or it MAY return the '6A80' (incorrect parameter in the command data field) error.

   If the eMRTD returns any error in response to UPDATE BINARY with the proprietary DO'C0, then the IS MUST send the standard [ISO/IEC 7816-4] UPDATE BINARY (odd INS) command with zero offset DO'54 and DO'53, without the DO'C0.
2) Subsequent UPDATE BINARY (odd INS, without DO'C0) commands SHOULD use the offset n+1 where n denotes the number of bytes written so far to the EF.Biometrics. I.e. the terminal SHOULD sequentially write the EF data without a gap or overlap between the two consecutive UPDATE BINARY commands.
3) READ BINARY command MAY be used after any UPDATE BINARY command to verify the data written to the EF.
4) The ACTIVATE command MUST finalize EF.Biometrics personalization by permanently disabling writing into the EF.

6.4.1 UPDATE BINARY Command

A contactless IC which supports the Additional Biometrics Application MUST support the UPDATE BINARY command with odd INS byte 'D7' according to the Table 34.
The value of the BER-TLV Offset Data Object in the command data field specifies the offset; the value of the BER-TLV Discretionary Data Object in the command data field specifies the data to be written; the value of the optional BER-TLV File Size Data Object in the command data field specifies the total EF size. The length fields of these BER-TLV data objects should be encoded as short as possible.

Table 34: UPDATE BINARY Command with odd INS

CLA	'0C' - '8C'
INS	'D7'
P1	File identifier
P2	'00 00' identifies the current EF
Lc	Length of the command data field
Data	Offset Data Object (tag '54') || Discretionary Data Object (tag '53') || File Size Data Object (tag 'C0') (optional)
Le	absent

Table 35: UPDATE BINARY Response

Data field	Absent
SW1-SW2	'9000', Checking or Execution error '6A84' (Not enough memory space in the file) '6A80' Incorrect parameters in the command data field (ex., DO'C0 not supported) '6982' Security status not satisfied: The EF.Biometrics is in EF LCS Activated

If the Inspection System does not follow the UPDATE BINARY sequence as specified in clause 6.4 (i.e. the first UPDATE BINARY does not start at offset 0), the eMRTD chip MAY terminate the UPDATE BINARY command with an error.

6.4.2 ACTIVATE Command

The ACTIVATE command initiates the transition of the currently selected Additional Biometrics EF from the Deactivated life cycle state (LCS) to the Activated LCS.

Table 36: ACTIVATE Command

CLA	'0C'
INS	'44'
P1	'00'
P2	'00'
Lc	Absent
Data field	Absent
Le	Absent

Table 37: ACTIVATE Response

Data field	Absent
SW1-SW2	'9000', Checking or Execution error

After successful execution of this command, the currently selected EF.Biometrics MUST be switched to the Activated LCS. In case an error occurs (SW different from '9000'), the currently selected EF.Biometrics MUST remain in the Deactivated state.

Immediately after successful execution of this command (Status Word = '9000'), the effective authorization required to perform an action on the EF.Biometrics MUST be the one corresponding to the Activated state (according to the table 17). The effective authorization corresponding to the Deactivated state MUST NOT raise any access right on the EF.Biometrics anymore.

6.5 Memory Management

6.5.1 File and Memory Management Command

FILE AND MEMORY MANAGEMENT (FMM) command initiates a query of the used or free memory size for the addressed EF. This command is provided for eMRTD as proprietary. This command may be used for checking the available free space for the addressed EF before writing or appending. Also this command may be used for getting the last appended record number for reading. P1 indicates the EF addressing method - current EF or file reference DO'51' can be used. P2 indicates the content of the query. The total number of bytes in the addressed EF with transparent or record structure and the number of existing or remaining records for the addressed record EF are provided. The total number of bytes comprises bytes available in the EF without any structural information. This number excludes any structural information that may be required by the eMRTD chip. The assumption for the number of remaining records is that the size of all remaining records is maximum.
After a successful FMM command, the referenced EF becomes the current EF.

