



Belgian eID Toolkit Developer's guide

For
Fedict

From
CSC Computer Sciences



Document History

<i>Date</i>	<i>Version</i>	<i>Author(s)</i>	<i>Reason for change</i>
12-02-04	1.0	M. Stern K. Bogaert	First version
19-02-04	1.0a	M. Stern	Modified some C types Added licensing info
27-02-04	1.0b	K. Bogaert	Changed certificate validation return codes Added OCSP/CRL and PIN function codes
02-03-04	1.0c	K. Bogaert	Added Java API specification
05-03-04	1.0d	M. Stern	Added some explanations
01-04-04	1.0e	M. Stern	Added some explanations
16-04-04	1.0f	M. Stern	Integration with the " <i>User's Guide</i> "
03-05-04	1.0g	K. Bogaert	Added the functions <i>SetRawData/GetRawData</i>
11-05-04	1.0h	M. Stern	Added " <i>Development environment matrix</i> "
18-06-04	1.0i	M. Stern	Removed "eidlib.jar", as it is part of the run-time
02-09-04	1.0j	M. Stern	Added some explanations and remark about thread safety
02-02-05	1.0k	M. Stern	Added C library <i>calling conventions</i> and <i>struct packing</i>
30-03-05	1.0l	M. Stern	Added explanation about concatenated fields
29-05-06	1.0m	K. Bogaert	Added new C API functions

Table of Contents

1.	Introduction.....	5
1.1	Development environment matrix	6
2.	Functional Specifications.....	7
2.1	Version handling and compatibility.....	7
2.2	PIN Entry	7
2.3	Remark on maximum length of parameters.....	7
2.4	Multi-threaded application.....	7
2.5	API Organisation	8
2.6	Initialisation and termination functions	8
2.6.1	BEID_Init.....	8
2.6.2	BEID_Exit.....	9
2.7	Identity functions	9
2.7.1	BEID_GetID	10
2.7.2	BEID_GetAddress	11
2.7.3	BEID_GetPicture	11
2.7.4	Off-line identity functions.....	12
2.8	General purpose high-level functions	13
2.8.1	BEID_BeginTransaction.....	13
2.8.2	BEID_EndTransaction.....	13
2.8.3	BEID_SelectApplication.....	14
2.8.4	BEID_ReadFile.....	14
2.8.5	BEID_WriteFile.....	14
2.8.6	BEID_VerifyPIN	15
2.8.7	BEID_ChangePIN.....	15
2.8.8	BEID_GetPINStatus	15
2.9	Low-level functions	17
2.9.1	BEID_GetVersionInfo	17
2.9.2	BEID_SendAPDU	17
2.9.3	BEID_FlushCache	18
2.10	PIN identification.....	18
2.10.1	PIN types.....	19
2.10.2	PIN usages	19
2.11	OCSP and CRL input policy parameters	19
2.12	Functions status.....	20

- 2.12.1 General return codes 20
- 2.13 Signature checking and certificates validation 21
 - 2.13.1 Signature checking 23
 - 2.13.2 Certificate checking and validation results 24
 - 2.13.3 OCSP and CRL used policies 25
- 3. Programming Interfaces 26
 - 3.1 C API 26
 - 3.1.1 Structures 26
 - 3.1.2 Functions 32
 - 3.2 Java API 39
 - 3.2.1 Data Classes 39
 - 3.2.2 Main Class 44
 - 3.2.3 Applet Class 46
 - 3.3 ActiveX API 49
 - 3.3.1 Interface Definitions 49
 - 3.3.2 Functions 52
- 4. Installation 57
 - 4.1 Installation Package 57
 - 4.2 Run-time 57
 - 4.3 C Library 58
 - 4.4 Java 58
- 5. License issues 59
 - 5.1 Disclaimer 59
 - 5.2 Third Party Licenses 60
 - 5.2.1 OpenSSL 60
 - 5.2.2 OpenSC 62
 - 5.2.4 libstdc 67

1. INTRODUCTION

This document describes the Application Programming Interface of the eID Toolkit used to retrieve the identity information from the Belgian Identity Card.

The programmer's guide to interact with the card cryptographic functions (Microsoft CSP/CryptoAPI & PKCS#11 are included in separate manuals).

The main goals of this toolkit are:

- To provide an easy way to retrieve the identity information from any version of a Belgian Identity Card, whatever data internal format
- To automate and hide all validation mechanisms; that is, all received data is automatically validated with the right mechanisms to ensure its consistency
- To provide an interface that as easy to use as possible to reduce the integration time in applications, and to increase the robustness. All functions are – as much as possible – self-sufficient; as an example, all identity functions will automatically
 - select the right application before reading the identity file
 - ensure they are not interrupted in the middle of a file read
 - interpret the contents of a file based on the card version
 - etc.

Section 2 is intended to all developers, whatever programming language they use. This section describes the parameters independently of any programming language.

Section 3 specifies the API for the different programming languages. This is a syntax only description.

Section 4 describes the package installation for the various platforms and the various development environments.

Demo programs are available in the package for all the languages. These are the best place to look for examples.

1.1 Development environment matrix

As the eID Toolkit offers several interfaces for the same functionalities, a choice has to be made in order to chose the best one.

This matrix gives the recommended interface for several of the common development environments and languages.

API	C	ActiveX	Java	Java applet
Development environment				
Java			✓	
C	✓			
VB, Delphi		✓		
.NET		✓		
VBA, Vbscript, etc.		✓		
Perl	✓	✓		
Web application		6		✓

Important remark:

To access the EID card from a Web page, the ActiveX should not be used, because

1. It only works with Microsoft Internet Explorer
2. It is not trusted by the browser and will be refused by most of the installations and users
3. The Java applet contains a user interface to interactively ask the user for an access confirmation

Moreover, the current ActiveX is not scriptable because the scripts are not able to access the memory allocated by the component; a wrapper that uses a buffer provided by the script should be made around it to be able to script it.

2. FUNCTIONAL SPECIFICATIONS

2.1 *Version handling and compatibility*

The Toolkit automatically handles all the different versions of the cards. There is no need, when using the Toolkit to worry about the internal way the card owns the data, as it will be available in an uniform way through the API.

A low-level function exist to get the various versions of the card's components, but this is only aimed at technical developers who need to access some very specific features of the card – a normal application should not worry about a card's version.

2.2 *PIN Entry*

Several functions accept a PIN reference input parameter. In case a PIN reference is provided, and the function gets an “access denied” when trying to access a card resource, the function automatically will ask the PIN to the user, and retry to access the resource (in case of a successful PIN verification).

This is a just-in-time PIN checking, because the PIN will only be asked when needed. For example, a permanent PIN may have been entered previously and is still valid. In this case, it will not be asked another time.

If the reader allows a secure PIN entry, the reader pinpad is used, otherwise the PC keyboard is used.

2.3 *Remark on maximum length of parameters*

The maximum length of the data returned by the functions is given in the type of the parameter:

- Bytes
- ASCII characters
- UTF-8 characters

For C developers, all ASCII and UTF-8 strings are null terminated. Beware that the given length does not include the ending ‘\0’ for null terminated strings.

2.4 *Multi-threaded application*

The library is not thread-safe. It is the calling application responsibility to not use the library simultaneously in parallel threads.

Remarks: The CSP is thread-safe, but you may not call the CSP in one thread and the Toolkit in another thread.

