SafeSign Identity Client

Administrator's Guide

A.E.T. Europe B.V.

IJsselburcht 3

NL - 6825 BS Arnhem

The Netherlands

Warning Notice

All information herein is either public information or is the property of and owned solely by A.E.T. Europe B.V. who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

This information is subject to change as A.E.T. Europe B.V. reserves the right, without notice, to make changes to its products, as progress in engineering or manufacturing methods or circumstances warrant.

Installation and use of A.E.T. Europe B.V. products are subject to your acceptance of the terms and conditions set out in the license Agreement which accompanies each product. Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/ or industrial property rights of or concerning any of A.E.T. Europe B.V. information.

Cryptographic products are subject to export and import restrictions. You are required to obtain the appropriate government licenses prior to shipping this Product.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, A.E.T. Europe B.V. makes no warranty as to the value or accuracy of information contained herein. The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, A.E.T. Europe B.V. reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

A.E.T. EUROPE B.V. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE INFORMATION CONTAINED HEREIN, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL A.E.T. EUROPE B.V. BE LIABLE, WHETHER IN CONTRACT, TORT OR OTHERWISE, FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER INCLUDING BUT NOT LIMITED TO DAMAGES RESULTING FROM LOSS OF USE, DATA, PROFITS, REVENUES, OR CUSTOMERS, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION CONTAINED IN THIS DOCUMENT.

SafeSign Identity Client is a trademark of A.E.T. Europe B.V. All A.E.T. Europe B.V. product names are trademarks of A.E.T. Europe B.V. All other product and company names are trademarks or registered trademarks of their respective owners.

Credit information:

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

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

Contact Information: A.E.T. Europe B.V.

IJsselburcht 3
NL-6825 BS
P.O. Box 5486
NL-6802 EL Arnhem
The Netherlands
Tel. +31-26-365 33 50
Tel. Support +31-26-365 35 43
Fax +31-26-365 33 51

info@aeteurope.nl / support@aeteurope.nl
http://www.aeteurope.com/

SafeSign Identity Client is a product developed by A.E.T. Europe B.V.

Document Information

Filename: **SafeSign Identity Client Administrator's Guide**

Document ID: **Administrator_Guide_SafeSign-IC_v3.2**

Project Information: **SafeSign Identity Client Administrator Documentation**

Document revision history

Version	Date	Author	Changes
1.0	19-12-2005	R.M. van Rijswijk	First edition for SafeSign Identity Client Version 2.2 for Windows (release 2.2.0)
2.0	29-08-2006	R.M. van Rijswijk	First edition for SafeSign Identity Client Version 2.3 for Windows (release 2.3.0)
2.1	11-01-2007	Drs C.M. van Houten	Edited for SafeSign Identity Client Version 2.3 for Windows (release 2.3.2)
3.0	22-09-2008	Drs C.M. van Houten	Edited for SafeSign Identity Client Version 3.0 for Windows (release 3.0.15)
3.1	28-12-2009	Drs C.M. van Houten	Edited for SafeSign Identity Client Version 3.0 for Windows (release 3.0.33)
3.2	07-07-2011	Drs C.M. van Houten	Edited for SafeSign Identity Client Version 3.0 for Windows (release 3.0.45 / 3.0.45-x64)

WE RESERVE THE RIGHT TO CHANGE SPECIFICATIONS WITHOUT NOTICE

Table of contents

Table of Figures

About the Product

SafeSign Identity Client is a software package that can be used to enhance the security of applications that support hardware tokens through PKCS #11 and Microsoft CryptoAPI.

The SafeSign Identity Client package provides a standards-based PKCS #11 Library and Cryptographic Service Provider (CSP), allowing users to store public and private data on a personal token, either a smart card, USB token or SIM card. It also includes the SafeSign Identity Client PKI applet, enabling end-users to utilise any Java Card 2.1.1 / Java Card 2.2 and higher compliant card with the SafeSign Identity Client middleware.

Combining full compliance with leading industry standards and protocols, with flexibility and usability, SafeSign Identity Client can be used with multiple smart cards / USB tokens, multiple Operating Systems and multiple smart card readers.

SafeSign Identity Client allows users to initialise and use the token for encryption, authentication or digital signatures and includes all functionality necessary to use hardware tokens in a variety of PKI environments.

SafeSign Identity Client comes in a standard version with an installer for the following Windows environments[1]:

Windows XP (Professional), Windows Vista, Windows 7, Windows Server 2003 and Windows Server 2008.

In principle, SafeSign Identity Client supports any PC/SC (including PC/SC 2.01) compliant smart card reader. However, to avoid power problems, smart card readers must be capable to provide at least a current of 60mA. PC/SC driver software is available from the web site of the smart card reader manufacturer.

For more information, refer to the latest SafeSign Identity Client Product Description

[1] Windows NT 4.0 is supported up to SafeSign Identity Client 1.0.9.04, in line with Microsoft's end-of-life policy.
Windows 98 and Windows ME are supported up to SafeSign Identity Client 2.3.0 (< 2.3.0), in line with Microsoft's end-of-life policy.
Windows 2000 is supported up to SafeSign Identity Client 3.0.33 (≤ 3.0.33), in line with Microsoft's end-of-life policy.

1 About this Administrator's Guide

1.1 Intended audience

This manual is intended for administrators who want to change the default configuration of the token and the default behaviour of SafeSign Identity Client (on a per-computer basis). After reading this manual, an administrator will be able, for example, to change the look of the SafeSign Identity Client Token Management Utility / Token Administration Utility and modify its features.

Regular users of SafeSign Identity Client are advised not to make such changes, as this may cause irreparable damage and may lead to malfunctioning of SafeSign Identity Client.

1.2 Prerequisites

1. You need to have sufficient rights and knowledge to modify the registry.
2. You need to have sufficient knowledge about tokens, and their internal workings to modify the registry correctly.
3. You need to have SafeSign Identity Client Standard Version 3.0.45 installed.
4. You are advised to read, and understand, the following material before modifying the registry: PKCS #11 and PKCS #15 Public-key Cryptography Standards. This material can be found at: http://www.rsasecurity.com/rsalabs/node.asp?id=2124
5. You should be in possession of (one of) the supported tokens, as described in the latest SafeSign Identity Client Standard Version Product Description.

1.3 Additional Warning

This manual contains information about modifying the registry. Before you modify the registry, make sure to back it up and understand how to restore the registry if a problem occurs.

Modification of the registry is done at your own risk. A.E.T. Europe B.V. cannot accept liability for any malfunctioning of SafeSign Identity Client or Windows (applications) due to changes in the registry.

2 General information

2.1 Profiles

For Java Card v2.1.1 / Open Platform 2.0.1 compliant Java smart cards, SafeSign Identity Client Version 3.0 for Windows allows you to initialise your token with multiple profiles (Minimal, Medium=Default and Maximal profile, when available). Moreover, it offers the possibilities to change the profile on the token, from default values to customised values. This means that it is possible to customise the way the token is initialised in terms of public and private space.

For Java Card v2.2+ / Global Platform 2.1.1 compliant Java smart cards, the SafeSign Identity Client applet supports dynamic use of memory (flexible allocation of space). This means that the SafeSign Identity Client Java applet is still initialised with a PKCS #15 file structure, but because the applet allows files to grow and shrink, and even to be created on the fly, not all memory that will be used during the life cycle of the card has to be allocated at initialisation time. Therefore, there is only one profile available, that is default for all Java Card v2.2+ cards that you should **not** edit (as it has already been optimised), at least in terms of public and private space (and related PKCS#15 structure related values).

2.2 Initialisation

During the initialisation of an uninitialized token, the amount of the public and private space is set (and any other PKCS #15 values). These values cannot be changed during the lifetime of the token. This means that during the lifetime of the token, the token keeps the so-called 'profile' that has been created during the initialisation.

For test (completed) tokens, it is possible to change the profile of the token during a re-initialisation of the token (i.e. replace the existing PKCS #15 structure with a new PKCS #15 structure). For series / production (completed) tokens, it is not possible to change a profile once it has been set during initialisation. You may only wipe its contents, while maintaining the PKCS #15 structure written on it during initialisation.

Note that test completed tokens are not intended to be used in production (environments). These tokens are usually only provided for testing and evaluation purposes. Users will generally be provided with series (completed) tokens, that may have the SafeSign Identity Client applet installed (in case of Java cards) and that may even be initialised. Also, it is recommended that for Java cards, the default GlobalPlatform key set is changed to a (customer) specific key set, so the applet(s) cannot be removed (without knowledge of this keyset).

2.3 Overview Registry entries

To change the default behaviour of SafeSign Identity Client, you will need to change the appropriate entries in the registry. To do so, you need a registry editor. This manual assumes that the registry editor used is the Microsoft application 'Registry Editor' ('regedit'). Microsoft provides this registry editor with your operating system. For the working of this registry editor please read the appropriate manuals from Microsoft.

After starting the 'regedit' application go to the entry point:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign]

Below this entry point all information to configure SafeSign Identity Client version 3.0 can be found:

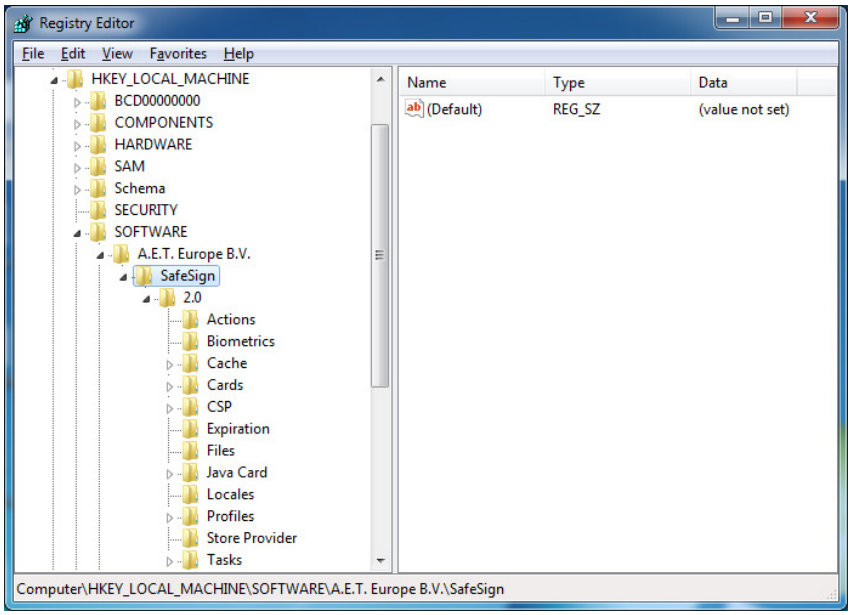

Figure 1: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\

Note

For 64-bit Windows Operating System, the SafeSign Identity Client registry entries can be found in [HKEY_LOCAL_MACHINE|Wow6432Node|SOFTWARE|A.E.T. Europe B.V.|SafeSign]

The next chapters will describe the registry entries for SafeSign Identity Client and their configuration in more detail.

3 SafeSign Identity Client Registry

3.1 General behaviour

With the setting found in

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0]

you can change some general behaviour of SafeSign Identity Client:

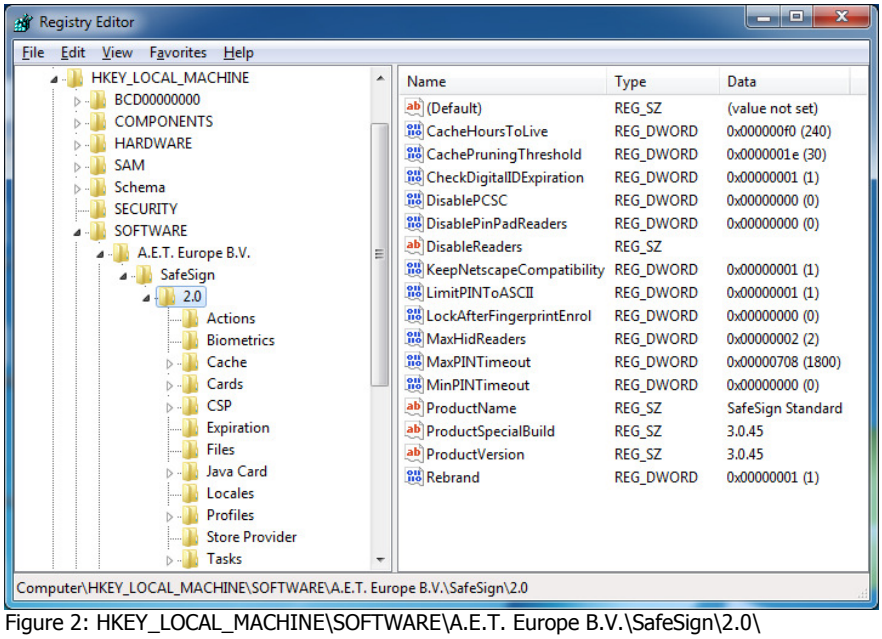

Figure 2: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\

The items an administrator may change are:

- **CheckDigitalIDExpiration**: this DWORD Value manages the appearance of the Certificate Expiration Warning. When changing the value from 1 to 0, SafeSign Identity Client will no longer automatically check the expiration of the certificates on the token, and will no longer warn the end user when certificates are about to expire.
- **DisablePinPadReaders**: this value manages the availability of secure pinpad readers. When changing the value from 0 to 1, SafeSign Identity Client will no longer offer secure PIN entry for all secure pinpad readers attached to the system. The *Enter PIN* dialog will appear, rather than the *PinPad* dialog.
- **DisableReaders**: this value manages the availability of PC/SC smart card readers. When entering the exact name (and slot number) of a particular smart card reader in this field (for example, 'OMNIKEY CardMan 3x21 0'), SafeSign Identity Client will no longer (try to) access the smart card in this reader[1]. Note that it is possible to disable multiple readers, by using a semi-colon (";") as separator. It does not work with a comma (",").

[1] This feature was implemented because some users (particularly of e-banking applications or of particular UMTS cards with PC/SC access) would experience problems 1) when their application requires exclusive access to the card and/or 2) where the card used is (also) recognised or supported by SafeSign Identity Client. In this case, the banking application would not be able to access the card. From SafeSign Identity Client version 3.0.15 onwards (≥ 3.0.15), it is possible to disable a particular smart card reader, so that this smart card reader is excluded from being accessed by SafeSign Identity Client, thus being available for (exclusive) access with the (banking / UMTS) card.

- **LimitPINtoASCII**: This DWORD Value limits the PIN entry to ASCII characters when initialising the token. This value has been implemented to prevent problems (in applications) with PINs with so-called diacrits[1]. When changing the value from 1 to 0, SafeSign Identity Client will no longer limit the PIN entry to ASCII characters only[2].
- **MaxPINTimeout**: This DWORD Value manages the maximum PIN Timeout value in seconds (1800 seconds is 30 minutes).
- **MinPINTimeout**: This DWORD Value manages the minimum PIN Timeout value in seconds. Although this seems to suggest that the minimum PIN Timeout is zero (0) seconds, the minimum PIN Timeout value in the Token Utility is set to 20 seconds[3].

Note

Note that the value DisablePCSC is deprecated as of SafeSign Identity Client version 3.0.33.

3.2 Actions

For different reasons (such as preventing an end-user from being able to accidentally delete his Digital ID), it could be useful to hide or remove some functionality of the Token Management Utility / Token Administration Utility from the user.

Note that SafeSign Identity Client already comes in two different versions: one for users and one for administrators, where the difference is not in the functionality of SafeSign IC, but in the utility provided with it; i.e. the Token Management Utility for end-users and the Token Administration Utility for administrators. The Token Management Utility contains a limited set of features, as compared to the Token Administration Utility. Licensed users are allowed to decide which version to distribute to their users and/or customers.

To hide or remove features in the Token Management Utility / Token Administration Utility, it is necessary to edit the registry. All entries below [HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Actions] reflect a part of the Token Management Utility / Token Administration Utility that can be hidden or made visible.

In principle, when the value of an item in the list of Actions is changed from 1 to 0, the particular Action will not be visible / available in the Token Utility anymore[4].

[1] If you take a look at the PKCS #11 specification, you will see that the PIN is stored as UTF-8 characters (CK_UTF8CHAR). Therefore, PINs that contain characters with diacrits (such as umlaut, accent grave, etc.) and such characters as ß and €, are converted into their relative UTF-8 encoding, which is at least 2 bytes long. That is why the text in the initialisation dialog states "PIN at least x bytes long", not "PIN at least x characters long".

[2] From SafeSign Identity Client version 2.3.2 until 3.0.15 (≥ 2.3.2 and ≤ 3.0.15), this function does not work when disabled. If disabled, it is not possible to enter such (UTF-8) characters as ß, ü and €.

[3] Setting the PIN Timeout to zero seconds would effectively disable the possibility to enter the PIN. The value of 20 seconds is believed to provide the best results.

[4] It is also possible to remove an item from the registry, rather than edit it.

