eID test infrastructure manual

MANUAL FOR WINDOWS, LINUX AND MAC OS X

VERSION 1.1

Disclaimer

Fedict is not responsible for any damage caused by erroneous or incomplete information in this document, nor by the software described in it.

Contents

1. Introduction 3

2. Physical cards and PCSC 3

3. Virtual cards and PCSC proxy 4

4. Generating virtual cards 5

5. Using the software 5

5.1 Control tools 5

5.2 Using the PCSC proxy lib 7

5.2.1 Dynamicly loaded PCSC lib 8

5.3 Summary: how to use a virtual card 9

6. OCSP and CRL 9

7. About the BE eID middleware 10

7.1 Version 3.X 10

7.2 Version 4.X and other Java applications 11

References 11

1. Introduction

The eID test infrastructure can help you to test your applications that use a Belgian eID card, kids card or foreigner card. These are the **main components** of the infrastructure:

- A test card, to be purchased
- A website to generate virtual cards for this test card, you can specify the contents for these virtual cards yourself
- Software for using these virtual cards in your test application (Windows XP and Vista, Fedora and Mac OS 10.4 and 10.5).

This manual will explain how to generate those virtual cards and how to work with the software for using those virtual cards.

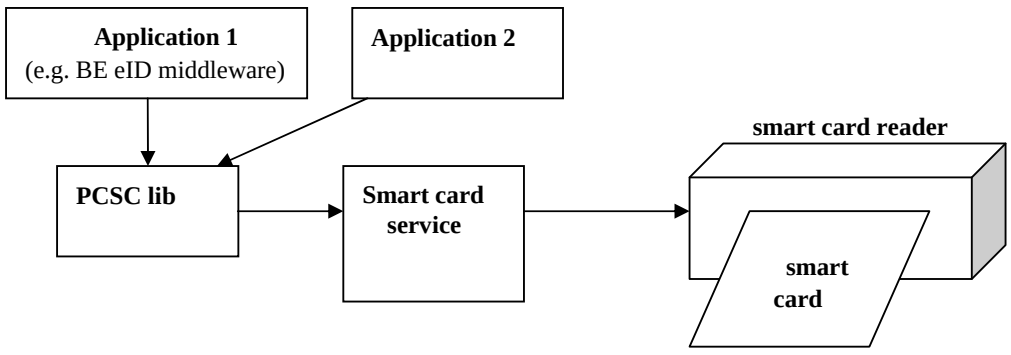
As will be explained below, the general idea is to let your test application or test lib use the **PCSC proxy library** instead of the real PCSC library. This PCSC proxy library will emulate virtual cards in virtual readers, and its behavior is specified by a control file that can be modified by means of **control tools** (a GUI and a command line application).

2. Physical cards and PCSC

An **eID card** has a simple file structure containing a number of directories and files. Each file and directory is identified by 2 bytes called the file ID.
For example, the root directory has a file ID *3F 00* (hex). The directory containing the ID, address, photo, .. files has file ID *DF 01*, while the directory containing the certificates has file ID *DF 00*. The photo file has ID 40 35, so it's full path is *3F 00 DF 00 40 35*.
See [4] for more details about the files on an eID card.
Apart from files, the eID card also contains a PIN and keys that are not visible in the file structure.
Each card has a unique chip number that can be used to identify the card.

PCSC [1] is the most common API to communicate with smart cards (via a smart card reader). PCSC is available on Windows, Linux and Mac OS X and provides functions such as:

- to list the readers connected to a PC
- check if a card is (still) present in a reader
- connect to a card in a reader, or disconnect from it
- send commands (called APDUs) to the smart cards and get back the responses