2.5 API Organisation

The functions are divided in 4 categories:

- **Initialisation and termination** functions, mandatory to initialise and terminate with the toolkit usage.
- **Identity** functions, used to retrieve the identity data (name, address, etc.) from the card.
- **General purpose high-level** functions, used to access information in a generic way (files, PIN), mainly in other applications than the identity one. There is no need to use these functions to access the identity data.
- **Low-level** functions, provided for developers that need very technical functions, or for debugging purpose. These should be ignored by the normal developers.

2.6 Initialisation and termination functions

2.6.1 BEID_Init

This function initialises the toolkit.

This function must be called before any other one.

The policy about certificate validation (either by using OCSP, or CRL) is given. It is valid for all further function calls until **BEID_Exit()** is called. The OCSP and CRL Mandatory flags are mutual exclusive. When both OCSP and CRL are used, OCSP will be tried first; if successful, the process will stop, otherwise the CRL is used.

Remark : when both OCSP and CRL values are equal to 0, the user certificates are not read for all further functions calls. This speeds up the reading.

Parameter	In	Out	Detail	Permitted values or format	Max. length
Reader Name	X		Reader name	<ul style="list-style-type: none"> ▪ Empty string or NULL for detecting automatically the first reader ▪ Reader PC/SC name ▪ "VIRTUAL" for a virtual reader (See 2.7.5) 	
OCSP	X		OCSP Policy	0=Not Used (default), 1=Optional, 2=Mandatory (see 2.11)	
CRL	X		CRL Policy	0=Not Used (default), 1=Optional, 2=Mandatory (see 2.11)	
PC/SC handle		X	PC/SC handle	This output parameter should be ignored by most of the developers; it is only provided for the ones willing to interact with the reader at the PC/SC level, for very technical purpose.	

2.6.2 BEID_Exit

This function cleans up all data used by the toolkit.

This function must be called at the end of the program.

Parameter	In	Out	Detail	Permitted values or format	Max. length

2.7 Identity functions


All the identity functions are self-sufficient. That is, there is no need to call any other function together with an identity function (except the initialization and termination ones).

No PIN has to be entered in order to read identity files.

All these functions can be called whatever state the card currently is – whether another DF (Data File) than the identity is selected, etc.

2.7.1 BEID_GetID

Parameter	In	Out	Detail	Permitted values or format	Max. length
Version		X	ID data version	SHORT	
CardNumber		X	Logical card number	ASCII	12
ChipNumber		X	Physical chip number	ASCII	16
ValidityDateBegin		X	Card validity date begin	ASCII: YYYYMMDD	8
ValidityDateEnd		X	Card validity date end	ASCII: YYYYMMDD	8
Municipality		X	Card delivery municipality	UTF-8	50
NationalNumber		X	National Number	ASCII	11
Name		X	Surname	UTF-8	80
FirstName1		X	1 st first name1	UTF-8	60
FirstName2		X	2 nd first name	UTF-8	30
FirstName3		X	3 rd first name (first letter)	UTF-8	1
Nationality		X	Nationality – ISO code	ASCII	3
BirthLocation		X	Birth location	UTF-8	50
BirthDate		X	Birth date	ASCII: YYYYMMDD	8
Sex		X	Sex	ASCII: M/F	1
NobleCondition		X	Noble condition	UTF-8	40
DocumentType		X	Document type	LONG 1: Belgian citizen 2: European Community citizen 3: non European Community citizen 7: bootstrap card 8: "habilitation/machtigings" card	
WhiteCane		X	White cane allowed (blind people)	Boolean	
YellowCane		X	Yellow cane allowed (partially sighted people)	Boolean	
ExtendedMinority		X	Extended minority	Boolean	
HashPhoto		X	Hash of the photo. This data is not relevant for most applications; it is only provided for technical ones.	Binary (SHA-1)	20
CertifCheck		X	Certificate checking and validation result	see 2.13	

 Some cards do not hold the first and second first names in a separate form, but both are concatenated; in this case, both are included in the field **FirstName1**, and **FirstName2** is empty.

1 Some cards do only contain the two first first names together; in this case, both of them will be returned in this field and the second first name will be empty.

2.7.2 BEID_GetAddress

Parameter	In	Out	Detail	Permitted values or format	Max. length
Version		X	Address data version	SHORT	
Street		X	Street2	UTF-8	80
Street number		X	Street number	ASCII	8
Box number		X	Box number	ASCII	4
Zip		X	Zip code	ASCII	6
Municipality		X	Municipality name	UTF-8	50
Country		X	Country ISO code	ASCII	3
CertifCheck		X	Certificate checking and validation result	see 2.13	

i Some cards do not hold the street and numbers in a separate form, but all are concatenated; in this case, all values are included in the field *Street*, and both *Street number* and *Box number* are empty.

2.7.3 BEID_GetPicture

Parameter	In	Out	Detail	Permitted values or format	Max. length
Picture		X	Picture, in JPEG format	BYTE stream	10 000
PictureLen	X	X	Length of the picture Byte Stream	Long : size in bytes <ul style="list-style-type: none"> ▪ IN: size of the given buffer ▪ OUT : actual size of returned data 	
CertifCheck		X	Certificate checking and validation result	see 2.13	

2.7.4 BEID_GetCertificates

This function returns the certificates on the card.

Parameter	In	Out	Detail	Permitted values or format	Max. length
CertifCheck		X	Certificates		

Remark: The certificates are not validated. (More info on validation See 2.13)

2 Some cards do also contain the number and the box number in the same field; in this case, all of them will be returned in this field and the other fields will be empty.

2.7.5 Off-line identity functions

Sometimes you may need to archive the data from the card, in order to not only validate them now, but also keep them with their signature, as a proof. In this case, you need two functions: one to read the raw data from the card, and one to give the raw data you stored as input to the identity functions.

BEID_GetRawData

This function returns the raw data files from the card. (ID, Address, Picture, RRN certificate, CardData, TokenInfo, Challenge/Response from internal authenticate). You need to store this data for later use.

Remark: The raw data is not checked, only read. You need to call the normal identity functions to validate them.

Parameter	In	Out	Detail	Permitted values or format	Max. length
RawData		X	Raw Data		

BEID_SetRawData

This function sets the raw data as input for the following identity functions. This allows to verify some identity data previously stored. Note that no card reader needs to be present in order to use that function.

In order to check the raw data, you need to:

- Call the *BEID_Init()* function with the parameter reader set to “VIRTUAL”
- Call the *BEID_SetRawData()* function
- Call the normal identity functions

Parameter	In	Out	Detail	Permitted values or format	Max. length
RawData	X		Raw Data		

BEID_GetRawFile

This function reads the raw data files from the card and returns them in one file. You need to store this data for later use.

Parameter	In	Out	Detail	Permitted values or format	Max. length
RawFile		X	Raw Data File		

Remark: the raw data file can be read with the beidgui application.

BEID_SetRawFile

This function sets the raw data file as input for the following identity functions. This allows to verify some identity data previously stored. Note that no card reader needs to be present in order to use that function.

In order to check the raw data, you need to:

- Call the **BEID_Init()** function with the parameter reader set to “VIRTUAL”
- Call the **BEID_SetRawFile()** function
- Call the normal identity functions

Parameter	In	Out	Detail	Permitted values or format	Max. length
RawFile	X		Raw Data File		

2.8 General purpose high-level functions

These functions provide an access – integrated with the toolkit – to general-purpose functions for application that need to perform other actions than simply accessing the identity data.

2.8.1 BEID_BeginTransaction

This function starts a transaction, waiting for the completion of all other transactions before it begins. No other application will have access to the card until **BEID_EndTransaction** is called.

This function must be called before other card operation functions that need to be grouped. This function is not needed to call any individual function, nor identity functions.