By default, all items are available for editing, in the Token Administration Utility:

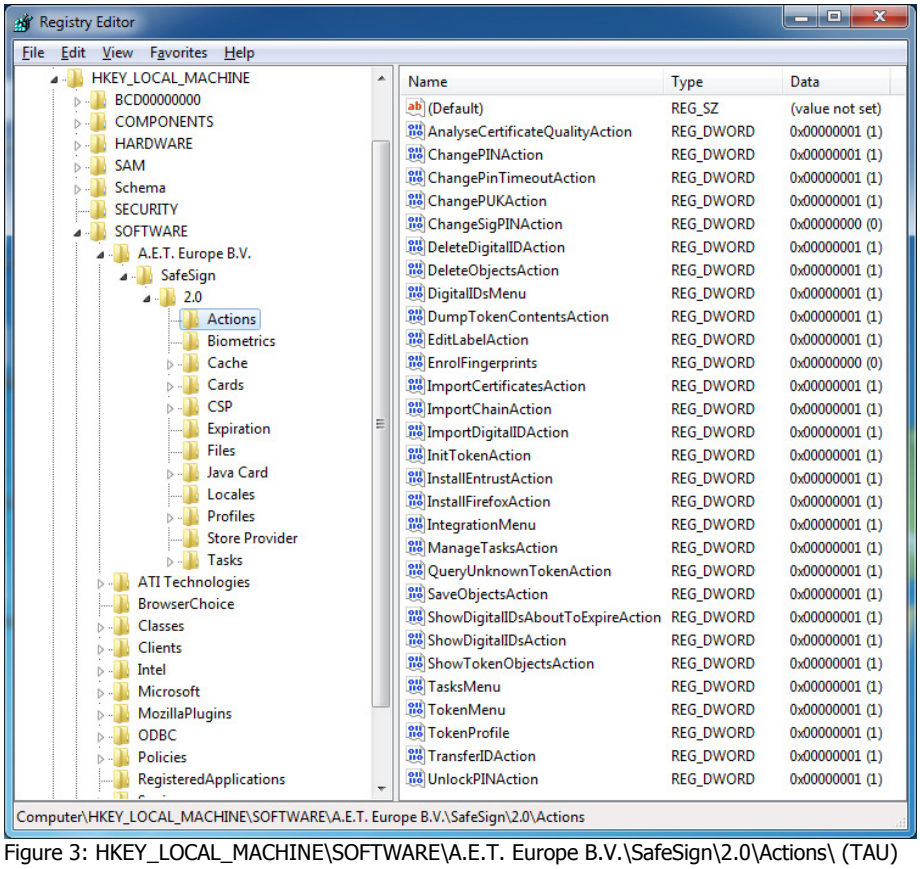

Figure 3: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Actions\ (TAU)

And in the Token Management Utility:

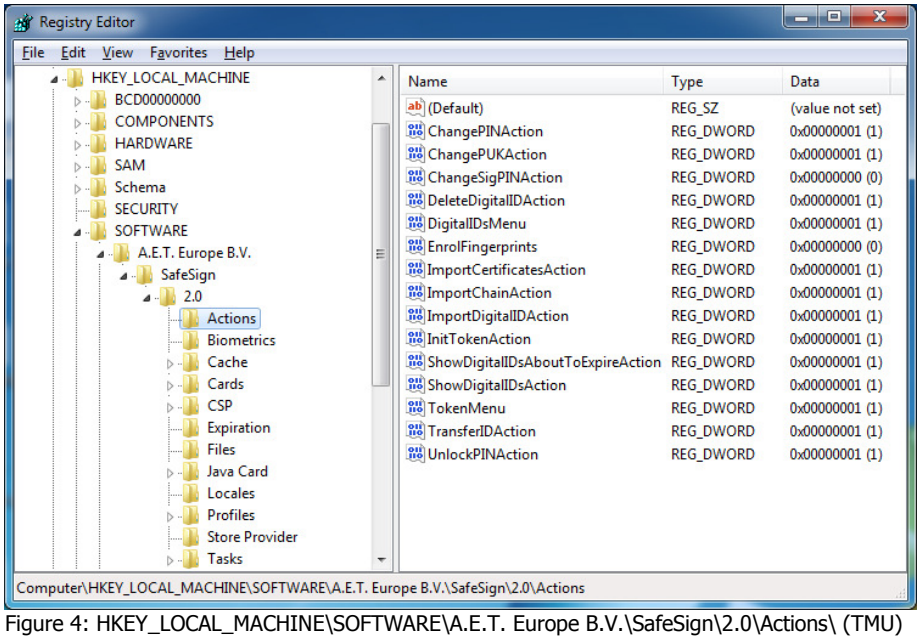

Figure 4: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Actions\ (TMU)

Note

Note that the item EnrolFingerprints (though present) is already deactivated, as this feature does not apply to (non-biometric) SafeSign IC Standard.

Note that the item ChangeSigPINAction is also deactivated, as this feature only applies to digital signature cards with a separate signature PIN (in which case it can be enabled).

Note that it is not possible to disable the Help menu.

In our example in paragraph 3.2.1 we will hide (deactivate) the *Change PIN* item (*ChangePINAction*) from the **Token** menu in the Token Administration Utility:

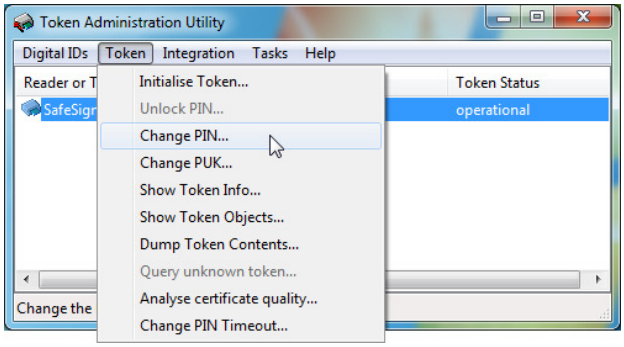

Figure 5: Token Administration Utility: Change PIN

In order to hide other functions, you should follow the steps below for the particular item you want to hide.

3.2.1 Hide Function: Change PIN

To hide the *Change PIN* function in the Token Administration Utility, the registry entry should be changed from 1 (visible) to 0 (hidden). In order to do so, go to:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Actions\ChangePINAction]:

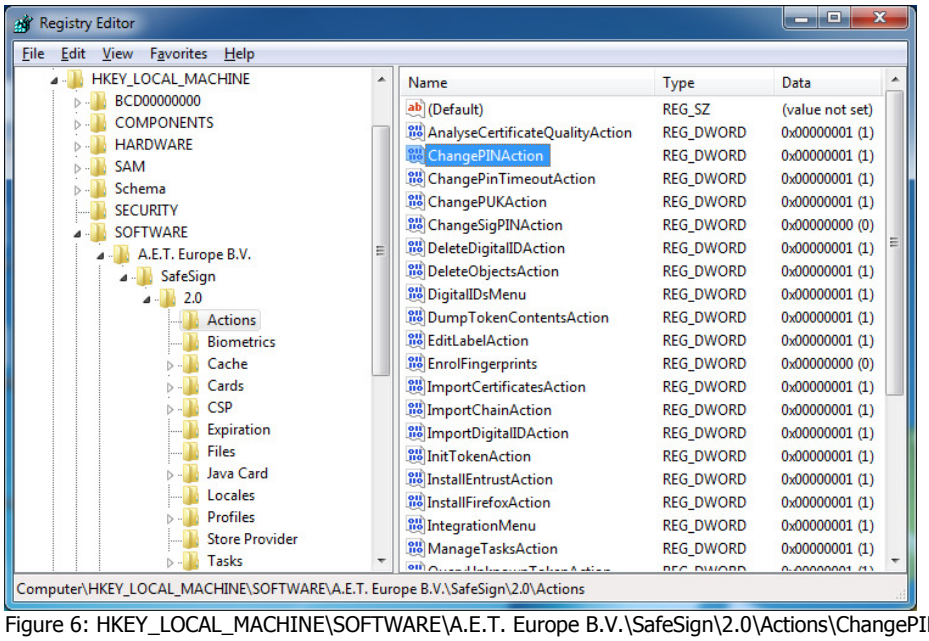

Figure 6: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Actions\ChangePINAction

➔ Right-click the entry and select **Modify**

After you have selected **Modify**, you are presented with the current value of the *ChangePINAction*:

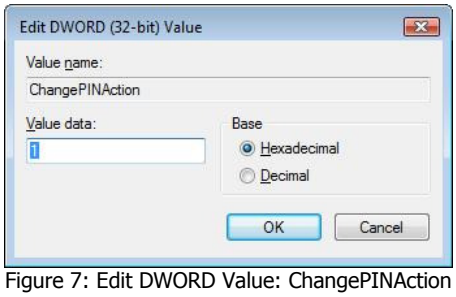

Figure 7: Edit DWORD Value: ChangePINAction Value data 1

In the *Value data* box, change the 1 to a 0 (as below):

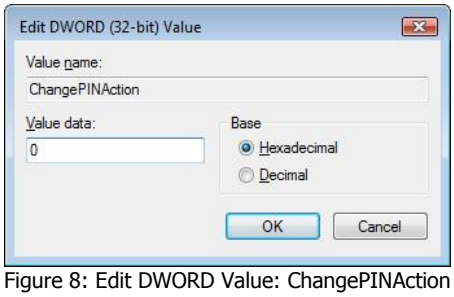

Figure 8: Edit DWORD Value: ChangePINAction Value data 0

➔ Click **OK**

The *ChangePINAction* entry will now look like this:

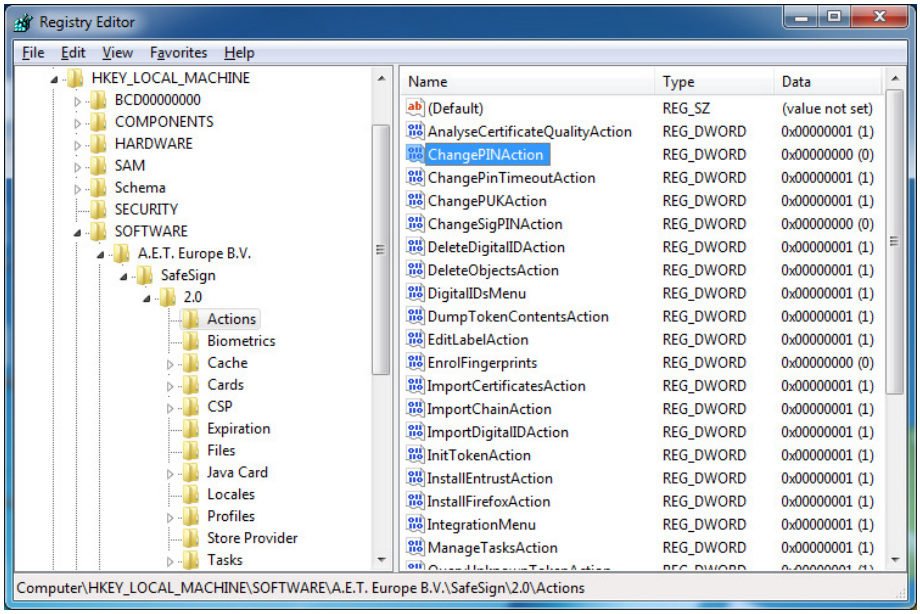

Figure 9: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Actions\ChangePINAction=0

When the Token Administration Utility is restarted, the *Change PIN* item is hidden from the **Token** menu:

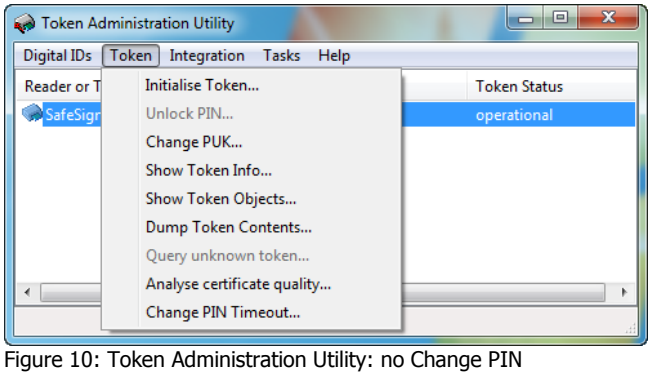

Figure 10: Token Administration Utility: no Change PIN

Thus: In accordance with the above example, in order to change an item from visible to hidden, or vice versa, you will need to change an entry from 1 to 0.

3.2.2 Functions

Not only the functions / items in the Token Management Utility / Token Administration Utility menus can be made hidden or visible (in analogy with the example of *ChangePINAction* above), but also the complete **Token**, **Digital IDs**, **Integration** and **Task** menu.

The menus can be made invisible with the following entries:

- **DigitalIDsMenu**: this entry refers to the **Digital IDs** menu of the Token Management Utility / Token Administration Utility. When changing this value from 1 to 0, this menu will be made invisible. Users will not be able to perform any of the functions under the **Digital IDs** menu.
- **TokenMenu**: this entry refers to the **Token** menu of the Token Management Utility / Token Administration Utility. When changing this value from 1 to 0, this menu will be made invisible. Users will not be able to perform any of the functions under the **Token** menu.
- **IntegrationMenu**: this entry refers to the **Integration** menu of the Token Administration Utility. When changing this value from 1 to 0, this menu will be made invisible. Users will not be able to perform any of the functions under the **Integration** menu.
- **TasksMenu**: this entry refers to the **Tasks** menu of the Token Administration Utility. When changing this value from 1 to 0, this menu will be made invisible. Users will not be able to perform any of the functions under the **Tasks** menu.

The functions for both the Token Management Utility and Token Administration Utility that can be edited are:

- *ChangePINAction*: this entry refers to the *Change PIN* action from the **Token** menu.
  When changing this value from1 to 0, this action will be made invisible.
- *ChangePUKAction*: this entry refers to the *Change PUK* action from the **Token** menu.
  When changing this value from1 to 0, this action will be made invisible.
- *DeleteDigitalIDAction*: this entry refers to the *Delete Digital ID* action (button) in the *Show Registered Digital IDs* dialog (**Token > Show Registered Digital IDs**).
  When changing the value from 1 to 0, this action will be made invisible.
- *ImportCertificatesAction*: this entry refers to the *Import Certificate* action from the **Digital IDs** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *ImportChainAction*: this entry refers to the *Import trust chain* action (button) in the *Digital IDs* dialog (**Digital IDs > Show Registered Digital IDs**).
  When changing the value from 1 to 0, this action will be made invisible.
- *ImportDigitalIDAction*: this entry refers to the *Import Digital ID* action from the **Digital IDs** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *InitTokenAction*: this entry refers to the *Initialise Token / Wipe Token* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.

- *ShowDigitalIDsAboutToExpireAction*: this entry refers to the *Check Expiration* action (button) in the *Digital IDs* dialog (**Digital IDs > Show Registered Digital IDs**).
  When changing the value from 1 to 0, this action will be made invisible.
- *ShowDigitalIDsAction*: this entry refers to the *Show Registered Digital IDs* action from the **Digital IDs** menu.
  When changing the value from 1 to 0, this action will be made invisible
- *TransferIDAction*: this entry refers to the *Transfer ID to token* action (button) in the *Digital IDs* dialog (**Digital IDs > Show Registered Digitals IDs**).
  When changing the value from 1 to 0, this action will be made invisible.
- *UnlockPINAction*: this entry refers to the *Unlock* action from the **Token** menu (which will only be available when the PIN is locked).
  When changing the value from 1 to 0, this action will be made invisible.

Note

Note that there is no entry for the menu item Clean Certificate Cache, but you can add it manually by creating a DWORD Value called CleanCertificateCacheAction, after which you can enable / disable the action.

The additional (unique) functions for the Token Administration Utility that can be edited are:

- *AnalyseCertificateQualityAction*: this entry refers to the *Analyse certificate quality* action from the **Token** menu.
  When changing this value from 1 to 0, this action will be made invisible.
- *ChangePinTimeoutAction*: this entry refers to the *Change PIN Timeout* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *ChangeSigPINAction*: this entry refers to the *Change Signature PIN* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *DeleteObjectsAction*: this entry refers to the *Delete Object* action (button) in the *Show Token Objects* dialog (**Token > Show Token Objects**).
  When changing the value from 1 to 0, this action will be made invisible.
- *DumpTokenContentsAction*: this entry refers to the *Dump Token Contents* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *EditLabelAction*: this entry refers to the *Edit Label* action (button) in the *Show Token Objects* dialog (**Token > Show Token Objects**).
  When changing the value from 1 to 0, this action will be made invisible.
- *InstallEntrustAction*: this entry refers to the *Install SafeSign in Entrust* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *InstallFirefoxAction*: this entry refers to the *Install SafeSign in Firefox* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *ManageTasksAction*: this entry refers to the *Manage tasks* action from the **Tasks** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *QueryUnknownTokenAction*: this entry refers to the *Query unknown token* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *SaveObjectsAction*: this entry refers to the S*ave object* action (button) in the *Show Token Objects* dialog (**Token > Show Token Objects**).
  When changing the value from 1 to 0, this action will be made invisible.
- *ShowTokenObjectsAction*: this entry refers to the *Show Token Objects* action from the **Token** menu.
  When changing the value from 1 to 0, this action will be made invisible.
- *TokenProfile*: this entry refers to the displaying of detailed information about the current token profile in the *Show Token Info* dialog.
  When changing the value from 1 to 0, this information will be made invisible.

3.3 Biometrics

Though no biometric functionality is included in SafeSign Identity Client Standard version 3.0, the *Biometrics* key is available:

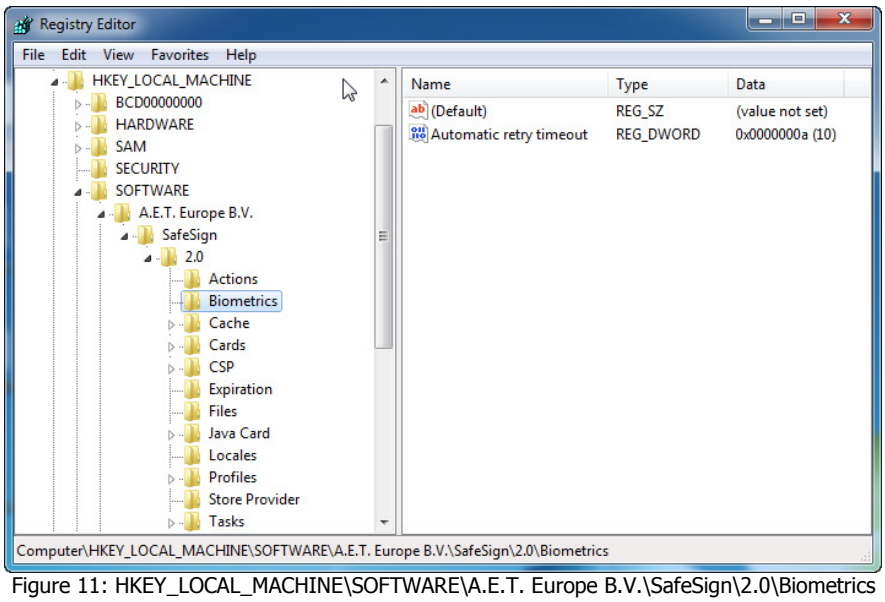

Figure 11: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Biometrics

In SafeSign Identity Client Standard version 3.0, the following value is present:

- *Automatic retry timeout*: this entry refers to the error dialogs that are used for errors that occur when using the G&D StarSign Bio Token. To facilitate ease of use, the "Yes" button can automatically be pressed in these dialogs after a configurable number of seconds. This value configures this number of seconds. The default value is 10.

