Belgian Electronic Identity Card content

Document Change History

Version	Date	Description
1.0		Internal version
1.1	14-01-2003	Version for distribution
1.2	29-01-2003	Changes in PKCS#15 files
1.3	10-02-2003	Changes in PKCS#15 files
1.4	21-02-2003	Changes in PKCS#15 files Specified the address file format
1.5	27-02-2003	Adapted the address file max. length
1.6	18-03-2003	Added the ***Preference*** file format Added ***W*** in the "sex" field from the ID file Changes in PKCS#15 files
1.7	24-03-2003	Corrected national number length from the ID file
1.8	04-04-2003	Changed birth date type to UTF-8 the ID file and added the dot (.) as separator for German
1.9	09-04-2003	Added birth months table Corrected the ***PKCS#15 Authority*** flag for the certificates
2.0	25-04-2003	Changed ***EF(ID#Address)*** length
2.1	26-05-2003	Added "document type" field from the ID file
2.3	03-09-2003	Corrected card number length in ID file Corrected address file length
2.4	26-11-2003	Minor corrections
2.5	19-12-2003	Added ***TokenInfo*** version detail Added optional version in files Added envisioned max. length for fields
2.6	24-12-2003	Typo correction in AID Added default values for version
2.7	20-01-2004	Generalised to include Applet V2 specs Added picture resolution
2.8a	16-03-2004	Detailed some explanations Removed unused files/keys

Table of content

1. Scope

This standard describes the specifications of the Belgian Electronic Identity Card files and objects. The third party applications that may be added on the card are not covered by this document.

Only the ***DF(BelPIC)*** and the ***DF(ID)*** are covered in this document.

1.1. Terms and definitions

For the purposes of this document, the following definitions apply:

authentication object directory file	optional elementary file containing information about authentication objects known to the PKCS#15 application
binary coded decimal	Number representation where a number is expressed as a sequence of decimal digits and then each decimal digit is encoded as a four bit binary number. Example – Decimal 92 would be encoded as the eight bit sequence 1001 0010.
cardholder	person for whom the card was issued
card issuer	organization or entity that issues smart cards and card applications
certificate directory file	optional elementary file containing information about certificate known to the PKCS#15 application
data object directory file	optional elementary file containing information about data objects known to the PKCS#15 application
dedicated file	file containing file control information, and, optionally, memory available for allocation, and which may be the parent of elementary files and/or other dedicated files
directory (DIR) file	optional elementary file containing a list of applications supported by the card and optional related data elements
elementary file	set of data units or records that share the same file identifier, and which cannot be a parent of another file
file identifier	2-byte binary value used to address a file on a smart card
master file	mandatory unique dedicated file representing the root of the structure NOTE – The MF typically has the file identifier $3F00_{16}$
object directory file	elementary file containing information about other directory files in the PKCS #15 application

path	concatenation of file identifiers without delimitation NOTE – If the path starts with the MF identifier ($3F00_{16}$), it is an absolute path; otherwise it is a relative path. A relative path shall start with the identifier '$3FFF_{16}$' or with the identifier of the current DF.
personal identification number (PIN)	4 to 8 digit number entered by the cardholder to verify that the cardholder is authorized to use a functionality of the card
private key directory file	optional elementary file containing information about private keys known to the PKCS#15 application
provider	authority who has or who obtained the right to create the MF or a DF in the card
public key directory file	optional elementary file containing information about public keys known to the PKCS#15 application
record	string of bytes which can be handled as a whole by the card and referenced by a record number or by a record identifier
private key directory file	optional elementary file containing information about private keys known to the PKCS#15 application
token	portable device capable of storing persistent data

1.2. Symbols, abbreviated terms and document conventions

1.2.1. Symbols

DF(x) Dedicated file *x*

EF(x) Elementary file *x*

1.2.2. Abbreviated terms

For the purposes of this document, the following abbreviations apply:

AID	Application Identifier
AODF	Authentication Object Directory File
BCD	Binary-Coded Decimal
CDF	Certificate Directory File
DER	Distinguished Encoding Rules
DF	Dedicated File (directory)
DODF	Data Object Directory File
EF	Elementary File
MF	Master File
ODF	Object Directory File
PIN	Personal Identification Number
PrKDF	Private Key Directory File
PuKDF	Public Key Directory File

2. Versions

2.1. Applet version

Some data – mainly the PINs - are linked to the version of the applet used. When applicable, we will refer in this document to "***Applet version x*** ". This version is actually the "***applet interface version*** " received with the command "***GetCardData*** ".

2.2. Card content versions

The main version of the card content is located in the file ***TokenInfo*** (see 5.3.1).

Two versions are available:

- ***Electrical personalisation version***: this number increases at every change – even minor – in the personalisation format
- ***Electrical personalisation interface version***: this number increases when a change in the personalisation format introduces an incompatibility of the file structure with the old format.

An application can thus use newer cards that have additional files, because the interface version will be the same.

Note that individual files may have an internal version number corresponding to the data in the file. The "***Electrical personalisation interface version***" should be used to check the file structure, the internal file version should be used to check the fields format in the file.

Note that these versions are not linked at all to the applet (see 2.1).

2.3. Electrical Personalisation Versions History

Version (Hexa)	Interface Version (Hexa)	Date	Description
00	00		▪ Initial version
01	00	01-01-2004	▪ New ATR: 3B 98 94 40 **0A** A5 03 01 01 01 AD 13 10 ▪ PKCS#15 files size adaptations ▪ Address file length extended to 117 bytes

3. Security Objects

3.1. PIN

Two types of PIN exist:

- ***Permanent***: once the PIN has been validated, its granted access right is permanent until current access condition specifically change (card reset, logoff, external authenticate…). This is the usual way of using a PIN.
- ***Transient***: the PIN access right granted is available for the next command only.