Typically, a transaction is used to select an application before accessing a file or a PIN.

Parameter	In	Out	Detail	Permitted values or format	Max. length

2.8.2 BEID_EndTransaction

This function completes a previously declared transaction, allowing other applications to resume interactions with the card.

This function must be called at the end of some card operation functions.

Parameter	In	Out	Detail	Permitted values or format	Max. length

2.8.3 BEID_SelectApplication

This function selects an application in the card.

Parameter	In	Out	Detail	Permitted values or format	Max. length
AID	X		Application AID	Byte Stream – card dependant (Ex. A000000177504B43532D3135)	
AIDLen	X		Application AID Length	Long: Length (in bytes) of provided AID	

2.8.4 BEID_ReadFile

This function reads a file on the card. If a PIN reference is provided, the PIN will be asked and checked if needed (just-in-time checking).

Parameter	In	Out	Detail	Permitted values or format	Max. length
FileID	X		Path to file to read from – relative to the current application	Byte Stream – card dependant (Ex. DF 00 50 32)	
FileIDLen	X		FileID Length	Long: Length (in bytes) of the given FileID	
OutData		X	Returned data		
OutDataLen	X	X	Returned data length	Long : size in bytes <ul style="list-style-type: none"> ▪ IN: size of the given OutData buffer ▪ OUT : actual size of returned data 	64 Kb
PIN	X		PIN protecting the file	See 2.10	

2.8.5 BEID_WriteFile

This function writes a file on the card. If a PIN reference is provided, the PIN will be asked and checked if needed (just-in-time checking).

Parameter	In	Out	Detail	Permitted values or format	Max. length
FileID	X		Path to file to be written – relative to the current application	Byte Stream – card dependant (Ex. DF 00 50 32)	
FileIDLen	X		FileID Length	Long: Length (in bytes) of the given FileID	
InData	X		Input data		
InDataLen	X		Input data length		
PIN	X		PIN protecting the file	See 2.10	

2.8.6 BEID_VerifyPIN

This function verifies a PIN.

Parameter	In	Out	Detail	Permitted values or format	Max. length
PIN	X		PIN protecting the file	See 2.10	
Pin	X		The given Pin	ASCII If empty or NULL then ask to user	12
NrTriesLeft		X	Number of tries left	Long: Number of tries remaining	

2.8.7 BEID_ChangePIN

This function changes a PIN.

Parameter	In	Out	Detail	Permitted values or format	Max. length
PIN	X		PIN protecting the file	See 2.10	
OldPin	X		The old Pin	ASCII If empty or NULL then ask to user	12
NewPin	X		The new Pin	ASCII If empty or NULL then ask to user	12
NrTriesLeft		X	Number of tries left	Long: Number of tries remaining	

2.8.8 BEID_GetPINStatus

This function retrieves the pin status. The result information may be signed by the card basic key (key '0x81' in the current application DF).

Parameter	In	Out	Detail	Permitted values or format	Max. length
PIN	X		PIN protecting the file	See 2.10	
NrTriesLeft		X	Number of tries left	Long: Number of tries remaining	
signature	X		Signature needed	Boolean	
signedStatus		X	Signed status	Byte Stream	256
signedStatusLen	X	X	Signed status length	Long : size in bytes <ul style="list-style-type: none"> ▪ IN: size of the given Byte Stream ▪ OUT : actual size of returned data: 256 	

2.8.9 BEID_GetPINs

This function retrieves the pins on the card.

Parameter	In	Out	Detail	Permitted values or format	Max. length
PINs		X	The returned PINs		

2.8.10 BEID_VerifyCRL

This function checks the certificates against the CRL.

Parameter	In	Out	Detail	Permitted values or format	Max. length
CertifCheck	X	X	Certificates to check	IN : certificates to validate OUT : validation status	
Download		X	Download CRL	Boolean	

2.8.11 BEID_VerifyOCSP

This function checks the certificates via OCSP.

Parameter	In	Out	Detail	Permitted values or format	Max. length
CertifCheck	X	X	Certificates to check	IN : certificates to validate OUT : validation status	

2.9 Low-level functions

Remark: The low-level functions are provided for applications that need to access specific technical functions not provided by the normal functions, or for debugging purpose. There is no need to use these low-level functions in normal applications.

2.9.1 BEID_GetVersionInfo

This function returns the version of the different components (applet, OS,...). The result information may be signed by the card basic key (key '0x81' in the current application DF).

Parameter	In	Out	Detail	Permitted values or format	Max. length
VersionInfo		X	See applet and card content specifications		
signature	X		Signature needed	Boolean	
signedStatus		X	Signed status	BYTE stream	128
signedStatusLen	X	X	Signed status length	Long : size in bytes <ul style="list-style-type: none"> ▪ IN: size of the given buffer: min. 128 ▪ OUT : actual size of returned data (128) 	

2.9.2 BEID_SendAPDU

This function sends an APDU command to the card.

Warning: when sending APDU to the card, the Toolkit cannot ensure the compatibility between the applet versions (except for the communication part). The call may not work in a next version of the applet.

If a PIN reference is provided, the PIN will be asked and checked if needed (just-in-time checking).

Remark: The command will first be tried without asking the PIN; if the command fails with a status "access denied", the PIN will be asked and the command will be retried.

Parameter	In	Out	Detail	Permitted values or format	Max. length
CmdAPDU	X		The command APDU to send to the card	Byte Stream – card dependant	256
CmdAPDULen	X		The input APDU command length		
PIN	X		PIN protecting the file	See 2.10	
RespAPDU		X	The response APDU received from the card		256
RespAPDULen	X	X	The response APDU command length	Long : size in bytes <ul style="list-style-type: none"> ▪ IN: size of the given buffer OUT : actual size of returned data 	

2.9.3 BEID_FlushCache

This function flushes the data cached in memory and on the disk.

Parameter	In	Out	Detail	Permitted values or format	Max. length

2.9.4 BEID_ReadBinary

This function reads a file from the card.

Parameter	In	Out	Detail	Permitted values or format	Max. length
FileID	X		Path to file to read from – relative to the current application	Byte Stream – card dependant (Ex. DF 00 50 32)	
Offset	X		Offset in the file to start read from		
Count	X		Number of bytes to read		
OutData		X	The response APDU received from the card	Byte Stream.	

2.10 PIN identification

When giving a PIN id to the Toolkit, you may chose to give either the PKCS#15 PIN id, or the Operating System PIN reference.

Additionally, in case the PIN is not a PIN registered in the Toolkit, you must provide a string describing why the PIN is needed (ex: “Personal data modification”). You must also give a short string (max. 3 characters) to be displayed on the card reader display, if any (ex: “MOD”).

Thus, a PIN is composed of 4 fields:

- The PIN type: PKCS#15 or Operating System (see 2.10.1)
- The PIN OS reference or : PKCS#15 id
- The usage code (see 2.10.2)
- The long description (in the user’s language)
- The short description (in the user’s language)

In order to enhance the security, and to standardise the information display, the description given by the application may be overwritten by the Toolkit if it can determine the exact usage.

If there is no PIN to check, a NULL value has to be used.

Remark: When a PIN is available in the PKCS#15 structure, it is safer to used the PKCS#15 reference instead of the OS one, as the OS may change in the future.