3.4 Cache

For performance reasons, SafeSign Identity Client stores all the certificate information from all tokens that are or were inserted, in the key:

[HKEY_CURRENT_USER\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cache\]

When a token is inserted that is already known in the cache and the contents of which have not changed, the information is not retrieved from the token, but from the registry entries, thereby increasing speed considerably.

Note that when de-installing / removing SafeSign Identity Client Standard version 3.0, the entries HKEY_CURRENT_USER\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cache will not be removed, to ensure that the certificate and token cache are available for future installed versions of SafeSign Identity Client.

SafeSign Identity Client does not cache private key information, because access to any private key information is never allowed (without logging in to the token with the PIN).

When the content of the token has changed between removal and insertion, the cache is updated the moment the token is inserted.

You should NOT edit / remove the entries (below)
[HKEY_LOCALMACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cache\]
[HKEY_CURRENT_USER\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cache\]
These entries have special privileges.

SafeSign will also store the SHA-1 fingerprint of the certificates on the local machine, for purposes of speed.

- On Windows XP and Windows Server 2003 in *C:|Documents and Settings|[name logged-on user]|Local Settings|Application Data|A.E.T. Europe B.V|SafeSign|2.0|Cache*
- On Windows Vista, Windows 7 and Windows Server 2008 in: *C:|Users|[name logged-on user]|AppData|Local| A.E.T. Europe B.V|SafeSign|2.0|Cache*

Note that you may not be able to see this directory, as these files are hidden by default.

Note

Note that from SafeSign Identity Client version 3.0.33 onwards, the option (in the Token Utility) to clean the certificate cache is included, when this has become corrupted and certificates are not registered anymore. This means that users do not need to go into the registry and delete items manually. However, users should only clear their cache when explicitly instructed to do so by their Helpdesk or system administrator.

The option Clean certificate cache will only remove the entry [HKEY_CURRENT_USER|SOFTWARE|A.E.T. Europe B.V.|SafeSign|2.0|Cache|], not the local certificate cache in the locations above (where the SHA-1 fingerprints are stored).

Note that as of SafeSign Identity Client version 3.0.45, the SafeSign Store Provider that took care of registering the certificates, is no longer included and that certificate registration (propagation) is left to the appropriate Microsoft processes and services.

3.5 Cards

Every token supported in SafeSign Identity Client can be found below the entry:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cards]

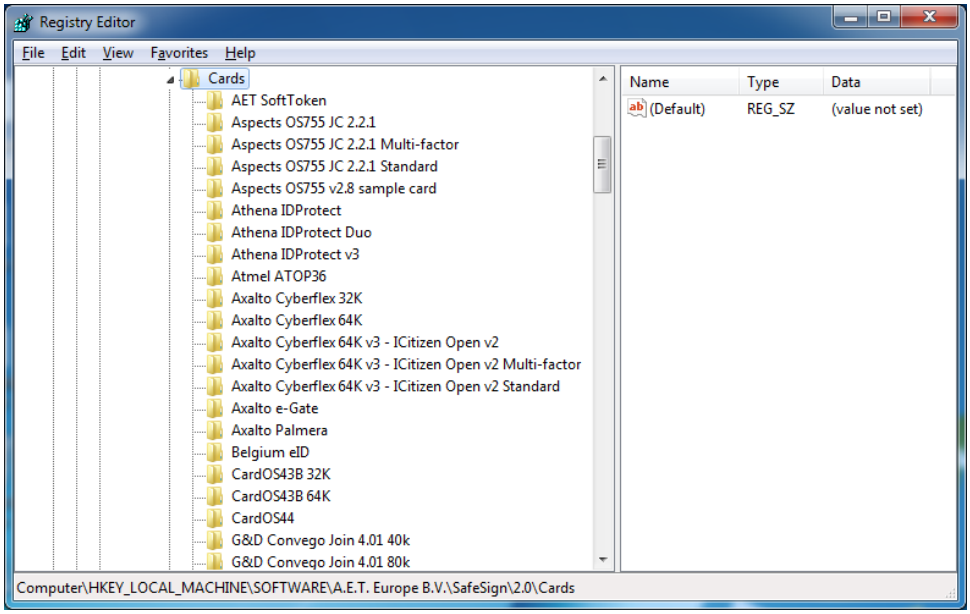

Figure 12: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cards

The separate entries for the tokens also mention the corresponding ModelID (see paragraph 3.11.1.2).

3.6 CSP

This entry has been included in the registry to be able to (a) configure the timeout for the selection of the default key container and to (b) disable applications from deleting key containers.

3.6.1 Default selection timeout

From SafeSign Identity Client version 2.3 onwards (≥ 2.3.2), it is possible to select the default key container to logon to Windows XP[1].

This means that when your token contains more than one Digital ID suitable for smart card logon (either for the same domain or for different domains), it is possible to select the Digital ID you want to use for smart card logon. The user will be presented with a dialog asking him to select the Digital ID he wants to use, if this is enabled in the registry (the DWORD value DefaultContainerSelection in the key [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\SafeSign Standard Cryptographic Service Provider]):

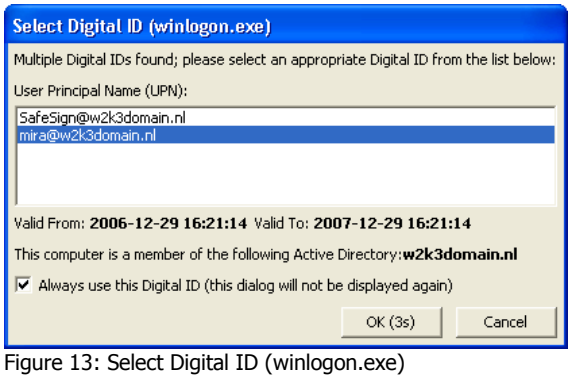

Figure 13: Select Digital ID (winlogon.exe)

In the CSP key, it is possible to select the time this dialog is displayed (when the user has selected that this dialog should not be displayed again[2]). This counter is shown in Figure 13 above.

The default value is 5 seconds:

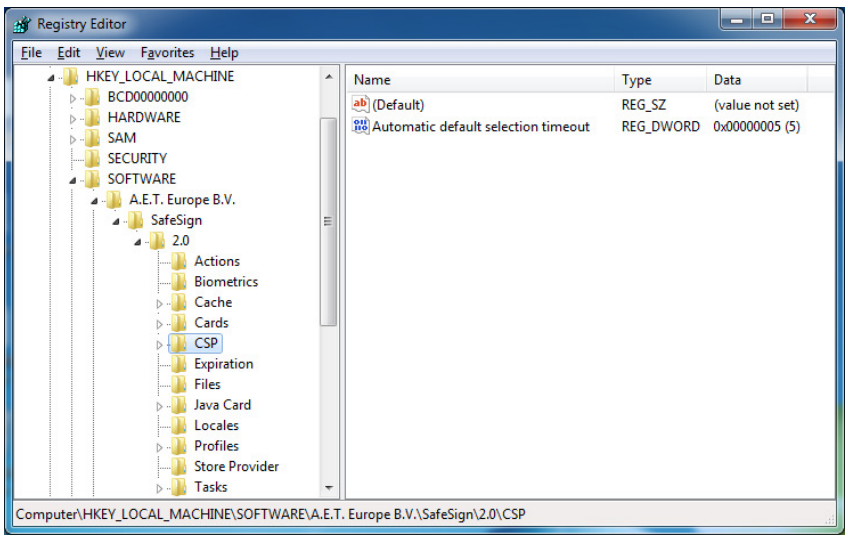

Figure 14: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\CSP\

When the value is set to 0 (zero), the selection dialog will not appear anymore.

[1] Note that in SafeSign Identity Client version 3.0.11, this feature was not available. This has been fixed in version 3.0.15 (≥ 3.0.15).
[2] This functionality has been included to allow users to be aware / reminded of the fact that their token contains multiple Digital IDs for logon (should they want to switch at a given time).

Note

Note that Microsoft made changes with regard to the use of a default key container from Windows Vista onwards.

Though the default selection timeout value can still be enabled on Windows Vista and higher, it will not work with the SafeSign Credential Provider (which does not support multiple certificates on one token), but only with the Microsoft Credential Provider, in which case it is not very useful, as the Microsoft Credential Provider will already display all certificates on the card in the first place.

See also section 4.2.1.

3.6.2 Delete key container

Some applications try to delete the (default) key container when writing a key pair on a token.

For example, the Microsoft 'xenroll' component (xenroll.dll) tries to delete the (default) key container, when requesting a key pair on a token that does not yet contain a Digital ID (in which case it cannot find a key container and enrollment fails) or when requesting a key pair on a token that already contains a (default) key container (in which case it will delete the existing key container).

This is why the xenroll.dll is already included in the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\CSP\

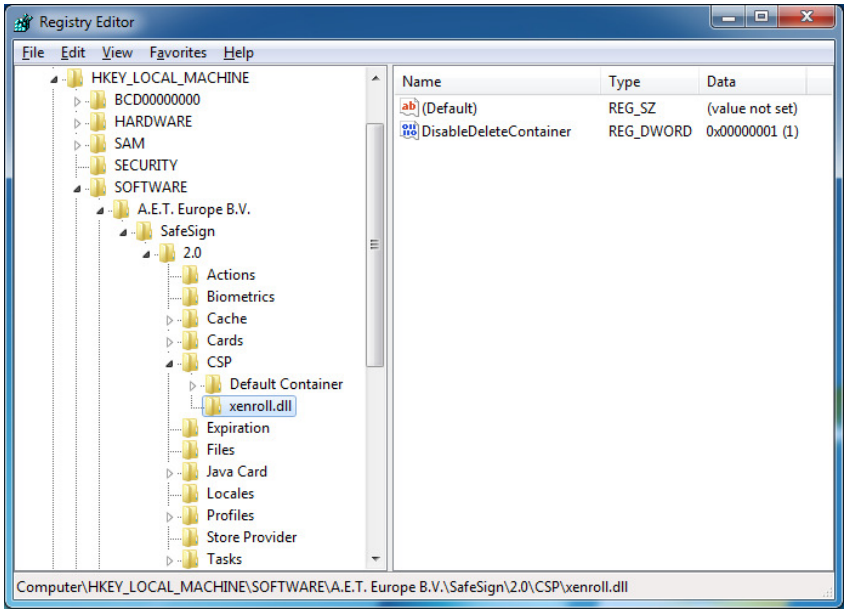

Figure 15: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\CSP\xenroll.dll

When the DWORD Value *DisableDeleteContainer* is set to 1, the executable or dll file cannot delete key containers.

Note

Note that Xenroll.dll is deprecated in Windows Vista and higher and replaced by CertEnroll.dll, which does not require the generation / setting of a (default) key container.

3.7 Expiration

By default SafeSign Identity Client warns an end-user that a certificate is about to expire the moment a token is inserted into reader / machine. The default days in advance that SafeSign Identity Client warns an end-user, is 30 days.

This value can be changed, by changing the *WarnDaysInAdvance* entry in:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Expiration\]:

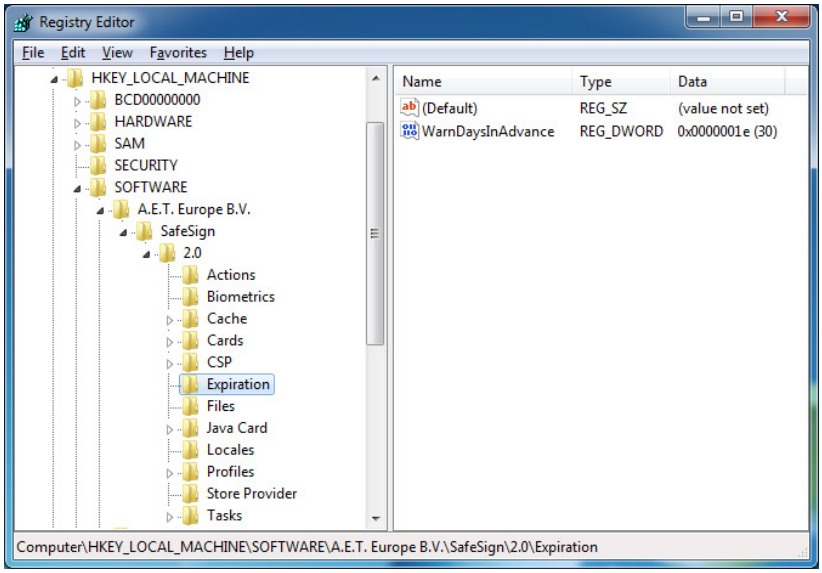

Figure 16: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Expiration\WarnDaysInAdvance

The actual message of the Certificate Expiration Warning can be changed by means of the language files for SafeSign Identity Client. These files are located in *C:\Program Files\A.E.T. Europe B.V.\SafeSign\Locales*, which contains the translation directories for the languages supported by SafeSign Identity Client.

Please be advised that the actual message of the Certificate Expiration Warning must and may only be changed / customised by the Administrator. Further editing of the language files by the administrator or the user is strictly forbidden and will make all warranties void.

For instructions on how to edit the language files, please contact support@aeteurope.nl

3.8 Files

The items in the registry entry

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Files\]

indicate where SafeSign Identity Client can find the necessary files for its correct functioning:

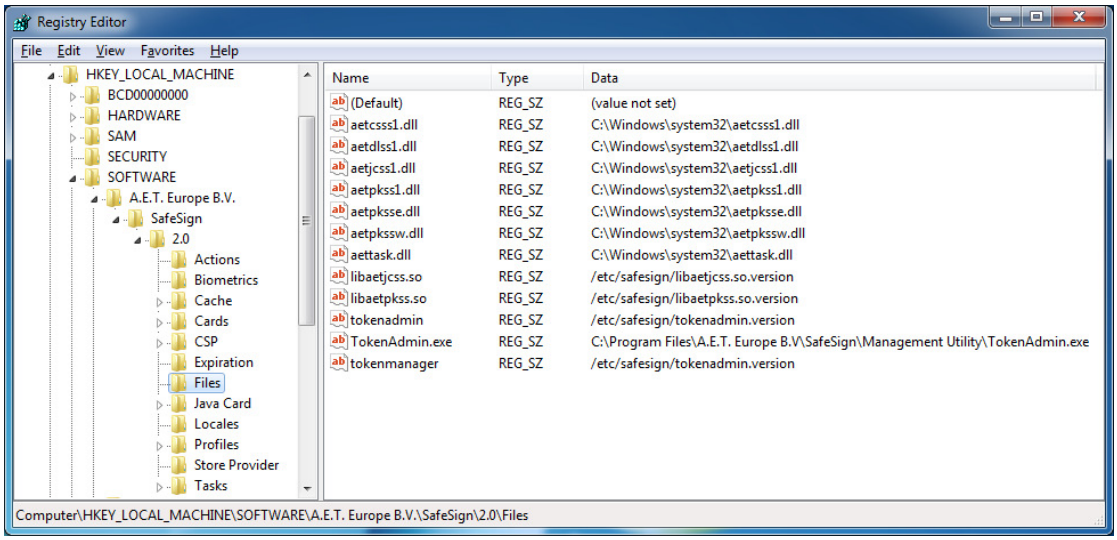

Figure 17: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Files\

If you install SafeSign Identity Client in the default location indicated during the installation of SafeSign Identity Client or in any other folder (that you created), the registry will reflect this. However, if you change the locations of some of the files after installation of SafeSign Identity Client, you need to reflect those changes in the corresponding registry settings.

This is also the place where the *Versions Info* item in the **Help** menu retrieves its information from.

3.9 GINA

When the SafeSign GINA is installed, the registry key used by the GINA is:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0**GINA**]

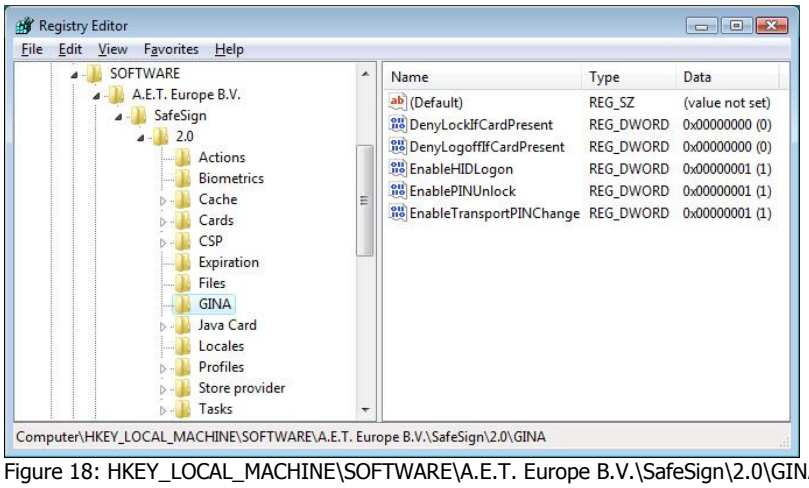

Figure 18: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\GINA\

Please note that this key is **not present** in a default SafeSign Identity Client installation!

Note on Windows Vista and higher

In Windows Vista and higher, the Microsoft GINA (msgina.dll) has been removed, and custom GINAs will not be loaded on systems running Windows Vista and later versions (including Windows 7 and Windows Server 2008). Consequently, the SafeSign GINA will not be loaded on Windows Vista and higher.

Although even on Windows XP, use of the SafeSign GINA is not required for a correct functioning of the secure pinpad readers that SafeSign Identity Client supports, the absence of a GINA does mean that it is not possible to unlock the PIN of the token or change the Transport PIN during logon, functionality that was implemented in the SafeSign GINA.

For this purpose, on Windows Vista and higher, a custom Credential Provider is required, which is available in SafeSign Identity Client version 3.0.40 and higher.

The SafeSign GINA can be configured to include the following functionality:

- *DenyLockIfCardPresent* / *DenyLogoffIfCardPresent*: these values, which are disabled by default, can be set to present the user with a visual and auditory warning when he tries to lock the PC or log off with the token still present;
- *EnableHIDLogon*: this value, which is enabled by default, will allow the user to log on with a StarKey 220 HID token; when it is disabled, it is not possible to log on with this token;
- *EnablePINUnlock*: this value, which is enabled by default, will allow the user to unlock (the PIN for) the token at Windows logon;
- *EnableTransportPINChange*: this value, which is enabled by default, will allow the user to change the Transport PIN for the token at Windows logon.