	PIN reference (Java Object)		Type (transient/permanent)	PUK	PIN_{Reset}	Max. trials before blocked
	Applet Version 1	Applet Version 2				
PUK	03	03	transient	-	-	12
PIN_{Reset}	02	02	transient	-	-	10
Activate	84	84	transient	-	-	15
Authentication	01	01	permanent	03	02	3
Non-repudiation	01	04	transient	03	02	3

3.2. Keys and Certificates

3.2.1. Keys and certificates relationships

	Private Key (Java Object)	Public Key (Java Object)	X.509 Certificates (Transparent file)
Basic	PrK#1		
Authentication	PrK#2	In Cert#2	Cert#2
Non-repudiation	PrK#3	In Cert#3	Cert#3
Citizens CA		In Cert#4	Cert#4
Root			Cert#6
Government CA			
Role CA		PuK#7	
RRN			Cert#8

Each key or certificate is indicated by means of a reference number (#). Some keys do not have a corresponding private/public key or certificate.

Remark: Puk#7 is not readable on the card.

3.2.2. Keys Access Control

Command on RSA key	*Reference (hex)*	*Generate Key*	*Get Key*	*Put Key*	*Erase Key*	*Activate Key*	*Deactivate Key*	*PSO: Compute Digital Signature*	*Internal Authenticate*	*External Authenticate*
PrK#1 (basic)	81	NEV	✗	✗	NEV	NEV	NEV	NEV	ALW	✗
PrK#2 (authentication)	82	CTV(3)	✗	✗	CTV(3)	CTV(3)	CTV(3)	CHV(PIN_{Auth})	NEV	✗
PrK#3 (non-repudiation)	83	CTV(3)	✗	✗	CTV(3)	CTV(3)	CTV(3)	CHV($PIN_{Non-Rep}$)	NEV	✗
PuK#7 (CA role)	87	✗	✗	CTV(8)	CTV(8)	CTV(8)	CTV(8)	✗	✗	ALW

✗	Not possible (forbidden by the card Operating System/applet)
NEV	Never
ALW	Always
CHV(x)	Card Holder Verification with PIN '**x**'
CTV(x)	Certificate Verification with Role '**x**'

3.2.3. Certificates and role identifiers

In compliance with ISO/IEC FDIS 7816–9 (sub–clause 7.4) card verifiable certificates will be applied in public key based authentication procedures. Such certificates contain certificate holder authorisations (e.g. role identifiers). This role identifier is used in the security conditions to be fulfilled for access to data or functions.

These certificates will be X.509 compliant, and thus will not use the ISO tags.

In the BELPIC application following roles are defined:

- ***Role R01***
  In this role the card architecture can be remotely updated and extended. This role is applied to delete application files in the EID card.

- ***Role R02***
  In this role the card architecture can be remotely updated and extended. This role is applied only to create new application files in the EID card.

- ***Role R03***
  In this role the card can be instructed to generate new keys pairs for authentication (PrK#2 and Puk#2) and/or non-repudiation (PrK#3 and PuK#3) and to store the generated private keys (Prk#2 and PrK#3) in the EID card.

- ***Role R04***
  In this role the card can be instructed to store new certificates for authentication (Cert#2) non-repudiation (Cert#3) and CA (Cert#4) in the EID card.

- ***Role R05***
  In this role the card can be instructed to store a new root certificate Cert#6 in the EID card.

- ***Role R06***
  Unused.

- ***Role R07***
  In this role the card can be instructed to update the ID files EF(ID#Adresse), EF(SGN#Adresse) in the EID card.

- ***Role R08***
  In this role the card can be instructed to store a new Role CA key PuK#7 in the EID card.

- ***Role R09***
  In this role the card can be instructed to block the EID card with the ***BLOCK CARD*** command.

The roles are retrieved from the certificates after an ***External Authentication with Certificate Verification***. A ***Mutual Authentication*** with ***Secure Messaging*** – with or without encryption – is advised to secure the connection.

4. Files

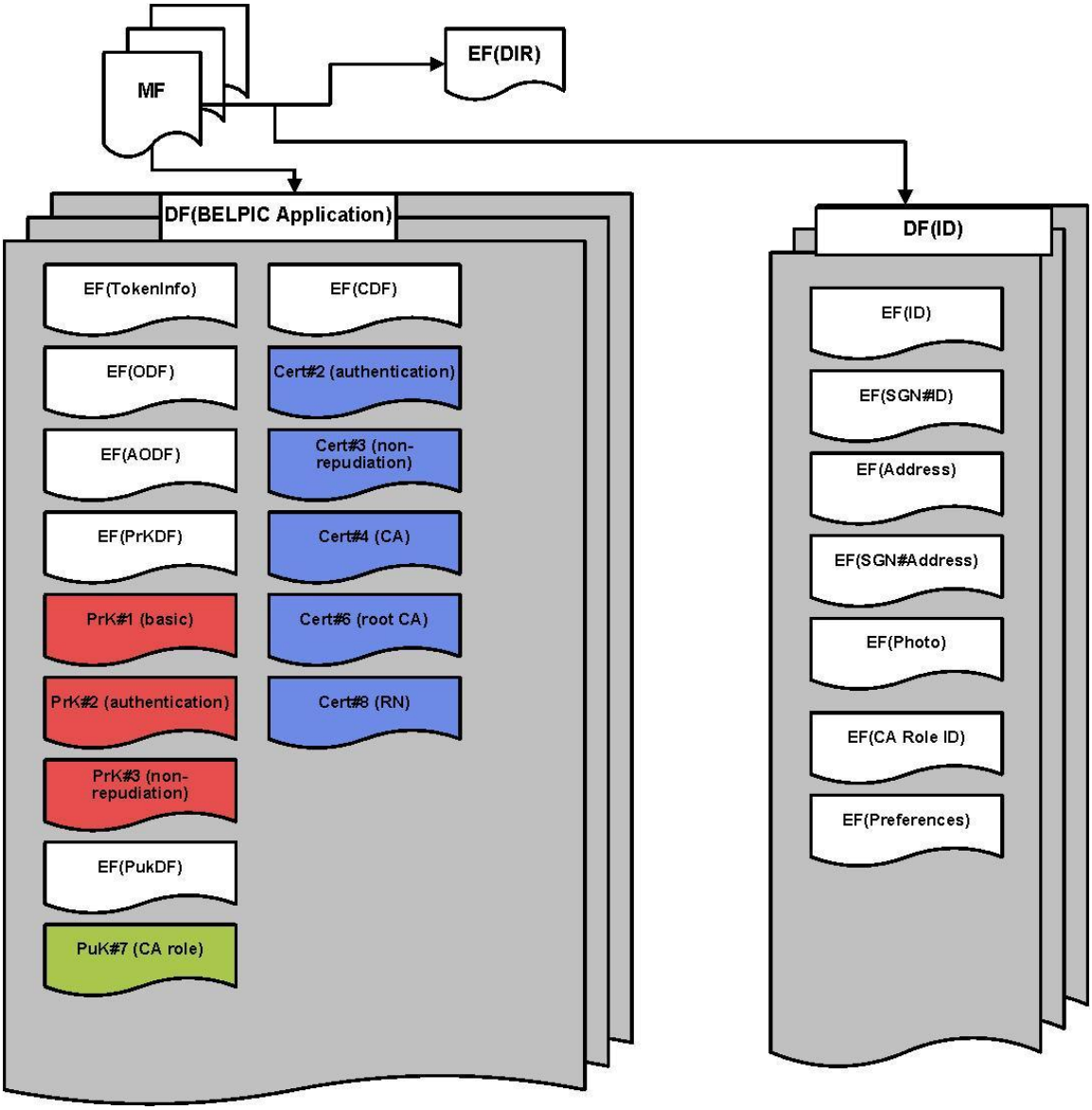
All EF file types are transparent, as defined in ISO/IEC 7816–4, sub–clause 5.1.3.