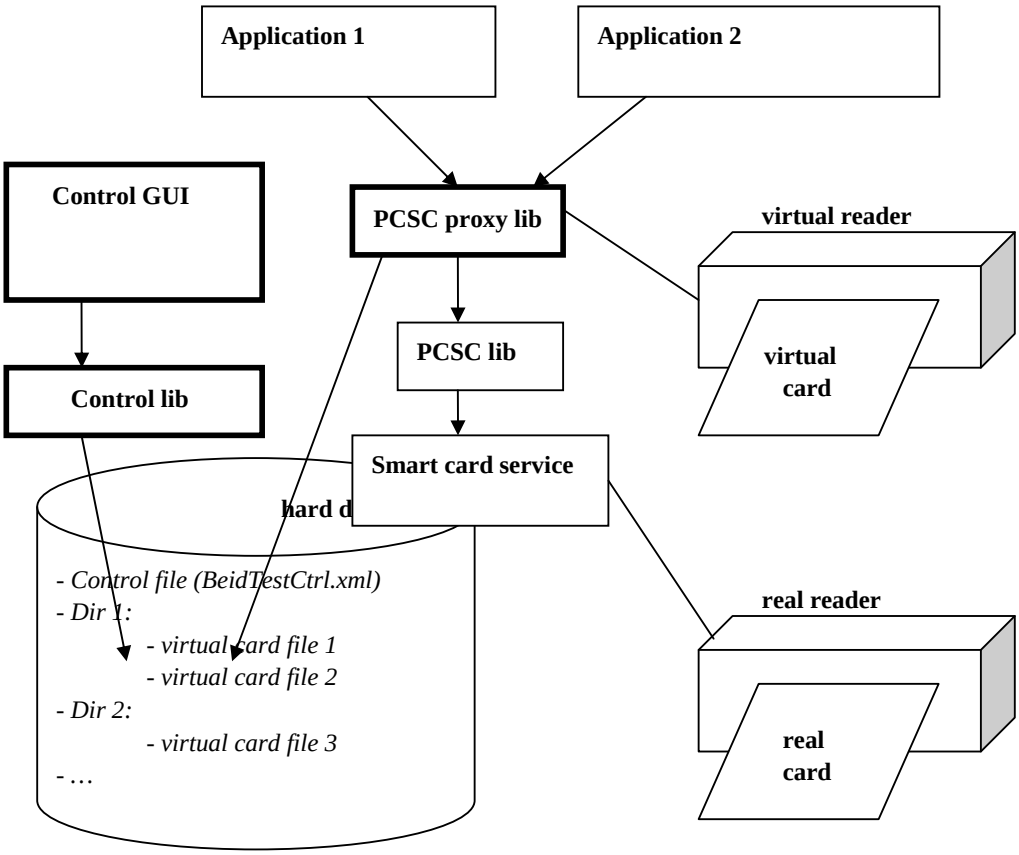


On Windows, the PCSC library is *%windir%\system32\winscard.dll*
On Mac OS X, the PCSC library is */System/Library/Frameworks/PCSC.framework/PCSC*
On Linux, the PCSC library may not be installed by default, but is usually put in */usr/lib/libpcsclite.so*

3. Virtual cards and PCSC proxy

A **virtual card** is in fact a simple XML file containing, apart from some extra information, the paths and contents of all the files that are on a normal card. You can make several virtual cards for a physical test card. The virtual cards don't contain PINs and keys; PIN and key related operations (PIN verify, PIN change, digital signatures) will be forwarded to the physical test card.

The software that is part of this eID test infrastructure contains a **PCSC proxy library** that will emulate virtual smart card readers containing those virtual cards. The PCSC proxy library has the same API as the PCSC library itself; so it is possible to replace the PCSC library by this PCSC proxy library as will be explained later.



The **behavior** of the PCSC proxy lib can be summarized as follows:

- For each physical card reader, a virtual card reader is emulated
- You can specify which virtual card should present (in the virtual reader) for each physical test card that is present in a physical reader. When you insert the physical card in the physical reader, the virtual card appears in the physical reader, and idem for removal.
- When you read data from the virtual card, the PCSC proxy lib returns the data from the virtual contents file for that virtual card.
- If no virtual card has been selected for a test card, an "unknown" card will be emulated that returns SW1/SW2 = 6D 00 to all commands sent to the card.
- For PIN and key related operations on the virtual card, the PCSC proxy lib will ask the physical test card to perform those operations (via the PCSC lib).