Note that after enabling / disabling the above features, you will first need to reboot your computer.

Note on the GINA in SafeSign Identity Client version 3.0.33

Note that on Windows XP, with the SafeSign GINA installed as part of SafeSign Identity Client version 3.0.33, the EnablePINUnlock and EnableTransportPINChange do not work (as opposed to previous versions of SafeSign Identity Client). This has been fixed in SafeSign Identity Client version 3.0.40 and higher.

3.9.1 Manually uninstalling the GINA

For administrative reasons, it may be necessary to manually uninstall the GINA from your Windows XP system. To do this, you will need to alter the following registry key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]

Under this key, there is a string value called *GinaDLL*.

If the SafeSign Identity Client GINA is installed, this value will contain the string "aetgina1.dll". To restore the default Microsoft GINA, you will need to change this value to "msgina.dll".

WARNING: modifying the GINA DLL on a running system without using the appropriate installer(s)/uninstaller(s) is extremely dangerous and may render your system inaccessible. It should only be done by experienced system administrators.

3.10 Locales

SafeSign Identity Client Version 3.0.45 for Windows supports the following languages:

Basque	Catalan
Chinese (Simplified)	Chinese (Traditional)
Croatian	Czech
Dutch	English
Finnish	French
German	Hungarian
Italian	Japanese
Korean	Portuguese: Portugal
Portuguese: Brazil	Russian
Serbian (Cyrillic and Latin)[1]	Spanish
Thai	Turkish

Multi-language support has been implemented such, to create utmost flexibility for both administrator and user. It may be imagined that an administrator, and not the user himself / herself, is installing SafeSign Identity Client on a user PC or on a central PC, for which he chooses a particular language. The user will then always be free to change the preferred language of SafeSign Identity Client.

The language of the InstallShield Wizard and the SafeSign Identity Client items in the Start menu, though this language can be selected upon installation of SafeSign Identity Client, is static and cannot be changed once selected (without de-installing or upgrading SafeSign Identity Client). This is due to limitations in Windows.

The language of SafeSign Identity Client and its utilities is dynamic and can be changed to any of the languages supported.

As of SafeSign Identity Client version 3.0.45, upgrading behaviour of SafeSign IC has been fixed, as there was en error upgrading, in another language than English. It is now also possible to upgrade SafeSign and select a different language for installation.

For this to work SafeSign Identity Client needs access to all the appropriate language files. The location of the language files is set in:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Locales]:

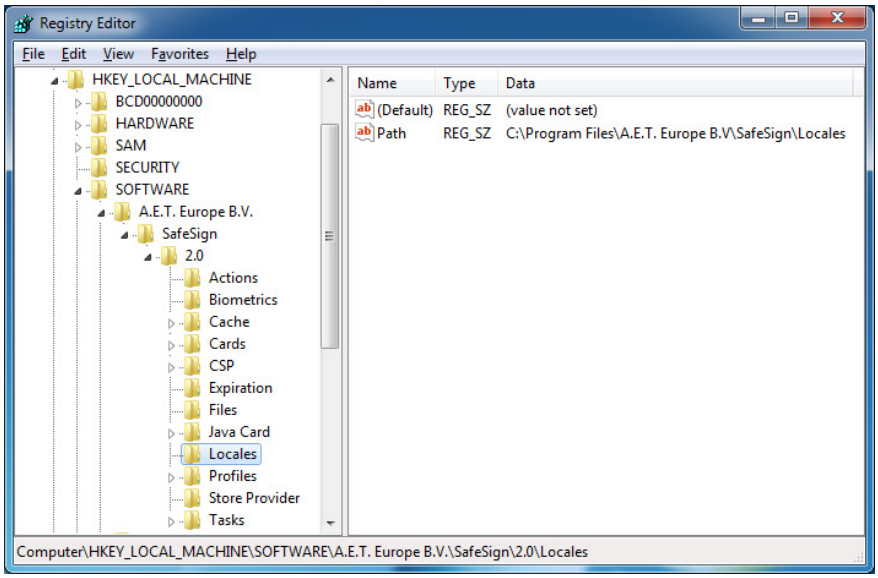

Figure 19: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Locales

[1] Note that although SafeSign IC supports both Serbian (Cyrillic) and Serbian (Latin), InstallShield (≤ 2010) does not support Serbian (Latin) and therefore, during installation, it is only possible to select Serbian (Cyrillic) as the language of the installation wizard.

3.11 Profiles

All the different kinds of profiles that are by default delivered with SafeSign Identity Client are stored in the entry:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\]

These profiles can be seen and used when you initialise a token with the use of the Token Management Utility / Token Administration Utility (in the *Initialise Token* dialog).

For Java Card v2.1.1 / Open Platform 2.0.1 compliant Java smart cards, there are in principle, three different profiles: Minimal, Medium (Default) and Maximal profile. You may edit these profiles at your own risk with custom (PKCS#15) values.

For Java Card v2.2+ / Global Platform 2.1.1 compliant Java smart cards, there is only one profile (Default) available, that you should **not** edit (as it has already been optimised)[1]. You may only edit such values in this profile as the Transport PIN and the maximum / minimum PUK and PIN length.

If for some reason you do not wish to use (any of) the profiles that SafeSign Identity Client provides[2], we strongly recommend creating your own token profile instead of changing the profiles that are by default delivered with SafeSign Identity Client and naming it accordingly (with a name that identifies it as a non-SafeSign Identity Client standard profile).

To create your own profile, follow the steps described in paragraph 3.11.1.

For a number of examples of custom values for your profile, refer to paragraph 3.11.2.

During initialisation of a token, the PKCS #11 library uses the *Active profile* entry in the registry to look up which token profile should be used for the initialisation of the token.

The registry string entry

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\Active profile]:

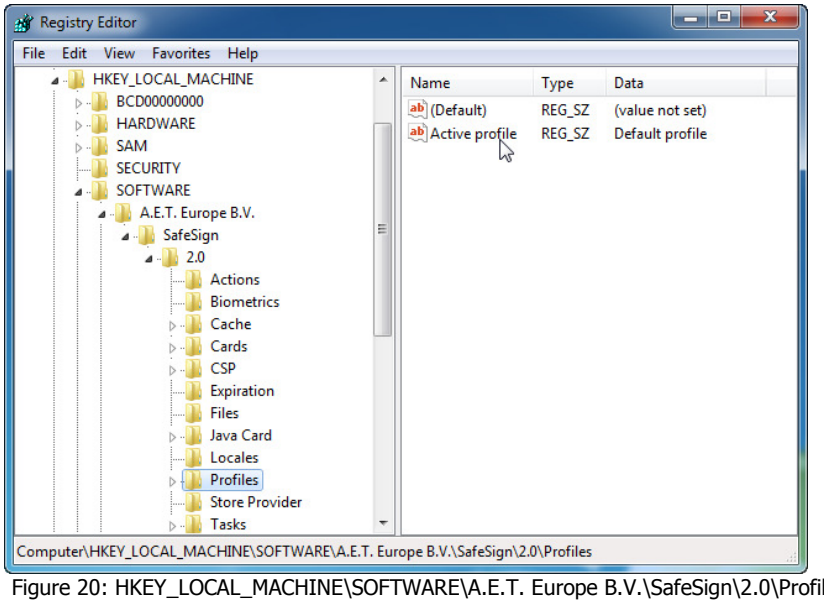

Figure 20: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\Active profile

should have the value of the name of the profile you wish to use during initialisation of a token, because a (domain) user will only be able to initialise the token according to the active profile set in the registry (the *Token Profile* drop-down list in the Token Management Utility will be greyed out, see Figure 28); i.e. he cannot choose which profile to use for initialisation.

[1] As already explained and mentioned in paragraph 2.1.
[2] For example, if you want to manipulate the amount of public and private space.

3.11.1 Create a new profile

Creating a new profile entails the following actions:

- Create a new profile key (paragraph 3.11.1.1)
- Determine to which token model(s) the profile should apply (paragraph 3.11.1.2)
- Set the minimum values for the profile (paragraph 3.11.1.3)

3.11.1.1 Create a new profile key

We recommend creating your own token profile by adding a new profile key, instead of changing the profiles that are by default delivered with SafeSign Identity Client, in order to avoid confusion.

In order to create your own profile, you first have to make an additional entry in the registry below:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\]

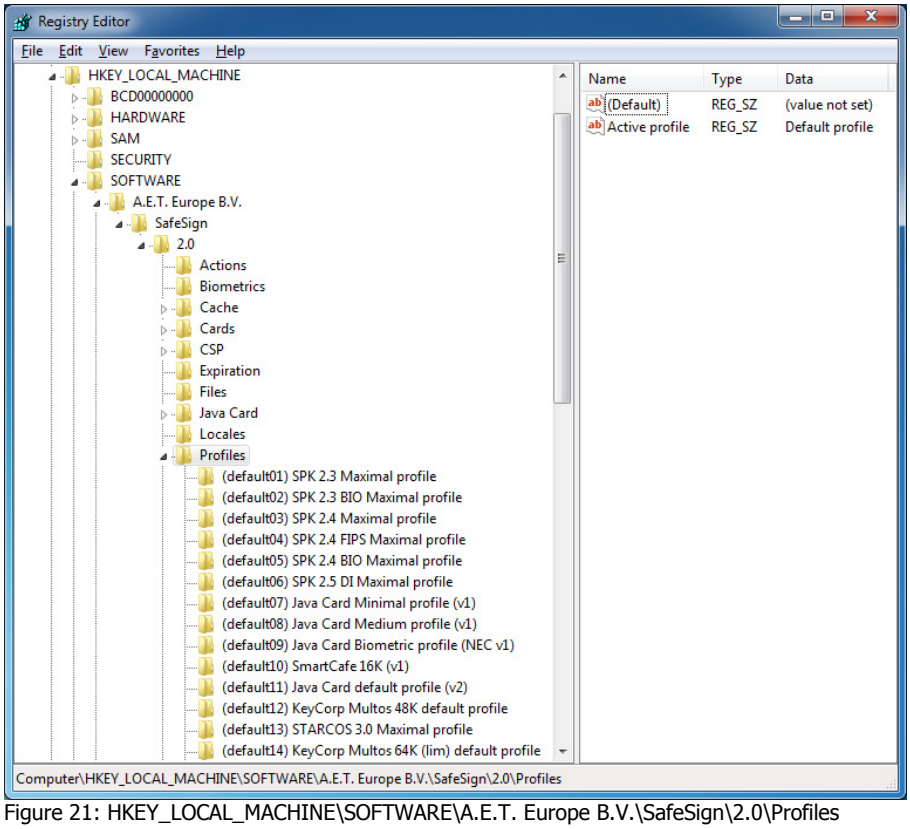

Figure 21: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles

Under the key

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\],

right-click the Profile key and select **New -> Key** (as below):

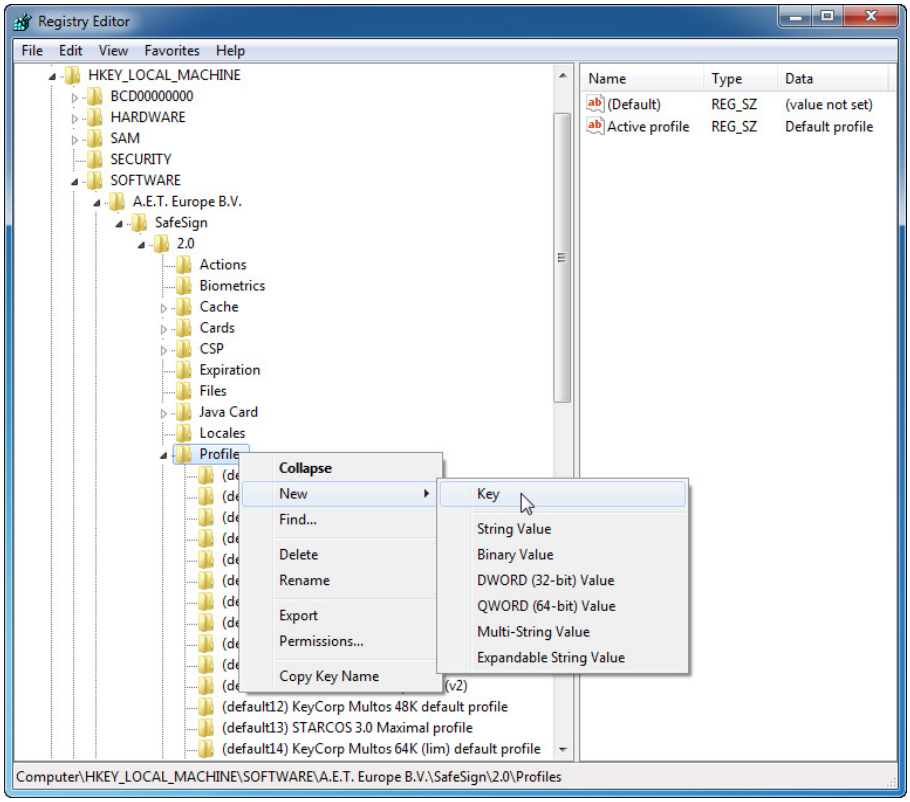

Figure 22: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles: New Key

After you have created a new key, you should name it, preferably with a name that identifies it as a non-SafeSign Identity Client standard profile:

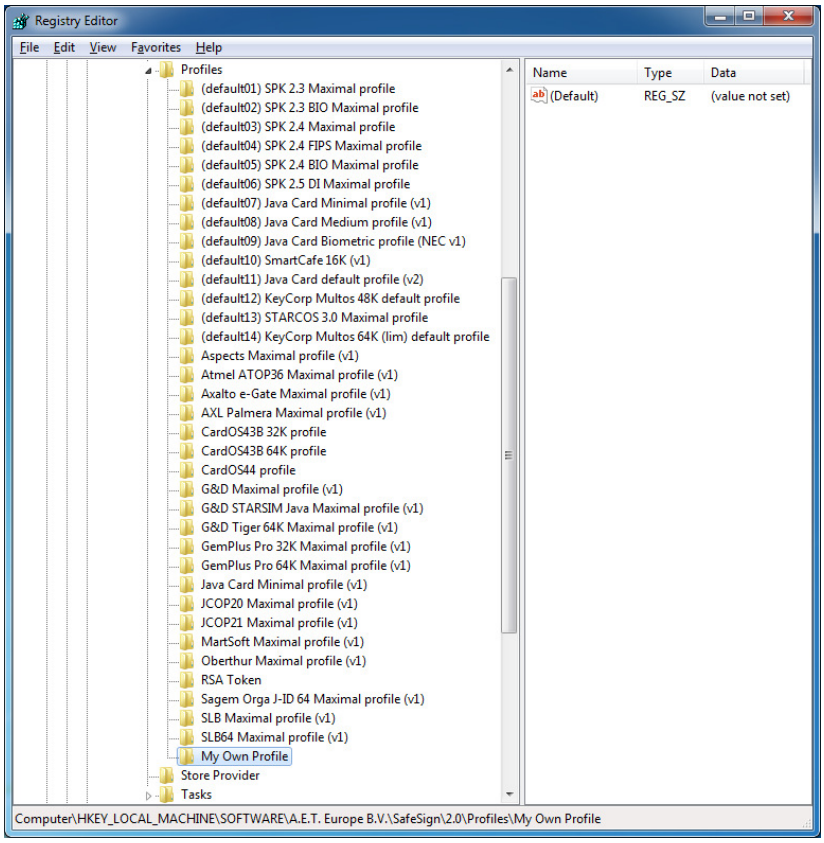

Figure 23: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\My Own Profile

3.11.1.2 Determine ModelID(s)

After you have created a new profile (paragraph 3.11.1.1), you should determine for which tokens you would like you profile to be valid. To do so, you should retrieve the token model ID.

Every token has a so-called 'ModelID' within SafeSign Identity Client. Below the entry

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cards]

you can find all the different tokens that are supported by SafeSign Identity Client. The separate entries for the tokens also mention the corresponding ModelID.

As an example, we will use the (IBM) JCOP 20 smart card, which has as its ModelID 'JC10':

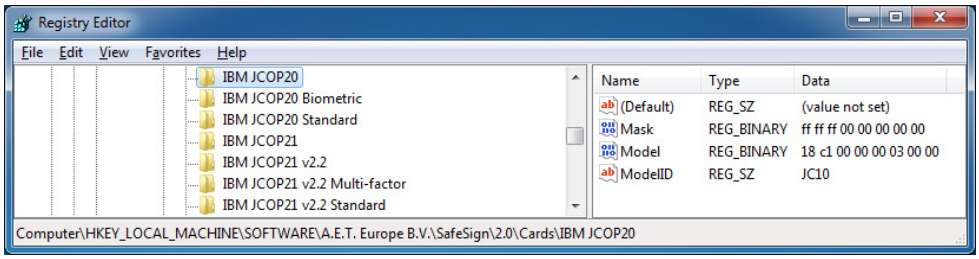

Figure 24: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cards\JCOP20 Standard

Note that a profile may be valid for more than one token. Carefully determine for which tokens your profile will be valid, by recording their ModelIDs.

3.11.1.3 Set minimum values for the profile

The profile you have created (paragraph 3.11.1.1), should contain at least two String Values: *ModelID* and *Name*, which both should have a valid data value.

