



crypto vision

cv act *sc/interface* V6.0

Manual

cv cryptovision GmbH • Munscheidstr. 14 • 45886 Gelsenkirchen • Germany
www.cryptovision.com • info@cryptovision.com • +49-209-167-2450

Table of Contents

1	Introduction	5
2	About This Manual	6
3	About Public Key Cryptography	7
3.1	What is Security Token Middleware?	7
3.2	What are Public Key Cryptographic Standards?	8
3.3	What are Cryptographic Interfaces?	10
4	About cv act <i>sc/interface</i>	12
4.1	Modules of cv act <i>sc/interface</i>	12
5	Supported Hardware	14
5.1	Security Tokens	14
5.2	Smart Card Readers	19
5.3	Citrix Terminal Server	23
6	Installing cv act <i>sc/interface</i>	25
6.1	Installing on Windows	25
6.2	Installing on Linux	30
6.3	Installing on OS X	37
7	Using the Manager Tool for Administration	38
7.1	cv act <i>sc/interface</i> Manager client interface	38
7.2	Creating Profiles	45
7.3	Generating and Importing Keys	51
7.4	Changing PINs	56
7.5	Unlocking Smart Cards / Unlock User PIN	57
8	Sample Configuration of a Smart Card for First Use	59
8.1	Preparing a Smart Card (Initialization and Personalization)	59
8.2	Create Card Profile	59
8.3	Generation and Import of Certificates	62
9	Using Biometrics	65
9.1	Supported Smart Cards and Reader	65

9.2	Profile	65
9.3	Enroll Fingerprint.....	67
9.4	Unlock and Delete a Fingerprint.....	68
9.5	Smart Card Reader Mapping	68
10	Advanced Functions.....	70
10.1	"Trusted Certificates", "Useful Certificates", "Certificates"	70
10.2	Directory "Data"	71
10.3	Function "Open Token"	72
10.4	Function "Delete all" and "Delete Certificate" / "Delete Data" / "Delete Secret key" / "Delete Container"	72
10.5	Function "Set Default Container"	73
10.6	Function "Show Certificate"	74
10.7	Function "Export Certificate".....	74
10.8	Function "Register Certificate"	74
10.9	Function "Check Private Key"	75
10.10	Function "Check Secret Key"	77
11	User Tool.....	78
11.1	Changing PINs	78
11.2	Offline PIN Reset.....	79
11.3	Smart Card Registration	80
11.4	Export Certificates	81
12	Register Tool	82
12.1	Start cv act sc/interface Manager and Start cv act sc/interface Utility.....	82
12.2	PKCS#11 register/unregister	83
12.3	Pause / Continue	84
12.4	Settings.....	84
12.5	Exit	85
13	Advanced Cryptographic Interface Configuration	86
13.1	CSP Module	86
13.2	Biometric-Login GINA-integration	88
13.3	Configuration Parameters	89

14	Minidriver	92
14.1	CMCK Certification	92
14.2	Minidriver Support with PACE	92
15	PKCS#11 Module.....	94
15.1	General	94
15.2	Smart Card Login to a Novell eDirectory	95
15.3	SSL- Authentication with Smart Card over Firefox / Safari.....	95
15.4	E-mail-Security with Smart Cards for Netscape Messenger	95
15.5	cryptovision products, for Instance cv act <i>s/mail</i> or cv act PKIntegrated	95
15.6	Support of Elliptic Curve Cryptography (ECC).....	95
15.7	Initialization by PKCS#11	96
15.8	Identification of Smart Card	98
	Glossary	99
	Password	101
	Appendix A: Reference for Developers	103
	Synopsis of specific functions	103
	Objects.....	106
	Mechanisms.....	108
	Appendix B: Debug	110
	PKCS#11 Logger (win)	110
	CSP Logger (win)	110
	Appendix C: Certificate Attributes (Key Usage).....	111
	Information / Export Restrictions.....	112

1 Introduction

Thank you for choosing cv act *sc/interface*!

The rising demand for electronic identity verification requires much more than simple user names and passwords for authentication and user identification. Additional validation methods are a must, and digital certificates paired with security tokens like smart cards are ideal for this purpose. As a very mature solution, smart cards or smart tokens have been widely deployed for years on bank cards and more recently on electronic ID cards. With many different token options, the hardware typically is not a limiting factor. In fact, successful projects depend much more on the middleware used.

A smart card middleware is software which connects a security token to PKI-enabled applications. Typically, the most useful middlewares are platform independent and support a broad number of applications across differing devices. In addition, middleware should utilize standardized protocols and advanced cryptographic methods.

cv act *sc/interface* is a powerful smart card middleware which connects virtually any PKI-enabled application (for instance: Windows, Outlook, Safari, Mozilla, etc.) to the desired token. It supports all relevant cryptographic interfaces for every major operating system: Microsoft CSP and Minidriver for Windows, TokenD for Mac OS, and PKCS#11 for Linux derivatives. With hardware support for over 50 card types, cv act *sc/interface* removes dependence on any single card vendor and provides unrivaled interoperability.

In supporting both RSA and Elliptic Curve Cryptography, cv act *sc/interface* meets the encryption standards recommended by National Security Administration of the United States NSA and the German BSI. With its platform independence, a modular architecture, and advanced cryptographic algorithm use, cv act *sc/interface* is the most versatile and mature product of its kind on the market.

2 About This Manual

This document is the operator manual intended for administrators of smart card and security token middleware. It assumes that there is a working installation of the various different certificate authorities and web services that comprise Public Key Infrastructure (PKI) where security tokens are to be integrated. For more specific installation instructions and reference materials for initial PKI configuration please refer to the documentation included from the original supplier.

This manual contains installation and usage instructions for administrators of cv act *sc/interface*. It is beyond the scope of this manual to explain how to configure 3rd party software of other suppliers. Please consult the documentation of the respective products prior to the installation and operation of cv act *sc/interface*.

Because cv act *sc/interface* supports a delegated administration model, different modules can be installed depending on the particular functions that are needed. If you intend to use cv act *sc/interface* Admin Edition, you will find the description of the administration tool helpful. It describes how to manage keys and certificates, how to change PINs, and how to unlock, initialize and personalize smart cards.

If you want to use cv act *sc/interface* User Edition, please refer to the description of the user tool. It describes how to change PINs and how to register the certificates on your smart card.

Furthermore, you will find more detailed information regarding the other modules of cv act *sc/interface*: Register Tool, CSP, Minidriver and PKSC#11 in the respective chapters. Other information about using biometrics and fingerprints along with a list of applications which can be employed with smart cards is included.

Further technical reference information is also included. Application developers will find information on how to access modules (e.g. accessing PKCS#11) of cv act *sc/interface* in appendix A. This is particularly helpful for developing custom applications. Appendix B contains a concise description of the certificate attributes, i.e. information about key employment.

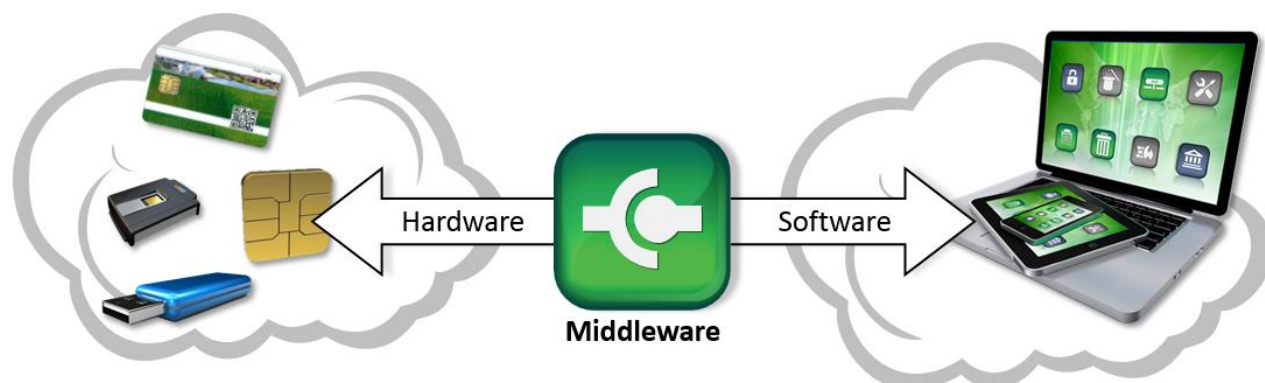
NOTE: To understand this manual you need a basic knowledge of IT security. You should be familiar with the following concepts: (digital) certificates, private, public, and secret keys, digital signature, PKI, etc. If you want to expand your knowledge of IT-security and cryptography, there is information available in the Solutions and Company areas of the cryptovision homepage: <http://www.cryptovision.com/>

3 About Public Key Cryptography

3.1 What is Security Token Middleware?

Operating system login, VPN access, secure web portal access, and similar applications need appropriate protection. Many enterprises still use passwords for this purpose, which are neither secure nor convenient. As a replacement for passwords, more and more enterprises issue smart cards or smart security tokens to their employees. These tokens are used for securely storing secret keys, which are used as a more secure replacement for simple passwords. These keys can also be used for encryption and digital signature.

In order to use a security token on a desktop, a security token middleware is necessary. Middleware is a software component that connects a smart card (or a smart token) with the desktop operating system and one or more applications. The core of the middleware is similar to a device driver, which delivers a high-level cryptographic interface to applications and which communicates the smart card via a (proprietary) low-level interface. In addition, many smart card middlewares include management tools for formatting, personalizing and performing similar tasks on the security token.



Security token middleware technology is far from trivial. As many users use the same card on different platforms, including both desktop and mobile, several operating systems need to be supported. In addition, there are dozens of different types of smart card or tokens all with differing interfaces that are required to be supported by the same middleware. Even on a single platform like Windows, there are different crypto interfaces like Cryptographic Service Provider or Minidriver.

3.2 What are Public Key Cryptographic Standards?

Because of the varying different methods for encrypting information and implementing public key infrastructures, it is essential to determine an agreed upon set of standards and practices in order for different operating systems, applications and security components to interact with one another. In the early 1990's the RSA Security Inc. firm first published their proposals for such standards to promote the use of their proprietary technology, including their RSA encryption algorithm. As such, these were not immediately established as industry standards but over time these Public Key Cryptography Standards ([PKCS](#)) became so widely used that even though RSA retains control over them, they began to be adopted by other organizations like the Internet Engineering Task Force ([IETF](#)) and more specifically the Public Key Infrastructure Working Group ([PKIX WG](#)). While there are many different PKCS standards, this manual will focus on those relevant to security token middleware

3.2.1 PKCS#7: Cryptographic Message Syntax Standard

The Cryptographic Message Syntax (CMS) is the IETF standard for cryptographically protecting messages or other forms of digital data. This includes digital signature, message digest, authentication and encryption of content. Because CMS is based on the syntax of PKCS#7 it forms the cornerstone for other standards such as Secure/Multipurpose Internet Mail Extensions (S/MIME) which are used for signing and encrypting e-mail and are needed for the interoperability of email and messaging platforms.

3.2.2 PKCS#10: Certification Request Standard

In any public key infrastructure, one of the most basic operations is the initial issuance of a digital certificate from a certification authority to participants in the PKI. A certificate request consists of a distinguished name, a public key, and other optional attributes that are digitally signed by the requesting entity. These certificate signing requests are then processed by a certification authority and then converted into a signed X.509 digital certificate which is then trusted throughout the PKI. The [PKCS#10](#) standard provides the proper Certificate Signing Request (CSR) syntax specifications to be used so that the diverse elements of a PKI, such as browsers, middleware, and applications can interoperate with a singular certification authority.

3.2.3 PKCS#11: Cryptographic Token Interface "Cryptoki"

The [PKCS#11](#) standard is one of the most mature and well established communication mechanisms for the interchange of data between devices which hold cryptographic information. Often referred to as "Cryptoki" (pronounced "crypto-key") this application programming interface was developed to facilitate the goal of technology independence amongst security tokens. It aims to present a common logical view of the cryptographic token that can be shared between multiple devices and multiple applications.

3.2.4 PKCS#12: Personal Information Syntax Standard

In order provide for the secure storage and, potentially transport, of cryptographically sensitive private keys and their corresponding certificates, the [PKCS#12](#) standard defines the syntax to be used and a file format to allow for the password-based symmetric encryption of content. PKCS#12 is a successor to the Microsoft Personal Information Exchange (PFX) format, although the two are often used interchangeably.

3.2.5 PKCS#15: Cryptographic Token Information Format Standard

cv act *sc/interface* implements an ISO 7816 compliant file system designed for different smart card platforms. This includes features required for [PKCS#15](#) or ISO 7816-15 ("Cryptographic Token Information Syntax Standard") based card profiles. Additionally, it can be used for IAS applications (Identification, Au-

thentication, Signatures) with optional support for biometric fingerprint match-on-card (MoC) using the Biomatch J™ package by Precise Biometrics. The PKCS#15 standard defines a hierarchical folder structure and defines file types to be used on a smart card which outlines the uses of key and certificate storage. Additional information regarding object specific properties describing key size and type is also stored. One advantage of this standardization is that every PKCS#15 compliant host software (e.g. a smart card middleware such as cv act [sc/interface](#)) is able to access to PKCS#15 compliant smart cards without any adjustments required on a file level.

3.2.6 ISO/IEC 7816-15:2004 Cryptographic Information Application

The PKCS#15 v1.1 standard is superseded by ISO 7816-15 effectively making it PKCS#15 v2, while providing backward compatibility for the older PKCS#15 revisions.

According to the [ISO/IEC 7816-15:2004](#) abstract:

“ISO/IEC 7816-15:2004 specifies a card application. This application contains information on cryptographic functionality. Further, ISO/IEC 7816-15:2004 defines a common syntax (in ASN.1) and format for the cryptographic information and mechanisms to share this information whenever appropriate.

ISO/IEC 7816-15:2004 supports the following capabilities:

- *storage of multiple instances of cryptographic information in a card;*
- *use of the cryptographic information;*
- *retrieval of the cryptographic information;*
- *cross-referencing of the cryptographic information with DOs defined in ISO/IEC 7816 when appropriate;*
- *different authentication mechanisms; and*
- *multiple cryptographic algorithms.”*

3.2.7 PACE (Password Authenticated Connection Establishment) by TR-03110

PACE (Password Authenticated Connection Establishment) is a mutual authentication mechanism between reader and chip based on a shared password, such as a secret PIN known only by the user, or a CAN (Card Access Number) printed on the smart card (as in the German electronic ID card). The method is used for initial configuration of a secure connection.

The protocol was developed by the german Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik) for the use for the [electronic ID card](#) and is described in the technical guideline [BSI-TR-03110](#).

PACE has the advantage that the password length has no influence on the security level of the encryption. This means even with the short and so not so secure PIN the data is strongly protected on the chip of the smart card and during transmission (Secure Messaging) by using a secure channel from cv act [sc/interface](#) to the smart card and a PIN will not be stored intermediately anywhere outside of the card.