2.10.1 PIN types

Value	C constant	Explanation
0	BEID_PIN_TYPE_PKCS15	PKCS15 PIN
1	BEID_PIN_TYPE_OS	OS PIN

2.10.2 PIN usages

Value	C constant	Explanation
0		Application defined
1	BEID_USAGE_AUTH	Authentication usage
2	BEID_USAGE_SIGN	Signature usage
...	...	Additional usages to come

2.11 OCSP and CRL input policy parameters

Value	C constant	Explanation
0	BEID_OCSP_CRL_NOT_USED	No checking CRL/OCSP
1	BEID_OCSP_CRL_OPTIONAL	Optional checking CRL/OCSP
2	BEID_OCSP_CRL_MANDATORY	Mandatory checking CRL/OCSP

2.12 Functions status

Each function returns a general return code specifying the type of error. In most of the cases, only this return code will be used. However, to get a detailed information about the technical reasons, the following status codes are also provided:

- The system error code (long)
- The PC/SC error code (long)
- The smart card Status Word (double byte)

2.12.1 General return codes

Value	C constant	Explanation
0	BEID_OK	Function succeeded
1	BEID_E_SYSTEM	Unknown system error (see system error code)
2	BEID_E_PCSC	Unknown PC/SC error (see PC/SC error code)
3	BEID_E_CARD	Unknown card error (see card status word)
4	BEID_E_BAD_PARAM	Invalid parameter (NULL pointer, out of bound, etc.)
5	BEID_E_INTERNAL	An internal consistency check failed
6	BEID_E_INVALID_HANDLE	The supplied handle was invalid
7	BEID_E_INSUFFICIENT_BUFFER	The data buffer to receive returned data is too small for the returned data
8	BEID_E_COMM_ERROR	An internal communications error has been detected
9	BEID_E_TIMEOUT	A specified timeout value has expired
10	BEID_E_UNKNOWN_CARD	The smart card is not recognized
11	BEID_E_KEYPAD_CANCELLED	Input on pinpad cancelled
12	BEID_E_KEYPAD_TIMEOUT	Timeout returned from pinpad
13	BEID_E_KEYPAD_PIN_MISMATCH	The two PINs did not match
14	BEID_E_KEYPAD_MSG_TOO_LONG	Message too long on pinpad
15	BEID_E_INVALID_PIN_LENGTH	Invalid PIN length
16	BEID_E_VERIFICATION	Error in a signature verification (see 2.13)
17	BEID_E_NOT_INITIALIZED	Toolkit not initialized
18	BEID_E_UNKNOWN	An internal error has been detected, but the source is unknown
19	BEID_E_UNSUPPORTED_FUNCTION	Function is not supported
20	BEID_E_INCORRECT_VERSION	The Toolkit version is incompatible with the calling interface. The program needs to be re-compiled.
21	BEID_E_INVALID_ROOT_CERT	Wrong Root certificate
22	BEID_E_VALIDATION	Error certificate validation (OSCP/CRL) (see 2.13)

2.13 Signature checking and certificates validation

Each function returning signed data always checks the signature, together with the integrity of whole certificate chain.

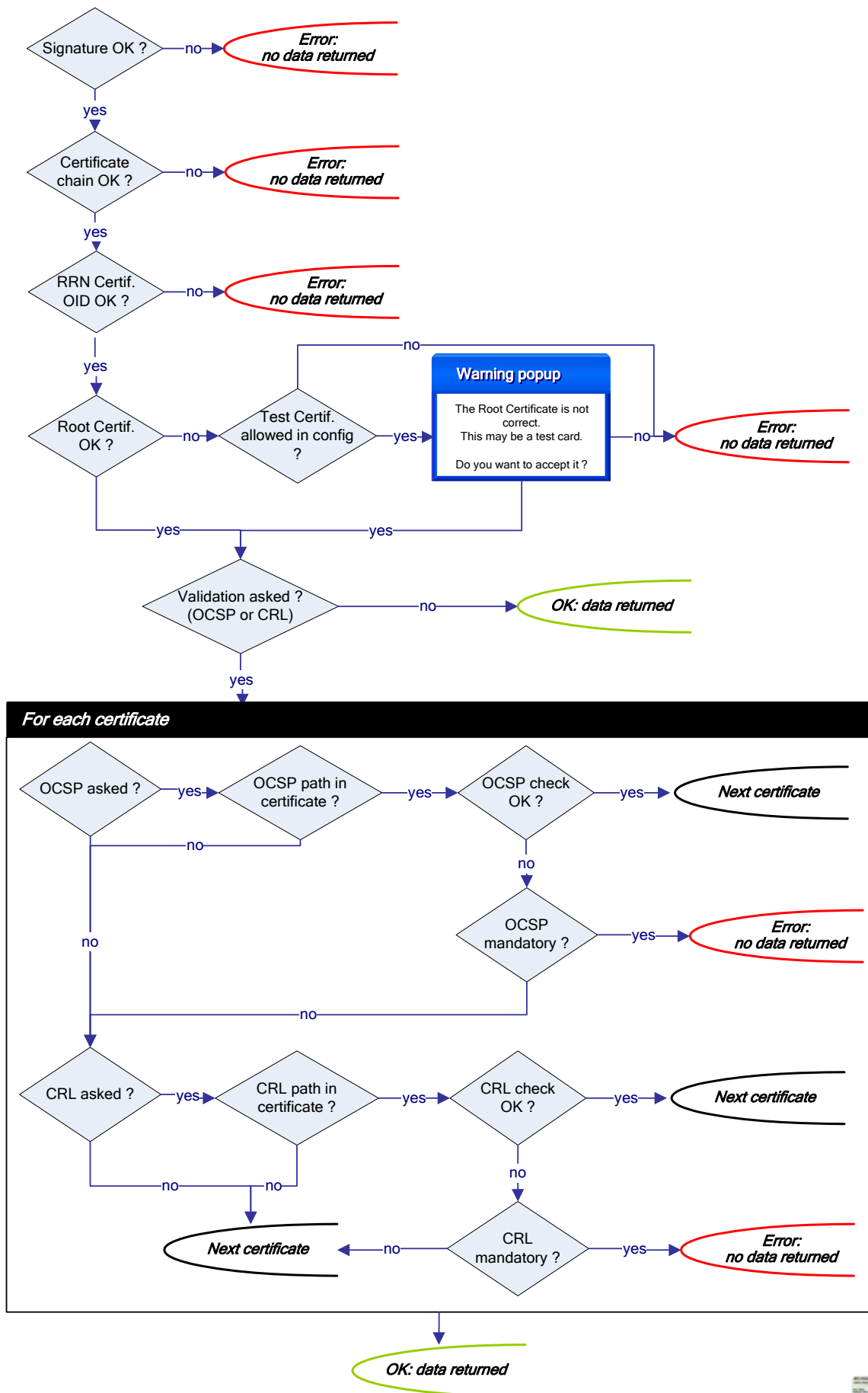
The function returns

- the status of the signature check (long)
- the global status of the certificate validation (long)
- for each certificate
 - the certificate
 - the certificate's label
 - the individual checking status
 - the individual validation status
 - the individual policy used: OCSP or CRL

In case of any error in the signatures or the validation chain, all functions exit with an error code, and **no data is returned**.

In case the root certificate is not the official one, a popup box will appear to notify that the *Root Certificate* is not the expected one. The user has the choice to temporarily accept this *Root Certificate* – this is needed to use the test cards that have a different *Root Certificate* than the official one.

Here is a diagram describing the signature checking workflow:



2.13.1 Signature checking

Value	C constant	Explanation
-1	BEID_SIGNATURE_PROCESSING_ERROR	Error verifying the signature.
0	BEID_SIGNATURE_VALID	The signature is valid.
1	BEID_SIGNATURE_INVALID	The signature is not valid.
2	BEID_SIGNATURE_VALID_WRONG_RRNCERT	The signature is valid but wrong RRN certificate.
3	BEID_SIGNATURE_INVALID_WRONG_RRNCERT	The signature is not valid and wrong RRN certificate.