To create these, right-click the profile key you have created ('My Own Profile') and select **New -> String Value** (as below):

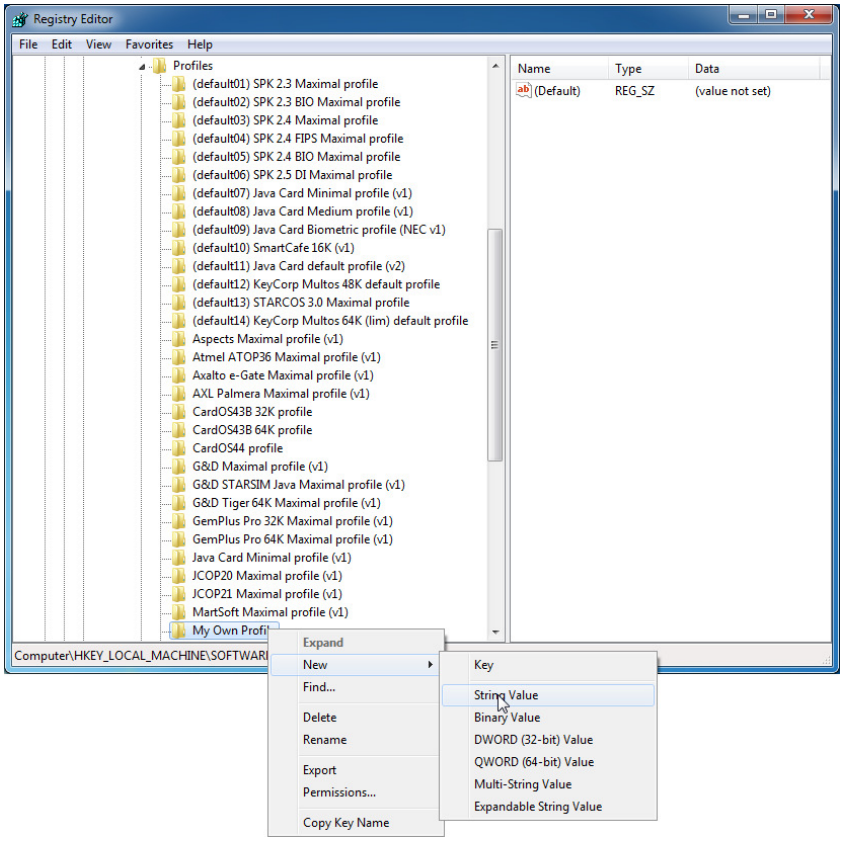

Figure 25: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\My Own Profile: New String Value

The new string value should be called *Name* and its *Value data* should be a name that easily identifies the profile ('My profile' in our example below):

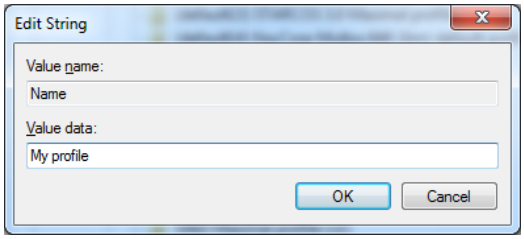

Figure 26: Edit String Value: Name Value data

This value will be shown in the Token Management Utility / Token Administration Utility *Initialise Token* dialog as (one of) the profile(s) to be selected (see Figure 28).

➔ Enter an identifying name in the *Value data* box and click **OK**

Now create a new string value called *ModelID*. The Value data contained in the String Value *ModelID* should list the exact ModelIDs (the ones you recorded earlier) that you want your profile to be valid for. Thus, if your profile is valid for multiple tokens, you should enter every ModelID for which your profile is valid in the *Value data* box, separated by a comma:

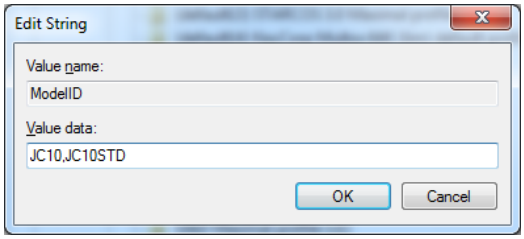

Figure 27: Edit String Value: ModelID

If during an initialisation of a token an error occurred with the profile you have created, this could mean that your profile is not valid for that token. In this case, please verify that you have entered all the correct values in the profile.

After creating the entries in the registry, you can verify that the profile works for tokens your have created it for, by using the Token Management Utility / Token Administration Utility.

In the Token Management Utility / Token Administration Utility, insert the token and select the *Initialise Token* item from the **Token** menu.

If you are able to select your profile in the *Token Profile* drop-down list in the *Initialise Token* dialog, you have made a valid match between the ModelID and the token(s) you wish to make your own profile for:

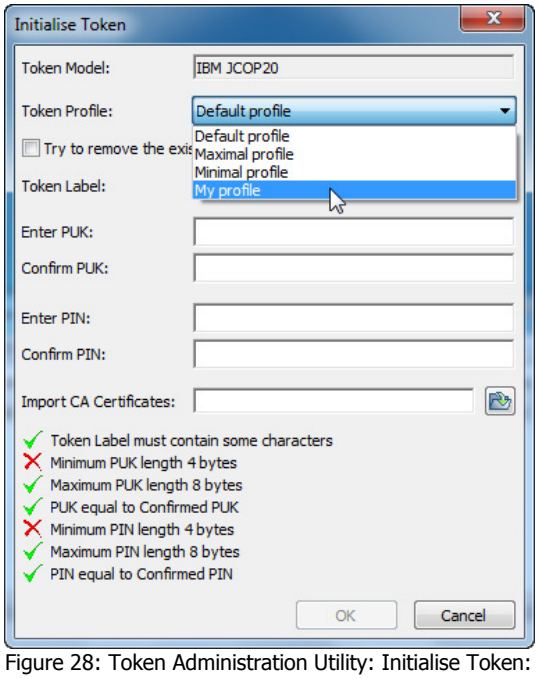

Figure 28: Token Administration Utility: Initialise Token: My profile

Apart from these minimum values for your profile, you can also set other customised values. Some examples of values that you may include in your profile are described in the next paragraph(s).

3.11.2 Values: Examples

There are a number of different values that can be set during the initialisation. Below you will find some examples of values that you may include in your own profile.

Note that when creating your own profile, for every value that you do not list in your profile, a default value is assumed.

3.11.2.1 Example 1: How to set a Transport PIN

A Transport PIN is a temporary PIN on the token that has to be changed into a personalised PIN code before a token can be used. Setting a Transport PIN can be useful for security reasons, for example when you want to be certain that a user has (consciously) set his / her own PIN prior to any signature token operations. A Transport PIN can be set during the initialisation of the token.

For the STARCOS range of cards and Java Card v2.1.1 cards supported, a Transport PIN is a PIN that contains fewer characters than a valid PIN code. This means that when a valid PIN is set to be at least 5 characters, the Transport PIN should contain no more than 4 characters.

For the Java Card v2.2+ / Global Platform 2.1.1 compliant Java smart cards, the Transport PIN should be equal to or larger than the personal PIN. It should not contain fewer characters. This means that when a valid PIN is set to be at least 4 characters, the Transport PIN should contain 4 characters or more (not less).

Note

Note that rather than setting the Transport PIN in the registry of a local machine, as described below, it is much more common to set the Transport PIN programmatically (as an expired PIN), in accordance with the PKCS #11 standard, which defines: "If a PIN is set to the default value, or has expired, the appropriate CKF_USER_PIN_TO_BE_CHANGED or CKF_SO_PIN_TO_BE_CHANGED flag is set to TRUE. When either of these flags are TRUE, logging in with the corresponding PIN will succeed, but only the C_SetPIN function can be called. Calling any other function that required the user to be logged in will cause CKR_PIN_EXPIRED to be returned until C_SetPIN is called successfully."

In the example below we will use the custom profile ('My Own Profile') that we have created above, for a Java Card v2.1.1 card, the IBM JCOP20.

In order to set a Transport PIN (in our example, '1234'), you should create:

A String Value with the name *TransportPin* and with Value data '1234';

A DWORD Value with the name *MinPinLen* (Minimum PIN Length) with the value of '5' (which stands for a minimum PIN length of 5 characters);

A DWORD Value with the name *MinPukLen* (Minimum PUK Length) with the value of '5' (which stands for a minimum PUK length of 5 characters):

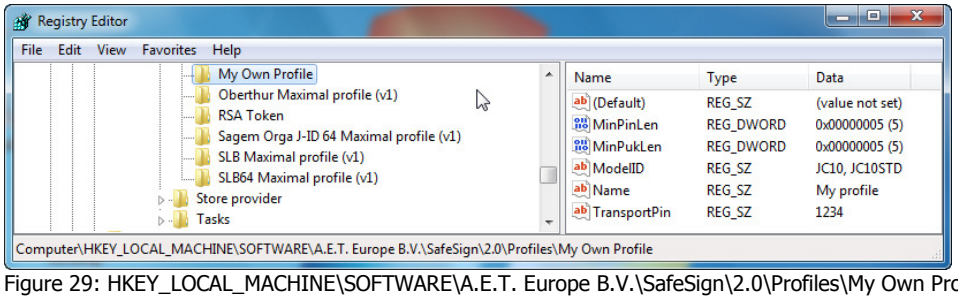

Figure 29: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\My Own Profile: Transport PIN

Note on the values for minimum PUK and PIN length

Note that as it is not possible to derive from the (PKCS#15) token information field the separate values for the PIN and the PUK, the minimum values for PIN and PUK should be kept the same.

This means that when you change the minimum values for PIN and PUK length, you should set both values and keep both values the same, i.e. you should not set a minimum PIN length of 6 characters and a minimum PUK length of 8 characters. Doing so may cause the token to be locked.

In the Token Management Utility / Token Administration Utility, upon initialising a token, you should enter a PUK, but you cannot set a PIN, for the (temporary) PIN for the token will be the Transport PIN:

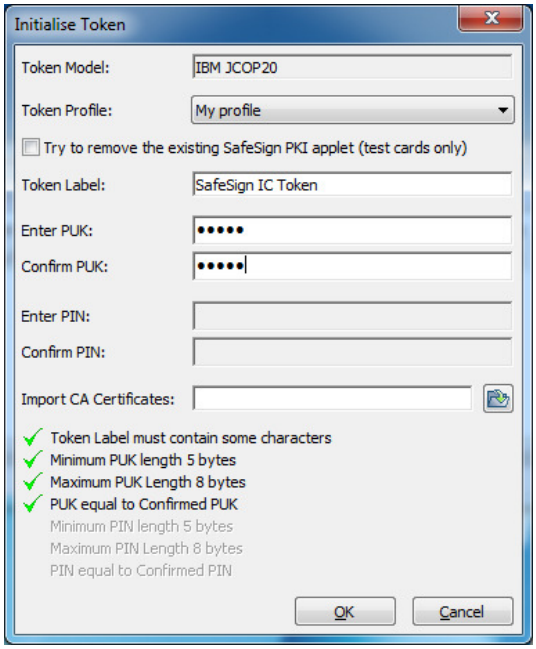

Figure 30: Token Administration Utility: Initialise Token with Transport PIN set

➔ Click **OK** to initialize the token

After a successful initialisation, the user (assuming that you, the Administrator, have initialised the token with the Transport PIN first) can use the Token Administration Utility to change the Transport PIN into a personalised PIN:

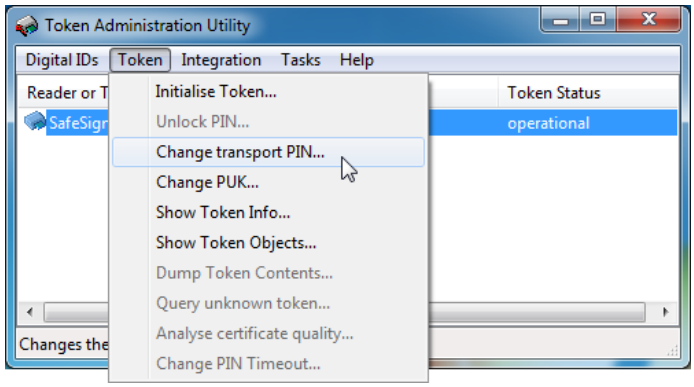

Figure 31: Token Administration Utility: Change Transport PIN

3.11.2.2 Example 2: Setting the value of public space

During initialisation of a token, the amount of public and private space is set on the token. It is possible to use different values for the amount of public and / or private space than the values SafeSign Identity Client uses by default (for the profile selected).

To customise the amount of public and / or private space for your own profile during initialisation, we recommend that you set the amount of public and / or private space (bytes) in the registry below your own profile key (created in paragraph 3.11.1). In this example we will set the value of public space, using a value of '100'.

Setting the amount of public space can be done by creating a DWORD Value with the name 'PublicSpace' under your profile key. The Value data for 'PublicSpace' should be the amount of public space you want to be created on the token during initialisation:

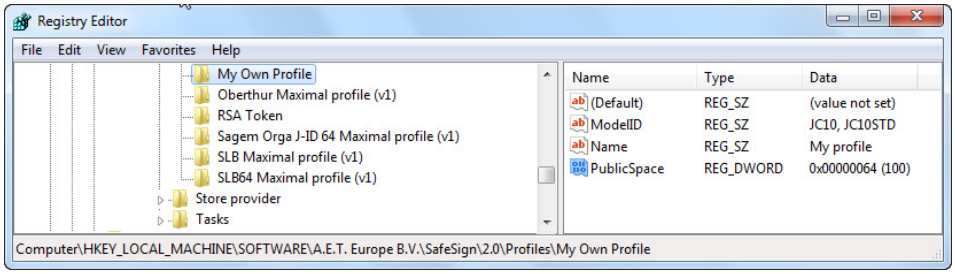

Figure 32: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\My Own Profile: Public space

When you do not set other values, for example the amount of private space, default values will be assumed.

Note on public / private space

Note that this means that when you set the amount of public space to a certain value, the amount of private space will not be increased / decreased proportionally, but will retain its default value. This would imply that you might not use the maximum amount of space available on the token. In this case, you could consider adding the difference between the default amount of public space and the amount of public space you created, to the private space.

Note that the private space is NOT used to store private keys. Private space is used for legacy applications like Entrust v6 who need to store data objects on the card that are PIN protected. You can decrease the private space in favour of increasing public space, for example to store data objects.

Note that if you want to increase private space, for example, to store data objects, you should take into account the DODF (data object directory file) in the PKCS #15 structure. You also need to allocate some extra space to the DODF in the profile.

After a successful initialisation, the token will have a public memory of 100 bytes.

This can verified in the Token Management Utility / Token Administration Utility, under **Token > Show Token Info**:

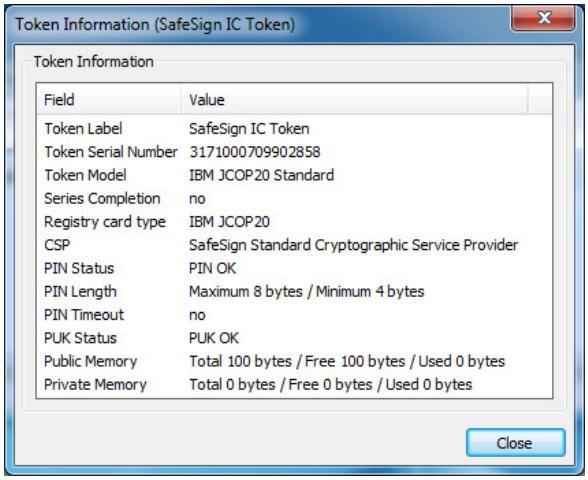

Figure 33: Token Administration Utility: Show Token Info

3.11.2.3 Example 3: Setting PIN / PUK length

It is possible to set customised values for minimum and maximum PIN and PUK length to be set on the token.

Note that the minimum and maximal PIN / PUK length is determined by the card:

- For the STARCOS range of cards and the Java Card v2.1.1 cards supported, SafeSign Identity Client Version 3.0 supports a minimum PIN/PUK length of 4 bytes[1] and a maximum PIN/PUK length of 8 bytes;
- For the Java Card v2.2+ cards supported, SafeSign Identity Client version 3.0 supports a minimum PIN/PUK length of 4 bytes and a maximum PIN/PUK length of 15 bytes.

This example will describe how to create a profile with a minimum PIN and PUK length of 6 characters.

Note on the values for minimum / maximum PUK and PIN length

Note that as it is not possible to derive from the (PKCS#15) token information field the separate values for the PIN and the PUK, the minimum values for PIN and PUK should be kept the same.

This means that when you change the minimum values for PIN and PUK length, you should set both values and keep both values the same, i.e. you should not set a minimum PIN length of 6 characters and a minimum PUK length of 8 characters. Doing so may cause the token to be locked.

[1] Minimum and maximum PIN/PUK length is in bytes (not characters), as according to the PKCS #11 specification, PIN is stored as UTF-8 characters (CK_UTF8CHAR). PINs that contain characters with diacrits (such as umlaut, accent grave, etc.) and such characters as ß and €, are converted into their relative UTF-8 encoding, which is at least 2 bytes long.

To set minimum PIN and PUK length, you should create the DWORD Values MinPinLen and MinPukLen in the profile you created (in paragraph 3.11.1). The Value data of the DWORD Values should contain minimum PIN and minimum PUK length respectively:

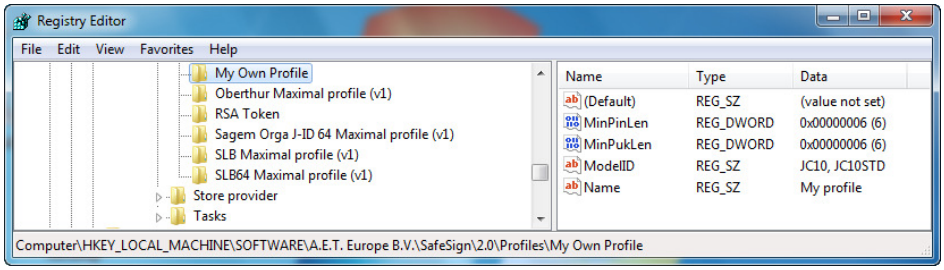

Figure 34: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Profiles\My Own Profile: Minimum PIN/PUK

In order to initialise the token, you should enter a PIN and PUK with a minimum length of 6 characters:

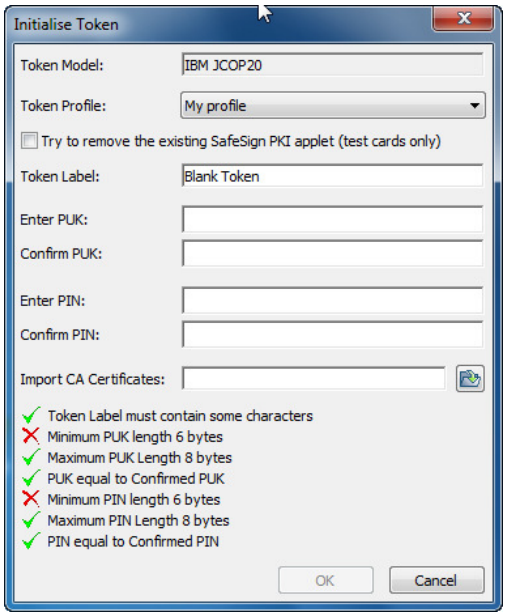

Figure 35: Token Administration Utility: Initialise Token with minimum PIN and PUK length set

3.11.3 Profile Values: Description

Below is a short description of the entries that may be edited. For a detailed description of all the attributes please see the PKCS #15 specification.