Table 38: FILE AND MEMORY MANAGEMENT (FMM) command

CLA	'8C'	Proprietary command with secure messaging
INS	'5F'	FILE AND MEMORY MANAGEMENT
P1	See Table 39	
P2	See Table 40	
Lc	Absent for encoding Nc = 0, present for encoding Nc > 0	
Data field	P1 = '00'	Absent
	P1 = '01'	File reference DO'51' (See Table 7 in ISO/IEC 7816-4:2013)
Le	'00'	Expected size of response data field

P1 specifies the EF selection method. P2 contains a bit mask specifying which information MUST be included in the response.

Table 39: Coding of P1 in the FFM command

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	Current EF
0	0	0	0	0	0	0	1	File reference DO'51 in the command data field
- Any other value is RFU.								

Table 40: Coding of P2 in the FFM command

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
-	-	-	-	-	-	-	1	Total number of bytes in the addressed EF
-	-	-	-	-	-	1	-	Number of remaining records in the addressed record EF
-	-	-	-	-	1	-	-	Number of existing records in the addressed record EF
x	x	x	x	x	-	-	-	0000 (any other value is RFU)
- Any other value is RFU.								

Table 41: Coding of DO'51 in the FMM command data field

Tag	Length	Value
'51'		
	1	Short EF identifier (bits b8 to b4 encode a number from one to thirty; bits b3 to b1 are set to 000)
	2	EF identifier

The FMM command response contains a set of DO's representing requested file and memory size information.

Table 42: FMM command response

Data field	Absent or control information according to P2. See Table 43.
SW1-SW2	'9000', Checking or Execution error as per Table 6 in [ISO/IEC 7816-4]

Table 43: File and Memory management

Tag	**Length**	**Value**		
'7F78'	Var.	File and memory management DOs		
		Tag	**Len**	**Value**
		81	Var	Total number of bytes in the addressed EF
		82	Var	Number of remaining records in the addressed record EF
		83	Var	Number of existing records in the addressed record EF

Note 1: The eMRTD chip MUST return only the Data objects in the file and memory management DO that are requested by means of P2.

Note 2: The FMM response data is valid only for the specified EF.FMM response data from different EFs may not be independent, e.g. if different EFs share the available memory. The IS should take this into account if combining FMM response data of different EFs.

Note 3: Secure Messaging tag '85' MUST be used for the FMM command data.