Files in the EID card is organised into a hierarchical structure according to ISO/IEC 7816–4.

The electronic signature and electronic identification applications are separated in the card by means of two application directories: ***DF(BELPIC)*** and ***DF(ID)***. Other applications DF may be added later.

4.1. File structure

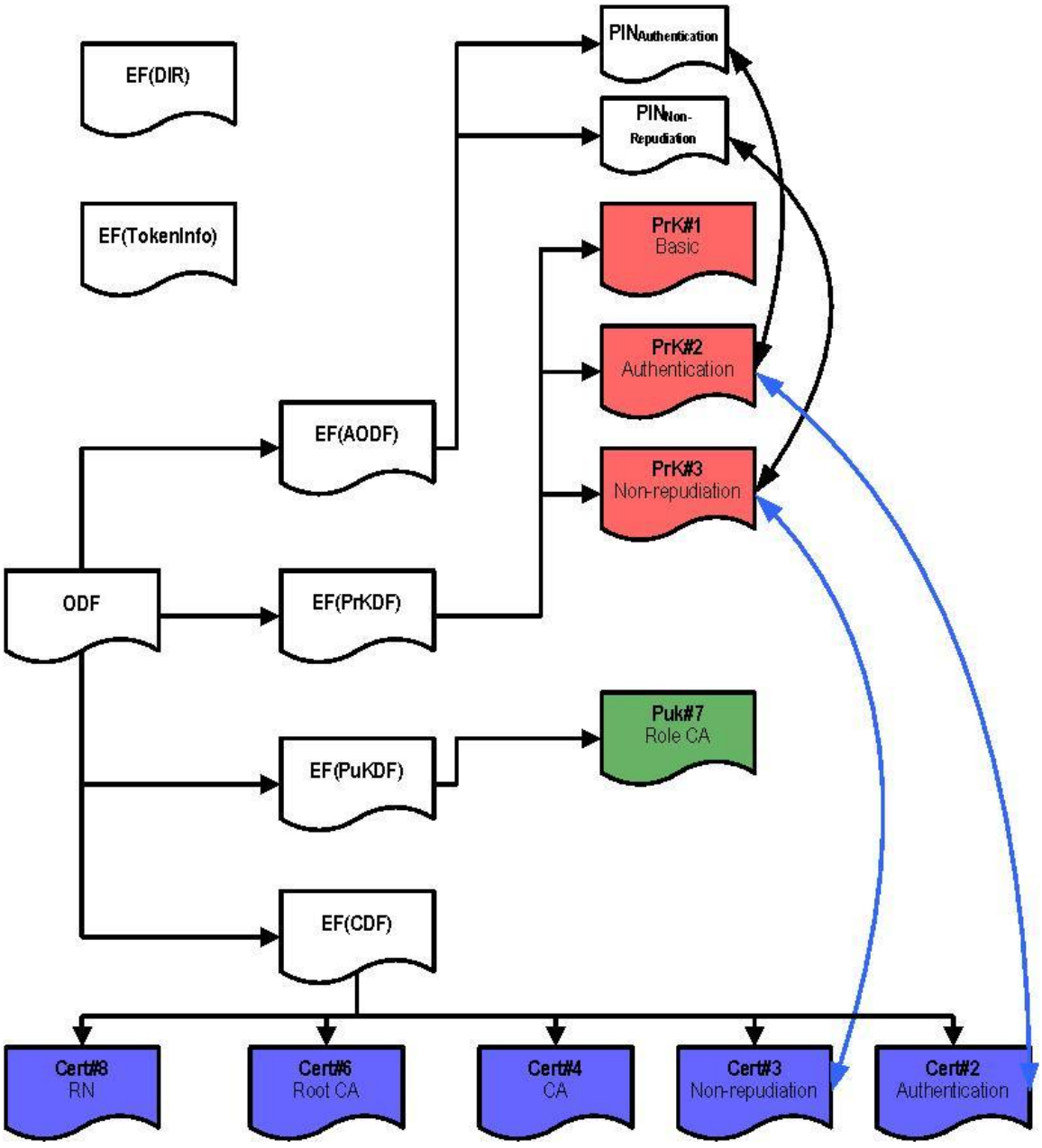
The file structure of the card is described in the figure below.