3.3 What are Cryptographic Interfaces?

In the current computing environment, most applications are intended to be run across many different platforms. For example, a single internet page is often targeted at users on Windows, Mac OS X, Linux and more. Further, the different browsers that run on each of the operating systems use different mechanisms for performing cryptographic operations. In order to provide independent development of applications suitable for use across differing platforms, a programmatic interface to perform cryptographic operations are used. The specific interface used depends largely on the operating system platform, as well as the calling application. cv act *sc/interface* provides the following cryptographic interfaces:

3.3.1 Cryptographic Service Provider

Within Microsoft operating systems, a Cryptographic Service Provider is a component that provides for generic or common cryptographic functions. While the operating system themselves include specific CSPs which are configured to be used with a corresponding certificate template and the defined cryptographic algorithms. By selecting a specific CSP, administrators can effectively determine what algorithms and key lengths are used with the certificates issued. Additional CSPs can be added, like the cryptovision cvCSP.DLL included with cv act *sc/interface* that include enhanced functionality like biometry or PIN caching.

3.3.2 Minidriver

Microsoft has designed a new interface to be used for accessing smart cards. This interface provides a consistent method for the usage of smart cards with the Base Smart Card Cryptographic Service Provider (Base CSP) or the Crypto Next Generation (CNG) Key Storage Provider (KSP) and the Smart Card Management Interface.

This minidriver is added to the Windows system during installation. The minidriver consists of very few smart card specific files. Further parts of the driver which are necessary for a smart card CSP are integrated in the operating system. More information is available in the "[Smart Card Minidriver Specification](#)" document from Microsoft.

Further information on the card minidriver that is part of cv act *sc/interface* is available in the chapter "Minidriver".

3.3.3 TokenD

TokenD is the defined interface for the use of smart cards with Mac OS, enabling the keys on smart cards to be used by applications like Logon, the Safari browser, or mail clients.

The Keychain is an encrypted container that holds passwords for multiple applications and secure services. Keychains are secure storage containers, which mean that when the keychain is locked, no one can access its protected contents. In Mac OS X, users can unlock a keychain; thus providing trusted applications access to the contents - by entering a single master password. Further information you find under in the "[Keychain Services Programming Guide](#)" from the Apple iOS Developer Library.

3.3.4 PKCS#11 Library

The PKCS#11 library of cv act *sc/interface* is delivered as either a dynamic link library for Windows (cvP11.dll) or a shared object (libcvP11.so) for Linux and Mac OS X. These libraries allow for operating system or kernel level interaction with the cryptographic token via the cryptoki API.

3.3.4.1 Virtual Slots

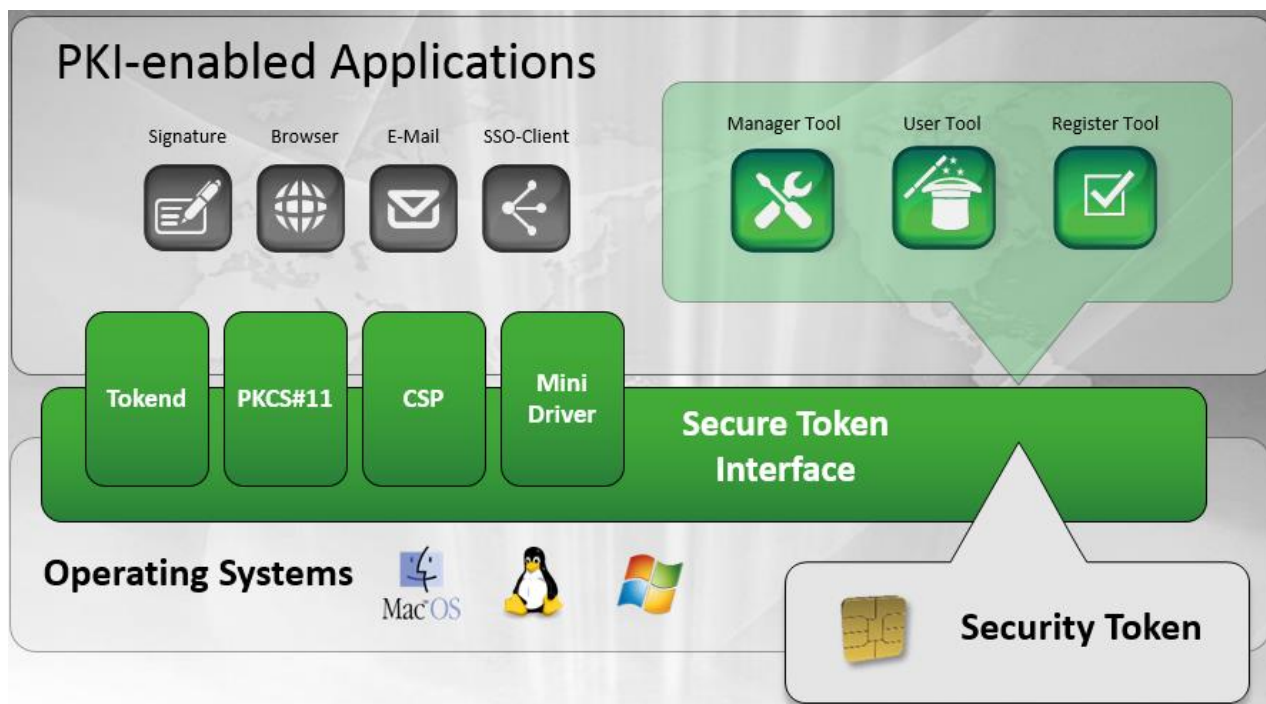
cv act *sc/interface* supports smart cards which provide multiple applications on the card, or multiple PIN's per application, such as D-Trust or SwissSign.

Because PKCS#11 is not intended to be used with more than one User-PIN, a compliant solution for multi-application scenarios was introduced by Nexus called "virtual slots" (see PKCS#11 v2.10 <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/pkcs11-v2-10/pkcs11v2-10.pdf>). By splitting a slot into multiple "virtual slots" exactly one User-PIN per virtual slot can be defined for any application which is capable of using PKCS#11. This enables a single smart card to provide multiple User-PIN's for key material. This is represented a single logical slot within cv act *sc/interface*, but to applications it appears that there are multiple slots and tokens.

4 About cv act *sc/interface*

4.1 Modules of cv act *sc/interface*

This illustration represents the architectural components of the middleware:



cv act *sc/interface* is comprised of the following modules:

- **Manager Tool:** This administration tool enables full card, key, and certificate management functions. It supports key generation, import and export of certificates, as well as other functions like generation of certificate signing requests. With the Manager tool it is possible to create different types of card profiles, set and change the smart card PIN, and unlock a smart card and enroll biometric credentials.
- **User Tool:** This specialized interface allows only user functions; enabling self-service user PIN changes and certificate registration on a smart card.
- **Register Tool:** This tool seamlessly registers user's certificates from a smart card into the Windows certificate store. It also can be used to register the PKCS#11 module for browser integration.
- **CSP Module:** The CSP (Cryptographic Service Provider) module allows a smart card login to the operating system and Windows domains and extends token use for other applications and services in Microsoft environments.

- **GINA-integration:** This legacy module enables smart card login to Windows by extending the Windows Graphical Identification and Authentication (GINA) library. This enables both smart card and biometric login through the CSP for 32-Bit Windows XP.
- **PKCS#11 Module:** The PKCS#11 module applications and services for applications support via the PKCS#11 Cryptographic Token Interface Standard and the use of the cryptoki application programmer interface. Examples of such programs include Mozilla-based applications, like Firefox or Thunderbird, as well as Linux and Novell environments. The PKCS#11 module of cv act *sc/interface* supports cryptographic algorithms that are based on both RSA and also elliptic curves (ECC). Further information is available in the chapter "PKCS#11 Module".
- **Minidriver Module:** With this interface smart cards can be used by the Microsoft Base Smart Card Cryptographic Service Provider (CSP) or the Crypto Next Generation (CNG) Key Storage Provider (KSP). It makes the Smart Card Management Interface available. Further information you find in the chapter "Card Minidriver". This module is available in both the 32-Bit and in the 64-Bit versions.
- **TokenD Module:** This plugin enables smart card use with the native cryptographic token handling in Mac OS operating system (Mac OS X v10.4 and higher). Further information is available in the chapter "TokenD".

The Administration Tool and the User Tool are parts of every installation package. The other modules will only be supported by the relevant platforms as shown here:

	PKCS#11	Register Tool	CSP	Minidriver	GINA	TokenD
Win 32-Bit	✓	✓	✓	✓	only Windows XP	--
Win 64-Bit	✓	✓	✓	✓	--	--
Linux	✓	--	--	--	--	--
Mac OS X	✓	--	--	--	--	✓

5 Supported Hardware

5.1 Security Tokens

Here you find a list of all smart cards and smart card operating systems which were successfully tested with this release of cv act *sc/interface*. So this security token are supported by cv act *sc/interface*.

	cvProfile	PKCS#15	Biometric	Minidriver	TokenD
<i>AustriaCard ACOS EMV A04 / A05</i>	✓	✓		✓	
<i>AustriaCard JCOP 21 V2.2</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 21 V2.3.1</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 31 V2.2</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 31 V2.2 contactless</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 31 V2.3.1</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 31 V2.3.1 contactless</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 31/72 V2.3.1</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 31/72 V2.3.1 contactless</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 41 V2.2.1</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 41 V2.2.1 contactless</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 41 V2.3.1</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 41 V2.3.1 contactless</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 41 V2.4</i>	✓	✓		✓	✓
<i>AustriaCard JCOP 41 V2.4 contactless</i>	✓	✓		✓	✓
<i>E.ON Card V1</i>		✓		✓	✓
<i>E.ON Card V1 contactless</i>		✓		✓	✓
<i>ePasslet-Suite 1.1 on JCOP V2.4.1R3</i>	✓	✓	✓	✓	✓

	cvProfile	PKCS#15	Biometric	Minidriver	TokenD
<i>ePasslet-Suite 1.1 on JCOP V2.4.1R3 with PACE Profile</i>	✓	✓	✓	✓	✓
<i>ePasslet-Suite 1.2 on JCOP V2.4.1R3</i>	✓	✓	✓	✓	✓
<i>ePasslet-Suite 1.2 on JCOP V2.4.1R3 with PACE Profile</i>	✓	✓	✓	✓	✓
<i>ePasslet-Suite 2.0 on JCOP V2.4.2R3</i>	✓	✓	✓	✓	✓
<i>ePasslet-Suite 2.0 on JCOP V2.4.2R3 with PACE Profile</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 3.1</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 3.1 contactless</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 3.2</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 4.0</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 5.0</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 6.0</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 6.0 SCP 03</i>	✓	✓	✓	✓	✓
<i>G&D STARCOS 3.0</i>		✓		✓	✓
<i>G&D STARCOS 3.1</i>		✓		✓	✓
<i>G&D STARCOS 3.2</i>		✓		✓	✓
<i>G&D STARCOS 3.4 (Swiss Health Card eGK)</i>		✓		✓	✓
<i>Gemalto TOP IM GX4</i>	✓	✓		✓	✓
<i>HID Crescendo C700</i>	✓	✓		✓	✓
<i>HID Crescendo C700 contactless</i>	✓	✓		✓	✓
<i>Infineon JCLX80 jTOP</i>		✓	✓	✓	✓
<i>Infineon JCLX80 jTOP contactless</i>		✓	✓	✓	✓
<i>NXP JCOP V2.1</i>	✓	✓	✓	✓	✓

	cvProfile	PKCS#15	Biometric	Minidriver	TokenD
<i>NXP JCOP V2.2</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.2 Contactless</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.3.1</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.2.1 IDptoken 200</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.4</i>	✓	✓	✓		✓
<i>NXP JCOP V2.4.1</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.4.2 R1</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.4.2 R2</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.4.2 R3</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.4.2 R3 SCP 03</i>	✓	✓	✓	✓	✓
<i>NXP JCOP V2.2 Certgate microSD</i>	✓	✓	✓	✓	✓
<i>Siemens CardOS M4.01a</i>	✓		✓	✓	
<i>Siemens CardOS V4.2</i>	✓	✓		✓	✓
<i>Siemens CardOS V4.2B</i>	✓	✓		✓	✓
<i>Siemens CardOS V4.2C</i>	✓	✓		✓	✓
<i>Siemens CardOS V4.3</i>	✓	✓		✓	✓
<i>Siemens CardOS V4.3B</i>	✓	✓		✓	✓
<i>Siemens CardOS V4.4</i>	✓	✓		✓	✓
<i>SwissSign SwissStick (CardOS M4.3B)</i>	✓	✓		✓	✓
<i>SwissSign suisseID (CardOS M4.3B)</i>	✓	✓		✓	✓
<i>SwissSign suisseID (CardOS M4.4)</i>	✓	✓		✓	✓

Here you find a list of all smart cards and smart card operating systems which should work with cv act *sc/interface*, but which were not tested in this release. Additional hardware testing of specific additional hardware can be performed upon customer request.

	cvProfile	PKCS#15	Biometric	Minidriver	TokenD
<i>AustriaCard ACOS EMV D01</i>	✓	✓			✓
<i>AustriaCard JACOS 2.4.1</i>	✓	✓	✓		✓
<i>G&D Mobile Security Card 3.x microSD™</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 3.2 StarSign Card Token 550 (USB)</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 64 cfg3</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 64 cfg8</i>	✓	✓	✓	✓	✓
<i>G&D Sm@rtCafé Expert 64 StarKey400 USB Token</i>	✓	✓	✓	✓	✓
<i>Gemalto GemXpresso Pro R3</i>	✓	✓		✓	✓
<i>Oberthur Cosmo V5.2D</i>		✓		✓	✓
<i>Siemens CardOS M4.01</i>	✓		Readonly		
<i>Siemens CardOS V4.2B contactless</i>	✓	✓		✓	✓

Notes:

- A supported token which is configured with middleware of another vendor, can also work with cv act *sc/interface*. For more information about the configuration of the cryptovision middleware please contact your contact person of cryptovision.
- For all JavaCards SCP 02 will be supported. For some JavaCards, like G&D Sm@rtCafé Expert 6.0 or NXP JCOP V2.4.2 R3 additionally SCP 03 will be supported.
- For G&D Sm@rtCafé Expert 3.1 and G&D Sm@rtCafé Expert 3.2 on StarSign Card token 550 (USB) the "Visa Fixed keys" and the key derivation "CPG2.04" are supported. For the G&D Sm@rtCafé Expert64 only the "Visa Fixed Keys" are supported.
- For using the biometric profile in case of CardOS M4.01a the Match-On-Card Package is required, in case of G&D smart cards the BioMatch J 3.0 Applet. For G&D smart cards only the PKCS#15-profile with biometrics is supported.
- All PINs of all smart cards mentioned above are supported. Therefore all keys stored on such a smart card can be used. Further information is available in the chapter Introduction" ➔ "virtual slots".

- In case of G&D StarSign Version 1.0, Siemens HiPath Version 1.6.2.1 (and above), A.E.T. SafeSign Version 2.3.0 (and above) and Nexus Personal Version 4.6.1 (and above) CardOS M4.01a is not supported.
- In case of Gemalto TOP IM GX4 and Infineon JTOP the Triple-DES function on the smart card is not supported.
- In case of StarCOS 3.0 only 128Bit Triple-DES keys are supported.
- If the G&D Mobile Security card is used, please configure the following ATR in the G&D configuration software "PC / SC Driver for Mobile Security Card" under "Communication Settings": 3B B9 18 00 C0 0A 31 FE 45 53 50 4B 32 35 44 49 90 00 33



5.2 Smart Card Readers

cv act *sc/interface* uses the PC/SC interface of the respective operating platform for data interchange between smart card, reader, and operating system. In theory, all PCSC 2.0-compliant smart card readers with appropriate drivers should work. With Apple OS X, and Linux derivatives "pcsc-lite" version 1.1.2 or newer is required.