6.6 Object Identifiers

6.6.1 Legacy and New Application Object Identifiers Summary

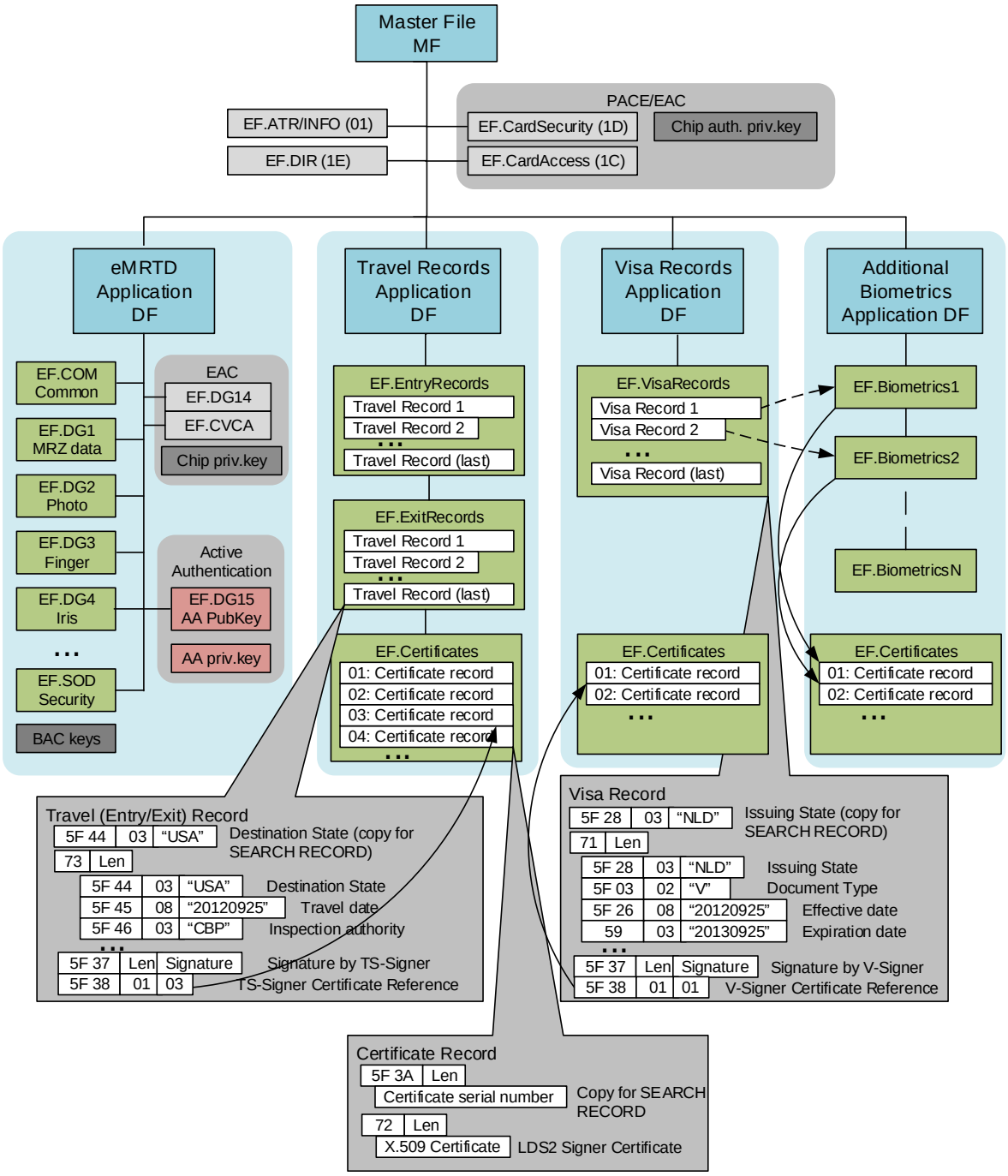
Table 44: LDS1.7, LDS1.8 and LDS2 OIDs

Object Identifier	Value	Comments
id-icao	2.23.136	ICAO OID
id-icao-mrtd	id-icao 1	eMRTD OID
id-icao-mrtd-security	id-icao-mrtd 1	
id-icao-ldsSecurityObject	id-icao-mrtd-security 1	LDS security object
id-icao-mrtd-security-cscaMasterList	id-icao-mrtd-security 2	CSCA master list
id-icao-mrtd-security-cscaMasterListSigningKey	id-icao-mrtd-security 3	
id-icao-mrtd-security-documentTypeList	id-icao-mrtd-security 4	document type list
id-icao-mrtd-security-aaProtocolObject	id-icao-mrtd-security 5	Active Authentication protocol
id-icao-mrtd-security-extensions	id-icao-mrtd-security 6	CSCA name change
id-icao-mrtd-security-extensions-nameChange	id-icao-mrtd-security-extensions 1	
id-icao-mrtd-security-extensions-documentTypeList	id-icao-mrtd-security-extensions 2	DS document type
id-icao-DeviationList	id-icao-mrtd-security 7	Defect List Base OIDs
id-icao-DeviationListSigningKey	id-icao-mrtd-security 8	
id-icao-lds2	id-icao-mrtd-security 9	LDS2 Object Identifiers
id-icao-lds2-travelRecords	id-icao-lds2 1	Travel Records application base OID