4.2. PKCS#15 files

The content of the ***DF(BELPIC)*** application directory files is compliant with PKCS#15 v1.1.

A directory file, ***EF(DIR)***, containing the AID (ISO/IEC 7816–5) for each application in the EID card is present in the ***Master File***. The PKCS#15 AID, and other AID, are also directly selectable.



The purpose of the figure above is to show the relationship between certain files ***EF(ODF)***, ***EF(AODF)***, ***EF(PrKDF)*** and ***EF(CDF)*** in the ***DF(BELPIC)***. Directory. ***EF(ODF)*** points to other EF.

EF(PrKDF) contains cross-reference pointers to authentication objects (PIN) used to protect access to the keys. Arrows between PIN and Private Keys indicate this.

Some certificates (***Cert#2*** & ***Cert#3***) contain a public key whose private key also resides on the card, so this certificates contain the same identifier as the corresponding private key. Arrows between Certificates and Private Keys indicate this.

4.3. Files identifiers and permissions

	Command	**File type Access method Meaning**
MF/DF	Activate	The MF or DF can be activated
	Deactivate	The MF ore DF can be deactivated
	Create	In the MF or DF files can be created
	Delete	In the MF or DF files can be deleted
EF	Activate	The object can be activated
	Deactivate	The object can be deactivated
	Read Binary	The content of the object can be read
	Update Binary	The content of the object can be updated
	Erase Binary	The content of the object can be erased

	Reference (hexa)	Activate	Deactivate	Select	Create	Delete	Read Binary	Update Binary	Erase Binary
MF	3F00	CHV($PIN_{activate}$)	NEV	ALW	CTV(2)	CTV(1)	✗	✗	✗
EF(DIR)	2F00	NEV	NEV	ALW	✗	✗	ALW	CTV(1) CTV(2)	CTV(1) CTV(2)
- DF(BELPIC): PKCS#15 ***AID***: A000000177504B43532D3135	DF00	NEV	NEV	ALW	NEV	NEV	✗	✗	✗
EF(ODF)	5031	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(TokenInfo)	5032	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(AODF)	5034	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(PrKDF)	5035	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(CDF)	5037	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(Cert#2) (authentication)	5038	CTV(4)	CTV(3)	ALW	✗	✗	ALW	CTV(4)	CTV(4)
EF(Cert#3) (non-repudiation)	5039	CTV(4)	CTV(3)	ALW	✗	✗	ALW	CTV(4)	CTV(4)
EF(Cert#4) (CA)	503A	CTV(4)	CTV(3)	ALW	✗	✗	ALW	CTV(4)	CTV(4)
EF(Cert#6) (root)	503B	CTV(5)	CTV(5)	ALW	✗	✗	ALW	CTV(5)	CTV(5)
EF(Cert#8) (RN)	503C	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
- DF(ID)	DF01	NEV	NEV	ALW	NEV	NEV	✗	✗	✗
EF(ID#RN)	4031	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV

	Reference (hexa)	Activate	Deactivate	Select	Create	Delete	Read Binary	Update Binary	Erase Binary
EF(SGN#RN)	4032	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(ID#Address)	4033	NEV	NEV	ALW	✗	✗	ALW	CTV(7)	CTV(7)
EF(SGN#Adress)	4034	NEV	NEV	ALW	✗	✗	ALW	CTV(7)	CTV(7)
EF(ID#Photo)	4035	NEV	NEV	ALW	✗	✗	ALW	NEV	NEV
EF(PuK#7) (Role CA)	4038	NEV	NEV	ALW	✗	✗	ALW	CTV(8)	CTV(8)
EF(Preferences)	4039	NEV	NEV	ALW	✗	✗	ALW	CHV($PIN_{authentication}$)	CHV($PIN_{authentication}$)

✗ Not possible (forbidden by the card Operating System/applet)

NEV Never

ALW Always

CTV(x) Certificate Verification with Role '**x**'

5. PKCS#15 information detail

5.1. PKCS#15 application selection

The EID card support direct application selection as defined in ISO/IEC 7816–4, Section 9 and ISO/IEC 7816–5, Section 6 (the full AID is to be used as parameter for a 'SELECT FILE' command).

The operating system of the card keeps track of the currently selected application and only allow the commands applicable to that particular application while it is selected.

When several PKCS#15 applications resides on one card, they are distinguished by their object identifier in their application template in ***EF(DIR)***.

5.2. MF directory contents

5.2.1. EF(DIR)

This file contains all application templates as defined in ISO/IEC 7816–5. Each application template (tag '61'H) for a PKCS#15 application must at least contain the following Data Objects:

- ***Application Identifier***: tag '4F', UTF-8 encoded
- ***Path***: tag '51', DER-encoded

Other tags from ISO/IEC 7816–5 may, at the application issuer's discretion, be present as well. In particular, it is recommended that application issuers include the following Data Objects:

- ***Application Label***: tag '50', UTF-8 encoded
- ***Discretionary Data Objects***: tag '51', DER-encoded

Encoding	ASN.1 Syntax
	Belpic (One entry per application)
	-- [APPLICATION 1] IMPLICIT SEQUENCE
	Application ID
4F 0C A0 00 00 01 77 50 4B 43 53 2D 31 35	-- [APPLICATION 15] IMPLICIT OCTET STRING
	Label
50 06 42 45 4C 50 49 43	-- [APPLICATION 16] IMPLICIT UTF8 String -- 'BELPIC'
	Path
51 04 ***3F 00 DF 00***	-- [APPLICATION 17] IMPLICIT OCTET STRING *-- MF/Belpic*
	Discretionary Data Object
73 05	-- [APPLICATION 19] IMPLICIT SEQUENCE
	ObjectID
06 03 60 38 02	-- OBJECT IDENTIFIER -- belgian citizen (2.16.56.2)

5.3. DF(BELPIC) Application directory contents

This DF is the directory of the BelPIC application.