In some cases, a hardware manufacturer may not implement a full PC/SC complaint driver package which may lead to possible errors in certain combinations of smart card, driver, and the firmware of smart card reader chipset. In such cases error messages will occur (e. g. "send error") and you should contact the vendor of your smart card reader to receive an update of the driver or firmware of the smart card reader.

When using virtual machines, there is an additional hardware layer. Thus dependencies can occur and the function of a reader cannot be guaranteed 100%.

Here you find the smart card readers which were successfully tested with this release of cv act *sc/interface*. So these readers are supported by cv act *sc/interface*.

- Omnikey Cardman 3121 USB
- Identive SCR3310v2.0

In the following subchapters you find lists of all smart card readers which should work with cv act *sc/interface*, but which were not tested in this release. Additional hardware testing of specific additional hardware can be performed upon customer request.

5.2.1 Smart Card Readers without PIN pads or USB-tokens

- ACS ACR100 SIMFlash (CCID)
- ACS ACR100 SIMFlash (HID)
- ACS ACR101 SIMicro (CCID)
- ACS ACR38 Smart Card Reader
- ACS ACR38DT DualKey
- ACS ACR38ET DualKey2
- ACS ACR38T Plug-in (SIM Sized) Card Reader
- Cherry SmartTerminal ST-1044U
- Cherry SmartTerminal ST-1210
- Gemalto GemPC Express
- Gemalto PC Twin Reader (USB/Serial)
- Omnikey Cardman 2020 USB
- Omnikey Cardman 3620 USB
- Omnikey 6121 USB
- Eutronsec SIMReader Combo
- Identive CLOUD 2700 F
- Identive CLOUD 4700 F
- Identive @MAXX® lite
- Identive @MAXX® NFC

- Identive @MAXX® prime
- Identive @MAXX® token SCT3511
- Identive SCT3522
- Identive SCR3311
- Identive SCR335
- Identive SCR241
- Identive SCR243
- Identive SCR331
- Identive SCR3310
- Identive SCR3320
- Identive SCR3321
- Identive SCR333
- Identive SCR3340
- Identive SCR335 USB
- Identive SPR532 serial/USB

5.2.2 Keyboards with Integrated Smart Card Readers

- ACS ACR38k Smart Keyboard
- Cherry FingerTIP ID Board G83-14400
- Cherry FingerTIP ID Board G83-14500
- Cherry FingerTIP ID Board G83-14600
- Cherry MultiBoard contactless G81-8072LUC
- Cherry SmartBoard G83-6610
- Cherry SmartBoard G83-6644
- Cherry SmartBoard Twin G83-6675

5.2.3 Contactless Smart Card Readers

- ACS ACR120
- ACS ACR122L VisualVantage NFC
- ACS ACR122S NFC
- ACS ACR122T NFC
- ACS ACR122U NFC
- ACS ACR1281U nPA
- ACS ACR128U Dual-Interface
- Cherry SmartTerminal ST-1275
- Omnikey CardMan 5321 RFID
- Identive CLOUD 4700 F
- Identive SCL010
- Identive SCL3711

- Identive SDI010
- Identive SDI110
- REINER SCT cyberJack® RFID basis

5.2.4 Smart Card Readers with Fingerprint Sensors

In such case the fingerprint sensor is used to read the finger print that is used for authentication instead of a PIN:

- ACS AET52
- ACS AET63
- ACS AET65
- Omnikey 7121 Biometric
- Precise Biometrics 250 MC
- Precise Biometrics SENSE™ MC
- Precise Biometrics SENSE™ MC-S
- Precise Biometrics Tactivo™ for iPhone (with supported card configuration and a special software – please contact your contact person at cryptovision)
- Precise Biometrics Tactivo™ for iPad with supported card configuration and a special software – please contact your contact person at cryptovision)

5.2.5 Smart Card Readers with PIN Pads

cv act *sc/interface* works with smart card readers with PIN Pad and the use of the PIN Pad for secure PIN entry. The following requirements have to be met:

It has to be a PC/SC reader.

The smart card reader has to support the FEATURE_VERIFY_PIN_DIRECT.