id-icao-lds2-travelRecords-application	id-icao-lds2-travelRecords 1	Travel Records AID
id-icao-lds2-travelRecords-signing	id-icao-lds2-travelRecords 2	LDS2-TS signer certificate
id-icao-lds2-travelRecords-access	id-icao-lds2-travelRecords 3	Authorization certificate extension
id-icao-lds2-visaRecords	id-icao-lds2 2	Visa Records application base OID
id-icao-lds2-visaRecords-application	id-icao-lds2-visaRecords 1	Visa Records AID
id-icao-lds2-visaRecords-signing	id-icao-lds2-visaRecords 2	LDS2-V signer certificate
id-icao-lds2-visaRecords-access	id-icao-lds2-visaRecords 3	Authorization certificate extension
id-icao-lds2-additionalBiometrics	id-icao-lds2 3	Additional Biometrics base OID
id-icao-lds2-additionalBiometrics-application	id-icao-lds2-additionalBiometrics 1	Additional Biometrics AID
id-icao-lds2-additionalBiometrics-signing	id-icao-lds2-additionalBiometrics 2	LDS2-B signer certificate
id-icao-lds2-additionalBiometrics-access	id-icao-lds2-additionalBiometrics 3	Authorization certificate extension
id-icao-spoc	id-icao-mrtd-security 10	SPOC Object Identifiers
id-icao-spocClient	id-icao-spoc 1	Client
id-icao-spocServer	id-icao-spoc 2	Server