MaxPinRetries: Is the number of incorrect PINs that the token can receive before it blocks. The minimum number of PIN retries is 1[1], the maximum is 15.

MaxPukRetries: Is the number of incorrect PUK that the token can receive before it blocks. The minimum number of PUK retries is 1, the maximum is 15.

MinPinLen: Is the minimum length a PIN code must have. Values can be between 1 and 8/15.

MinPukLen: Is the minimum length a PUK code must have. Values can be between 4 and 8/15.

PrivateSpace: Is the amount of available private space after initialisation.[2]

PublicSpace: Is the amount of available public space after initialization.[3]

TransportPin: When the value TransportPin is set, a token will have this value set as transport PIN after initialization.

Note

Note that the entries SafeBootPublicSpace and SafeBootPrivateSpace were included to ensure interoperability with SafeBoot.

Note that the entry IgelClientPublicSpace was included to ensure interoperability with Igel thin clients.

3.12 Store provider

3.12.1 Background and history

In early SafeSign Identity Client versions, up to and including version 2.1.3 (≤ version 2.1.3), certificates were registered for Microsoft applications by means of the SafeSign Identity Client Certificate Registration Utility (aetcrss1.exe). This separate utility took care of registering / copying certificates into the Microsoft My Store.

Though fast and reliable in its operation, it could occur that by the time an application wanted to use the certificate(s) on token (either during logon or at a later stage), the certificate was not registered yet by the Certificate Registration Utility (because it was not started yet at the time the application looked for the certificate). Applications that were reported to show this behaviour are Cisco VPN and Outlook Web Access. By the time these applications check for the availability of the certificate in the My Store, the certificates were not yet registered in this store by the Certificate Registration Utility. This could cause an application not to be able to find a certificate and fail (for example, in signing an e-mail message).

This is the main reason why the SafeSign Identity Client CryptoAPI Store Provider was implemented in SafeSign Identity Client version 2.3.2 and onwards (≥ version 2.3.2), replacing the Certificate Registration Utility (part of which remains to do certificate expiration checking, now called Certificate Expiration Check Utility), to ensure that certificates on the token are immediately available when the token is inserted. This means that applications can make use of the certificates on the token immediately and do not have to wait until the Certificate Registration Utility has finished registering the certificates. When the token is removed, the certificates are no longer available. This process is much more accurate: it is no longer possible to have a situation where certificates are not available one moment and are available the next. Furthermore, all this has been implemented such as to guarantee that users will not experience a decrease in performance and speed.

[1] Setting the number of PIN/PUK retries to 0, would make initialization impossible.
[2] Should be set only for Java Card v2.1 cards.
[3] Should be set only for Java Card v2.1 cards.

As of SafeSign Identity Client version 3.0.45 (≥ version 3.0.45), the SafeSign Store Provider has been removed and certificates are now no longer registered by SafeSign Identity Client, but by the appropriate Windows Services. This has been done in order to prevent compatibility problems when registering certificates and to be compliant with Microsoft policy (ensuring that certificates are available in the appropriate place at all times). Note that this service does not deregister certificates once they are propagated to the appropriate store (as can be seen also when using the Microsoft minidriver solution).

Note

Note that when the token is removed, the Microsoft Internet Explorer Personal certificate store will still say that the certificate has a private key. This is caused by the Microsoft Certificate Propagation wizard not verifying or updating the association of the private key with the certificate at a later stage, when the token is not present.

For more information, please refer to:

http://technet.microsoft.com/en-us/library/ff404288.aspx

http://technet.microsoft.com/en-us/library/ff404304(WS.10).aspx

3.12.2 CryptoAPI Store Provider

The Store Provider settings can be found in the registry key

HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Store provider:

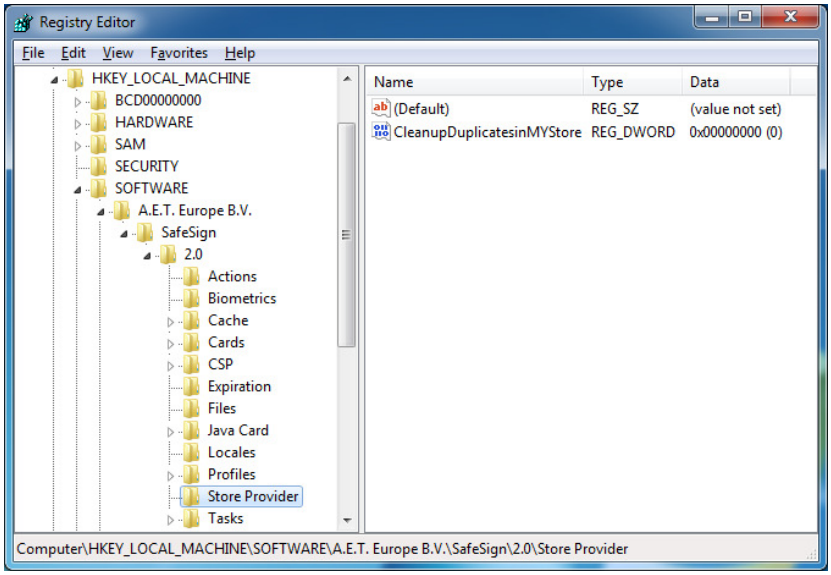

Figure 36: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Store provider

Before SafeSign IC version 3.0.45, all applications that enumerated the My Store would be added below this key. The purpose of this was to allow the Store provider to (temporarily) lock the My Store, until the application had finished enumerating the certificates, so that other applications could not interfere with this process.

As of SafeSign IC version 3.0.45, the Store Provider is no longer included, but the registration of certificates is left to Microsoft Certificate Propagation. The only value in this entry is CleanupDuplicatesInMYStore, which is deprecated (as it belonged to the functionality of the SafeSign Store Provider) and will be removed in a future release.

3.13 Task Manager

Below the entry

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Tasks]

You can modify the settings of the Task Manager (in SafeSign Identity Client Version 2.3.0 or higher), which is included in the Token Administration Utility (only):

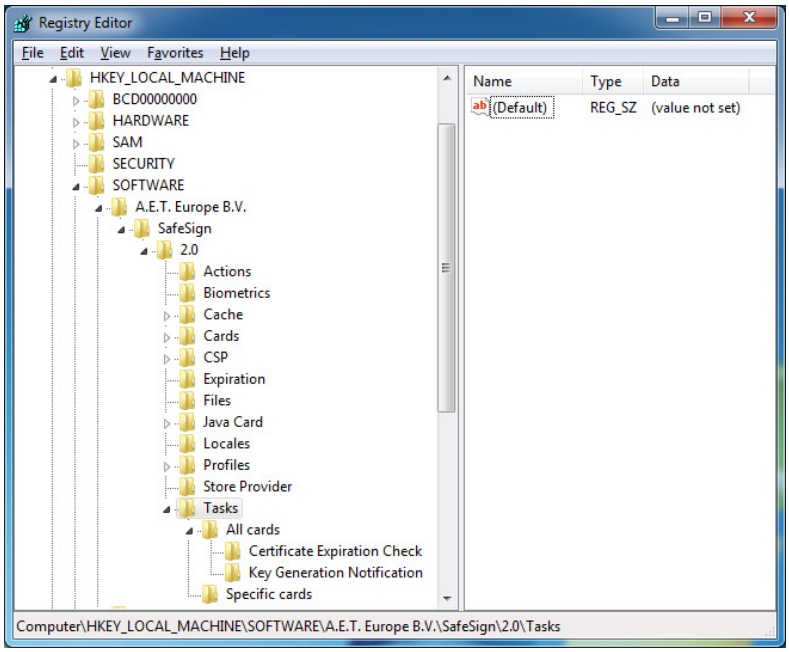

Figure 37: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Tasks\

It is possible to create two types of tasks in the Task Manager:

1. Launch an application when a token is inserted: for example, open Internet Explorer ((on a particular (secure) web site)) or set up a Remote Desktop Connection / Citrix connection;
2. Launch a plug-in when a token is inserted: for example, to change the Transport PIN of the token.

These tasks can apply either to all tokens or to a specific token (identified on the basis of token label and serial number). This is reflected in the Tasks registry key. Below the entry 'All cards' you will find those tasks that apply to all tokens; below the entry 'Specific cards' you will find those tasks that apply to one (or more) specific token(s).

For example, when you have defined as a task to have Internet Explorer to open, when a(ny) token is inserted (and named the task "Internet Explorer"), you will find this task under the entry 'All cards' as follows:

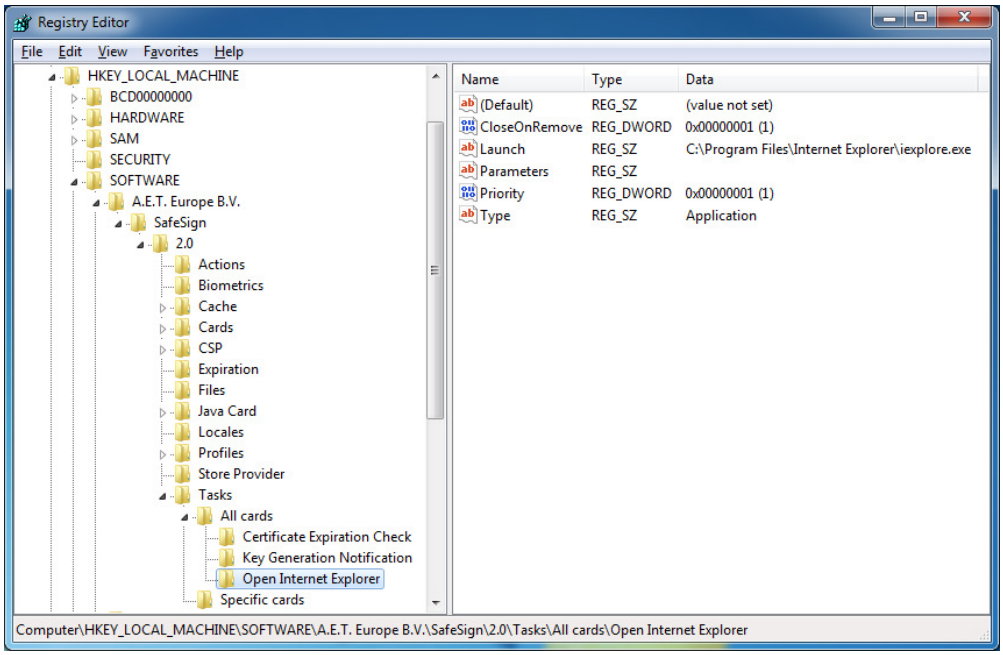

Figure 38: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Tasks\All cards\Internet Explorer

When this Task has been defined (during creation of the task in Task Manager) to close when the token is removed, the value for *CloseOnRemove* will be set to 1 (as above).

Note

Note that when selecting the option to close the application when the token is removed, the Task Manger will try to close the application launched, when possible. However, there are some scenarios in which this is not possible, for example when launching the remote desktop application (mstsc.exe) with parameters to connect to a particular session. In that case, the SafeSign Task Manager cannot close the session for the user or the application itself.

Note

It is possible to define a parameter to open the task. These parameters are application-specific. For example, in order to open a specific web page in Internet Explorer, you should enter: open http:// [name of web page] when creating the Task with the Task Wizard.

3.13.1 Predefined tasks

The tasks predefined for 'All cards' are the 1) Certificate Expiration Check, which appears when certificates are expired, and the 2) Key Generation Notification, which appears when a key pair is generated.

When both tasks are deselected, the process ' aetcrss1.exe' will automatically be ended, so that it does not interfere with other processes.

If you want to disable one of the tasks, for example certificate registration checking, you can remove the task (*Remove task*) from the Task menu of the Token Administration Utility, but we recommend to deselect the Task in the *Manage tasks* dialog:

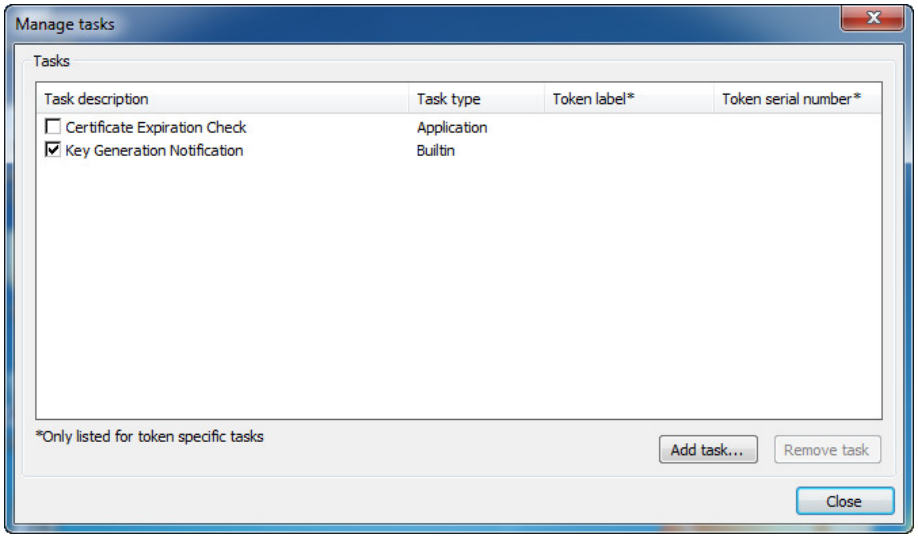

Figure 39: Token Administration Utility: Manage tasks

3.13.1.1 Certificate Expiration Check

From SafeSign Identity Client version 2.3, release 2.3.2 onwards, the Certificate Expiration Check Utility is part of the Task Manager, i.e. it is pre-defined as a task:

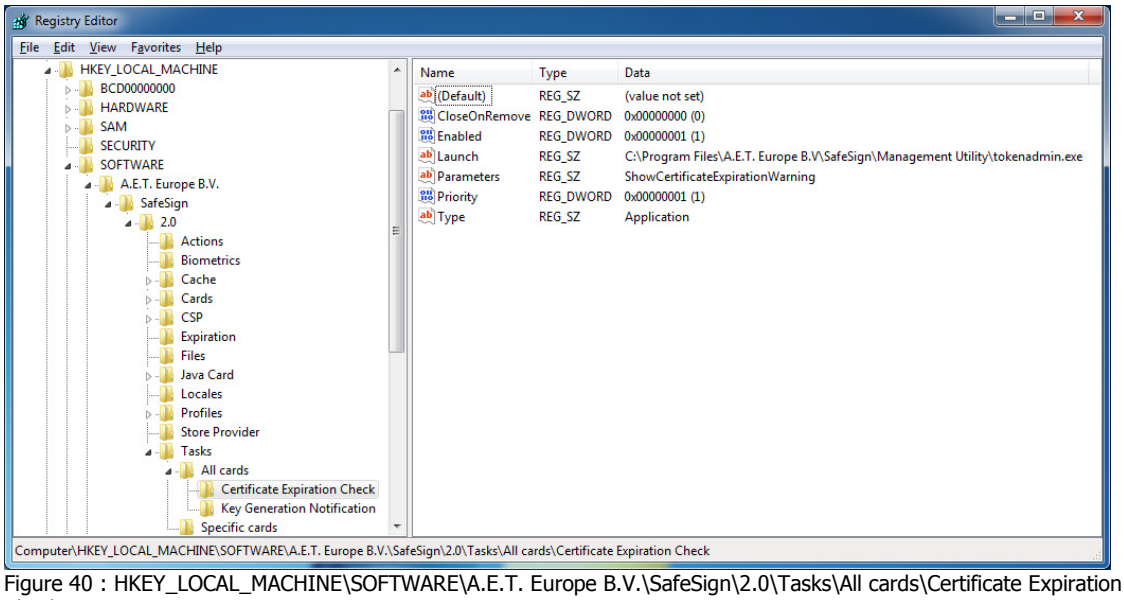

Figure 40 : HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Tasks\All cards\Certificate Expiration Check

Note that this task is not set to close upon token removal (the value for *CloseOnRemove* is set to 0). This should not be edited, to ensure proper certificate expiration checking.

3.13.1.2 Key generation Notification

As of SafeSign Identity Client version 3.0.40, the dialog that is displayed when keys are generated (for example, in Internet Explorer), is part of the Task Manager, i.e. it is pre-defined as a (built-in) task:

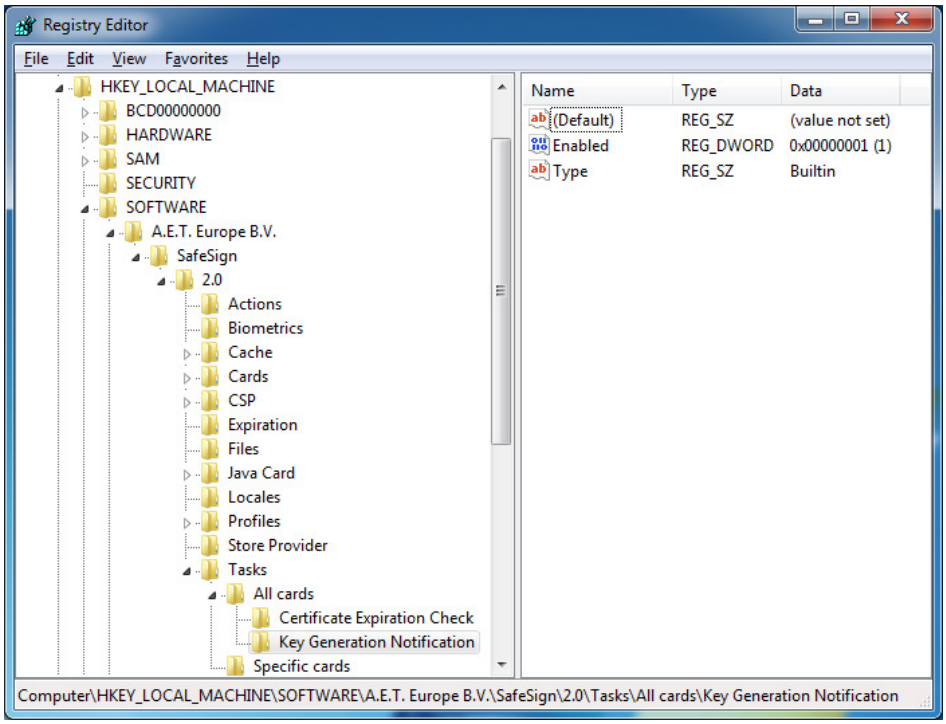

Figure 41: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Tasks\All cards\Key Generation Notification

4 CSP Integration

4.1 Supported tokens

SafeSign Identity Client supports a number of different tokens (see section 3.5). These can be found in the key:

[HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Cards]

To have Microsoft recognise the tokens (i.e. to associate a token's ATR with the Cryptographic Service Provider that supports it), for use with for example Smart Card logon, all tokens known by SafeSign Identity Client have to be known in the registry for Windows as well.