Secure PIN entry is configured by a registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface]
"Enable_Secure_PIN_Entry"=dword:00000001
```

If this key is set to the value "0", the pinpad is not available. When the key is "1" the PIN Pad can be used. This configuration affects CSP- and PKCS#11 Module, but not the Administration Tool or the User Tool. If the key is absent, the default value "1" is used. This means, if a PIN Pad reader is installed, it is used automatically.

Notes:

- Please note that there is an problem with the use of Outlook and an reader with PIN Pad. Your allocation can freeze.
- Smart card logon always requires the PIN, so that you cannot use an PIN pad reader for smart card logon.

Here is a list of known smart card readers with PIN Pad:

- ACS ACR83 PINeasy Smart Card Reader
- ACS ACR88 PIN-Pad Reader
- ACS ACR880 GPRS Portable Smart Card Terminal
- Cherry SmartTerminal ST-2000U
- Omnikey 3821
- REINER SCT cyberJack® RFID comfort
- REINER SCT cyberJack® RFID standard
- REINER SCT cyberJack® e-com plus
- REINER SCT cyberJack® secoder
- Gemalto PC Pinpad Reader
- KAAN TriB@nk

Notes (Known Issues):

- KAAN TriB@nk is not supported in combination with the G&D Sm@rtCafé Expert 64 and Gemalto TOP IM GX4 smart cards.
- The Gemalto PC Pinpad reader supports only a maximal PIN length of 8, which means that longer PINs cannot be inserted. Under Windows there is no possibility to retrieve this maximal PIN length (right now). Under Linux this works from the CCID-Version 1.3.12 release 8 May 2010. This issue is discussed on the following web-site: <http://ludovicrousseau.blogspot.com/2010/05/how-to-know-pin-sizes-supported-by.html>

5.2.6 ExpressCard & PCMCIA Readers

- ACS ACR92
- Cherry SmartReader SR-4044
- Cherry SmartReader SR-5044
- Omnikey 4040 PCMCIA
- Omnikey 4321 ExpressCard 54
- Identive SCR243
- Identive SCR3340

5.2.7 Mobile Readers

- Identive @MAXX ID-1
- Identive SCR3500
- Identive SCL3711
- Precise Biometrics Tactivo™ for iPhone (with supported card configurations)
- Precise Biometrics Tactivo™ for iPad (with supported card configurations)

5.3 Citrix Terminal Server

cv act *sc/interface* is tested with speciell configurations of the following components.

5.3.1 Tested configuration of XenDesktop 5

- Windows 2008 R2 Server
- XenDesktop 5 with Windows 7 64bit (via VMWare)
- CSP Module of cv act *sc/interface* (cvCSP) – **IMPORTANT:** the cvCSP must be first installed on the client and on the server, before the Citrix components are installed. Please run the installation of cv act *sc/interface*. For more information to the requirements of Citrix see <http://support.citrix.com/proddocs/topic/xendesktop-71/cds-smart-cards-sys-reqs.html>
- Reader: Identive SCR 3310
- JCOP2.4.1 and CardOS 4.3
- Applications:
 - Smart card logon to XenApp 5 via WebInterface (Internet Explorer 10 and Firefox 24.0)
 - Smart card logon to Windows 7 and Windows XP
 - Certification Enrollment (Internet Explorer 10 and Firefox 24.0)
 - SSL (Internet Explorer 10 and Firefox 24.0)
 - PDF signing with Adobe 11

5.3.2 Tested Configuration of XenApp 6.5

- Windows 2008 R2 Server
- XenApp 6.5 with Windows 7 FatClient(VM)
- CSP Module of cv act *sc/interface* (cvCSP) – **IMPORTANT:** the cvCSP must be first installed on the client and on the server, before the Citrix components are installed. Please run the installation of cv act *sc/interface*. For more information to the requirements of Citrix see <http://support.citrix.com/proddocs/topic/xendesktop-71/cds-smart-cards-sys-reqs.html>
- Reader: Identive SCR3310v2
- JCOP2.4.1 and CardOS 4.3
- Applications:
 - Smart card logon to XenApp 6.5 via WebInterface (Internet Explorer 10 and Firefox 24.0)
 - Certification Enrollment (Internet Explorer 10 and Firefox 24.0)
 - SSL (Internet Explorer 10 and Firefox 24.0)
 - PDF signing with Adobe 9

Note: The drivers of an USB Token must be HotPlugEnable (Citrix requirement)

5.3.3 Tested Version of CitrixReceiver

- CitrixReceiver: 4.1.0.56 (Fileversion: 14.1.0)

6 Installing cv act *sc/interface*

Because cv act *sc/interface* is designed to function on every major desktop operating system, the installation procedure varies depending on the platform. In the following sections, the different operating system installation procedures will be described in further detail.

6.1 Installing on Windows

The functionality of cv act *sc/interface* is targeted at two main groups, either administrators of the security tokens who will perform full token lifecycle management functions like initially configuring the tokens for first use, key and certificate management, and end users of the tokens. Due to this role distinction, the Manager and the User modules are split into different applications and use different software setup files. These setup files are also split further into specific versions for 32 bit and 64 bit chip architectures. The cv act *sc/interface* installation media is delivered as a compressed archive file complete with subfolders. For Windows, the relevant subfolders are as follows:

- installation_admin
- installation_admin_x64
- installation_user
- installation_user_x64
- minidriver
- support
- windows

The setup files are contained in the installation_admin or installation_user folders for the respective chip architectures. The other subfolders contain files and resources useful for advanced installations or for verbose logging versions useful for application programmers and diagnostic troubleshooting.