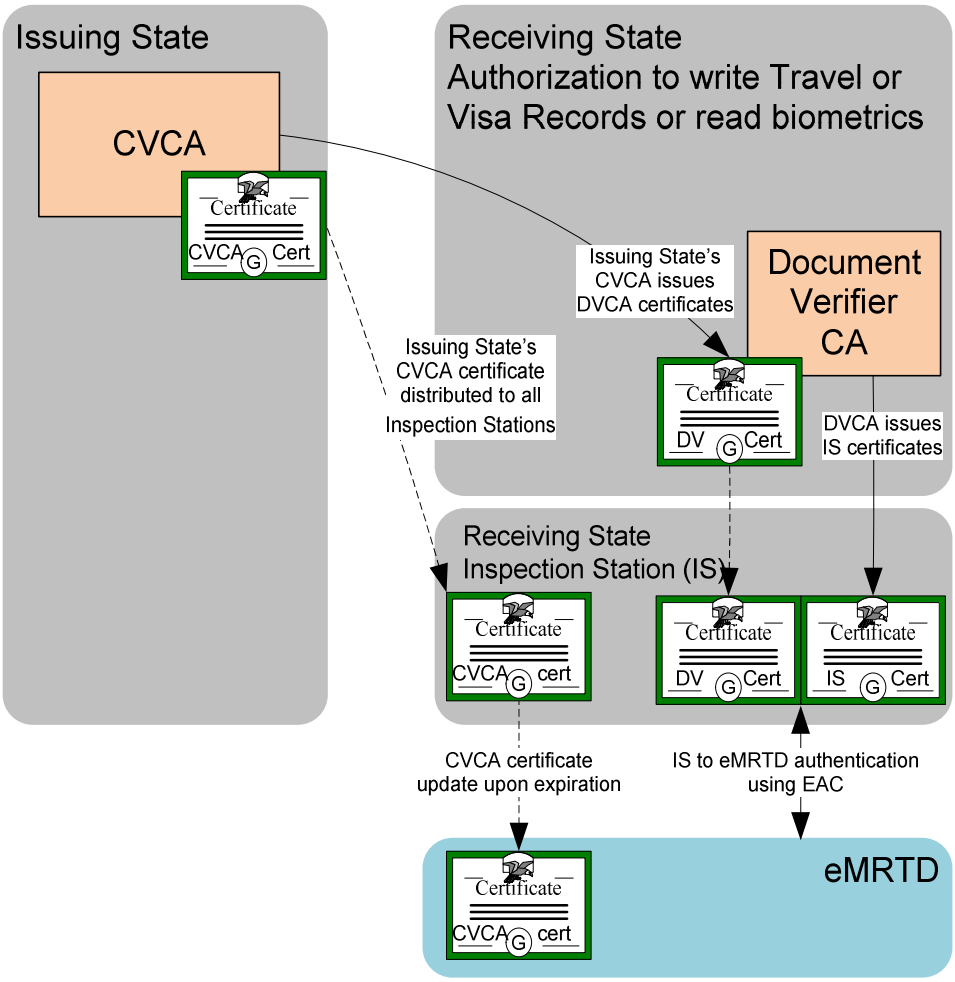
6.6.2 ASN.1 Specifications

```
-- LDS2 Travel Records application Object Identifiers
id-icao-lds2-travelRecords                  OBJECT IDENTIFIER    ::= {id-icao-lds2 1}
id-icao-lds2-travelRecords-application      OBJECT IDENTIFIER    ::= {id-icao-lds2-travelRecords 1}
id-icao-lds2-travelRecords-signing  OBJECT IDENTIFIER    ::= {id-icao-lds2-travelRecords 2}
id-icao-lds2-travelRecords-access   OBJECT IDENTIFIER    ::= {id-icao-lds2-travelRecords 3}

-- LDS2 Visa Records application Object Identifiers
id-icao-lds2-visaRecords                    OBJECT IDENTIFIER    ::= {id-icao-lds2 2}
id-icao-lds2-visaRecords-application        OBJECT IDENTIFIER    ::= {id-icao-lds2-visaRecords 1}
id-icao-lds2-visaRecords-signing    OBJECT IDENTIFIER    ::= {id-icao-lds2-visaRecords 2}
id-icao-lds2-visaRecords-access     OBJECT IDENTIFIER    ::= {id-icao-lds2-visaRecords 3}

-- LDS2 Additional Biometrics application Object Identifiers
id-icao-lds2-additionalBiometrics   OBJECT IDENTIFIER    ::= {id-icao-lds2 3}
id-icao-lds2-additionalBiometrics-application OBJECT IDENTIFIER    ::= {id-icao-lds2-additionalBiometrics 1}
id-icao-lds2-additionalBiometrics-signing OBJECT IDENTIFIER    ::= {id-icao-lds2-additionalBiometrics 2}
id-icao-lds2-additionalBiometrics-access OBJECT IDENTIFIER     ::= {id-icao-lds2-additionalBiometrics 3}
```