All the different tokens are listed in:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Calais\SmartCards]:

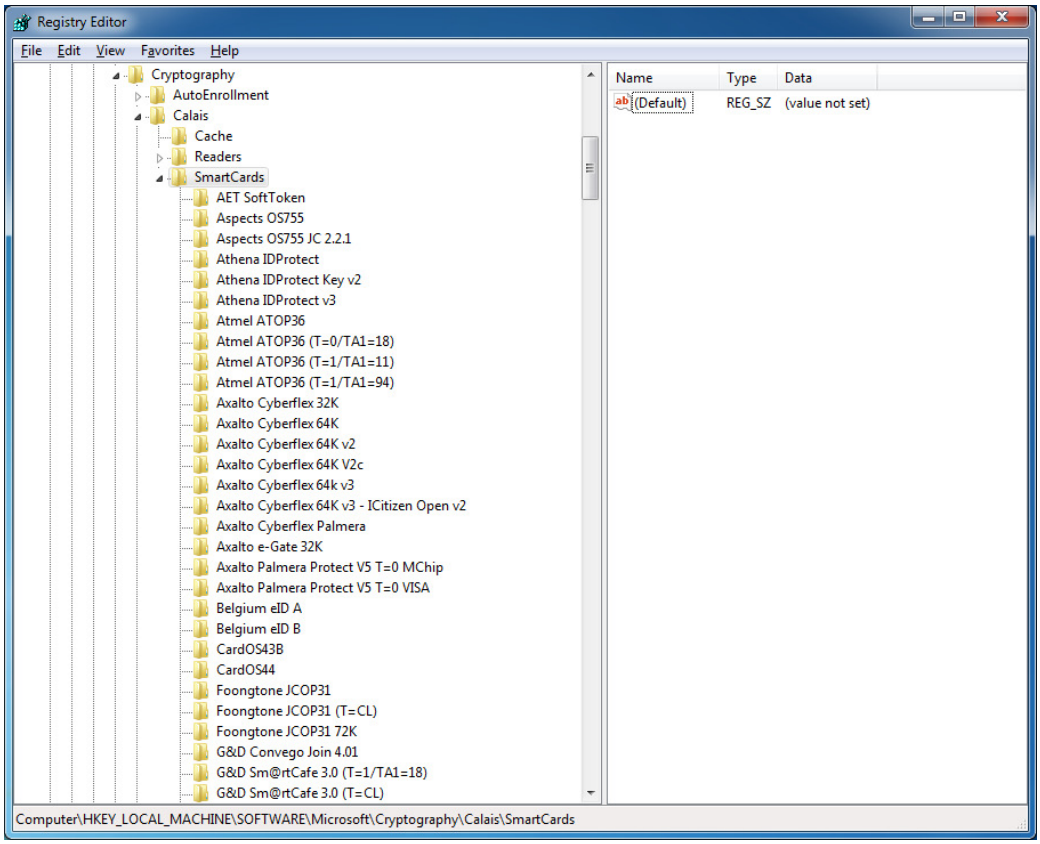

Figure 42: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Calais\SmartCards\

Note

For 64-bit Windows Operating Systems, the Smart Card entries can be found both in [HKEY_LOCAL_MACHINE|SOFTWARE|Microsoft|Cryptography|Calais|SmartCards] and [HKEY_LOCAL_MACHINE|Wow6432Node|SOFTWARE|Microsoft|Cryptography|Calais|SmartCards]

Without the presence of the ATRs in both registry keys on a 64-bit Windows machine, Windows will start looking for drivers for the Smart Card, trying to download and install the smart card minidrivers for the card through Plug and Play services.

4.2 Reference to the CSP

Apart from the fact that all the tokens have to be known in the registry to enable integration with Microsoft, it is also necessary for Microsoft to recognise the token, to know which DLL (Dynamic Link Library) it has to use to make the integration possible.

In order to do so, Microsoft uses a so-called CSP (Cryptographic Service Provider). SafeSign Identity Client provides such a CSP. This library makes it possible for Microsoft applications, or applications that use the SafeSign Identity Client CSP, to communicate with the token.

As of SafeSign Identity Client version 3.0.40, for Windows Vista and higher (including Windows 7 and Windows Server 2008), support for AES encryption / decryption has been implemented. SafeSign Identity Client now offers both a type 1 CSP (PROV_RSA_FULL) and a type 24 CSP (PROV_RSA_AES), supporting AES-128, AES-192 and AES-256. This is reflected in the registry as well.

SafeSign Identity Client registers the SafeSign Identity Client CSP in:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\]:

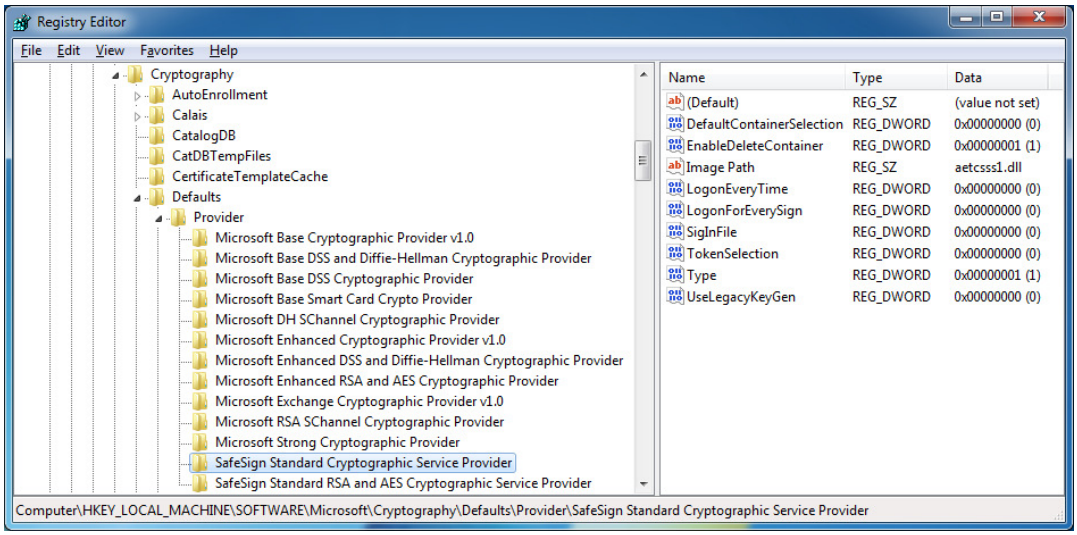

Figure 43: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Type 1

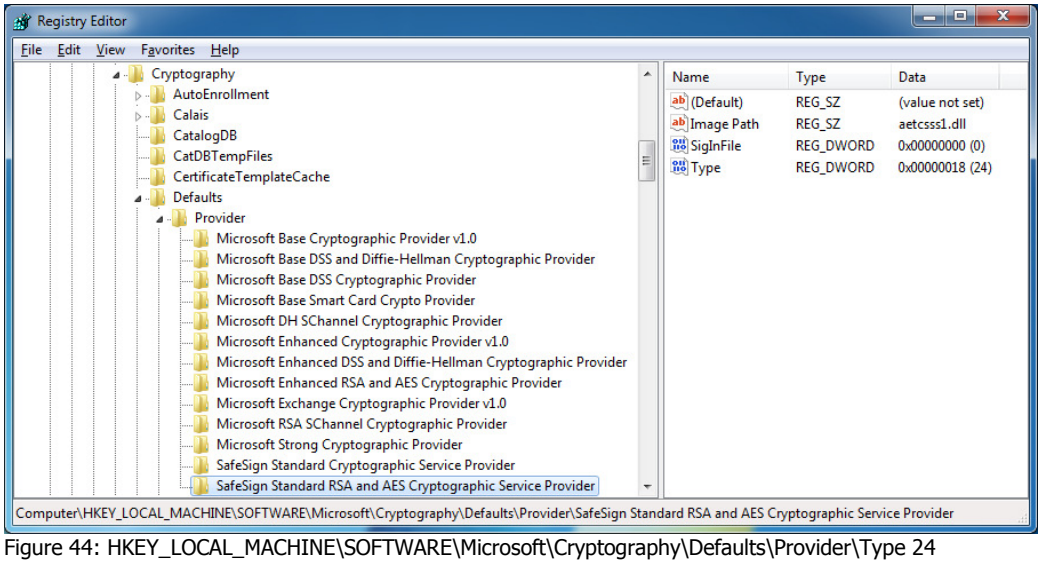

Figure 44: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Type 24

Note

For 64-bit Windows Operating Systems, reference to the SafeSign CSP can be found in: [HKEY_LOCAL_MACHINE|SOFTWARE|Microsoft|Cryptography|Defaults|Provider] and [HKEY_LOCAL_MACHINE|Wow6432Node|SOFTWARE|Microsoft|Cryptography|Defaults|Provider]

Below is a description of the relevant entries below this entry point (SafeSign CSP Type 1).

4.2.1 DefaultContainerSelection

4.2.1.1 Windows XP

Before Windows Vista, a smart card could support only one certificate for logon, because only one container on the smart card could be marked default.

Thus, in accordance with the Microsoft standard, when there are two certificates on the token, SafeSign Identity Client (from version 2.3.2 onwards) automatically detects which certificate is the smart card logon certificate (by searching the certificates for the smart card logon OID). If the smart card logon certificate is detected, this certificate automatically becomes the smart card logon certificate. If both certificates are smart card logon certificates, the certificate that is placed latest on the smart card becomes the smart card logon certificate.

However, it is possible in SafeSign Identity Client (version 2.3.2 and higher) to select the default key container. This means that when a token contains more than one Digital ID suitable for smart card logon, you can select which certificate to use for logon. This means that you can use one card with two different smart card logon certificates, one for logging on as administrator to the domain, one for logging on as 'normal' user to the domain. In practice, if the value *DefaultContainerSelection* is enabled (note that by default, it is not enabled), this means that a dialog will be displayed, allowing the user to select the Digital ID he wants to use for Windows logon. Microsoft VPN may also ask to select the Digital ID.

Thus, even before Vista, SafeSign Identity Client provided users with the ability for a smart card to contain more than one certificate for logon and to select which certificate to use for logon. Also, in combination with the GINA, it was possible to change the PIN and unblock the token at logon.

4.2.1.2 Windows Vista and higher

In Windows Vista and higher (including Windows 7 and Windows Server 2008), Winlogon supports multiple logon certificates and containers on the same smart card. The Microsoft GINA (msgina.dll) has been removed, and custom GINAs will not be loaded on systems running Windows Vista and later versions. Both the password credential provider (i.e. entering username and password to log on) and the smart card credential provider (i.e. entering a smart card and its appropriate PIN) are provided by default, and the ability to support custom authentication mechanisms will require the creation of a custom credential provider.

SafeSign IC versions prior to version 3.0.33, will not display multiple logon tiles, but only the default key container[1].

When using SafeSign Identity Client version 3.0.33 or higher, if a token contains more than one certificate, these will be displayed at logon, identified by the tiles that are displayed. Therefore, users will be able to use their SafeSign IC token with multiple certificates during logon.

As of SafeSign Identity Client version 3.0.40, SafeSign IC has implemented a custom Credential Provider. The SafeSign Credential provider is a smart card credential provider, interacting with the SafeSign IC components. In fact, when the SafeSign Credential Provider is installed, the Microsoft Credential Provider will be deregistered, to ensure that users can benefit from all the features of the SafeSign IC Credential provider. The value *DefaultContainerSelection* can still be enabled, but it will not work with the SafeSign Credential Provider, but only with the Microsoft Credential Provider, in which case it is not very useful, as the Microsoft Credential Provider will already display all certificates on the card in the first place.

[1] The only way on Vista to have two or more logon tiles to appear on Vista logon, is to use the option ForceReadingAllCertificates that is part of the new Windows 2008 Server Group Policy settings that can be used on a per-computer basis. When enabled, it allows you to force read of all certificates from the smart card, regardless of the supported feature that is set in the CSP. This policy is applicable whenever the smart card credential provider or the credential UI is called.

Note on the limitations of the Credential Provider

Note that the current SafeSign Credential Provider does not support multiple certificates on one token. When you have more than one certificate on a token, it is recommended not to install the SafeSign Credential Provider, but to use the Microsoft Credential Provider instead, in which case it will behave as described above: when a token contains more than one certificate, these will be displayed at logon, identified by the tiles that are displays.

Also, the SafeSign Credential Provider does not support PLAP / Single Sign-On[1]. This means that when setting up a (Microsoft) VPN connection, the SafeSign Credential Provider will not available. For this reason, Microsoft VPN does not work with SafeSign Identity Client version 3.0.40 and higher, when the SafeSign Credential Provider is installed.

4.2.2 EnableDeleteContainer

Some applications delete the default key container from the token, for example, during an enrol or recovery operation. For example, Entrust 7 deletes the old certificate profiles from the token, when doing a key enroll operation.

This setting is by default turned on. When the entry *EnableDeleteContainer* in the registry, which is located at (Figure 43):

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\SafeSign Standard Cryptographic Service Provider\EnableDeleteContainer],

is changed from 1 to 0, the key container will not be deleted.

4.2.3 LogonEveryTime

When access to the private credentials on the token is requested in Microsoft applications, for instance when signing an e-mail message in Microsoft Outlook, you will be presented with the SafeSign Identity Client Login dialog to enter the PIN for the token (when access to the token is requested).

Under some Windows Operating platforms (e.g. Windows 98) you will have to enter the PIN every time access to the token is required (as long as the application requesting access is open), whereas under other platforms (e.g. Windows XP), you do not have to enter your PIN every time, even when the application is closed and opened again in the meantime[2].

However, it is possible to force the PIN entry for those platforms where the PIN is not requested every time, by activating the *LogonEveryTime* registry key.

The feature LogOnEveryTime requires the PIN to be entered each time when accessing the private credentials on the token. It triggers on the fact that a logon to a token is required (not on what is on the token itself).

When the entry *LogonEveryTime* in the registry, which is located at (Figure 43):

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\SafeSign Standard Cryptographic Service Provider\LogonEveryTime],

is changed from 0 to 1, it forces the user to enter there PIN code every time an application wants to use the token.

Note that this may result in the user having to enter the PIN for his/her token multiple times, for example when accessing a secured site (https://), where the web page may contain multiple items for which authentication is required.

[1] Single Sign-On (SSO) API represents a set of methods used to obtain EAP method specific credentials for a network user or computer account in a secure fashion without having to raise multiple UI instances.

[2] Note that this has nothing to do with SafeSign Identity Client caching the PIN. SafeSign Identity Client never caches the PIN in any way (for obvious security reasons)! This behaviour (that you do not have to enter your PIN every time on some platforms) is due to the fact that whereas the CSP loads the PKCS #11 Library and tries to maintain a logged-in session with the PKCS #11 Library as long as possible, the CSP itself is loaded by the implementation of the CryptoAPI on the Windows system involved, which can also unload the CSP. If this loading / unloading of the CSP is such that the CSP is closed (unloaded) after each use, this would mean that the user will have to enter the PIN every time the functionality of the CSP is asked for. This is different on all Windows platforms and cannot be influenced by SafeSign Identity Client.

4.2.4 LogonForEverySign

It is possible to force the PIN entry for every signing operation, by activating the *LogonForEverySign* registry key.

Unlike LogOnEveryTime, which merely triggers on the fact a logon to the token is required, LogonForEverySign looks at the X.509 certificate content, to determine whether it has the signing / authentication attribute set.

When the entry *LogonForEverySign* in the registry, which is located at (Figure 43):

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\SafeSign Standard Cryptographic Service Provider\LogonForEverySign],

is changed from 0 to 1, it forces the user to enter there PIN code every time an application wants to use the token for a signing operation.

Note that this may result in the fact that the logon dialog does not (always) appear (although it may be expected), if it was not a signing operation being performed, but rather a decryption operation.

4.2.5 TokenSelection

According to the Microsoft CSP standard, private and public keys should be generated on the first token the system encounters when there are multiple smart card readers installed and attached to the system. By turning on the feature TokenSelection, SafeSign Identity Client enhances the standard capabilities of the standard CSP specification by offering the possibility to choose on which token a key is generated when there are multiple smart card readers installed and attached. By default this feature is turned off.

When the entry *TokenSelection* the registry, which is located at:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\SafeSign Standard Cryptographic Service Provider\TokenSelection],

is changed from 0 to 1, the user will be allowed to select the token to generate a key pair on:

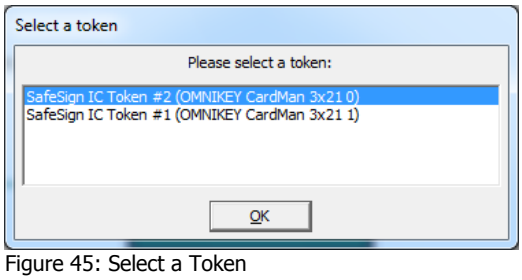

Figure 45: Select a Token

4.2.6 Friendly name

Note that from SafeSign Identity Client version 3.0.45 onwards, it is not possible to set or edit the friendly name in the way described below.