2.13.2 Certificate checking and validation results

Value	C constant	Explanation
0	BEID_CERTSTATUS_CERT_VALIDATED_OK	Validation has occurred successfully.
1	BEID_CERTSTATUS_CERT_NOT_VALIDATED	No validation has been done.
2	BEID_CERTSTATUS_UNABLE_TO_GET_ISSUER_CERT	Unable to get issuer certificate
3	BEID_CERTSTATUS_UNABLE_TO_GET_CRL	Unable to get certificate CRL
4	BEID_CERTSTATUS_UNABLE_TO_DECRYPT_CERT_SIGNATURE	Unable to decrypt certificate's signature
5	BEID_CERTSTATUS_UNABLE_TO_DECRYPT_CRL_SIGNATURE	Unable to decrypt CRL's signature
6	BEID_CERTSTATUS_UNABLE_TO_DECODE_ISSUER_PUBLIC_KEY	Unable to decode issuer public key
7	BEID_CERTSTATUS_CERT_SIGNATURE_FAILURE	Certificate signature failure
8	BEID_CERTSTATUS_CRL_SIGNATURE_FAILURE	CRL signature failure
9	BEID_CERTSTATUS_CERT_NOT_YET_VALID	Certificate is not yet valid
10	BEID_CERTSTATUS_CERT_HAS_EXPIRED	Certificate has expired
11	BEID_CERTSTATUS_CRL_NOT_YET_VALID	CRL is not yet valid
12	BEID_CERTSTATUS_CRL_HAS_EXPIRED	CRL has expired
13	BEID_CERTSTATUS_ERR_IN_CERT_NOT_BEFORE_FIELD	Format error in certificate's notBefore field
14	BEID_CERTSTATUS_ERR_IN_CERT_NOT_AFTER_FIELD	Format error in certificate's notAfter field
15	BEID_CERTSTATUS_ERR_IN_CRL_LAST_UPDATE_FIELD	Format error in CRL's lastUpdate field
16	BEID_CERTSTATUS_ERR_IN_CRL_NEXT_UPDATE_FIELD	Format error in CRL's nextUpdate field
17	BEID_CERTSTATUS_OUT_OF_MEM	Out of memory
18	BEID_CERTSTATUS_DEPTH_ZERO_SELF_SIGNED_CERT	Self signed certificate
19	BEID_CERTSTATUS_SELF_SIGNED_CERT_IN_CHAIN	Self signed certificate in certificate chain
20	BEID_CERTSTATUS_UNABLE_TO_GET_ISSUER_CERT_LOCALLY	Unable to get local issuer certificate
21	BEID_CERTSTATUS_UNABLE_TO_VERIFY_LEAF_SIGNATURE	Unable to verify the first certificate
22	BEID_CERTSTATUS_CERT_CHAIN_TOO_LONG	Certificate chain too long

23	BEID_CERTSTATUS_CERT_REVOKED	Certificate revoked
24	BEID_CERTSTATUS_INVALID_CA	Invalid CA certificate
25	BEID_CERTSTATUS_PATH_LENGTH_EXCEEDED	Path length constraint exceeded
26	BEID_CERTSTATUS_INVALID_PURPOSE	Unsupported certificate purpose
27	BEID_CERTSTATUS_CERT_UNTRUSTED	Certificate not trusted
28	BEID_CERTSTATUS_CERT_REJECTED	Certificate rejected
29	BEID_CERTSTATUS_SUBJECT_ISSUER_MISMATCH	Subject issuer mismatch
30	BEID_CERTSTATUS_AKID_SKID_MISMATCH	Authority and subject key identifier mismatch
31	BEID_CERTSTATUS_AKID_ISSUER_SERIAL_MISMATCH	Authority and issuer serial number mismatch
32	BEID_CERTSTATUS_KEYUSAGE_NO_CERTSIGN	Key usage does not include certificate signing
33	BEID_CERTSTATUS_UNABLE_TO_GET_CRL_ISSUER	Unable to get CRL issuer certificate
34	BEID_CERTSTATUS_UNHANDLED_CRITICAL_EXTENSION	Unhandled critical extension

2.13.3 OCSP and CRL used policies

Value	C constant	Explanation
0	BEID_POLICY_NONE	No policy used
1	BEID_POLICY_OCSP	OCSP policy used
2	BEID_POLICY_CRL	CRL policy used
3	BEID_POLICY_BOTH	OCSP and CRL policy used

3. PROGRAMMING INTERFACES

3.1 C API

All output buffers must be allocated by the calling application.

All functions use the *C calling convention*, not *Pascal* convention.

Struct packing is set to 8.

3.1.1 Structures

```
typedef struct {
    long general;           // General return code
    long system;           // System error (errno)
    long pcsc;              // PC/SC error
    BYTE cardSW[2];        // Card status word
} BEID_Status;

typedef struct {
    BYTE certif[...];      // Byte stream encoded certificate
    long certifLength;     // Size in bytes of the encoded certificate
    char certifLabel [...]; // Label of the certificate (Authentication, Signature, CA, Root,...)
    long certifStatus;     // Validation status code (see 2.3.3 Certificate checking and validation results)
} BEID_Certif;
```

```
typedef struct {  
    long usedPolicy;           // Policy used (see 2.13.3)  
    BEID_Certif_certificates[...]; // Array of BEID_Certif structures  
    long certificatesLength;   // Number of elements in Array  
    long signatureCheck;      // Status of signature (for ID and Address) or  
                               // hash (for Picture) on retrieved field (see 2.13)  
} BEID_Certif_Check;
```

```
typedef struct {  
    // Get Card Data  
    BYTE SerialNumber[16];  
    BYTE ComponentCode;  
    BYTE OSNumber;  
    BYTE OSVersion;  
    BYTE SoftmaskNumber;  
    BYTE SoftmaskVersion;  
    BYTE ApplicationVersion;  
    ushort GlobalOSVersion;  
    BYTE ApplicationInterfaceVersion;  
    BYTE PKCS1Support;  
    BYTE KeyExchangeVersion;  
    BYTE ApplicationLifeCycle;  
    // TokenInfo Data  
    BYTE GraphPerso;  
    BYTE ElecPerso;  
    BYTE ElecPersoInterface;  
    BYTE Reserved;  
} BEID_VersionInfo;
```

```
typedef struct {
    short version;
    char cardNumber[...];
    char chipNumber[...];
    char validityDateBegin[...];
    char validityDateEnd[...];
    TCHAR municipality[...];
    char nationalNumber[...];
    TCHAR name[...];
    TCHAR firstName1[...];
    TCHAR firstName2[...];
    TCHAR firstName3[...];
    char nationality[...];
    TCHAR birthLocation[...];
    char birthDate[...];
    char sex[...];
    TCHAR nobleCondition[...];
    Long documentType;
    BOOL whiteCane;
    BOOL yellowCane;
    BOOL extendedMinority;
    BYTE hashPhoto[...];
} BEID_ID_Data;
```

```
typedef struct {
    short version;
    TCHAR street[...];
    char streetNumber[...];
    char boxNumber[...];
    char zip[...];
    TCHAR municipality[...];
    char country[...];
} BEID_Address;
```

```
typedef struct {
    BYTE *data;
    unsigned long length;
} BEID_Bytes;
```

```
typedef struct {
    long pinType;           // PIN Type (see 2.10.1)
    BYTE id;               // PIN reference or ID
    long usageCode;        // PIN Usage (see 2.10.2)
    char *shortUsage;      // May be NULL for usage known by the middleware
    char *longUsage;       // May be NULL for usage known by the middleware
} BEID_Pin;
```

```
typedef struct { ... } BEID_Raw;
```

```
typedef struct
{
    long pinType;           // BEID_PIN_TYPE_PKCS15 or BEID_PIN_TYPE_OS
    BYTE id;               // PIN reference or ID
    long usageCode;        // Usage code (BEID_USAGE_AUTH, BEID_USAGE_SIGN, ...)
    long triesLeft;
    long flags;
    char label [BEID_MAX_PIN_LABEL_LEN];
} BEID_Pin_Info;
```

```
typedef struct
{
    BEID_Pin_Info pins[BEID_MAX_PINS]; // Array of BEID_Pin_Info structures
    long pinsLength;                  // Number of elements in Array
} BEID_Pins;
```