- When you change the visibility of the readers (e.g. hide this physical readers) then this change will take effect the next time an SCardListReaders() call is done.
- When you switch to another virtual card, this is NOT automatically noticed by the PCSC proxy lib. You should do (an *SCarddisconnect()* and then) an *SCardConnect()* before you can use the new virtual card.

Which virtual card is to be used for a certain physical test card is specified in a **control file**. The PCSC proxy library reads out this control file when data must be read from a card in a virtual reader.

Additionally, the PCSC proxy lib can hide the physical or virtual readers, this can also be specified in the control file.

The control file is a system-wide xml file on a fixed location:

- Windows: *%ALLUSERSPROFILE%\BeidTestCtrl.xml* (usually *%ALLUSERSPROFILE%* is *C:\Documents and Settings\All Users* on Windows XP and *C:\ProgramData* on Vista).
- Linux, Mac OS X: */usr/share/BeidTestCtrl.xml*

4. Generating virtual cards

Once you have purchased a test card, you can generate virtual cards for this test card on [2].

In order to enter the part of the website where you can generate virtual card, you need to **authenticate yourself with your test card**. This is probably the hardest part. The BE eID middleware [3] contains software and documentation for various browsers such as Internet Explorer, Firefox and Safari.

Once you logged in with your test card, the actual virtual card generation is easy: you can select a template and modify it according to you needs.

There is a separate document explaining how to insert **Unicode** characters.

A special value is the **friendly name**. This one won't be put in any of the files on the virtual card, but can instead be used later to recognize and distinguish the virtual cards you generated. Therefore it is advisable to use a meaningful, unique name, e.g. "Lena foreigner" or "Tommy 1997"

When you submit the form, the web site will generate the files for the virtual card with the data in the form. This includes a.o. the ID file, the address file, the signatures over those files, the photo file and the certificates. Save the generated XML file to you hard disk; this is your virtual card.

The public key in the Authentication and Signature **certificates** is the same as in the certificates of the physical test card; as a result certificates on the virtual cards match with the Authentication and Signature keys on the physical test card. If you perform an authentication or signature with the virtual card (which doesn't contain keys or PINs), the PCSC proxy lib forwards these commands to the physical test card.

5. Using the software

After installing the software that is part of this eID test infrastructure, the following main components are present:

- the PCSC proxy lib
- control tools to manage the control file

5.1 Control tools

As explained earlier, the PCSC proxy lib emulates virtual readers and virtual smart cards, based on info in a system-wide control file. The control tools provides a simple way to manage the control file. More specifically, you can:

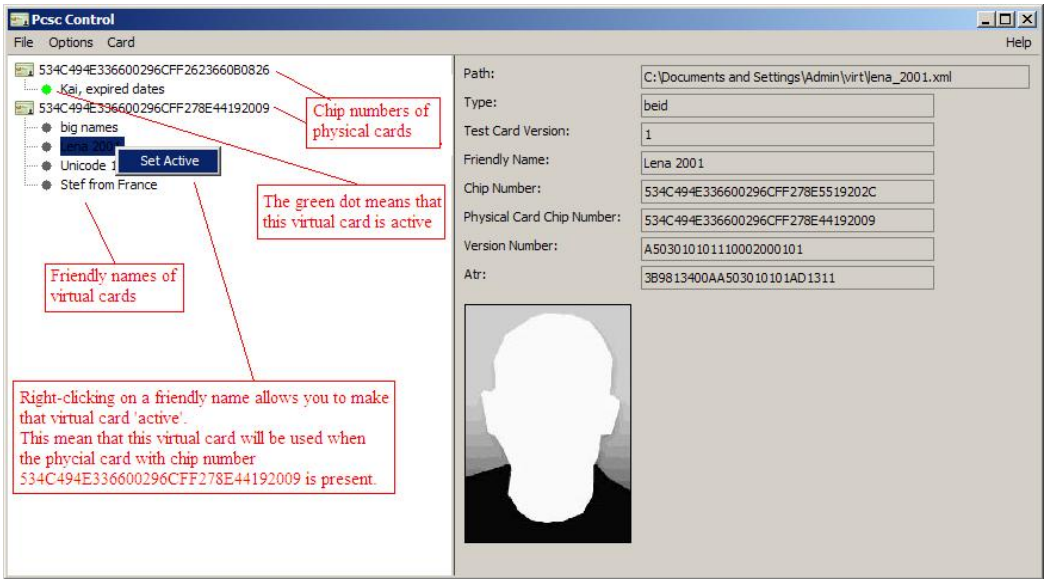
- Add and remove directories that contain virtual card files (adding such a directory is the first thing you should do).
- List the virtual cards that are available for a physical test card
- Select which virtual card will be used by the PCSC proxy lib (“Set Active”)
- Set the reader visibility:
    - show only the virtual readers
    - show only the physical readers
    - show all, the virtual readers first
    - show all, the physical readers first