No operation is available on this data file.

5.3.1. EF(TokenInfo)

This file contains generic information about the token as such and it's capabilities. This information includes the token serial number, file types for object directory files, algorithms implemented on the token, etc.

OFFSET	Encoding	ASN.1 Syntax
00	30 27	-- SEQUENCE
		Version
02	02 01	-- INTEGER
04	00	-- 0
		Serial Number
05	04 10	-- OCTET STRING
07	**{16 bytes}**	**-- chip serial number**
		Application Label
17	**80** 06	**-- [0] Label IMPLICIT UTF8 String**
19	42 45 4C 50 49 43	-- "BELPIC"
		TokenFlags
1F	03 02	-- BIT STRING
21	04 30	-- prnGeneration(2), eidCompliant (3)
23	**9E** 04	**-- [30] BELPIC Application IMPLICIT INTEGER**
25	**{4 bytes}**	**-- Version**

Version bytes:

- Graphical personalisation version (default = 0)
- Electrical personalisation version (default = 0)
- Electrical personalisation interface version[1] (default = 0)
- Reserved for future use (0)

[1] This is used to indicate to an application which file system organisation is used. This value only changes when a new version is no more compatible with the previous one.

5.3.2. EF(ODF)

The Object Directory File (***ODF***) is a transparent elementary file, which contains pointers to other elementary files (***PrKDF***, ***PuKDF***, ***CDF***, ***AODF***) of the EID card. The information is presented in ASN.1 syntax according to PKCS #15.

An application using the EID card must use this file to determine how to perform security services with the card.

OFFSET	Encoding	ASN.1 Syntax
00	**A0 0A**	**-- [0] Private Keys**
		Path
02	30 08	-- SEQUENCE
		Path
04	04 06	-- OCTET STRING
06	***3F 00 DF 00 50 35***	***-- MF/Belpic/PrKDF***
0C	**A4 0A**	**-- [4] Certificates**
		Path
0E	30 08	-- SEQUENCE
		Path
10	04 06	-- OCTET STRING
12	***3F 00 DF 00 50 37***	***-- MF/Belpic/CDF***
18	**A8 0A**	**-- [8] Authentication Objects**
		Path
1A	30 08	-- SEQUENCE
		Path
1C	04 06	-- OCTET STRING
1E	***3F 00 DF 00 50 34***	***-- MF/Belpic/AODF***

Remark: The AODF path will be removed in a future version, as it is the default path.

5.3.3. EF(AODF)

This elementary file (Authentication Object Directory File) contains generic authentication object attributes such as allowed characters, PIN length, PIN padding character, etc. It also contains the pointers to the authentication objects themselves (in the case of PINs, pointers to the DF in which the PIN file resides). The authentication objects are used to control access to other objects such as keys. The content of this file is according to PKCS#15.

OFFSET	Encoding	ASN.1 Syntax
		PIN Cardholder (One entry per PIN)
00	30 33	-- SEQUENCE
		Common Object Attributes
02	30 0F	-- SEQUENCE
		Label
04	0C 09	-- UTF8 String
06	42 61 73 69 63 20 50 49 4E	-- "Basic PIN"
		Common Object Flags
0F	03 02	-- BIT STRING
11	06 C0	-- private(0), modifiable(1)
		Common Authentication Object Attributes
13	30 03	-- SEQUENCE
		Authority ID
15	04 01	-- OCTET STRING
17	01	-- '01'
18	A1 1B	-- [1] Pin Attributes
1A	30 19	-- SEQUENCE
		Pin Flags
1C	03 02	-- BIT STRING
1E	*02 0C*	*-- initialized(4), needs-padding(5)*
		PinType
20	0A 01	-- ENUMERATED
22	*00*	*-- bcd(0)*
		Min Length
23	02 01	-- INTEGER
25	04	-- 4
		Stored Length
26	02 01	-- INTEGER
28	*08*	*-- 8*
29	80 01	-- [0] Pin Reference IMPLICIT INTEGER
2B	*01*	*-- 1*
		Pad Char
2C	04 01	-- OCTET STRING
2E	*FF*	*-- 'FF'*
		Path
2F	30 04	-- SEQUENCE
		Path
30	04 02	-- OCTET STRING
33	*3F 00*	*-- 'MF'*

5.3.4. EF(PrKDF)

This transparent elementary file (Private Key Directory File) contains general key attributes such as labels, intended usage, identifiers etc. It also contains the pointers to the keys themselves. The keys reside in the BELPIC application directory on the card.

OFFSET	Encoding	ASN.1 Syntax
		Private Authentication Key
00	30 3A	-- SEQUENCE
		Common Object Attributes
02	30 17	-- SEQUENCE
		Label
04	0C 0E	-- UTF8 String
06	41 75 74 68 65 6E 74 69 63 61 74 69 6F 6E	-- "Authentication"
		Common Object Flags
14	03 02	-- BIT STRING
16	06 C0	-- private(0), modifiable(1)
		Authority ID
18	04 01	-- OCTET STRING
1A	01	-- '01'
		Common Key Attributes
1B	30 0F	-- SEQUENCE
		Identifier
1D	04 01	-- OCTET STRING
1F	02	-- '02'
		KeyUsageFlags
20	03 02	-- BIT STRING
22	05 20	-- Sign(2)
		Key Access Flags
24	03 02	-- BIT STRING
26	03 B8	-- sensitive(0)
		alwaysSensitive(2)
		neverextractable(3)
		local(4)
28	02 02	*KeyReference*
2A	00 82	-- INTEGER
		-- '82'
2C	A1 0E	-- [1] Private RSA Key Attributes
		Path
2E	30 0C	-- SEQUENCE
		Path
30	30 06	-- SEQUENCE
		Path
32	04 04	-- OCTET STRING
34	3F 00 DF 00	-- MF
		Modulus Length
38	02 02	-- INTEGER
3A	04 00	-- 1024