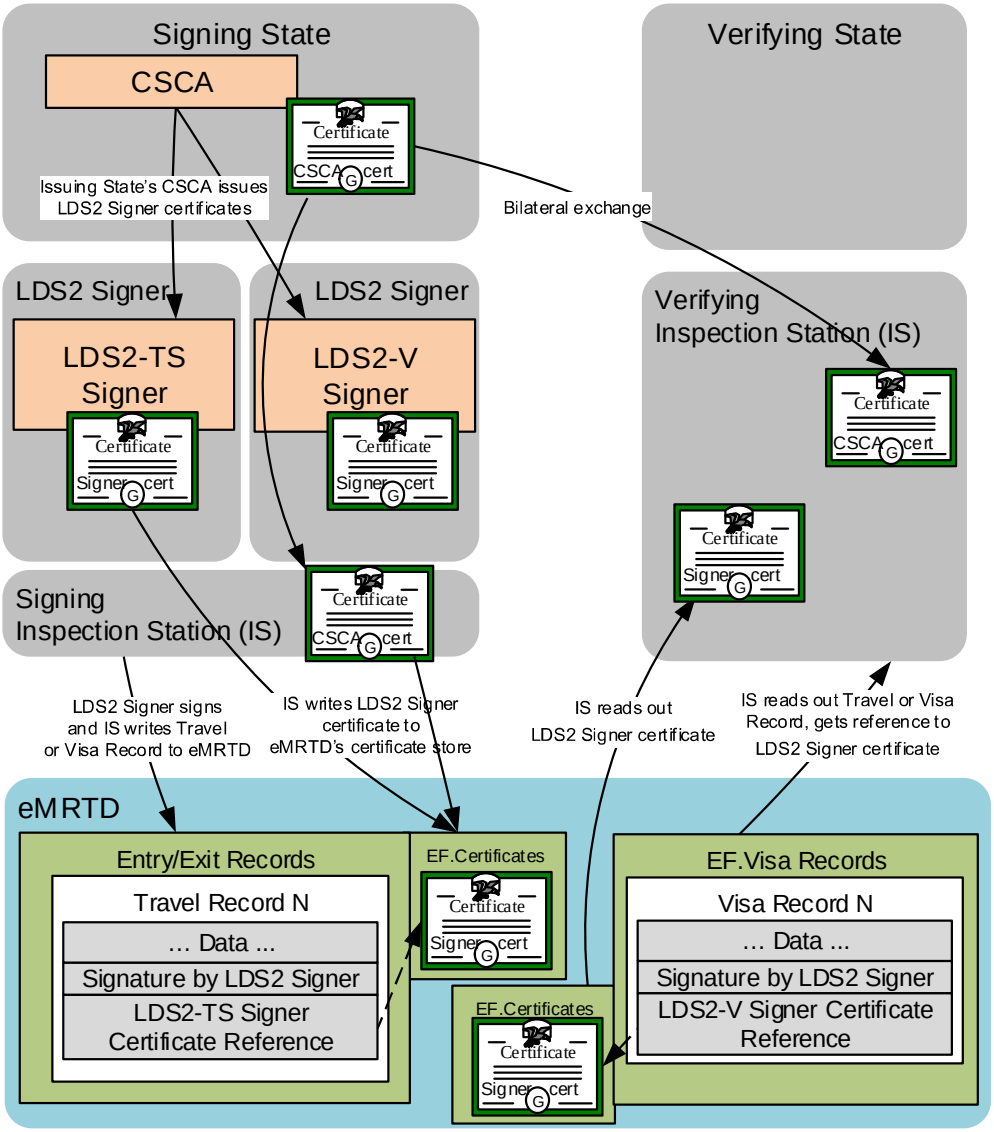
Annex A FILE STRUCTURES SUMMARY



Annex B LDS AUTHORIZATION SUMMARY



Annex C LDS DIGITAL SIGNATURE SUMMARY



Annex D EXAMPLE READING TRAVEL RECORDS

1) FMM command retrieving the number of Entry Records

CLA	INS	P1	P2	Lc	Data	Le
80	5E	01	04	04	51 02 01 01	00

CLA: Proprietary class / no secure messaging
INS: FMM
P1: 01 - EF identifier in command data field
P2: 04 - Return existing number of records in a record EF
Lc: 04
Data: DO'51 containing Entry Records EF identifier '0101'
Le : 00 (Short Le)

Response: File and Memory Management DO representing the number of records in the EF

Data	SW1-SW2
7F78 03 83 01 FD	90 00

The DO in the response data contains the last record number which can be used in the next READ RECORD command (P1).

Ex., last record number '00' means that there are no records in this file, response 'FD' means that number of records is 253 (maximum number of records is 254).

2a) READ RECORD command retrieving the last Travel Record from the retrieved list

The following command can be used to retrieve a single record using record number returned by the FMM command:

CLA	INS	P1	P2	Le
00	B2	FD	04	00 00 00

CLA : Interindustry class / no secure messaging
INS : READ RECORD(S)
P1 : Record number from the previous command's response
P2 : Record number in P1 / read record P1
Le : 00 00 00 (Extended Le) - read entire record

Response: Record - 253 (0xFD)

Data	SW1-SW2
5F44 Len <Data> || 73 Len <Data> || 5F37 Len <Data> || 5F38 Len <Data>	90 00

2b) READ RECORD retrieving last 2 Travel Records from retrieved the list

The following command can be used to retrieve 2 (or more) records from the list returned by FMM command. Reading several records in one APDU exchange improves performance. The number of records that can be retrieved by a single command can be determined from extended length information in EF.ATR/INFO and maximum size of Travel Record.

CLA	INS	P1	P2	Le

00	B2	FC	05	00 00 00

CLA : Interindustry class / no secure messaging
INS : READ RECORD(S)
P1 : Decremented Record number from the FMM response (253 - 1 = 252 = 'FC')
P2 : Record number in P1 / read all records from P1 up to the last
Le : 00 00 00 (Extended Le) - read entire record

Response: Last 2 records - 252 (0xFC) and 253 (0xFD)

Data	SW1-SW2
5F44 Len <Data> || 73 Len <Data> || 5F37 Len <Data> || 5F38 Len <Data> || 5F44 Len <Data> || 73 Len <Data> || 5F37 Len <Data> || 5F38 Len <Data>	90 00

Annex E EXAMPLE SEARCHING RECORDS BY STATE

1) SEARCH RECORD command searching Travel Record(s) by Destination State

CLA	INS	P1	P2	Lc	Data	Le
00	A2	00	F8	Var	7F 76 Len 51 01 01 A1 0B 80 01 00 B0 06 02 01 03 02 01 03 A3 07 B1 05 81 03 xx xx xx	00