6.1.1 Supported Windows Versions

Here you find the Microsoft operating systems which were successfully tested with this release of cv act *sc/interface*. So these Microsoft operating systems are supported by cv act *sc/interface*:

- Windows XP with Service Pack 3
- Windows 7 XP with Service Pack 1
- Windows Server 2008 R2 with Service Pack 1
- Windows 8.0 (Minidriver only)
- Windows 8.1 (Minidriver only)

Here you find a list of the Microsoft operating systems which should work with cv act *sc/interface*, but which were not tested in this release. Additional testing of specific additional OS can be performed upon customer request:

- Windows Server 2003 with Service Pack 3
- Windows Vista with Service Pack 2
- Windows Server 2008 with Service Pack 2
- Windows 8.0 (CSP)

- Windows 8.1 (CSP)
- Windows Server 2012

6.1.2 Running the Windows Administrator Setup

Start the file SETUP.EXE as a user with administrator rights. Follow the on screen installation instructions as guided by the setup wizard. The following screen shots illustrate this process.

Please note: The installation of cv act *sc/interface* requires administrative rights. If you start the installation as a regular user, you will be prompted by the installation process for alternative credentials. For further details please refer to the respective operating system documentation or ask your system administrator.



Click next or press enter to begin the installation process.



By continuing you accept the terms specified as part of the license agreement.



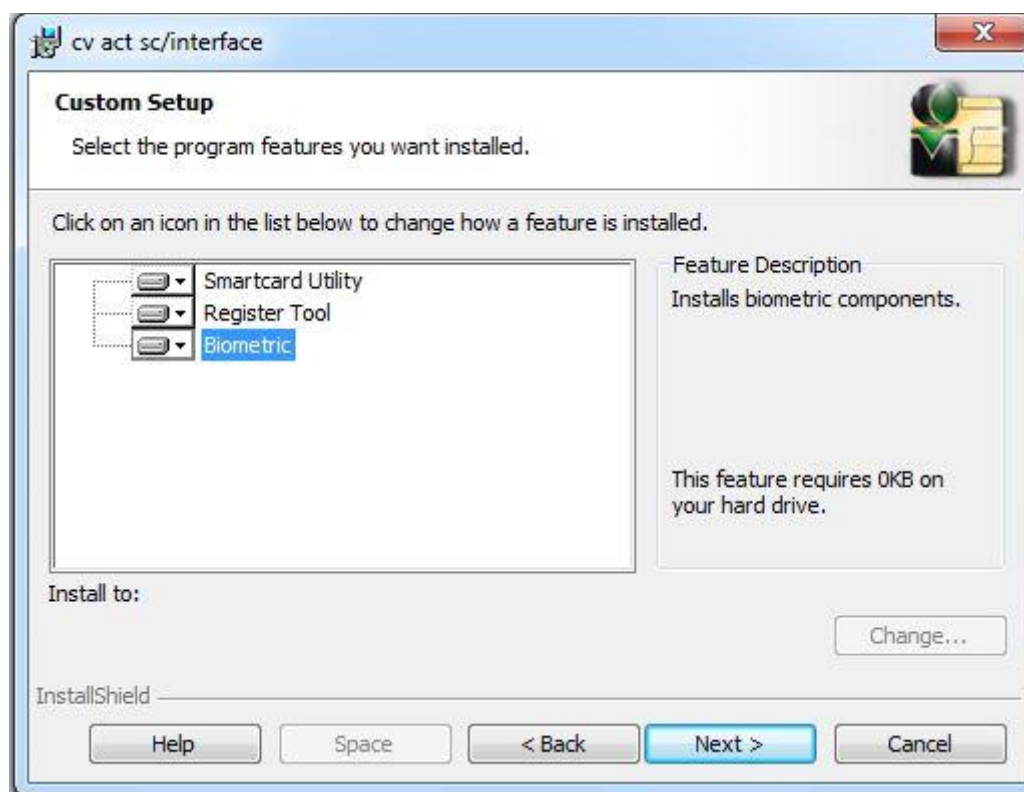
For applications that require enhanced functionality for the cryptographic service provider, like use of class 2 or 3 secure PIN pad card readers or extended PIN caching then cryptovision Cryptographic Service Provider is recommended. For legacy support of default installations of Windows XP, Windows 2000 or Windows Server 2003, it is required that you use the cryptovision Cryptographic Service Provider.

However, if you want to install the Minidriver on Windows XP, Windows 2000 or Windows Server 2003, you must first install the Microsoft Base Smart Card Cryptographic Service Provider Package: x86 (KB909520). You can download the relevant update for chip architecture at: <http://www.microsoft.com/en-us/download/search.aspx?q=KB909520>

If you intend on using cv act *sc/interface* with more than one type of security token or you plan on automating the distribution of the User module, then it is recommended that you chose the Smart Card Mini-driver Option as it will install the relevant minidriver files for all supported tokens. While this takes more disk space on the client, it ensures the maximum compatibility for systems supporting multiple types of tokens. For these systems where the distribution of the Minidriver is automated, it is recommended that the Windows “Smart Card Plug and Play” service be disabled to prevent Windows from attempting to re-install a driver on reinsertion of a smart card. This service can be disabled via Group Policy with the following example, in Windows 2008 Server:

Administrative Templates | Windows Components | Smart Card “Plug & Play smart card enabled services

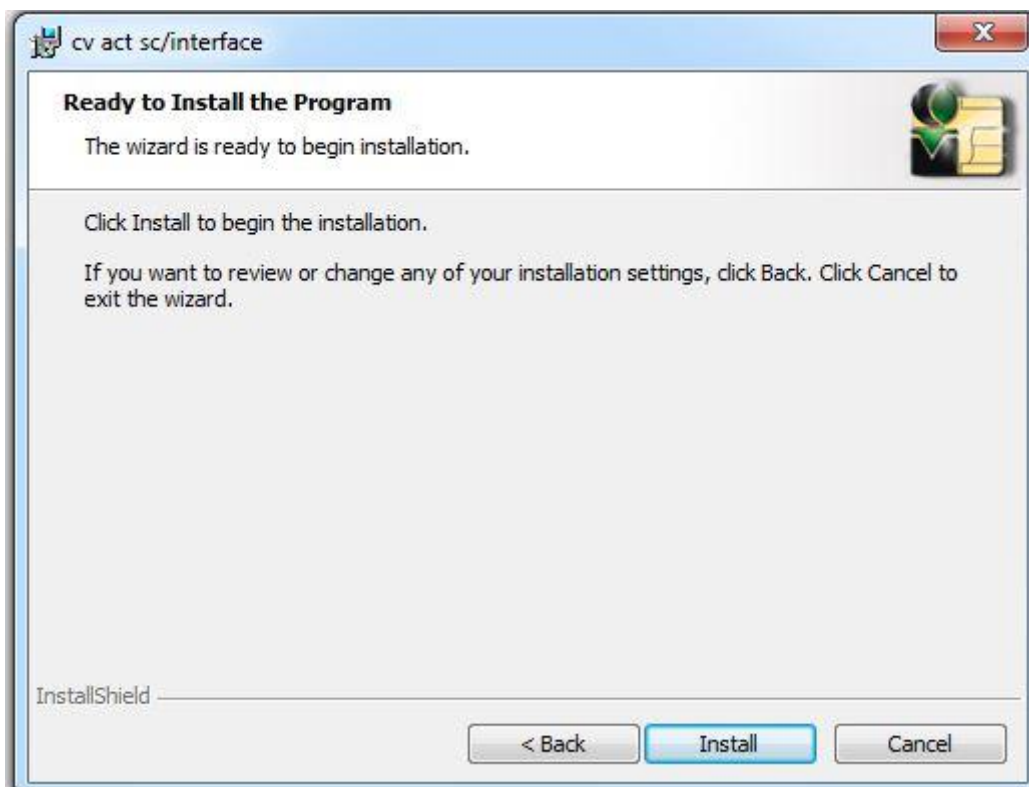
For a minimal installation, where only the token specific files are installed the Smart Card Minidriver – Device Driver Setup can be selected. However, since this method involves the automated download of token specific files, it requires online access to Windows Update.



If you purchase a license that includes the Precise Match-On-Card™ package, you can optionally include these files.



In addition to the middleware license agreement, you will need to acknowledge acceptance of the additional license terms additional biometric components which can be accepted at this screen.



Click Install to finalize the Setup Wizard configuration and begin the actual installation process.



Once the setup wizard has completed click on Finish to close the installation wizard.

6.1.3 Running the Windows User Setup

Start the file SETUP.EXE as a user with administrator rights. Follow the on screen installation instructions as guided by the setup wizard. Because the setup wizard files are based on the same installation procedure as the Administrator setup described in the preceding chapter with only a minor difference of installing the Smartcard Utility instead of the Manager Tool, the screen shots for this process have been omitted.

Please note: The installation of cv act *sc/interface* requires administrative rights. If you start the installation as a regular user, you will be prompted by the installation process for alternative credentials. For further details please refer to the respective operating system documentation or ask your system administrator.

6.2 Installing on Linux

Unlike the Windows procedure, the Linux installation is not based on administrative role but rather on the target distribution and chip architecture. In the cv act *sc/interface* installation media there is a Linux subfolder that contains different installation files for the different Linux derivatives.

6.2.1 Supported Linux Versions

The current cv act *sc/interface* installation media contains a .deb based installer file. Additional .rpm based install media can be produced upon request.

Here you find the Linux Distribution based on Linux (Kernel 2.6, 3.2) which is successfully tested with this release of cv act *sc/interface*. So this is supported by cv act *sc/interface*:

- Ubuntu 12.04.3 LTS (32/64 bit)

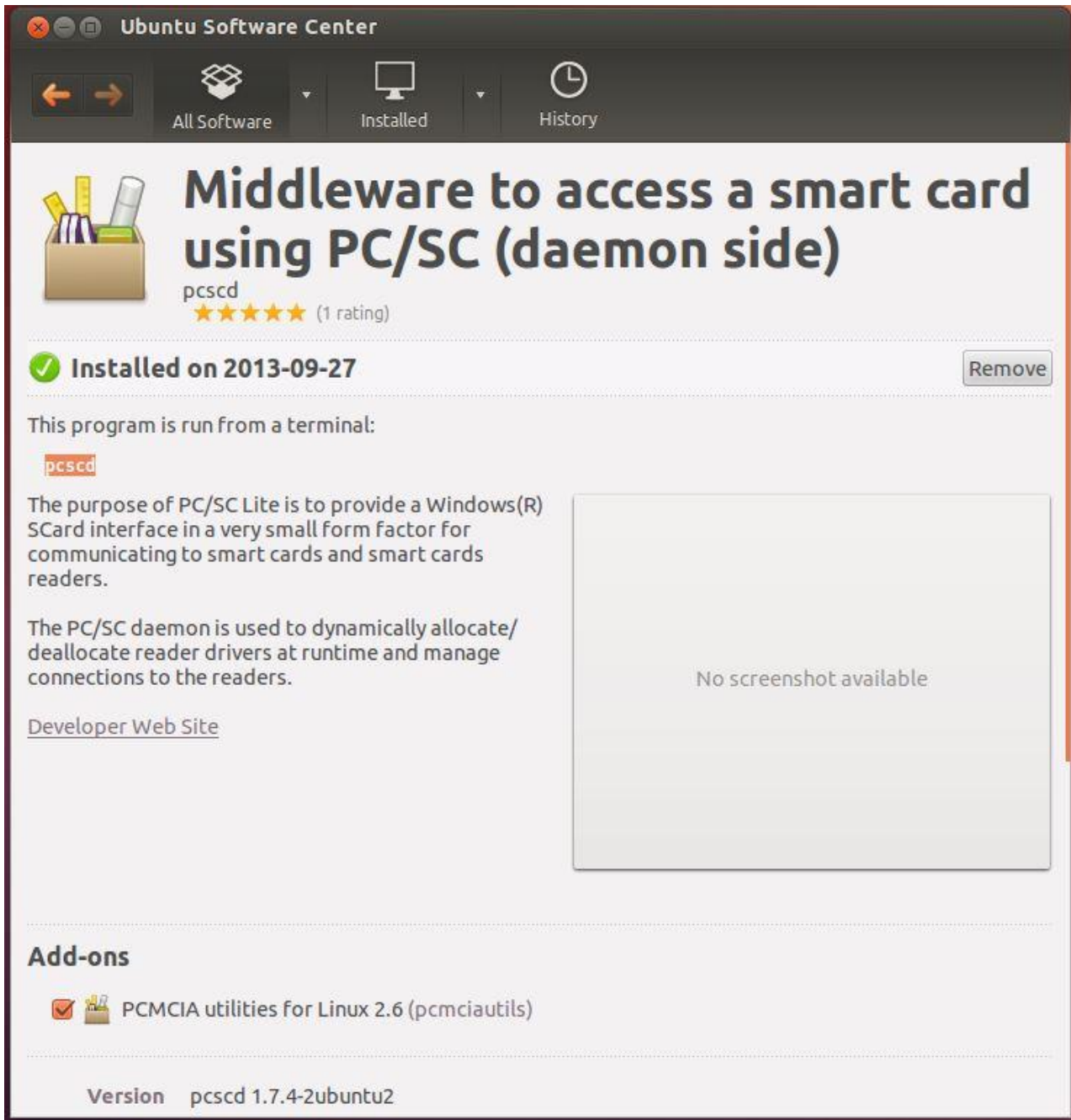
Here you find a list of the Linux Distributions based on Linux (Kernel 2.6, 3.2) which should work with cv act *sc/interface*, but which were not tested in this release. Additional testing of specific additional OS can be performed upon customer request:

- Ubuntu 10, 11, 12 LTE
- SLES 11 (OpenSuse 11.2) (64 bit)
- Fedora 15-19
- Debian 6, 7
- Red Hat Enterprise Linux 6

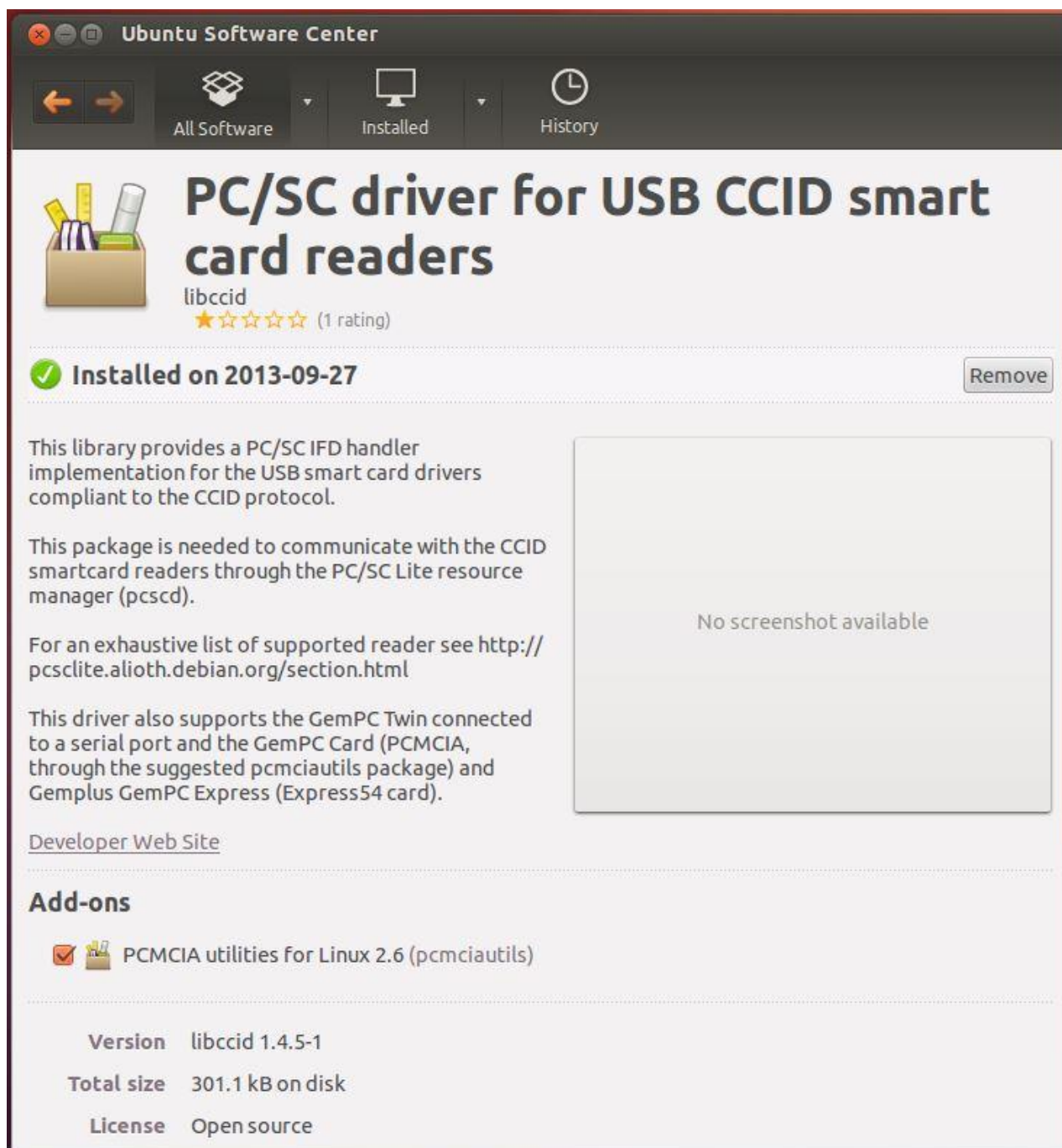
6.2.2 Sample Linux Distribution Installation: Ubuntu Setup

To begin the setup procedure, copy the relevant source installation files to the target workstation from the install media. Browse the installation media to the Linux subfolder which contains different installation files for the different Linux derivatives based on both target distribution and chip architecture. In this example, we will perform an installation of the cv act *sc/interface* Utility and cv act *sc/interface* Manager on Ubuntu Linux 12.4 LTS release.

Before the installation of cv act *sc/interface* can take place, you need to insure that the appropriate pre-requisites are installed. In the Ubuntu Software Center search for and install the pcscd package. This PC/SC daemon is needed for smart card reader drivers to be allocated at runtime thereby enabling connections to the reader.

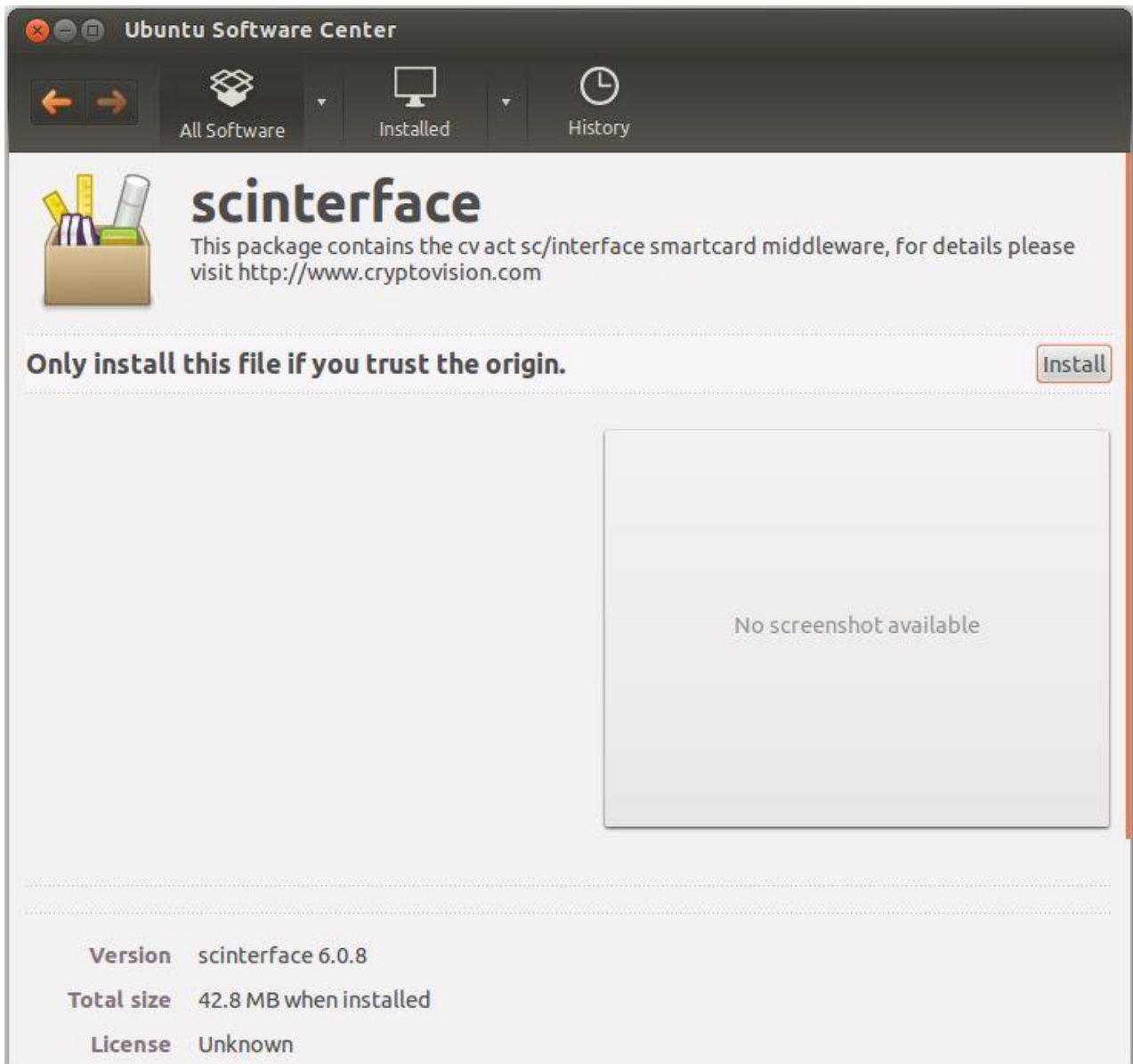


In addition to the PC/SC Lite installation, the CCID protocol compliant USB smart card drivers must also be installed. This libccid package can also be installed from the Ubuntu Software Center.

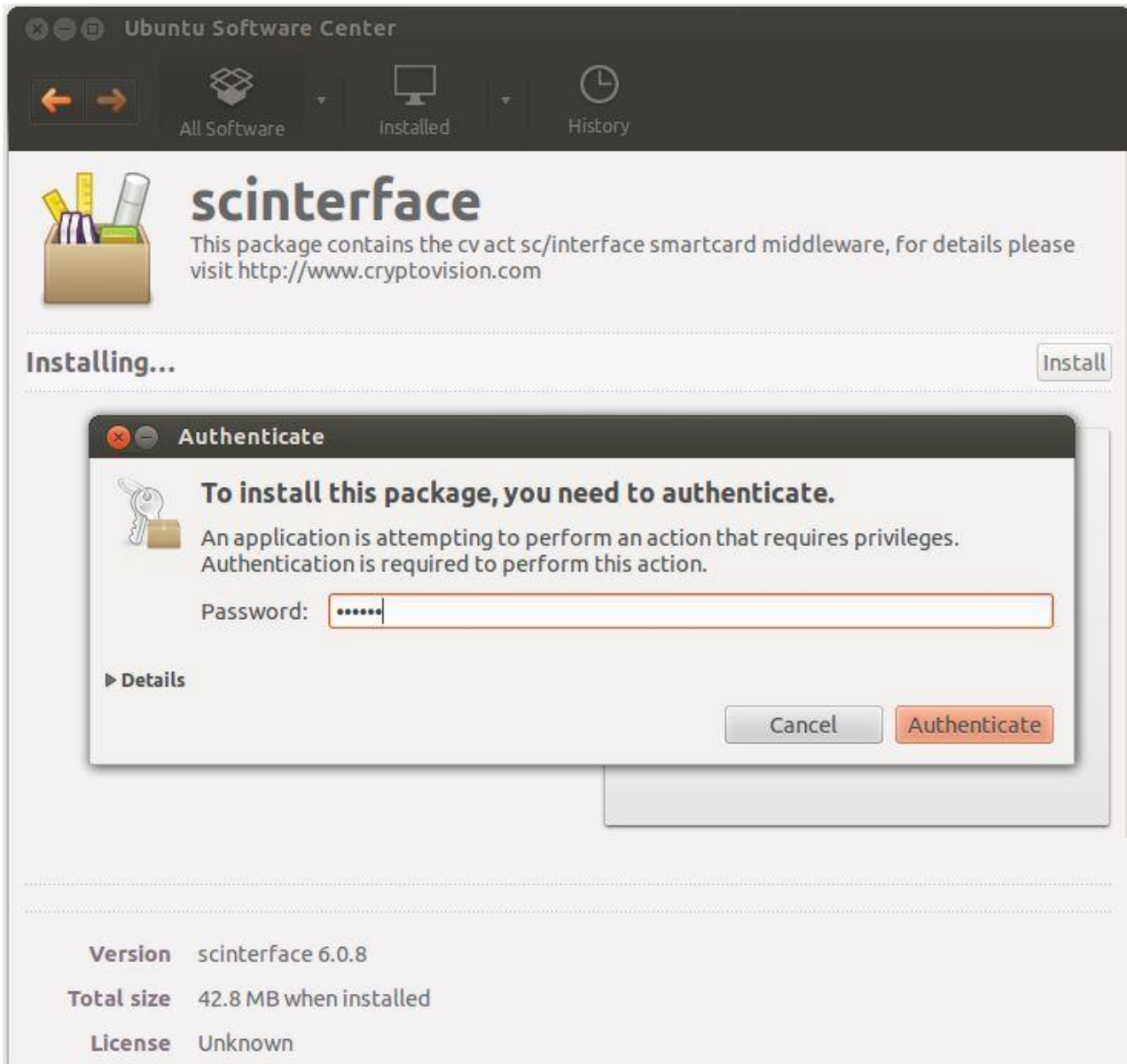


If the smart card reader you intend on using that is not listed as supported <http://pcsc-lite.alioth.debian.org/ccid/section.html> you will need to install the proprietary drivers for your specific smart card reader. Search the support section of the hardware vendor's website for the latest drivers and installation instructions.

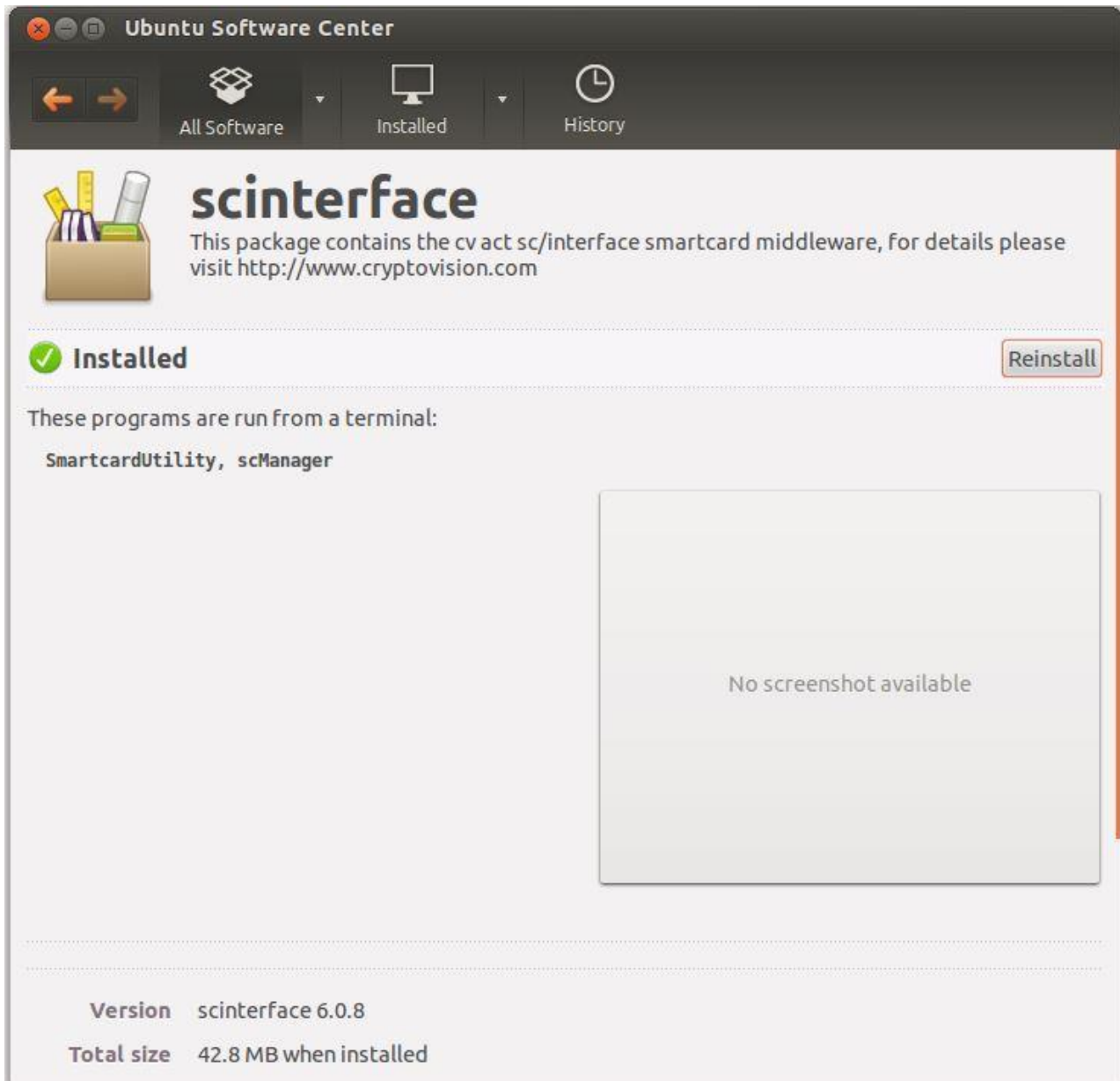
Once these prerequisites and suitable drivers are installed, installation of cv act *sc/interface* can begin. Copy the appropriate installation files from the linux subdirectory of the install media. In this example we will be installing on an Intel chip based Ubuntu machine so we will copy the *scInterface-6.0.x.x-Ubuntu12-i686.deb* file to the target client. Right click on this file and select the Open with Ubuntu Software Center option to begin the installation.



You will be prompted to authenticate with an account with appropriate privileges to install software. Enter a password and click Authenticate to proceed with the installation.



After the installation is complete you will receive acknowledgement that the process is finished and that the scManager SmartcardUtility interfaces are now available from a terminal console prompt.



6.3 Installing on OS X

Similar to the Linux installation, the OS X is not based on administrative role but rather on the target operating system version. In the cv act *sc/interface* installation media there is a macosx subfolder that contains different installation files for the different Mac OS X versions.

6.3.1 Supported OS X Versions

Here you find the OS X Versions which are successfully tested with this release of cv act *sc/interface*. So these are supported by cv act *sc/interface*:

- 10.7.x Lion (Intel 64 Bit)
- 10.8.x Mountain Lion (Intel 64 Bit)
- 10.9.x OS X Mavericks (Intel 64 Bit)

Here you find a list of the OS X Versions which should work with cv act *sc/interface*, but which were not tested in this release. Additional testing of specific additional OS can be performed upon customer request:

- 10.5.x Leopard (Intel 32Bit)
- 10.6.x Snow Leopard (Intel 32/64Bit)

Please note that since the Mac OS X 10.7 release, Apple no longer includes SmartCard Services components in the default installation. In order to use smart cards with requires the installation of additional SmartCard Services components which are be found at:

<http://smartcardservices.macosforge.org/trac/wiki/installers>

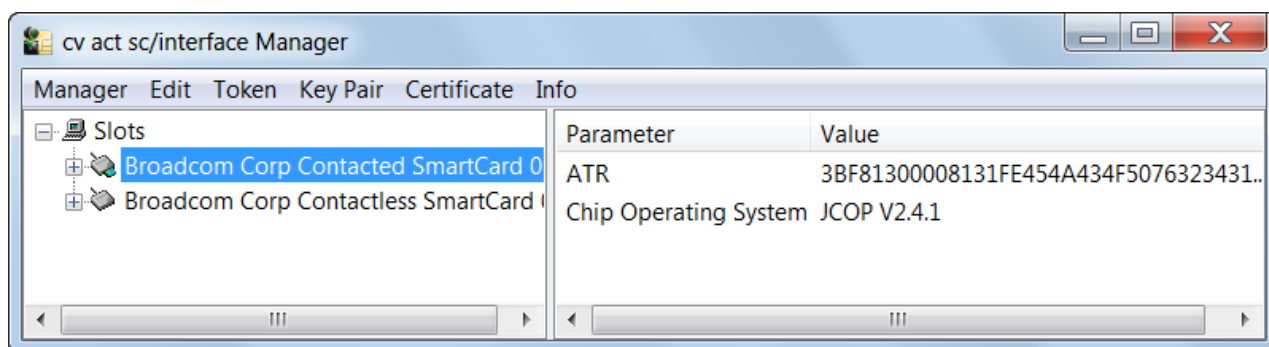
Please note that Apple do not support TokenD.

7 Using the Manager Tool for Administration

By installing the Administrator version of cv act *sc/interface*, you can perform all sorts of administration tasks like profile creation, PIN change, unlock smart cards, generation of keys and more. This chapter describes the Manager user interface first, then followed by a specification of the applications you can leverage with this Administration Tool.

7.1 cv act *sc/interface* Manager client interface

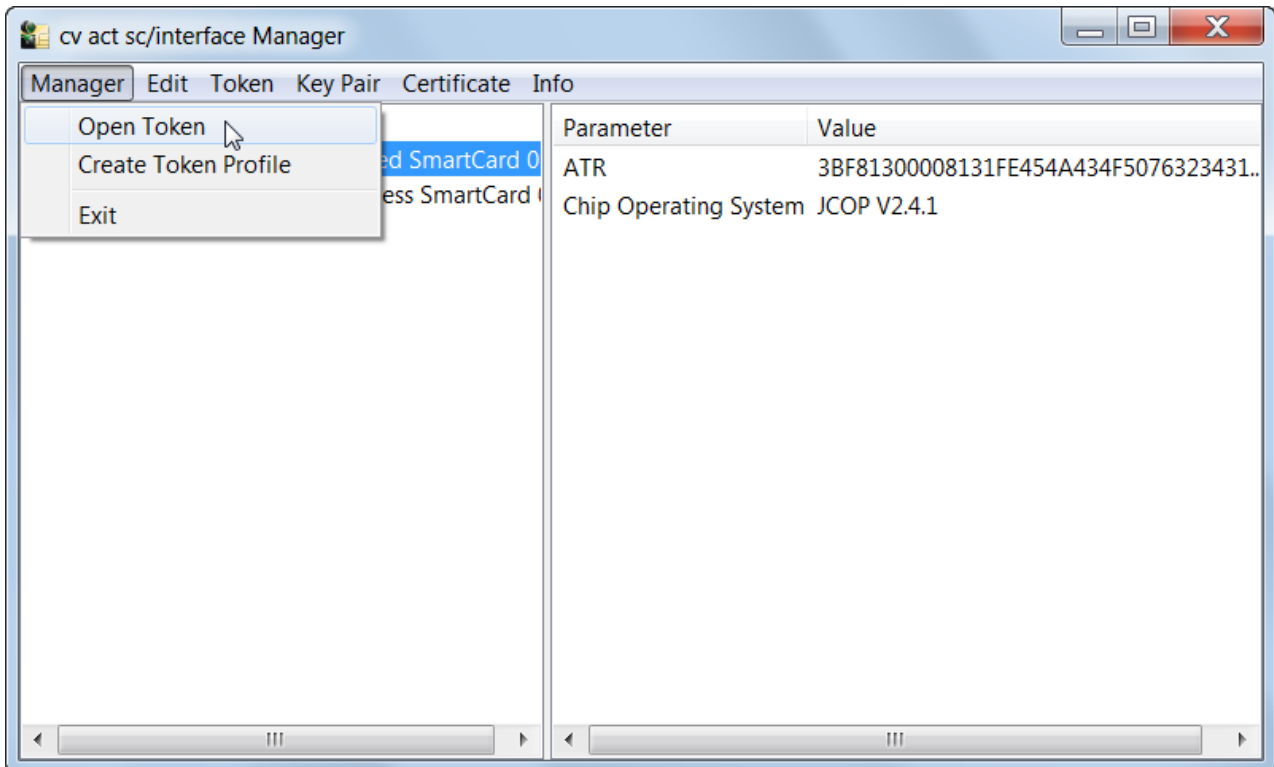
After opening the administration tool of cv act *sc/interface* you will see the following interface () depending on the reader):



Note: The administration tool supports the usage of CTRL+C for copying. In such a way e. g. the ATR or the Token Label can be copied.

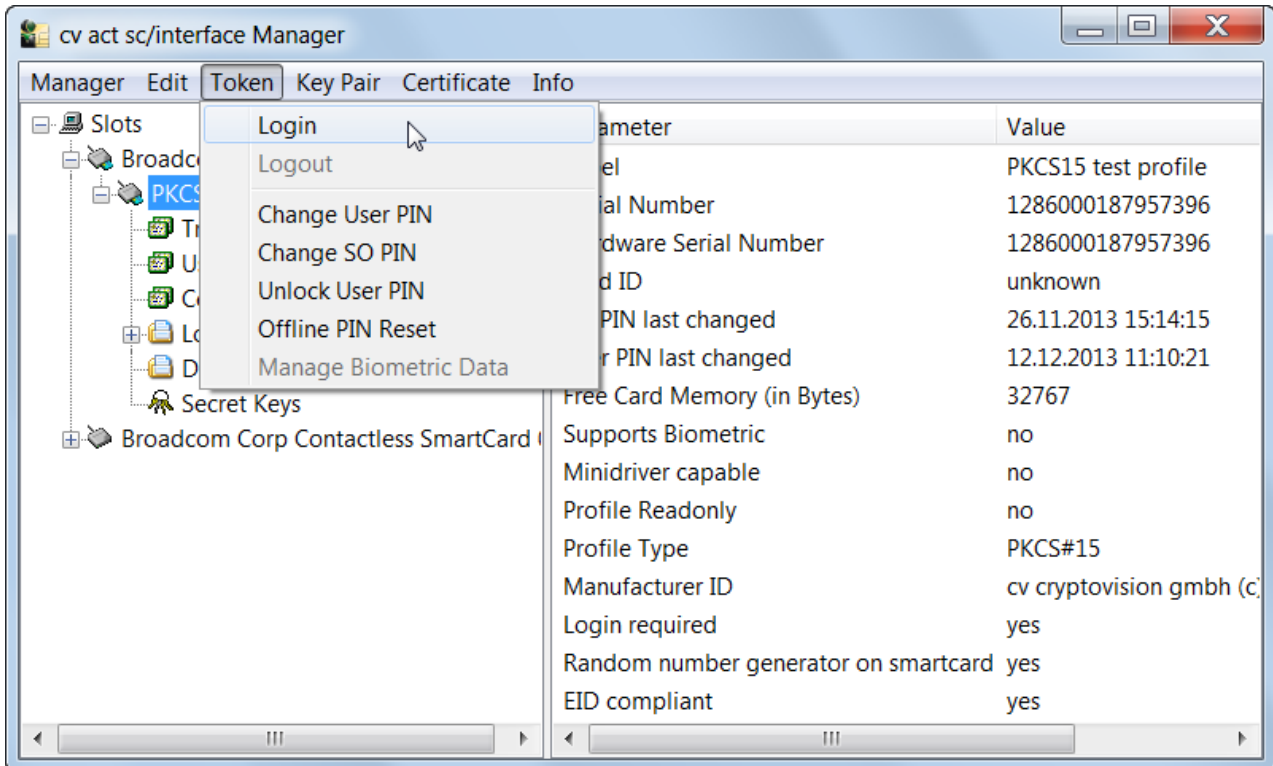
Note: You can connect or disconnect a card reader during operation of the administration tool. The availability of a reader is detected automatically.

After you insert a smart card into the card reader, you can select the smart card reader and "Open Token" with a right mouse click. Alternatively you have the possibility to select this function under the "Manager" menu.

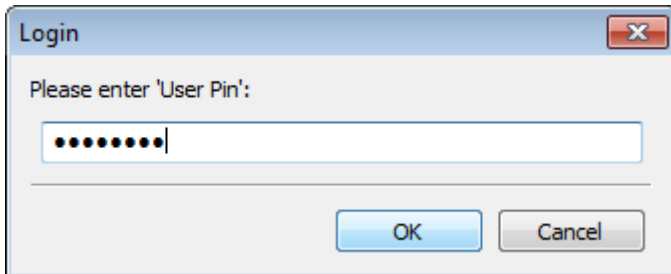