OFFSET	Encoding	ASN.1 Syntax
		Private Non-repudiation Key
3C	30 39	-- SEQUENCE
		Common Object Attributes
3E	30 15	-- SEQUENCE
		Label
40	0C 09	-- UTF8 String
42	53 69 67 6E 61 74 75 72 65	-- "Signature"
		Common Object Flags
4C	03 02	-- BIT STRING
4E	06 C0	-- private(0), modifiable(1)
		Authority ID
50	04 01	-- OCTET STRING
52	01	-- '01'
		UserConsent
53	02 01	-- INTEGER
55	01	-- 15
		Common Key Attributes
56	30 10	-- SEQUENCE
		Identifier
58	04 01	-- OCTET STRING
5A	03	-- '02'
		KeyUsageFlags
5B	03 03	-- BIT STRING
5D	06 00 40	-- NonRepudiation(9)
		Key Access Flags
60	03 02	-- BIT STRING
62	03 B8	-- sensitive(0) alwaysSensitive(2) neverextractable(3) local(4)
64	02 02	***KeyReference***
66	00 83	-- INTEGER -- '83'
68	**A1 0E**	**-- [1] Private RSA Key Attributes**
		Path
6A	30 0C	-- SEQUENCE
		Path
6C	30 06	-- SEQUENCE
		Path
6E	04 04	-- OCTET STRING
70	**3F 00 DF 00**	**-- MF**
		Modulus Length
74	02 02	-- INTEGER
76	**04 00**	**-- 1024**

5.3.5. EF(PuKDF)

This transparent elementary file (Public Key Directory File) can be regarded as directories of public keys known to the PKCS #15 application. They contain general key attributes such as labels, intended usage, identifiers, etc. When applicable, it contains cross-reference pointers to authentication objects used to protect access to the keys. Furthermore, they contain pointers to the keys themselves. Private keys corresponding to public keys must share the same identifier. The keys reside in the BELPIC application directory on the card.

As no private keys are used through the PKCS#15 interface, this file does not exist.

5.3.6. EF(CDF)

This transparent elementary file contains attributes and pointers to the authentication certificate (Cert #2), non-repudiation signature certificate (Cert #3) and CA certificate (Cert#4). Information in this file contains certificate attributes such as labels, key identifiers, pointers to certificate files etc. The format of the file is specified in PKCS#15.

OFFSET	Encoding	ASN.1 Syntax
		Authentication Certificate
00	30 32	-- SEQUENCE
		Common Object Attributes
02	30 17	-- SEQUENCE
		Label
04	0C 0E	-- UTF8String
06	41 75 74 68 65 6E 74 69 63 61 74 69 6F 6E	-- "Authentication"
		Common Object Flags
14	03 02	-- BIT STRING
16	06 C0	-- private(0), modifiable(1)
		AuthID
18	04 01	-- OCTET STRING
1A	01	-- '01'
		Common Certificate Attributes
1B	30 06	-- SEQUENCE
		Identifier
1D	04 01	-- OCTET STRING
1F	02	-- '02'
20	83 01	**--[3] ImplicitTrust IMPLICIT BOOLEAN**
22	00	-- False
23	**A1 0C**	**-- [1] 509CertificateAttributes**
		Path
25	30 0A	-- SEQUENCE
		Path
27	30 08	-- SEQUENCE
		Path
29	04 06	-- OCTET STRING
2B	**3F 00 DF 00 50 38**	**-- MF/Belpic/Cert#2(auth)**

OFFSET	Encoding	ASN.1 Syntax
		Non-Repudiation Certificate
31	30 2A	-- SEQUENCE
		Common Object Attributes
33	30 12	-- SEQUENCE
		Label
35	0C 09	-- UTF8String
37	53 69 67 6E 61 74 75 72 65	-- "Signature"
		Common Object Flags
40	03 02	-- BIT STRING
42	06 C0	-- private(0), modifiable(1)
		AuthID
44	04 01	-- OCTET STRING
46	01	-- '01'
		Common Certificate Attributes
47	30 06	-- SEQUENCE
		Identifier
49	04 01	-- OCTET STRING
4B	03	-- '03'
4C	**83 01**	**--[3] ImplicitTrust IMPLICIT BOOLEAN**
4E	00	-- False
4F	**A1 0C**	**-- [1] X509CertificateAttributes**
		Path
51	30 0A	-- SEQUENCE
		Path
53	30 08	-- SEQUENCE
		Path
55	04 06	-- OCTET STRING
57	**3F 00 DF 00 50 39**	**-- MF/Belpic/Cert#3(non-rep)**

OFFSET	Encoding	ASN.1 Syntax
		Certification Authority Certificate
5D	30 26	-- SEQUENCE
		Common Object Attributes
5F	30 0B	-- SEQUENCE
		Label
61	0C 02	-- UTF8String
63	43 41	-- "CA"
		Common Object Flags
65	03 02	-- BIT STRING
67	06 C0	-- private(0), modifiable(1)
		AuthID
69	04 01	-- OCTET STRING
6B	01	-- '01'
		Common Certificate Attributes
6C	30 09	-- SEQUENCE
		Identifier
6E	04 01	-- OCTET STRING
70	04	-- '04'
		Authority
71	01 01	-- BOOLEAN
73	FF	-- True
74	**83 01**	**--[3] ImplicitTrust IMPLICIT BOOLEAN**
76	00	-- False
77	**A1 0C**	**-- [1] X509CertificateAttributes**
		Path
79	30 0A	-- SEQUENCE
		Path
7B	30 08	-- SEQUENCE
		Path
7D	04 06	-- OCTET STRING
7F	**3F 00 DF 00 50 3A**	**-- MF/Belpic/Cert#4(CA)**