The are 2 control tools: a GUI and a command line tool.

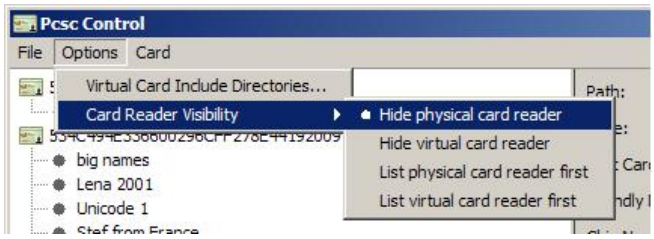
To start the **control GUI**:

- On Windows: *Start – Program – eID Test Infra – EidTestInfra-control*
- On Linux: */usr/share/eidtestinfra/pcsccontrol.sh*
- On Mac: *Applications – eidtestinfra – pcsccontrol.app*

In the left panel of the control GUI, the chip numbers of all physical test cards are visible; and for each physical test card the friendly names of the virtual cards that were generated with that physical test card. You can set one of the virtual cards active by right-clicking on its friendly name and selecting ***Set Active***. The active virtual card is the one that will be used by the PCSC proxy lib.
In the right panel, some info about the selected virtual card is shown.



You can add or remove directories in which the control tool will look for virtual cards in the ***Options*** menu. In this ***Options*** menu, you can also set the card reader visibility.



In the ***Card*** menu, you can **unblock** a physical test card.

Note

The list of directories in which is looked for virtual cards is *per-user* while the control file is *system-wide*. When you open the control GUI and the control file contains an active card with a path that is not present in your own directory list, the control Gui will ask if it should add this directory.

If multiple users use this software, it is up to those users to ensure that the directories and virtual files in those directories can be read by everyone.

To use the **command line tool**, do the following to see the command line options:

- On Windows: open a command window (cmd.exe) and type in *cd C:\Program Files\Eid Test Infrastructure*. Then type in *pcsccontrol.bat –h*
- On Linux, open a command shell and type in */usr/share/eidtestinfra/pcsccontrol.sh –h*
- On Mac OS X, open a Terminal window (*Applications – Utilities – Terminal*) and type in */usr/share/eidtestinfra/pcsccontrol.sh –h*

The command line tool offers the same functionality as the control GUI, except for unblocking test cards.

5.2 Using the PCSC proxy lib

In order to be able to use virtual cards in your application, ***the PCSC proxy lib should be used instead of the normal PCSC lib*** that is present on your PC. The way to do this depends on the OS you are working on. Since version 1.2 of the software, there as also a 64 bit version included.

On Windows

Copy the *C:\Program Files\Eid Test Infrastructure\proxy32\winscard.dll* or *C:\Program Files\Eid Test Infrastructure\proxy64\winscard.dll* to the directory where your (test) application is; which one depends whether your (test) application is a 32 or 64 bit binary.

Due to the way Windows searches for DLLs [5], your (test) application will now automatically use the proxy PCSC lib instead of the physical PCSC lib.