3.1.2 Functions

```
BEID_Status // return code
    BEID_Init(
        char *ReaderName, // Reader Name
        long OCSP, // OCSP policy certificates checking & validity
        long CRL, // CRL policy certificates checking & validity
        long *CardHandle, // output PC/SC handle
    );
```

```
BEID_Status // return code
    BEID_Exit(
    );
```

```
BEID_Status // return code
    BEID_GetID(
        BEID_ID_Data *IDData, // output data
        BEID_Certif_Check *CertifCheck // certificates checking & validity
    );
```

```
BEID_Status // return code
    BEID_GetAddress(
        BEID_ID_Address *Address, // output address data
        BEID_Certif_Check *CertifCheck // certificates checking & validity
    );
```

```
BEID_Status // return code
    BEID_GetPicture(
        BEID_Bytes *Picture, // output picture (JPEG format)
        BEID_Certif_Check *CertifCheck // certificates checking & validity
    );
```

```
BEID_Status // return code
    BEID_GetRawData(
        BEID_Raw *Raw // output Raw Data
    );
```

```
BEID_Status // return code
    BEID_SetRawData(
        BEID_Raw *Raw // input Raw Data
    );
```

```
BEID_Status // return code
BEID_GetVersionInfo(
    BEID_VersionInfo *pVersionInfo,
    BOOL Signature, // Signature needed
    BEID_Bytes *SignedStatus, // Signature 256 bytes
);
```

```
BEID_Status // return code
BEID_BeginTransaction();
```

```
BEID_Status // return code
BEID_EndTransaction();
```

```
BEID_Status // return code
BEID_SelectApplication(
    BEID_Bytes *Application // Application ID
);
```

```
BEID_Status // return code
BEID_FlushCache();
```

```
BEID_Status // return code
BEID_SendAPDU(
    BEID_Bytes *CmdAPDU, // Command APDU
    BEID_Pin *pin, // Pin Reference
    BEID_Bytes *RespAPDU // Response APDU
);
```

```
BEID_Status // return code
BEID_VerifyPIN(
    BEID_Pin *pin, // Pin Reference
    char *Pin, // Pin code
    long *TriesLeft, // Tries remaining
);
```

```
BEID_Status // return code
BEID_ChangePIN(
    BEID_Pin *pin, // Pin Reference
    char *OldPin, // Old Pin code
    char *NewPin, // New Pin code
    long *TriesLeft // Tries remaining
);
```

```
BEID_Status // return code
BEID_GetPINStatus(
    BEID_Pin *pin, // Pin Reference
    long *TriesLeft, // Tries remaining
    BOOL bSignature, // Signature needed
    BEID_Bytes *SignedStatus // Signature
);

BEID_Status // return code
BEID_ReadFile(
    BEID_Bytes *FileID, // File to read relative path
    BEID_Bytes *OutData, // Returned data
    BEID_Pin *pin, // Pin Reference
);
```

```
BEID_Status // return code
BEID_WriteFile(
    BEID_Bytes *FileID, // File to write relative path
    BEID_Bytes *InData, // Write buffer
    BEID_Pin *pin, // Pin Reference
);
BEID_Status // return code
BEID_GetCertificates(
    BEID_Certif_Check *CertifCheck // Certificates
);

BEID_Status // return code
BEID_GetRawFile(
    BEID_Bytes *RawFile // Raw file
);

BEID_Status // return code
BEID_SetRawFile(
    BEID_Bytes *RawFile // Raw file
);
BEID_Status // return code
BEID_GetPINs(
    BEID_Pins *Pins // The PINS
);
```

```
BEID_Status // return code
BEID_VerifyCRL(
BEID_CertificateCheck *ptCertificateCheck, // Certificates to check
BOOL bDownload // Download CRL
);

BEID_Status // return code
BEID_VerifyOCSP(
BEID_CertificateCheck *ptCertificateCheck // Certificates to check
);

BEID_Status // return code
BEID_ReadBinary(
BEID_Bytes *FileID, // File to read path
int iOffset, // Offset in file to start from
int iCount, // Number of bytes to read
BEID_Bytes *OutData // Returned Data
);
```

3.2 Java API

In order to stay compatible with the native code implementing the access to the card, the Java implementation does not use any exception handling; instead, all errors are returned in the error codes, as described in 2.12.

Package: `be.belgium.eid;`

3.2.1 Data Classes

```
public class BEID_Address
{
    public BEID_Address()
    public short getVersion()
    public String getStreet()
    public String getStreetNumber()
    public String getBoxNumber()
    public String getZip()
    public String getMunicipality()
    public String getCountry()
}

public class BEID_Raw { ... }
```

```
public class BEID_Bytes
{
    public BEID_Bytes()
    public byte[] getData()
}
```

```
public class BEID_Certif_Check
{
    public BEID_Certif_Check()
    public int getUsedPolicy()
    public BEID_Certif getCertificate(int Index)
    public int getCertificatesLength()
    public int getSignatureCheck()
}
```

```
public class BEID_Certif
{
    public BEID_Certif()
    public byte[] getCertif()
    public String getCertifLabel()
    public int getCertifStatus()
}
```

```
public class BEID_Long
{
    public BEID_Long()
    public long getLong()
}
```

```
public class BEID_Pin
{
    public BEID_Pin()
    public void setPinType(int pinType)
    public void setId(short id)
    public void setUsageCode(int usageCode)
    public void setShortUsage(String shortUsage)
    public void setLongUsage(String longUsage)
}
```

```
public class BEID_Status
{
    public BEID_Status()
    public int getGeneral()
    public int getSystem()
    public int getPcsc()
    public byte[] getCardSW()
}
```

```
public class BEID_VersionInfo
{
    public BEID_VersionInfo()
    public byte[] getSerialNumber()
    public short getComponentCode()
    public short getOSNumber()
    public short getOSVersion()
    public short getSoftmaskNumber()
    public short getSoftmaskVersion()
    public short getAppl etVersion()
    public int getGlobal OSVersion()
    public short getAppl etInterfaceVersion()
    public short getPKCS1Support()
    public short getKeyExchangeVersion()
    public short getAppl icationLifeCycle()
    public short getGraphPerso()
    public short getEl ecPerso()
    public short getEl ecPersoInterface()
    public short getReserved()
}
```

```
public class BEID_ID_Data
{
    public BEID_ID_Data()
    public short getVersion()
    public String getCardNumber()
    public String getChipNumber()
    public String getValidityDateBegin()
    public String getValidityDateEnd()
    public String getMunicipality()
    public String getNationalNumber()
    public String getName()
    public String getFirstName1()
    public String getFirstName2()
    public String getFirstName3()
    public String getNationality()
    public String getBirthLocation()
    public String getBirthDate()
    public String getSex()
    public String getNobleCondition()
    public int getDocumentType()
    public boolean getWhiteCane()
    public boolean getYellowCane()
    public boolean getExtendedMinority()
    public byte[] getHashPhoto()
}
```