OFFSET	Encoding	ASN.1 Syntax
		Root Certificate
85	30 28	-- SEQUENCE
		Common Object Attributes
87	30 0D	-- SEQUENCE
		Label
89	0C 04	-- UTF8String
8B	52 6F 6F 74	-- "Root"
		Common Object Flags
8F	03 02	-- BIT STRING
91	06 C0	-- private(0), modifiable(1)
		AuthID
93	04 01	-- OCTET STRING
95	01	-- '01'
		Common Certificate Attributes
96	30 09	-- SEQUENCE
		Identifier
98	04 01	-- OCTET STRING
9A	06	-- '06'
		Authority
9B	01 01	-- BOOLEAN
9D	FF	-- True
9E	**83 01**	**--[3] ImplicitTrust IMPLICIT BOOLEAN**
A0	00	-- False
A1	**A1 0C**	**-- [1] X509CertificateAttributes**
		Path
A3	30 0A	-- SEQUENCE
		Path
A5	30 08	-- SEQUENCE
		Path
A7	04 06	-- OCTET STRING
A9	**3F 00 DF 00 50 3B**	**-- MF/Belpic/Cert#6(Root)**

6. Application information detail

Remark: the maximum size of the files is only indicated when they can be later updated and they have to be created with a maximum size larger than the current one.

6.1. Files length

The files that can never be modified are created with the exact size to fit the content.

In case a file can be modified, its size is specified in the document.

6.2. TLV format

Some files are encoded in a simplified ***TLV*** format:

- a tag identifying the data: 1 byte
- the length of the data[2]:
  - <= 255: 1 byte
  - > 255: multiple bytes:
    - FF | x – 255
    - FF | FF | x – 510
    - FF | FF | FF | x – 765
    - …
- the data: *x* bytes

Encoding type:

- All data is either pure binary, or UTF-8 containing Unicode characters.
- UTF-8 strings are not null-terminated.
- UTF-8 containing multi-byte characters is referred as ***UTF-8***.
- When data contains only 7-bits characters, it is referred as ***ASCII***, although they are fully compatible with the ***UTF-8/Unicode*** conventions.
- The actual data may be followed by padding bytes (‘0’). They have to be ignored.

[2] Nor the tag, nor the length are counted in this length.

Fields length:

- All tables contain the maximum size ("*Max. # bytes*") for each individual field.
- The tables also contain the envisioned maximum size ("*Envisioned Max. # bytes*") for each field; this is intended to anticipate some possible changes in the data size. All applications should use this value for their internal buffers allocation.

Optional fields:

- Some fields are mandatory; in this case, the "***Default value***" column contains "***M*** ".
- If the field is optional, the "***Default value***" column contains the value to assign when the field is not present.

6.3. Identity data

6.3.1. DF(ID)

This transparent data file contains all files related to the citizen and to information that is managed by the National Register.

6.3.2. EF(ID#RN)

This transparent elementary file contains all permanent information about the citizen, such as issuing country, issuing authority, issuing date, validity period, name, address, birth date, etc. This is known as the 'ID file'.

This file contains most of the information that is graphically personalised on the card plastic.

It is formatted in simplified ***TLV*** format (see 6.2).

The file structure version corresponding to this document worth ***0***, and does not appear in the file.

File contents and format						
Tag (decimal)	**Tag (hexa)**	**Current Max. # bytes (decimal)**	**Data**	**Encoding type**	**Default value**	**Envisioned Max. # bytes (decimal)**
0	**00**	2	File structure version	Binary	**0**	2
1	**01**	12	Card Number	ASCII	M	12
2	**02**	16	Chip Number	Binary	M	16
3	**03**	10	Card validity date begin: ***DD.MM.YYYY***	ASCII	M	10
4	**04**	10	Card validity date end: ***DD.MM.YYYY***	ASCII	M	10
5	**05**	(42) *50	Card delivery municipality	UTF-8	M	80
6	**06**	11	National Number	ASCII	M	11
7	**07**	(62) *90	Name	UTF-8	M	110
8	**08**	(52) *75	2 first given names	UTF-8	-	95
9	**09**	3	First letter of 3rd given name	UTF-8	M	3
10	**0A**	(50) *65	Nationality	UTF-8	M	85
11	**0B**	(40) *60	Birth location	UTF-8	M	80
12	**0C**	12	Birth date: ***DD mmmm YYYY*** or ***DD.mmm.YYYY*** (German) See "Table 1" below	UTF-8	M	12
13	**0D**	1	Sex ***M***: man ***F***/***V***/***W***: woman	ASCII	M	1
14	**0E**	(21) *30	Noble condition	UTF-8		50
15	**0F**	1	Document type ***1***: Belgian citizen ***2***: European Community citizen ***3***: non European Community citizen ***7***: bootstrap card ***8***: "habilitation/machtigings" card	ASCII	M	2
16	**10**	1	Special status ***0***: No status ***1***: White cane (blind people) ***2***: Extended minority ***3***: White cane + extended minority ***4***: Yellow cane (partially sighted people) ***5***: Yellow cane + extended minority	ASCII	**0**	2
17	**11**	20	hash photo	Binary (SHA-1)	M	20

- These fields are UTF-8 multi-bytes characters. The exact maximum length cannot be known by advanced and is only estimated. The number between brackets represents the maximum number of UTF-8 characters.

Birth months												
French	JAN	FEV	MARS	AVR	MAI	JUIN	JUIL	AOUT	SEPT	OCT	NOV	DEC
Dutch	JAN	FEB	MAAR	APR	MEI	JUN	JUL	AUG	SEP	OKT	NOV	DEC
German	JAN	FEB	MÄR	APR	MAI	JUN	JUL	AUG	SEP	OKT	NOV	DEZ

Table 1

6.3.3. EF(SGN#ID)

This transparent elementary file contains the signature of the ***EF(ID#RN)*** by the National Register.