Remark

If you made a DLL that uses the PCSC library instead of the (test) application itself, you still have to copy the PCSC proxy lib to the same dir as where your (test) application resides. It doesn't matter where the DLL itself is.

For example, to use the PCSC proxy lib with the eID-Viewer of the BE eID middleware, you should copy the PCSC proxy lib to *C:\Program Files\Belgium Identity Card* where the eID-Viewer (beid35gui.exe) is, not to *%windir%\system32* where the beid35cardlayer.dll is

On Linux

Add the location of the PCSC proxy lib to your LD_LIBRARY_PATH variable. This can usually be done by the following command:

export LD_LIBRARY_PATH=/usr/share/eidtestinfra/:$LD_LIBRARY_PATH

Depending on whether your (test) application is 32 or 64 bit, you have to do
/usr/share/eidtestinfra/setproxy.sh 32
or
/usr/share/eidtestinfra/setproxy.sh 64

On Mac OS X

If your test application or test lib doesn't contain the full path to the PCSC lib, you can add the location of the PCSC proxy lib to your DYLD_LIBRARY_PATH variable as follows:

export DYLD_LIBRARY_PATH==/usr/share/eidtestinfra/:$DYLD_LIBRARY_PATH

By default however, the linker puts the full path to the PCSC lib in you library or application. The solution in this case is to use *install_name_tool* to renames the path to the PCSC lib into the path to the PCSC proxy lib. A script, ***/usr/share/eidtestinfra/proxyfy.sh*** has been made to facilitate this operation, as well as a script, ***/usr/share/eidtestinfra/unproxyfy.sh***, to undo the rename. As a parameter, this script accepts the path to the application or library that has to be 'proxy-fied' or "un-proxy-fied".

Remark 1

Remark: on Mac, applications are typically put in an application (.app) directory, the application itself is typically in the *Contents/MacOS/* dir.

For example, if you made a test.app, you should proxy-fy *test.app/Contents/MacOS/test*.

Remark 2

If you made a dylib that uses the PCSC library instead of the (test) application itself, you should "proxy-fy" the dylib, not the (test) application.

For example, to use the PCSC proxy lib with the eID-Viewer of the BE eID middleware, you should proxy-fy the *libbeidcardlayer.dylib*:
sudo /usr/share/eidtestinfra/proxyfy.sh /usr/local/lib/libbeidcardlayer.dylib

Remark 3

To check against which libraries an application or lib is linked to, you can use the following command:

otool –L <path>

In which <*path*> is the path to the application or lib.

General Remark

Please make sure to use the 32 bit proxy for a 32 bit application and vice versa for 64 bit. A 32 bit pcsc proxy lib can't be loaded by a 64 bit program and vice versa; you won't see an error or a crash but the virtual card will not be shown because the pcsc proxy lib isn't loaded.

5.2.1 Dynamicly loaded PCSC lib

In some rare cases, the normal PCSC lib is loaded dynamically at runtime (using ***LoadLibrary()*** on Windows or ***dlopen()*** on Linux/Mac). If, in addition, the path to the normal PCSC lib is hardcoded and the application/library can't be (easily) rebuild, then the techniques described in the previous chapter can't be used.

The only case we currently know of is the javax.smartcardio pacackage in Sun's JRE 1.6 on Linux (and Mac?). On Windows, the normal PCSC lib is presumably also loaded dynamically but probably not with an absolute path; so if you copy the proxy PCSC lib to the same dir as java.exe then all works fine).

On Linux, the following solution has been provided. Instead of only setting the LD_LIBRARY_PATH, you also have to set the LD_PRELOAD variable like this:

export LD_PRELOAD="/usr/share/eidtestinfra/libdl_proxy.so /usr/share/eidtestinfra/libpcsclite.so"

The *libdl_proxy.so* contains an implementation of dlopen() that replaces attempts to open the normal PCSC lib by the proxy PCSC lib.