3.2.2 Main Class

```
public class eidlib
{
    public static BEID_Status BEID_Init(String ReaderName, int OCSP, int CRL, BEID_Long CardHandle)

    public static BEID_Status BEID_Exit()

    public static BEID_Status BEID_GetID(BEID_ID_Data IDData, BEID_Certif_Check CertifCheck)

    public static BEID_Status BEID_GetAddress(BEID_Address Address, BEID_Certif_Check CertifCheck)

    public static BEID_Status BEID_GetPicture(BEID_Bytes Picture, BEID_Certif_Check CertifCheck)

    public static BEID_Status BEID_GetRawData(BEID_Raw RawData)

    public static BEID_Status BEID_SetRawData(BEID_Raw RawData)

    public static BEID_Status BEID_GetVersionInfo(BEID_VersionInfo VersionInfo, int Signature, BEID_Bytes SignedStatus)

    public static BEID_Status BEID_BeginTransaction()

    public static BEID_Status BEID_EndTransaction()

    public static BEID_Status BEID_SelectApplication(byte[] Application)
```

```
public static BEID_Status BEID_VerifyPIN(BEID_Pin PinData, String Pin, BEID_Long TriesLeft)

public static BEID_Status BEID_ChangePIN( BEID_Pin Pin, String oldPin, String newPin,
                                           BEID_Long TriesLeft)

public static BEID_Status BEID_GetPINStatus( BEID_Pin PinData, BEID_Long TriesLeft, int Signature,
                                             BEID_Bytes SignedStatus)

public static BEID_Status BEID_ReadFile( byte[] FileID, BEID_Bytes OutData, BEID_Pin PinData)

public static BEID_Status BEID_WriteFile(byte[] FileID, byte[] InData, BEID_Pin PinData)

public static BEID_Status BEID_FlushCache()

public static BEID_Status BEID_SendAPDU(byte[] CmdAPDU, BEID_Pin PinData, BEID_Bytes RespAPDU)
}
```

3.2.3 Applet Class

```
public class BEID_Applet
{
    public BEID_Applet()
    public int InitLib(String strReader) // If null or empty, applet parameter is used.
    public int ExitLib()
    public String getCardNumber()
    public String getChipNumber()
    public String getValidityDateBegin()
    public String getValidityDateEnd()
    public String getIssMunicipality()
    public String getNationalNumber()
    public String getName()
    public String getFirstName1()
    public String getFirstName2()
    public String getFirstName3()
    public String getNationality()
    public String getBirthLocation()
    public String getBirthDate()
    public String getSex()
    public String getNobleCondition()
    public int getDocumentType()
    public boolean getWhiteCane()
    public boolean getYellowCane()
    public boolean getExtendedMinority()
```

```
public String getStreet()
public String getStreetNumber()
public String getBoxNumber()
public String getZip()
public String getMunicipality()
public String getCountry()
public byte[] GetPicture()
public int SetRawData(byte[] IDData, byte[] SigIDData, byte[] AddrData,
                    byte[] SigAddrData,
                    byte[] PictureData, byte[] RNDData, byte[] cardData,
                    byte[] tokenInfoData,
                    byte[] challengeData, byte[] responseData)

public byte[] GetRawIDData()
public byte[] GetRawSigIDData()
public byte[] GetRawAddrData()
public byte[] GetRawSigAddrData()
public byte[] GetRawPictureData()
public byte[] GetRawCardData()
public byte[] GetRawTokenInfoData()
public byte[] GetRawRNDData()
public byte[] GetRawChallengeData()
public byte[] GetRawResponseData()
}
```

Applet Parameters:

Name	Values	Name
Card Reader name	Reader	String
OCSP checking	OCSP	See 2.11
CRL checking	CRL	See 2.11

HTML snippet:

```
<applet  
  codebase = ". "  
  archive = "eidlib.jar"  
  code     = "be.belgium.eid.BEIDApplet.class"  
  name     = "BEIDApplet"  
  width   = "0"  
  height  = "0"  
  hspace  = "0"  
  vspace  = "0"  
>  
<param name="Reader" value="">  
<param name="OCSP" value="0">  
<param name="CRL" value="0">  
</applet>
```

3.3 ActiveX API

3.3.1 Interface Definitions

Interface definitions are in IDL format.

Control Name: **EIDLibCtrl.EIDlib**

Interface IRetStatus

```
{  
    long * GetGeneral ( );  
    long * GetPCSC( );  
    long * GetSystem( );  
    VARIANT * GetCardSW( );    ' VARIANT type : VT_ARRAY | VT_UI1  
}
```

Interface IMapCollection

```
{  
    VARIANT * GetValue( [IN] BSTR strKey );  
    void SetValue( [IN] BSTR strKey, [IN] VARIANT vtValue );  
    long * GetCount( );  
}
```

Interface ICertif

```
{  
    VARIANT * GetCertif( );          ' VARIANT type : VT_ARRAY | VT_UI1  
    BSTR * GetLabel( );  
    long * GetStatus( );            ' Validation status code see 2.13  
}
```

Interface ICertifCheck

```
{  
    long * GetPolicy( );  
    long * GetSignatureCheck( );    ' Validation status code see 2.13  
    VARIANT * GetCertificates( );  ' Array of ICertif, VARIANT type : VT_ARRAY | VT_DISPATCH  
}
```

Interface IPin

```
{  
    void SetPinType([IN] long Type);    ' BEID_PIN_TYPE_PKCS15 or BEID_PIN_TYPE_OS  
    void SetID([IN] VARIANT ID);       ' VARIANT type : VT_UI1  
    void SetUsageCode([IN] long usageCode);  
    void SetshortUsage([IN] BSTR shortUsage); ' May be NULL for usage known by the middleware  
    void SetlongUsage([IN] BSTR longUsage); ' May be NULL for usage known by the middleware  
}
```

Interface IRaw

```
{  
    void SetIDDData([in] VARIANT vtIDD); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetIDDData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetIDSi gData([in] VARIANT vtIDSi gData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetIDSi gData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetAddrData([in] VARIANT vtAddrData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT * GetAddrData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetAddrSi gData([in] VARIANT vtAddrSi gData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetAddrSi gData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetPi ctureData([in] VARIANT vtPi ctureData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetPi ctureData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetRNDData([in] VARIANT vtRNDData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetRNDData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetCardData([in] VARIANT vtCardData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetCardData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetTokenI nfoData([in] VARIANT vtTokenI nfoData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetTokenI nfoData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetChal I engeData([in] VARIANT vtChal I engeData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT *GetChal I engeData(); ' VARIANT type : VT_ARRAY | VT_UI1  
    void SetResponseData([in] VARIANT vtResponseData); ' VARIANT type : VT_ARRAY | VT_UI1  
    VARIANT * GetResponseData(); ' VARIANT type : VT_ARRAY | VT_UI1  
}
```

3.3.2 Functions

Function definitions are in IDL format.

IRetStatus *

```
Init(  
  [IN] BSTR strReaderName,  
  [IN] long lOCSP,  
  [IN] long lCRL,  
  [OUT] long * plHandle  
);
```

IRetStatus *

```
Exit(  
);
```

‘ The IMapcollection contains Key/Value pairs of the returned data.