As the ***EF(ID#RN)*** file contains the hash of the picture, the picture is also implicitly signed.

Signature format: SHA-1, PKCS#1 v. 1.5, RSA.

6.3.4. EF(ID#Address)

This transparent elementary file contains the information about the citizen‘s residence.

It is formatted in simplified ***TLV*** format (see 6.2).

The file structure version corresponding to this document worth ***0***, and does not appear in the file.

Current file length: 117 bytes

File contents and format					
Tag (decimal)	**Max. # bytes (decimal)**	**Data**	**Encoding type**	**Default value**	**Envisioned Max. # bytes (decimal)**
0	2	File structure version	Binary	**0**	2
1	60	Street + number	UTF-8	M	80
2	4	ZIP code	ASCII	M	6
3	47	municipality	UTF-8	M	67

6.3.5. EF(SGN#Address)

This transparent elementary file contains the signature of the ***EF(ID#Address)*** by the National Register.

EF(SGN#ID) is first appended to ***EF(ID#Address)*** before signing, in order to ensure the consistency with the file ***EF(ID#RN)***.

Signature format: SHA-1, PKCS#1 v. 1.5, RSA.

6.3.6. EF(ID#Photo)

This transparent elementary file contains the citizen's picture in the standard JPEG format.

As the ***EF(ID#RN)*** file contains the hash of the picture, the picture is also implicitly signed when signing this file.

Current picture resolution: width: 140 pixels, height: 200 pixels, grey levels: 8 bits

Remark: The resolution and colour encoding are included in the JPEG format. It is advisable to dynamically use these, as they could change in the future.

6.3.7. EF(PuK#7 ID)

As the ***PuK#7*** (***CA Role***) is not readable, we need a way to know which key is stored.

This transparent elementary file contains the ***CA Role Public Key*** hash (SHA-1).

6.3.8. EF(Preference)

This file is present in the current cards but it will be dropped in future releases; therefore it should NOT be used anymore.

This transparent elementary file contains additional information belonging to the citizen such as display preferences, etc. This file can be updated on request of the citizen only.

Max. size: to be defined

It is formatted in a similar manner as Windows INI files:

```
[Section]
Parameter=Value
Parameter=Value|Value|Value
…
[Section]
Parameter=Value
Parameter=Value|Value
…
```

- All lines are separated by a *Line Feed* (code '0A'); when reading the file, a *Carriage Return* (code '0D') must also be accepted instead of a *Line Feed*, or in supplement of it
- the file ends at the first null character (code '0') – if any
- ***Section*** and ***Parameter*** are case insensitive and may only contain letters 'A' to 'Z' (or 'a' to 'z'), digits, and underscores
- no blanks or spaces are allowed before or after the **=** between ***Parameter*** and ***Value***
- Applications must not make any assumptions about the order of the ***Section*** and ***Parameter***
- Applications must know about the data encoding (string, base64, number, etc.) of the ***Value*** field.
    - multiple values for a parameter are separated by a '**|**'
    - a real '**|**' inside a value is preceded by a backslash '****'
    - Strings cannot be enclosed between quotes in ***Value***
    - no blanks or spaces are allowed before or after the '**|**' separating the multiple ***Values*** (that is, they will be interpreted as meaningful characters)
- Guidelines for applications:
    - ***Value*** should only contain 7-bits ASCII characters strings
    - Numbers should use a dot '**.**' as decimal separator
    - Non-7-bit ASCII string should be UTF-8 and base64 encoded
    - ***Section***, ***Parameter***, and ***Value*** should be maximum 3 characters long, unless a ***Value*** must absolutely be longer
    - No comments should be used
- Sections and values unknown to an application must be ignored and may not be altered or removed when rewriting the file to the card.
- All the following pre-defined values are optional; the file may thus also be totally empty

Section: [Gen]				
Parameter	**Description**	**Values**	**Multiple values allowed**	**Order meaningful**
LG	Language Language the citizens prefers to use	**en**: English **nl**: Dutch **fr**: French **de**: German	yes	yes
DE	Display Enhancement The citizen has a deficient vision and prefers some display enhancement	**bd**: bigger display **va**: vocal aid **ac**: alternate colours (aimed at colour blinds)	yes	no

7. Public and Private Keys detail

7.1. Private RSA Key #1

This file contains the private RSA ***Basic Key***. It is involved in the ***Internal Authentication*** process.

7.2. Private RSA Key #2

This file contains the private RSA ***Authentication Key***.

7.3. Private RSA Key #3

This file contains the private RSA ***Non-Repudiation Key***.

The userConsent element in ***PrKDF*** contains value 1 for this key i.e. the cardholder must manually enter the corresponding PIN for each private key operation.

7.4. Public RSA Key #7

This file contains the public RSA ***Role CA Key*** used for external and mutual authentication.

8. Certificates detail

All certificates stored in the card are DER encoded (not Base 64).

8.1. Certificate #2

This file contains the citizen's X.509 ***Authentication Certificate*** containing the public key corresponding to the private RSA ***Authentication Key*** (Private RSA Key #2).

8.2. Certificate #3

This file contains the citizen's X.509 ***Non-Repudiation Certificate*** containing the public key corresponding to the private RSA '***Non-Repudiation Key***' (Private RSA Key #3).

8.3. Certificate #4

This file contains the X.509 ***Citizen's CA Certificate*** used to sign the ***Authentication Certificate*** (#2) and the ***Non-Repudiation Certificate*** (#3).

8.4. Certificate #6

This file contains the X.509 ***ROOT CERTIFICATE*** used to sign the ***Citizen's CA Certificate*** (#4) and the ***Government CA Certificate*** (not in the card).

8.5. Certificate #8

This file contains the X.509 ***RRN Certificate***. This certificate corresponds to the private key used to sign the files ***EF(ID#RN)*** and ***EF(ID#Address)***.