On Mac OS X, a similar solution is provided:

export DYLD_FORCE_FLAT_NAMESPACE=1

export DYLIB_INSERT_LIBRARIES=/usr/share/eidtestinfra/libdl_proxy.so:/usr/share/eidtestinfra/PCSC

On Windows, no ready-to-use solution is provided; however see the *loadlibraryproxy* directory in the *Eid Test Infrastructure* installation directory for an explanation in case this problem might occur.

5.3 Summary: how to use a virtual card

The following should be done in order to use a virtual card in your application/lib:

- Your application/lib should use the PCSC proxy lib instead of the real PCSC lib. How to do this depends on the OS, see section 5.2.
- The physical test card that was used to generate the virtual card should be inserted in a smart card reader.
- The control file should be present and should contain an entry with the chip number of the physical test card and the location of the virtual card. You can use the control tool for this. See section 5.1. Also: make sure that the virtual card readers are visible.

6. OCSP and CRL

The virtual cards contain certificates, just like the real eID cards. They are generated by a test PKI and, like the real PKI, an OCSP responder is available and CRLs are generated. More specifically, the following certificates are present:

- the 2 user certificates (the Authentication and Signature certificate), for which a private key is present on the physical test card
- the intermediate CA certificate (the ‘foreigner’ or ‘citizen’ certificate)
- the root certificate
- the RRN certificate, this is the certificate of the National Register that can be used to verify the signatures on the ID and Address files.

When you generate a virtual card, the Authentication and Signature certificates are created based on the name and validity dates you filled in on the web form; and using the public keys of the physical test card.

You can modify the **certificate status** of the Authentication and Signature certificates on the card by logging in with your physical test card to the website [2], much the same way as for generating virtual cards. There are 3 different values for a certificate status:

- Active
- Suspended (‘on hold’)
- Revoked

Note

The Revoked status is irreversible: if you change the status of a certificate to Revoked, you cannot change it back again.

After you changed the status of a certificate, it is immediately propagated to the OCSP responder, and it will appear in the next CRL and delta CRL. A CRL is generated every hour, a delta CRL every 15 minutes. (The values differ form the real CRLs and delta CRLs that are generated every day).

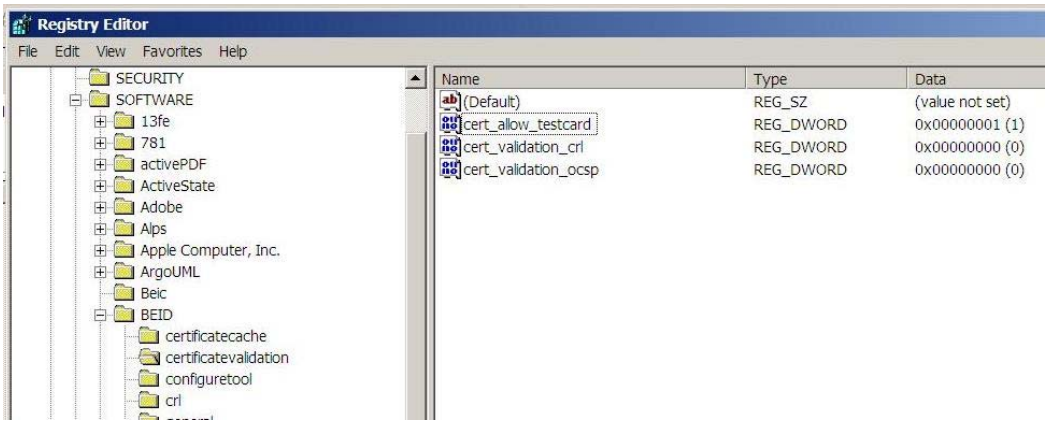
7. About the BE eID middleware

If you want to use the BE eID middleware with physical and virtual test cards, you should read this section.

7.1 Version 3.X

A. By default, only official eID, foreigner and kids cards can be read by the middleware. To **enable the reading of test cards**, you should set the ***cert_allow_testcard*** config option to ***1***:

- On ***Windows***: open the Registry (run regedit.exe) and go to *HKEY_LOCAL_MACHINE\SOFTWARE\BEID\certificatevalidation*. In the right panel, right-click and select New – DWORD Value. Enter ***cert_allow_testcard*** as name. Then double-click on this name and give ***1*** as value.