A certificate can have a 'friendly name', so that (end-)users can easily identify a certificate in the Microsoft Certificate store. An example of a friendly name for a certificate is 'my AET support certificate'. The content of the friendly name is derived from the content of a certificate.

In SafeSign Identity Client version 2.1.x, no friendly name is set when the certificate is suitable for smart card logon. This was implemented because the Microsoft 'winlogon' process also registers certificates, but without a friendly name, possibly causing problems with some applications (as the certificate selected for authentication may not be available under the same identifier anymore, depending on whether the Microsoft 'winlogon' process registered the certificate or SafeSign Identity Client did).

In SafeSign 2.3.2 and higher, smart card logon certificates are registered with a friendly name again. However, this led to problems with Outlook Web Access (and maybe other applications as well).

From SafeSign Identity Client 3.0 onwards (≥ 3.0.11) therefore, smart card logon certificates are not registered with a friendly name, but it is possible to enable friendly name registration for smart card user / logon certificates.

In order to enable a friendly name being set for smart card logon certificates, you should add the DWORD Value 'EnableFriendlyNameRegistration" in the Store Provider registry key:

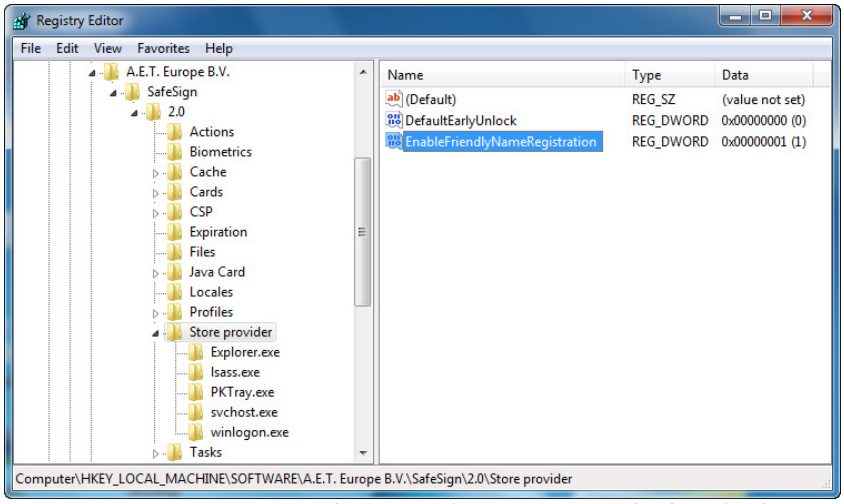

Figure 46: HKEY_LOCAL_MACHINE\SOFTWARE\A.E.T. Europe B.V.\SafeSign\2.0\Store provider\EnableFriendlyNameRegistration

When this value is set to 1, smart card logon certificates get a friendly name.

As of SafeSign Identity Client version 3.0.45, certificates are no longer registered by SafeSign, but by the Microsoft Certificate Propagation Service.

4.2.6.1 Edit friendly name value

Note that in SafeSign Identity Client version 3.0.33, it is not possible to edit the friendly name in the way described below.

In SafeSign Identity Client for Windows, the standard value for the friendly name is the CN of the subject and the CN of the issuer. Furthermore, it is possible to configure the friendly name.

To manipulate the friendly name of a certificate in the Microsoft Certificate store you should create a *string* value with the name *LabelFormat*.

This entry is not created by default, and should be manually created under:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\SafeSign Standard Cryptographic Service Provider\]

The following certificate values can be used to populate to content of the friendly name:

%N --> issuer CN
%n --> subject CN
%C --> issuer C (country)
%c --> subject C
%E --> issuer E (e-mail)
%e --> subject E
%L --> issuer L (locality)
%l --> subject L
%U --> issuer OU
%u --> subject OU
%O --> issuer O (organisation)
%o --> subject O
%S --> issuer S (state or province)
%s --> subject S

An example string value would be: "Certificate for %c issued by %N from %L (%C)". This would result in: "Certificate for support issued by A.E.T. Europe B.V. from Arnhem (NL)".

Appendix A: Install parameters

It is possible to perform a silent / unattended installation of SafeSign Identity Client and to select which components / features to install by means of command line options.

The components of SafeSign Identity Client version 3.0.45 32-bits (x86) that can be installed such are:

1. Locales: SafeSign Identity Client translations
2. CSP
3. CSP_Library
4. CSP_Signature
5. KSP : KeyStoreProvider, AET's implementation of Microsoft CNG
6. Common_Dialogs: SafeSign Identity Client Common Dialogs
7. PKCS11
8. GINA
9. CredentialProvider
10. Entrust
11. Firefox
12. Task_Manager
13. TAU
14. Documentation: License Agreement

The components of SafeSign Identity Client version 3.0.45 64-bits (x64) that can be installed such are:

1. Locales
2. CSP
3. CSP_Library
4. Documentation
5. x64
6. x64CSP
7. x64CredProv
8. x64KSP
9. x64User
10. x64Task
11. KSP
12. CSP_Signature
13. Common_Dialogs
14. Entrust
15. Firefox
16. PKCS11
17. x64TAU

A user guide for silent installation is available upon request.

Appendix B: Token structure

In this chapter you can find an overview on how (approximately) the space on different kind of tokens is divided.

On the STARCOS SPK range of tokens and Java Card v2.1 tokens, a fixed amount of space is allocated on the card for use with SafeSign Identity Client. Traditionally, this space was allocated for the SafeSign Identity Client PKCS #15 structure at card initialisation time.

For the Java Card v2.2 and higher supported tokens, the SafeSign Identity Client Java applet is still initialised with a PKCS #15 file structure, but because the applet allows files to grow and shrink, and even to be created on the fly, not all memory that will be used during the life cycle of the card has to be allocated at initialisation time.

Thus, it becomes much easier to have other applets co-existing with the SafeSign Identity Client Java applet on one card, even when these other applets are loaded after the SafeSign Identity Client applet has been loaded and initialised on the card, thus better leveraging the multi-application features of a Java Card.

The picture below shows how the STARCOS SPK 2.3 smart card memory is divided:

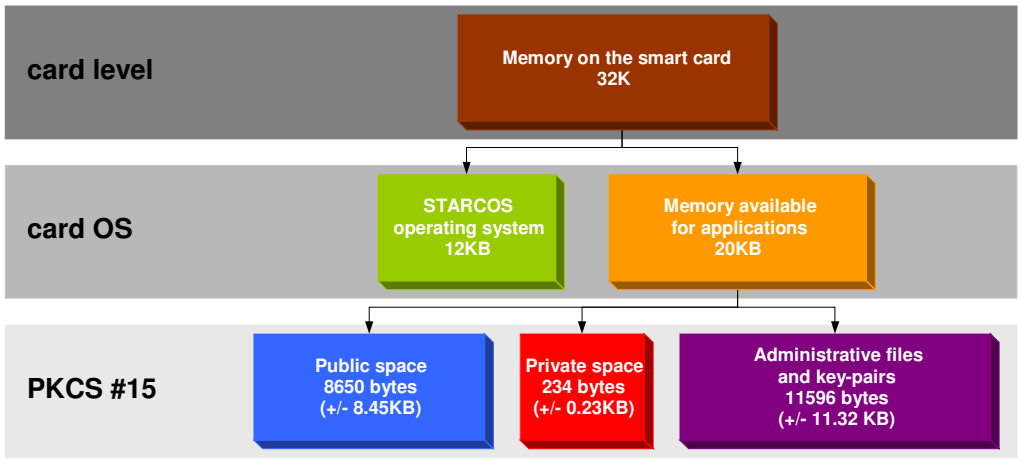

Figure 47: Card structure STARCOS SPK 2.3 smart card

This picture shows that all the free memory available on the card is taken up by the SafeSign Identity Client PKI applet and the PKCS#15 file structure written on the card. This means that there is no free space on the card for other applets. However, the public space on the card can be adjusted to fit, for example, data objects.

The picture below shows how the G&D Sm@rtCafé (Java Card v2.1) smart card memory is divided:

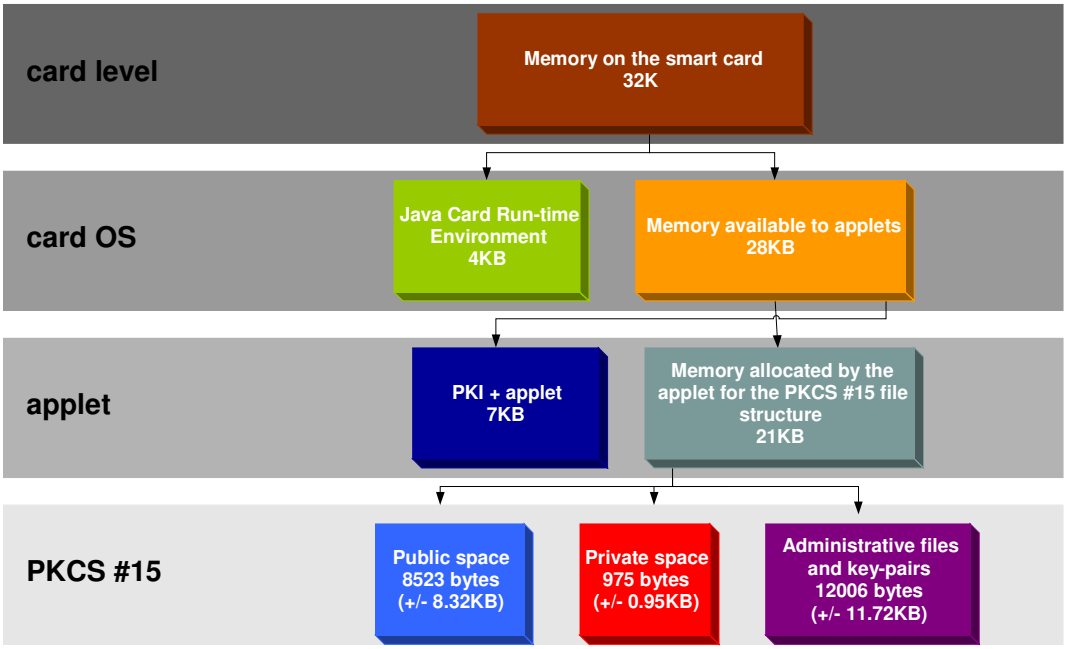

Figure 48: Card structure G&D Sm@rtCafé Java smart card

This picture shows that all the free memory available on the card is taken up by the SafeSign Identity Client PKI applet and the PKCS#15 file structure written on the card. This means that there is no free space on the card for other applets. However, the public space on the card can be adjusted to fit, for example, data objects.

Appendix C: Token middleware and the PC/SC layer

In this chapter you can find how SafeSign Identity Client middleware interacts between applications and smart card readers.

If we look at the standard architecture of middleware, we can divide the token applications into two different categories.

Category one is the category of applications that have a CSP interface (Cryptographic Service Provider). Examples of such applications are: Microsoft Internet Explorer or Microsoft Outlook Express. The CSP 'standard' is not an open standard.

The second category of applications is the applications with a PKCS #11 interface. Such applications are: Firefox or Thunderbird. In contrast to the Microsoft CSP standard, the PKCS #11 is an open standard.

"Below" the SafeSign Identity Client middleware is the PC/SC layer and the IFD handler. The PC/SC layer is a standard layer that interacts between the SafeSign Identity Client middleware and IFD handler. This PC/SC layer makes it possible for the SafeSign Identity Client middleware to talk via a standard interface to the smart card reader and the smart card.

The IFD handler is the driver for the smart card reader. This smart card reader can be a classical smart card reader or can have a USB format.

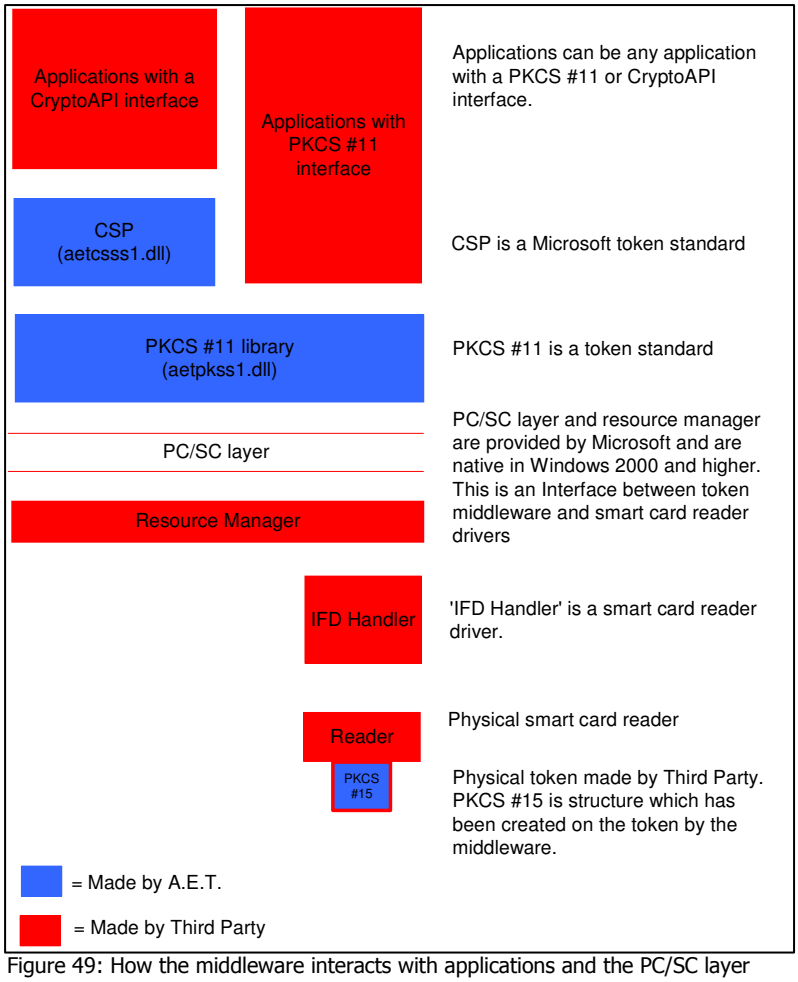

Figure 49: How the middleware interacts with applications and the PC/SC layer

Citrix[1] and Windows 2003 terminal service support the use of tokens. This means that tokens can be used within Windows 2003 terminal service and Citrix for smart card authentication.

Both implementations depend on the fact that they forward the PC/SC layer on the server to the client. From the middleware point of view, which needs only to be installed on the server, the smart card reader attached to the clients looks like it is attached on the server.

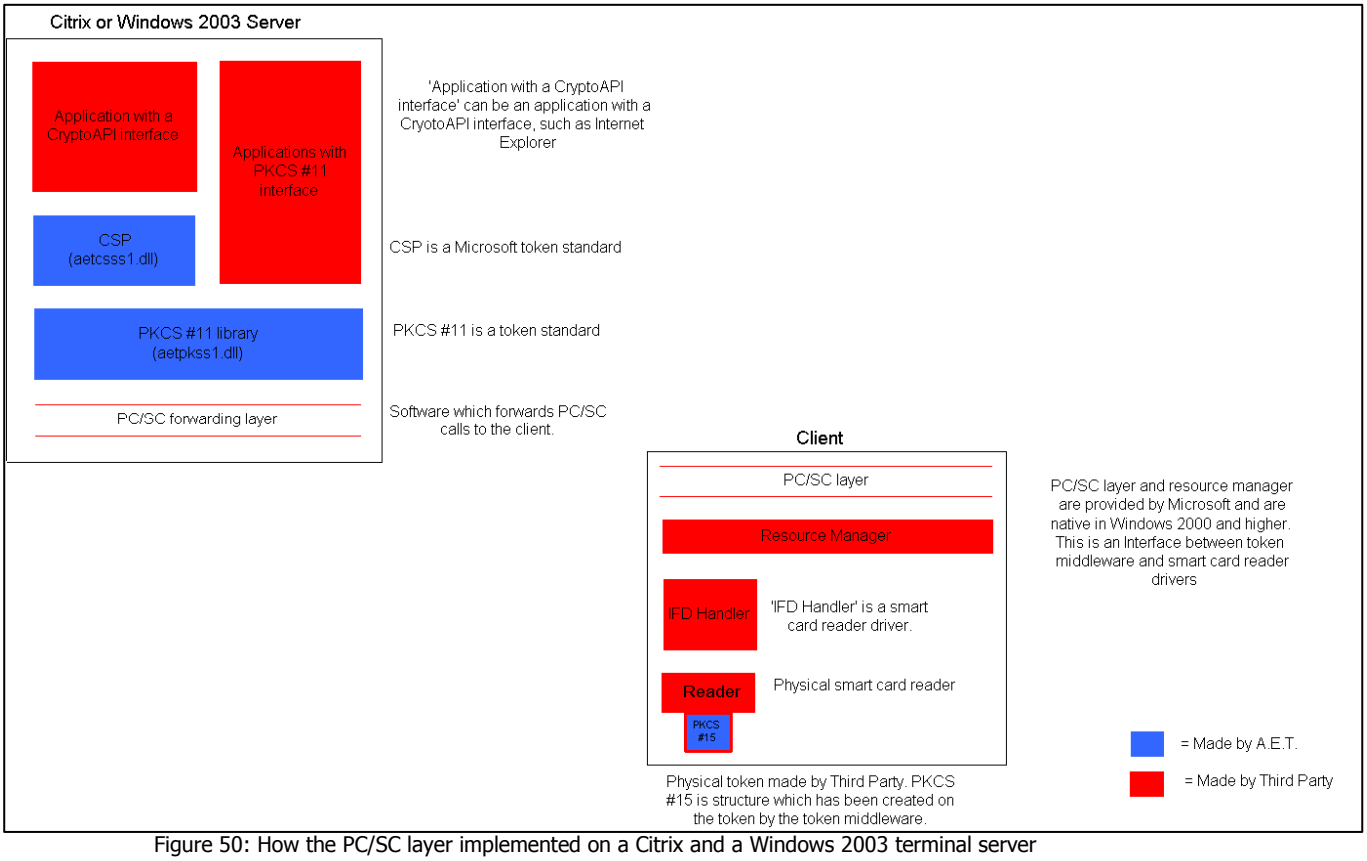

Figure 50: How the PC/SC layer implemented on a Citrix and a Windows 2003 terminal server

[1] Citrix Presentation Server.