This option will display the public information, e.g. name of the smart card, the profile, and the free card memory. Furthermore, certificates, public keys, containers and data are displayed. In this case a smart card is displayed, which contains two applications and therefore two virtual slots are displayed.

To display sensitive data of the smart card, you must login on the card. For this, select the item "Login" in the "Token" menu:

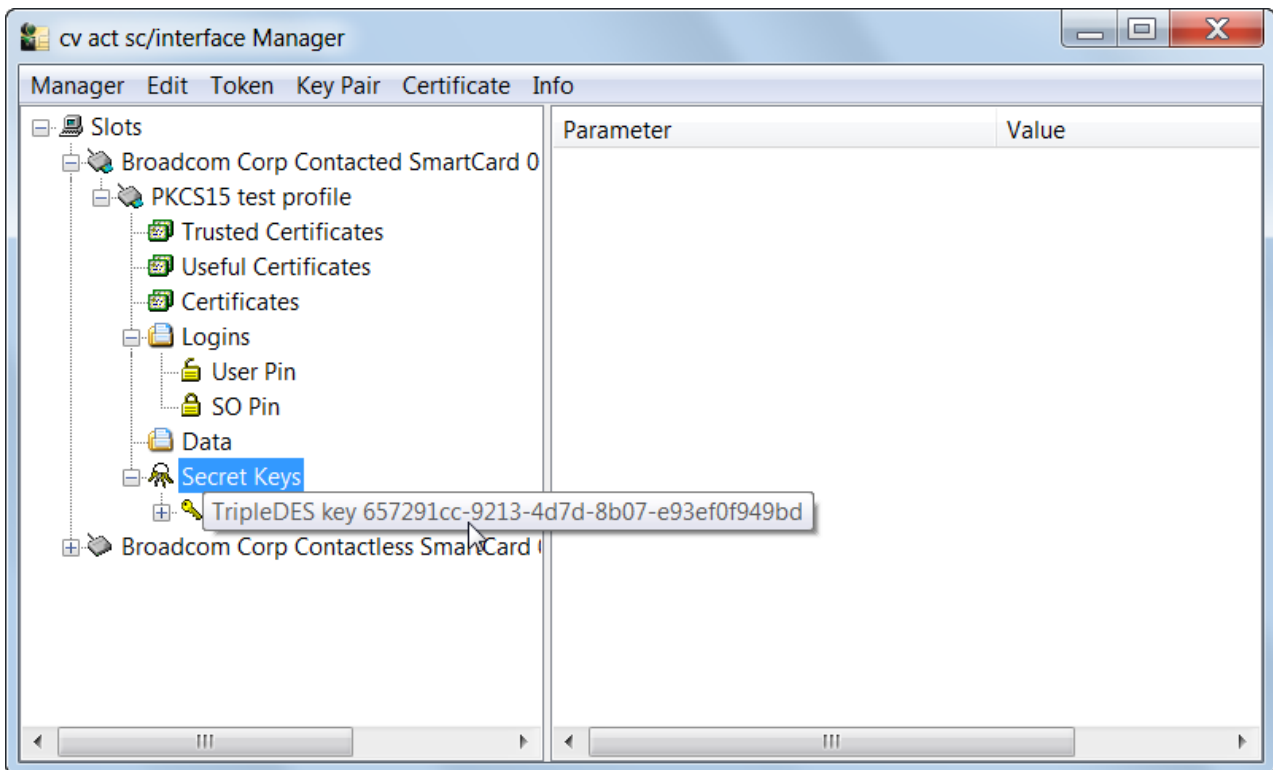


Now you are asked to insert your User PIN:



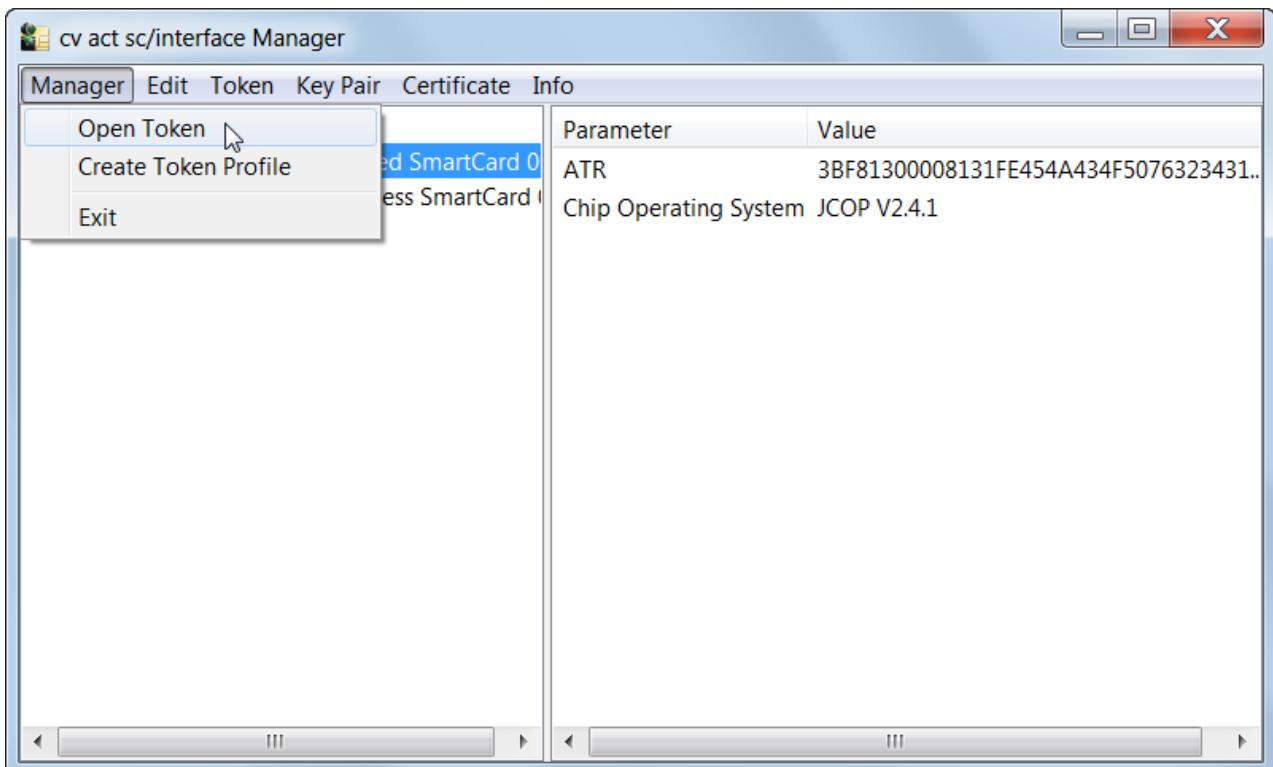
Here cv act *sc/interface* offers you the possibility to register the smart card based certificates in the Windows certificate store. The registration happens via dialogue; prompting for each certificate whether the registration of this certificate is desired. This dialog only appears if your certificates are not yet registered.

After login Private Keys and Secret Keys appear in the interface:



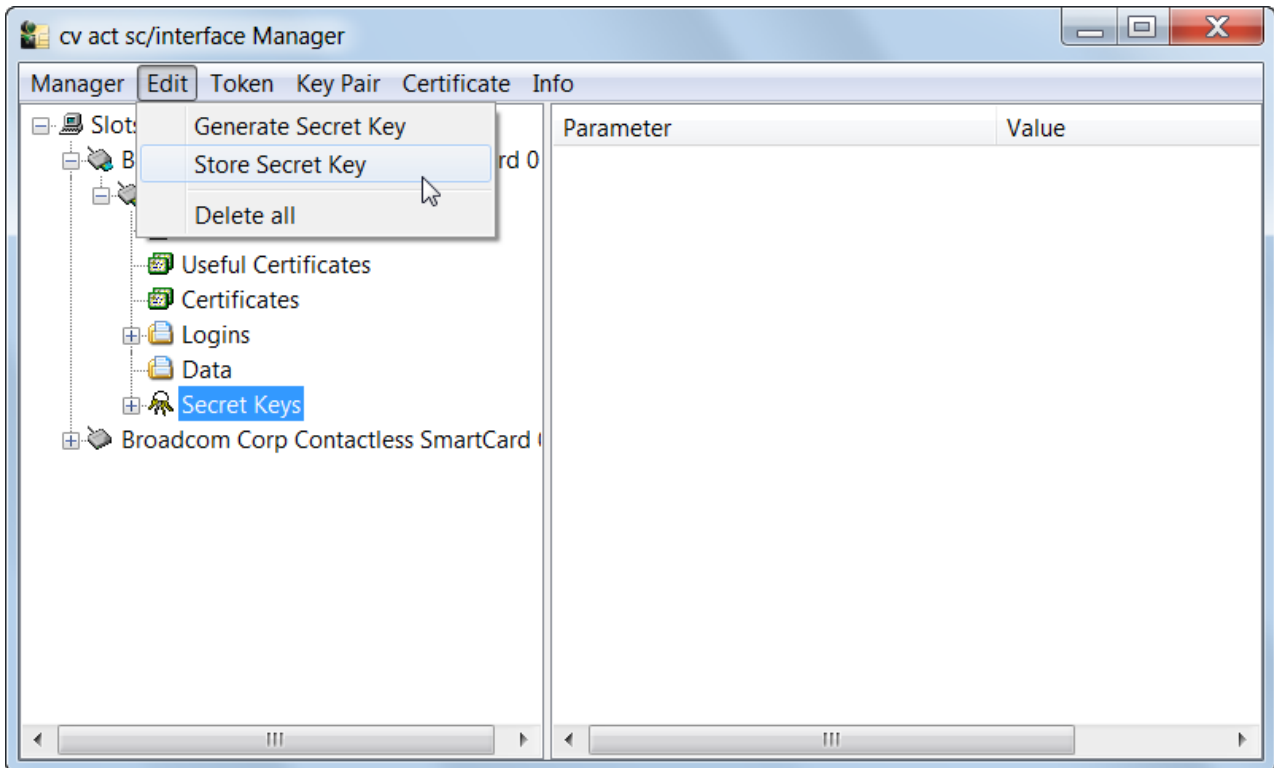
One can generate several keys with the corresponding certificates on the card. These will merge in the appropriate containers. The functions available will be described in the following sections in more detail.

7.1.1 "Manager" Menu



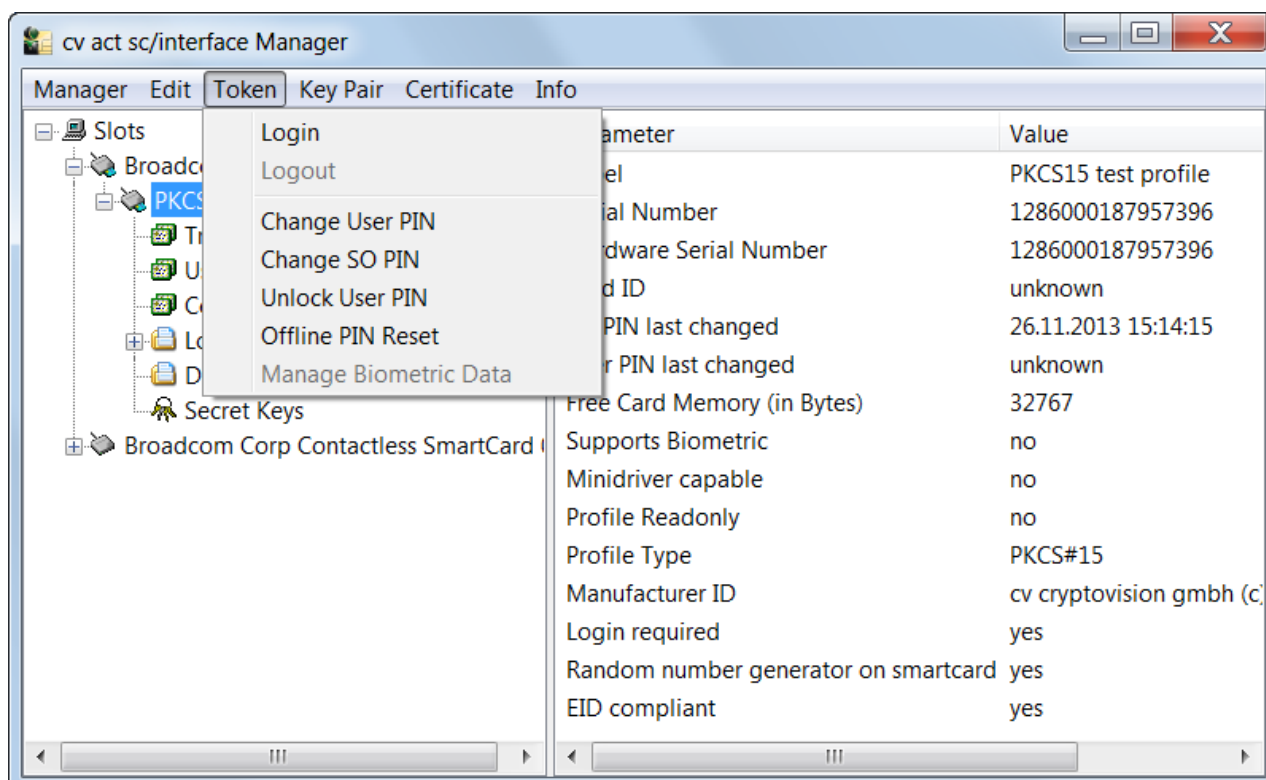
From this menu you access general functions for operating this tool. You can open a token, Create a token profile or close the cv act *sc/interface* Manager.

7.1.2 "Edit" Menu



From this menu you can call specific functions the smart card provides.

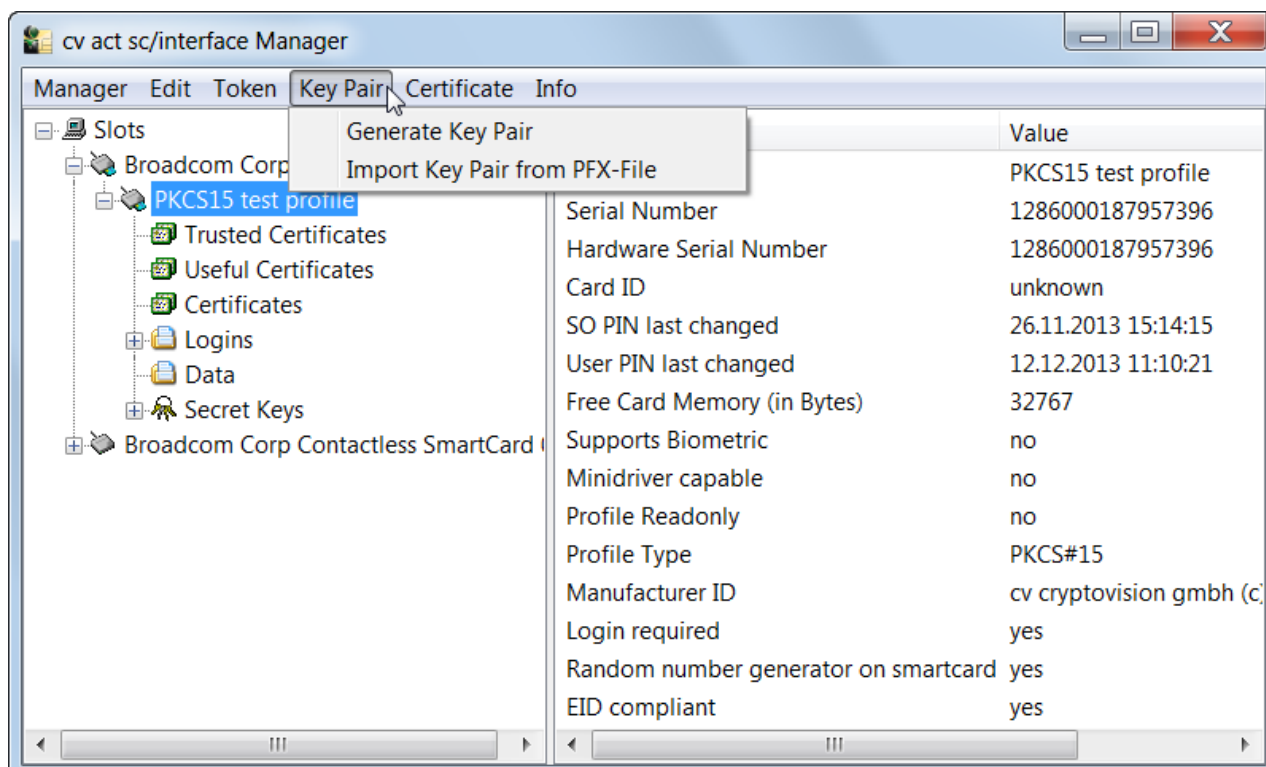
7.1.3 "Token" Menu



From this menu you can use the functions regarding the token itself.

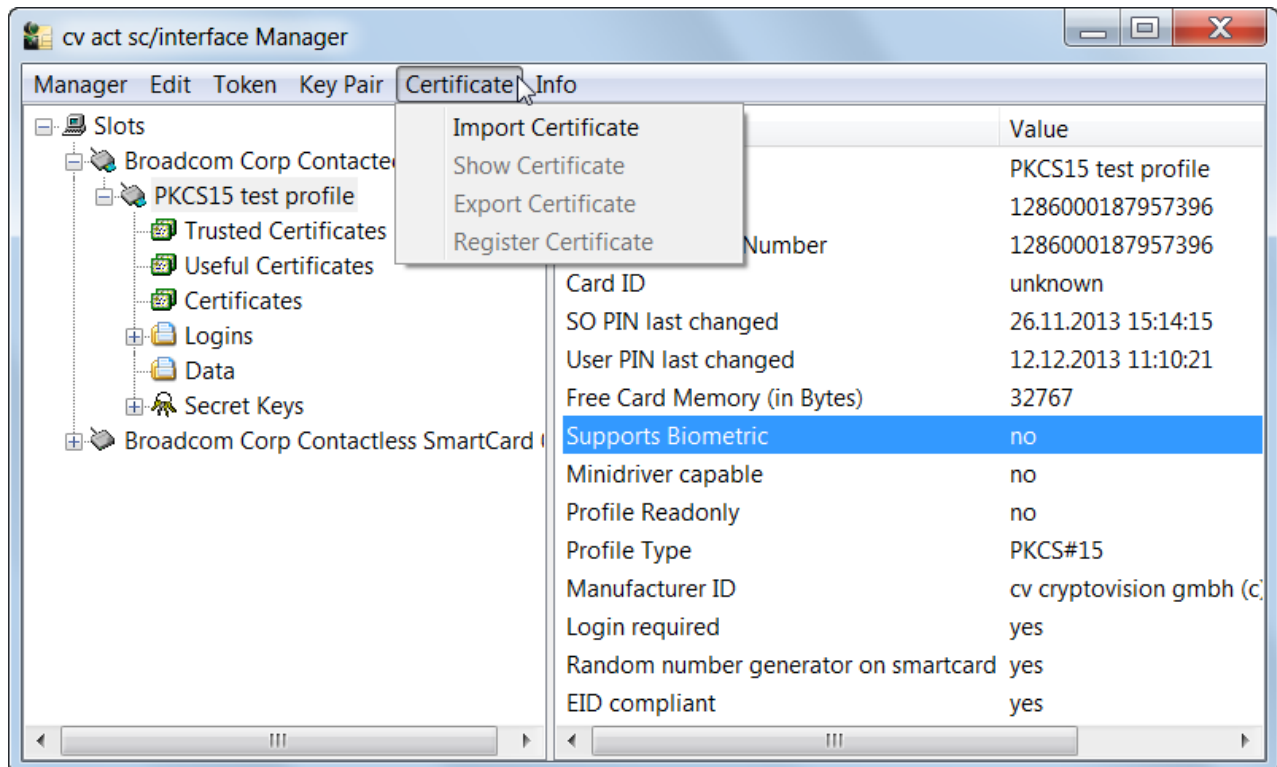
User-PIN and SO-PIN are stored per virtual slot, this is why "unlocking smart cards" equates to "unlocking of a slot" in case of smart cards with multiple applications.

7.1.4 "Key Pair" Menu



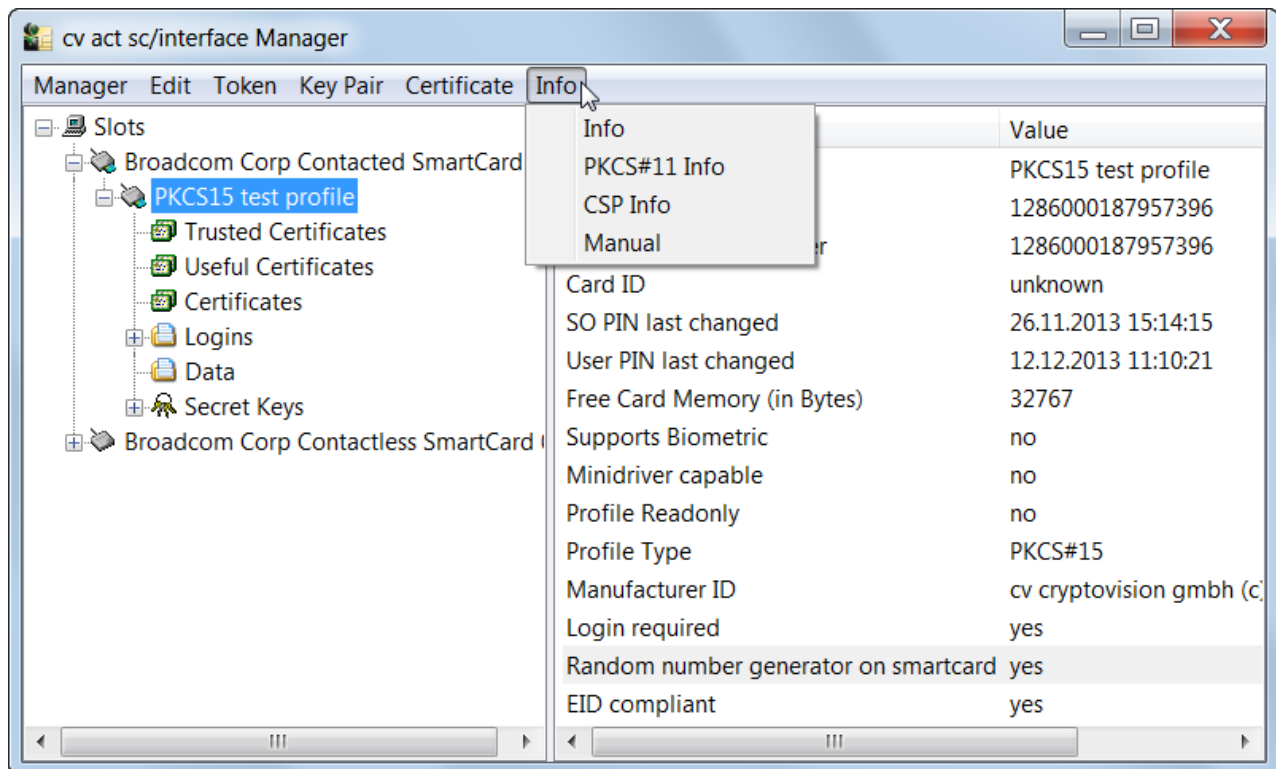
This menu provides functions regarding the key pair on inserted smart card including key pair generation and import of a key pair from a password protected file.

7.1.5 "Certificate" Menu



This menu provides access to functions specific to working with the certificates on the smart card.

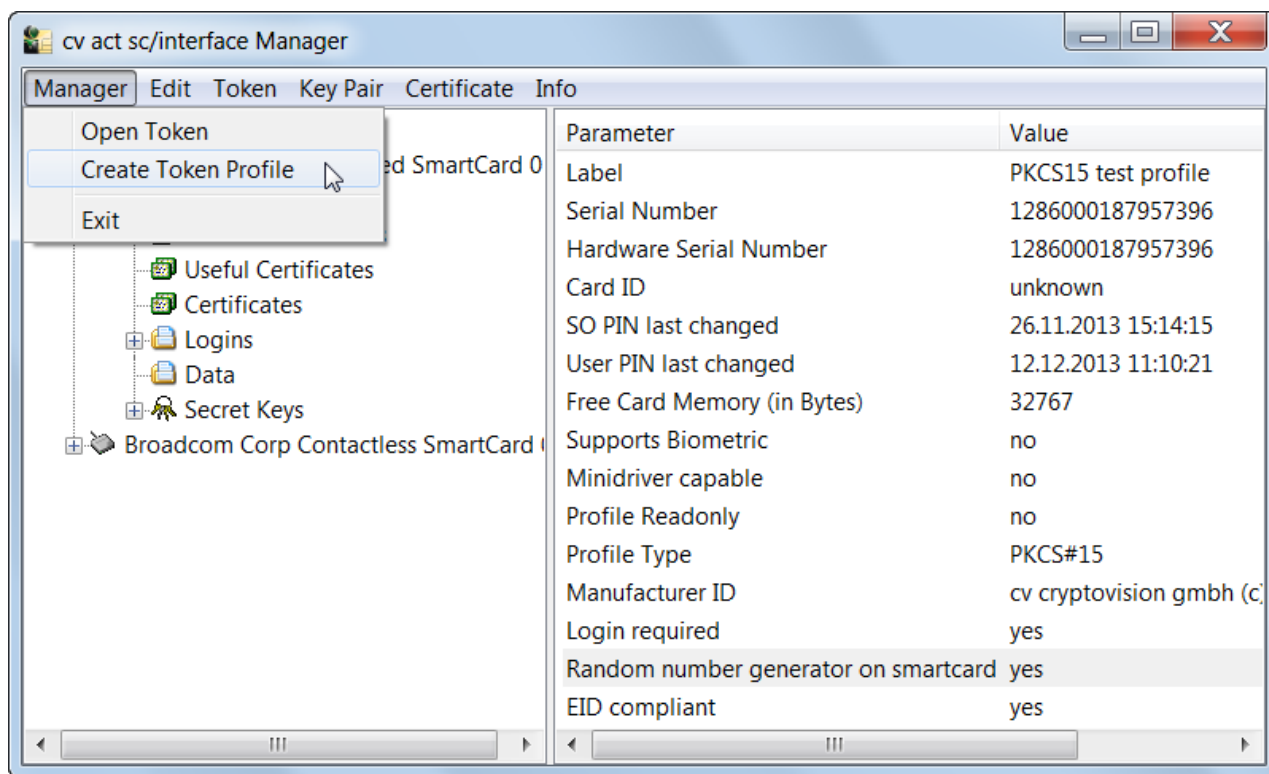
7.1.6 "Info" Menu



Selecting the "Info" menu allows you to obtain information about the version of the modules of cv act *sc/interface* and about the producer cv cryptovision GmbH as well as an online link to this manual.

7.2 Creating Profiles

In order to prepare a smart card for use, a profile must be created on the smart card. These profiles can be setup with the "Manager" menu item "Create Token Profile". Both PKCS#15 and cv profiles can be selected by default. If a smart card has an appropriate biometrics applet loaded, a PKCS#15 biometric profile can also be created.

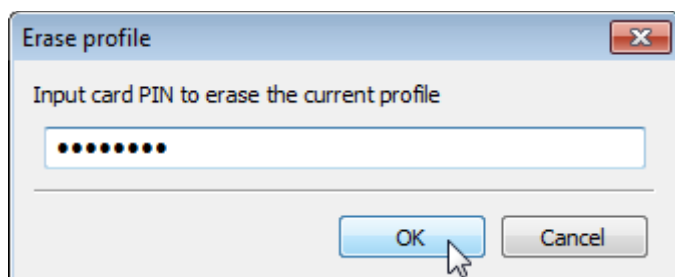


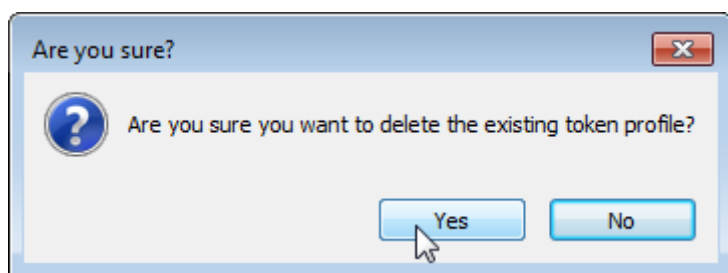
7.2.1 Smart Cards with existing profiles

If there is already a profile on the card and you want to create a new one, the existing one will be deleted as a first step.

Depending on the chosen profile which exists on the card please proceed as follows:

- Enter the assigned card PIN if you have created the card profile
- Enter the default card PIN (0987654321) if there is a Siemens profile on the smart card
- Enter the default card PIN (87654321) if there is a StarCOS profile on the smart card
- For all other profiles, please read the instructions of the vendor or the PINs are either not necessary or the profile cannot be deleted.
- If there is a Nexus profile on the smart card, the profile cannot be deleted.





The remaining steps follow the guidelines described in "in case of an empty smart card".

7.2.2 In the case of an empty smart card

During setup of a profile the following parameter must be defined (Token Label and User PIN are set to default values):

- Profiles (supported profiles are displayed, e.g. "PKCS#15 profile" and "cv profile")
- Token Label (the default value depends on the type of the smart card)
- Card PIN
- SO-PIN
- User-PIN (default value "11111111", eight times one)
- Serial Number
- Challenge Response PIN (please observe chapter "Minidriver")
- Minidriver compatible

7.2.3 Supported PIN-length

The following minimum and maximum PIN-lengths are available during initialization of a smart card by the administration tool:

	User	SO	Admin/Card
ACOS	4/8	8/8	8/8
CardOS	4/10	8/10	10/10
JavaCard	4/10	8/10	10/10 (only cvProfile)
StarCOS	4/8	8/8	8/8

7.2.4 Default values

These fields from the screenshot below are preconfigured with certain values:

[illegible]

This pre-configuration is subject to change. There are two options for this configuration:

- `scManager.ini`:

This file is part of cv act *sc/interface* and has to be modified. The file has to be stored in the same folder as scManager.exe

- Configuration via Registry:

The following registry keys can be used. They are subject to modification according to special requirements:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv_cryptovision\sc interface]
```

```
[HKEY LOCAL MACHINE\SOFTWARE\cv cryptovision\sc interface\keys]
```

```
"crkey"="0000000000000000000000000000000000000000000000000000000"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface\profile]
```

```
"userpin"="11111111"
```

```
"cardpin"="sopin"
```

```
"#cardpin"="0987654321"
```

```
"remindpinchange"="true"
```

```
"minidriver"="true"
```

```
"usehwsnr"=dword:00000001
```