- On ***Linux*** and ***Mac OS X***, you should add the following to the [certificatevalidation] section of the config file */usr/local/etc/beid.conf*:
  cert_allow_testcard=1
  Below is an example of a part of a config file:
  [certificatevalidation]
  cert_validation_crl=0
  cert_validation_ocsp=0
  cert_allow_testcard=1

Remark: instead of the *system-wide* config settings, you could also modify the cert_allow_testcard option in the *per-user* config settings. On Windows, this means using *HKEY_CURRENT_USER* instead of *HKEY_LOCAL_MACHINE*. On Linux, it means using file *~/.config/beid.conf* and on Mac OS X it means using file *~/Library/Preferences/beid.conf*.

B. If you want to see the **chip number** of your physical or virtual card, you can use the eID-Viewer of the middleware: the ***Extra*** tab contains the chip number.

C. You can also use the eID-Viewer to find out **which (virtual) readers are present**: the Options menu contains a drop-down list with all card readers found on the PC.

D. The middleware caches the **OCSP** responses and **CRL**s. As a result, you may not immediately see the changes you made to the status of the certificates on your virtual cards. If you don't want to wait, you can remove the cache files: On Windows, you can find the cache files in C:\Program Files\Belgium Identity Card\eidstore\crl. On Linux and Mac OS X, they are in /tmp/crl/. It it safe to delete those directories if no application is running that uses the middleware.

E. The current eID-Viewer only shows 2 different OCSP results: ***revoked*** (which also includes the "suspended" state) and ***valid***.

F. It is advisable not to modify the virtual card content files because the middleware caches some of them because it assumes they can never change. In general, modifying the virtual card content files will cause problems because most file are protected by signature, so modifying as little as one bit will invalidate those signatures, which is detected by the middleware.

7.2 Version 4.X and other Java applications

The current middleware is in fact a Java application that is started by a small binary (exe).

There is no more need to allow test cards like for the 3.X versions.

On the other hand, the eid-viewer no longer has a '*select reader*" box so you can't simply tell whether the virtual card reader is visible. (If it should be and it is not, it is a typically a sign that the pcsc proxy lib hasn't been loaded).
What you can do in this case is go to "Help" -> "Show Log Tab" and set the select box to "ALL" and then restart the eid-viewer. Then the log view should contain a line starting with
[BelgianEidViewer] Scanning card terminal: XXX
that tells which (virtual) reader will be used by eid-viewer to read out a (virtual) card.

On ***Windows***, since it is in fact Java that has to load the PCSC proxy, you have **to copy the pcsc proxy lib to the same dir as where *java.exe*** is located. In practice, there are often multiple jre and jdk dirs that contain a "*bin*" dir having a *java.exe* and it is not always clear which *java.exe* is used in every case.

Therefore, we advise to try to copy the pcsc proxy to each directory after the other until the virtual card is found; or otherwise copy the pcsc proxy to all directories containing *java.exe*.


REMARK: please make sure to copy the 64 bit pcsc proxy lib to the dir(s) containing the 64 bit java.exe; and vice versa for the 32 bit case.
A 32 bit pcsc proxy lib can't be loaded by a 64 bit java.exe and vice versa; you won't see an error or a crash but the virtual card will not be shownbecause the pcsc proxy lib isn't loaded.

On ***Linux*** and ***Mac OS X***, please see the section "Dynamically loaded PCSC-library".

References

[1] PCSC API: http://msdn.microsoft.com/en-us/library/aa374731(VS.85).aspx#smart_card_functions

[2] eID test infrastructure web site: https://env.dev.eid.belgium.be/

[3] BE eID middleware: http://eid.belgium.be

[4] Contents of an eID card:
http://www.ibz.rrn.fgov.be/fileadmin/user_upload/CI/eID/5%20aspects%20techniques/nl/belgian_electronic_identity_card_content_v2.8.a.pdf

[5] http://msdn.microsoft.com/en-us/library/ms682586.aspx