CLA: Interindustry class / no secure messaging
INS: SEARCH RECORD(S)
P1: record number = 00
P2: Search through multiple EFs
Lc: length of command data field
Data: DO'7F76' - Record handling DO
DO'51' - File reference DO (EF.EntryRecords short identifier '01')
DO'A1' - Search configuration template
DO'80' - Search configuration parameter: '00' (search all records)
DO'B0' - Search window template
DO'02' - Offset: '03'
DO'02' - Number of bytes: '03'
DO'A3' - Search string template
DO'B1' - Search string DO
DO'81' - Search string (country code): xx xx xx
Le: 00 (Short Le)

Response: DO'7F76' – Record handling DO
DO'51' - EF.EntryRecords short identifier '01'
One or more DO'02' containing matching record numbers

Data	SW1-SW2
7F 76 Len 51 01 01 02 01 03 02 01 04	90 00

Annex F EXAMPLE WRITING TRAVEL RECORD AND CERTIFICATE

1) SEARCH RECORD command searching EF.Certificates by a Certificate Serial Number

IS checks if LDS2-TS Signer certificate with required serial numbers exists in EF.Certificates. The following command can be used for searching certificates:

CLA	INS	P1	P2	Lc	Data	Le
00	A2	00	F8	Var	7F 76 Len 51 01 1A A1 0B 80 01 30 B0 06 02 01 03 02 01 {Search string size} A3 Len B1 Len 81 Len xx xx .. xx xx	00

CLA: Interindustry class / no secure messaging
INS: SEARCH RECORD(S)
P1: record number = 00
P2: Search through multiple EFs
Lc: length of command data field
Data: DO'7F76' - Record handling DO
DO'51' - File reference DO (EF.Certificates short identifier '1A')
DO'A1' - Search configuration template
DO'80' - Search configuration parameter: '30' (stop if record found)
DO'B0' - Search window template
DO'02' - Offset: '03'
DO'02' - Number of bytes: Search string size
DO'A3' - Search string template
DO'B1' - Search string DO
DO'81' - Search concatenation of country code and certificate serial number: xx xx .. xx xx
Le: 00 (Short Le)

Response: DO'7F76' - Record handling DO
DO'51' - EF.Certificates short identifier '1A'
DO'02' - contains matching record number

Data	SW1-SW2
7F 76 06 51 01 1A 02 01 **01**	90 00

or warning code 62 82 if no record matches the search criteria:

SW1-SW2
62 82

If an EF.Certificate record matches the search criteria, the IS can optionally use the returned record number ('01') in a READ RECORD command to check whether the certificate is the correct one. If no EF.Certificate record matches the search criteria, the IS writes the certificate into EF.Certificates using the APPEND RECORD command in step 2) and finally writes the entry record using step 3).

2) APPEND RECORD command writing Certificate

IS writes LDS2-TS Signer certificate into EF.Certificates. The following command can be used for writing certificates:

CLA	INS	P1	P2	Lc	Data	Le
00	E2	00	D0	00 XX XX	5F3A Len {certificate serial number} || 72 Len {X.509 certificate}"	Absent

CLA: Interindustry class / no secure messaging
INS: APPEND RECORD
P1: 00 (any other value is invalid)
P2: short EF identifier (=0x1A)
Lc: Record length (Extended Lc)
Data: Record data

Response: success or error code

SW1-SW2
90 00

3) APPEND RECORD command writing Travel Record

IS generates Travel Record using reference to LDS2-TS Signer certificate and writes it into EF.EntryRecords using the following command:

CLA	INS	P1	P2	Lc	Data	Le
00	E2	00	08	00 XX XX	5F44 Len {destination state} || 73 Len {Entry travel record} || 5F37 Len {Signature} || 5F38 Len {Cert Ref}	Absent

CLA: Interindustry class / no secure messaging
INS: APPEND RECORD
P1: 00 (any other value is invalid)
P2: short EF identifier (=0x01)
Lc: Record length (Extended Lc)
Data: Record data

Response: success or error code

SW1-SW2
90 00