‘ The keys are predefined in order to not redefine the interface on new versions

‘ Keys: CardNumber, ChipNumber, BeginValidityDate, EndValidityDate, IssuingMunicipality, NationalNumber, Name, FirstName1, FirstName2, FirstName3, Nationality, BirthPlace, BirthDate, Gender, NobilityTitle, DocumentType, WhiteCane, YellowCane, ExtendedMinority

IRetStatus *

```
GetID(  
  [OUT] IMapCollection ** ppMapCollection, ‘ Allocated by the Toolkit  
  [OUT] ICertificate ** ppCertificate ‘ Allocated by the Toolkit
```

```
);
```

‘ Keys: Street, HouseNumber, BoxNumber, ZIPCode, Municipality, Country

I RetStatus *

GetAddress(

[OUT] I MapCol I ecti on ** ppMapCol I ecti on, ‘ Allocated by the Toolkit

[OUT] I Certi fCheck ** ppCerti fCheck ‘ Allocated by the Toolkit

);

‘ Keys: Picture

I RetStatus *

GetPi cture(

[OUT] I MapCol I ecti on ** ppMapCol I ecti on, ‘ Allocated by the Toolkit

[OUT] I Certi fCheck ** ppCerti fCheck ‘ Allocated by the Toolkit

);

I RetStatus *

GetRawData (

[OUT] I Raw ** ppRaw ‘ Allocated by the library

);

I RetStatus *

SetRawData (

[IN] I Raw * pRaw ‘

);



```
I RetStatus * GetVersionInfo(  
    [IN] BOOL bSignature,  
    [OUT] IMapCollection ** ppMapCollection,    ' Allocated by the Toolkit  
    [OUT] VARIANT * pvSignature                ' VARIANT type : VT_ARRAY | VT_UI1  
);  
  
I RetStatus *  
    BeginTransaction();  
  
I RetStatus *  
    EndTransaction();  
  
I RetStatus *  
    FlushCache();  
  
I RetStatus *  
    SelectApplication(  
    [IN] VARIANT vtApplication    ' VARIANT type : VT_ARRAY | VT_UI1  
);
```

IRetStatus *

SendAPDU(

[IN] VARIANT vtCommand, ' VARIANT type : VT_ARRAY | VT_UI1

[IN] IPin * pin,

[OUT] VARIANT * vtResponse ' VARIANT type : VT_ARRAY | VT_UI1

);

IRetStatus *

VerifyPin(

[IN] IPin * pin,

[IN] BSTR strPin,

[OUT] Long * plTriesLeft

);

IRetStatus *

ChangePin(

[IN] IPin * pin,

[IN] BSTR strOldPin,

[IN] BSTR strNewPin ,

[OUT] Long * plTriesLeft);

IRetStatus *

GetPinStatus(

[IN] IPin * pin,

[IN] BOOL bSignature,

[OUT] VARIANT * pvSignature, ' VARIANT type : VT_ARRAY | VT_UI1

[OUT] Long * plTriesLeft);

IRetStatus *

ReadFile(

[IN] IPin * pin,

[IN] VARIANT strFileID, ' VARIANT type : VT_ARRAY | VT_UI1

[OUT] VARIANT * pvtData ' VARIANT type : VT_ARRAY | VT_UI1

);

IRetStatus *

WriteFile(

[IN] IPin * pin,

[IN] VARIANT strFileID, ' VARIANT type : VT_ARRAY | VT_UI1

[IN] VARIANT vtData ' VARIANT type : VT_ARRAY | VT_UI1

);

4. INSTALLATION

4.1 Installation Package

The installation package³ contains the following files:

C library	
<i>OS</i>	Directory containing the run-time for the various Operating Systems <ul style="list-style-type: none"> ▪ <i>Windows</i> ▪ <i>Linux</i>
eidlib.h, eiddefines.h	C library header file to be included by C calling programs
<i>OS</i> /beidlib. <i>x</i>	C library to link with (shared libraries entry points), where <i>x</i> is the shared library extension <ul style="list-style-type: none"> ▪ <i>LIB</i> for Visual C++
test.cpp	C test program
ActiveX	
Windows\VB	VB example of calling the ActiveX control
Java	
Java/Test.java	Java test program
Java/BEIDCard.html	JavaScript test page

4.2 Run-time

The run-time is needed for all environments.

It has to be installed as described in the document “*Belgian eID Run-time Users guide*”.

³ This package may be included in an archive, like a ZIP, TAR, or GZ file

4.3 C Library

The application must include the Toolkit include file “**eidlib.h**”.

The application must be linked with the shared library “**libbeid.so**” or “**beidlib.dll**” if the linker supports a direct link with a shared library (like *GCC*), or with the library entry points “**beidlib.lib**” or “**libbeid.a**” if it doesn't (like *Visual C++*).

4.4 Java

The toolkit is compatible with all Sun's Java Virtual Machine from 1.2.

5. LICENSE ISSUES

The eID Toolkit uses several third-party libraries or code.

Redistributions in any form of the eID Toolkit – even embedded in a compiled application – must reproduce all the eID Toolkit and third-party's copyright notices, list of conditions, disclaimers, and any other materials provided with the distribution.

5.1 *Disclaimer*

This eID Toolkit is provided by the Belgian Government “as is”, and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Belgian Government or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this Toolkit, even if advised of the possibility of such damage.

However, the Belgian Government will ensure the maintenance of the Toolkit – that is, bug fixing, and support of new versions of the Electronic Identity card.

5.2 Third Party Licenses

5.2.1 OpenSSL

This Toolkit uses the OpenSSL Toolkit developed by the OpenSSL Project (<http://www.openssl.org/>).

Here is a copy of the license (from <http://www.openssl.org/source/license.html>):

LICENSE ISSUES

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (ey@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (ey@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-). 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]



5.2.2 OpenSC

This Toolkit uses the OpenSC Toolkit developed by the OpenSC Project (<http://www.opensc.org/>).

Here is a copy of the license (from from the distribution package <http://www.opensc.org/cgi-bin/cvsweb/opensc/COPYING?rev=1.2&content-type=text/x-cvsweb-markup>):

GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

5.2.4 libstdc

This library uses the libstdc++ run-time libraries developed by the Gnu CC Project (<http://www.gnucc.org/>).

Here is a copy of the license (from http://gcc.gnu.org/onlinedocs/libstdc++/17_intro/license.html):

The Code: Runtime GPL

The source code of libstdc++-v3 is distributed under version 2 of the GNU General Public License (http://gcc.gnu.org/onlinedocs/libstdc++/17_intro/COPYING), with the so-called "runtime exception," as follows (or see any header or implementation file):

As a special exception, you may use this file as part of a free software library without restriction. Specifically, if other files instantiate templates or use macros or inline functions from this file, or you compile this file and link it with other files to produce an executable, this file does not by itself cause the resulting executable to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the executable file might be covered by the GNU General Public License.

Hopefully that text is self-explanatory. If it isn't, you need to speak to your lawyer, or the Free Software Foundation.

Q: So any program which uses libstdc++ falls under the GPL?

A: No. The special exception permits use of the library in proprietary applications.

Q: How is that different from the GNU {Lesser,Library} GPL?

A: The LGPL requires that users be able to replace the LGPL code with a modified version; this is trivial if the library in question is a C shared library. But there's no way to make that work with C++, where much of the library consists of inline functions and templates, which are expanded inside the code that uses the library. So to allow people to replace the library code, someone using the library would have to distribute their own source, rendering the LGPL equivalent to the GPL.

Q: I see. So, what restrictions are there on programs that use the library?

A: None. We encourage such programs to be released as open source, but we won't punish you or sue you if you choose otherwise.