Configurations set via registry override the file `scManager.ini`.

In case of modification, there are some aspects that must be observed:

1. The Challenge Response PIN (crkey) is a two-key TripleDES (ABA) in case of ACOS smart cards. For all other smart cards it is three-key (ABC) TripleDES. In both cases the key consists of three times eight hex-bytes, for ACOS the first and the last eight bytes have to be the same.

2. User PIN, Card PIN and SO PIN have to be chosen within the smart card specific boundaries.
3. If no SO-PIN or no Card-PIN is defined in the file, the respective fields remain empty.
 - If no User-PIN is defined in the file, the default User-PIN is used (see above table).
 - If the option "use HW serial number" is set to true, the checkbox (see next figure) is activated and the field for entering the serial number is deactivated.
4. In case of "Remind PIN change" the value false can be selected. In this case there is no warning by the register tool after creation of the profile if the user doesn't change the User-PIN.
5. Starting with cv act *sc/interface* 4.0.1 a profile can be created on smart cards with Java-operating systems which don't use Visa Fixed Keys. In such case the relevant keys have to be included in the file scmanager.ini. One set of keys has to look like this:

```
[javacard]
# VISA-Fixed Keyset
#      enc                                mac                                dek
Keyset=404142434445464748494a4b4c4d4e4f,404142434445464748494a4b4c4d4e4f,404142434445464748494a4b4c4d4e4f
keyset=...
keyset=...
```

Create Token Profile

Profile: PKCS#15 profile

Token Label: Test Profile

Card PIN:

SO PIN:

Confirm SO PIN:

User PIN:

Confirm User PIN:

Serial Number: ☐ Use Hardware SN

Challenge Response PIN: ☒ 11111111111111111111

Minidriver compatible: ☐

Session PIN support: ☒

The Card PIN is defined to consist of 10.

- ✓ The SO-PIN has to consist of at least 4.
- ✓ The SO-PIN shall not exceed 10.
- ✓ The SO-PIN was correctly verified.
- ✓ The user PIN has to consist of at least 4.
- ✓ The user PIN shall not exceed 10.
- ✓ The user PIN was correctly verified.

The serial number shall have not more than 16 and at least one alpha-numeric digits.

- ✓ The challenge response PIN must have exactly 48 hexadecimal digits.

OK Cancel

The Card PIN and the SO PIN are important PINs which enable administrative security functions. By using the Card PIN a smart card can be deleted; with the SO PIN the smart card can be unlocked. Therefore you should not assign "simple" PINs. The input of the SO PIN is **also not displayed** in plain text, but through * in the input mask. The input must be also confirmed by entering the SO PIN a second time.

Altogether, there are 3 PINs on the card (Card-PIN, SO PIN and User PIN). After initialization the User PIN is "11111111" (8 ones). These should be changed by the administrator or the user.

The usage of the hardware serial number as serial number of the smart card is configured by checking the relevant checkbox.

Applications and services requiring access to the smart card, for example, a browser which will be used to enroll key pair and certificate to the smart card, must be started after the profile is fully created.

7.2.5 PKCS#15 smart card Profiles of other Vendors

cv act *sc/interface* supports existing ISO7816 based PKCS#15 smart card profiles.

There is a difference between ReadWrite and ReadOnly access.

1. ReadWrite
 - Full functionality
 - Imported certificates are stored in "Certificates"
2. ReadOnly
 - Write access is prohibited

Depending on the smart cards and vendors the following points are important:

3. Siemens HiPath from version 1.6.2.1
 - ReadWrite in CardOS V series
 - ReadOnly in CardOS M series
 - Attention: Key pairs generated or imported by cv act *sc/interface* will not be displayed by the middleware Siemens HiPath from Version 1.6.2.1
4. A.E.T. SafeSign from version 2.3.0
 - ReadOnly
5. A.E.T. SafeSign from Version 2.3.0 with StarCOS
 - ReadWrite
6. G & D StarSign 1.0
 - ReadWrite
7. Nexus Personal from version 4.6.1
 - ReadOnly

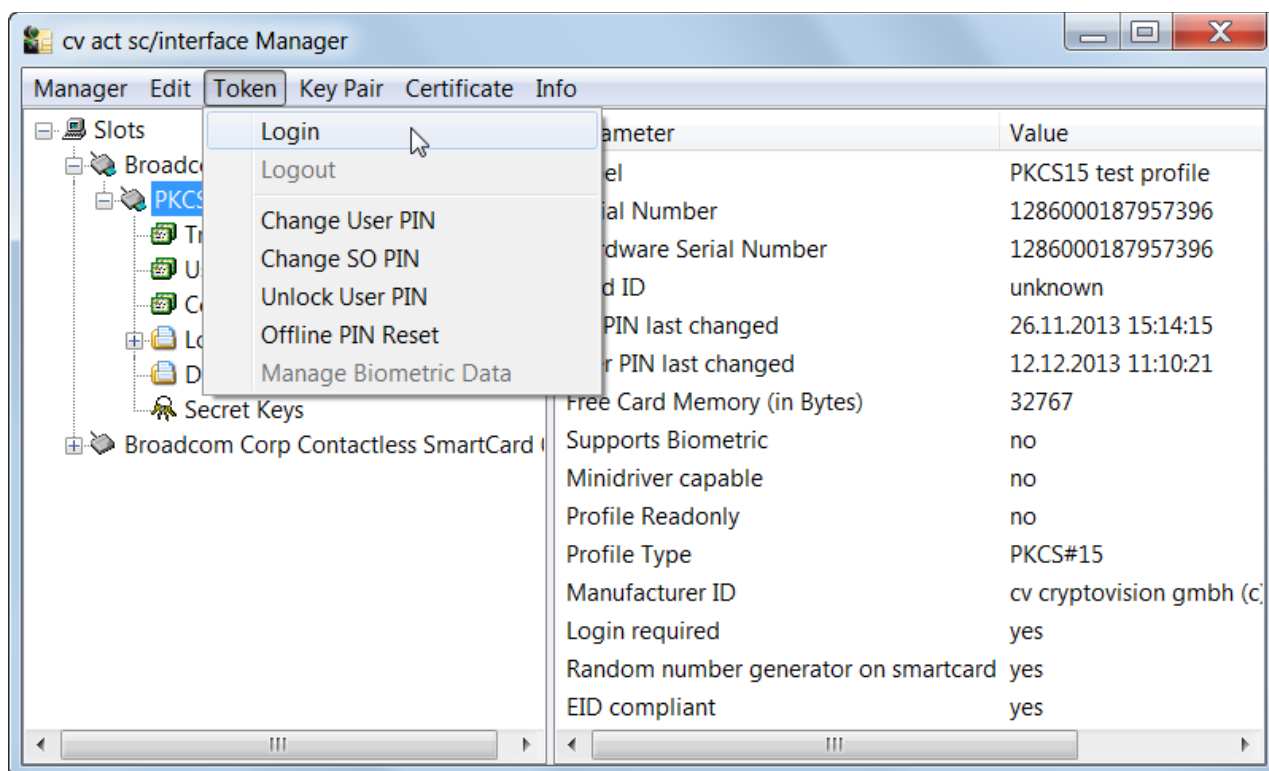
7.3 Generating and Importing Keys

To use the smart card for digital signatures or encryption, you need a key pair which is composed of a private and a public key. The private key must be stored very secretly and the public key must be accessible to communication partners by a certificate. These keys and certificates can be generated and managed by the administration tool.

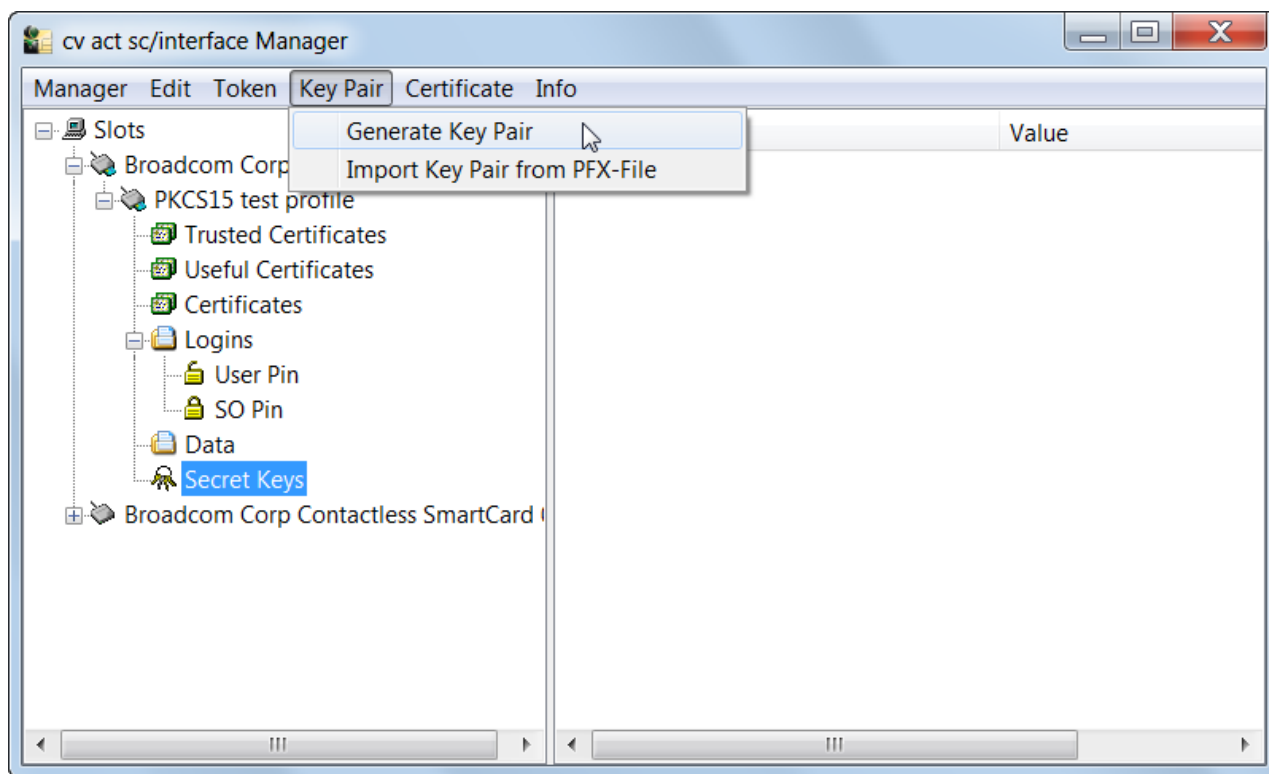
In principle there are several possibilities:

1. You can generate keys pairs (keys comprising private and public keys and secret keys) with the administration tool of cv act *sc/interface*.
2. You can generate keys by using the PKCS#11 Module, the CSP, or the Minidriver. Both are included in cv act *sc/interface*. The relevant procedures are described more closely in chapter 13, 14 or 15.
3. You already own a key and/or key pair. Then, you can import the key pair if necessary together with certificate as a PFX-file. You can store Secret Keys by importing them e.g. with "Copy and Paste".

To perform this function, you must first login to the smart card. Select from the menu "Token"->"Login".

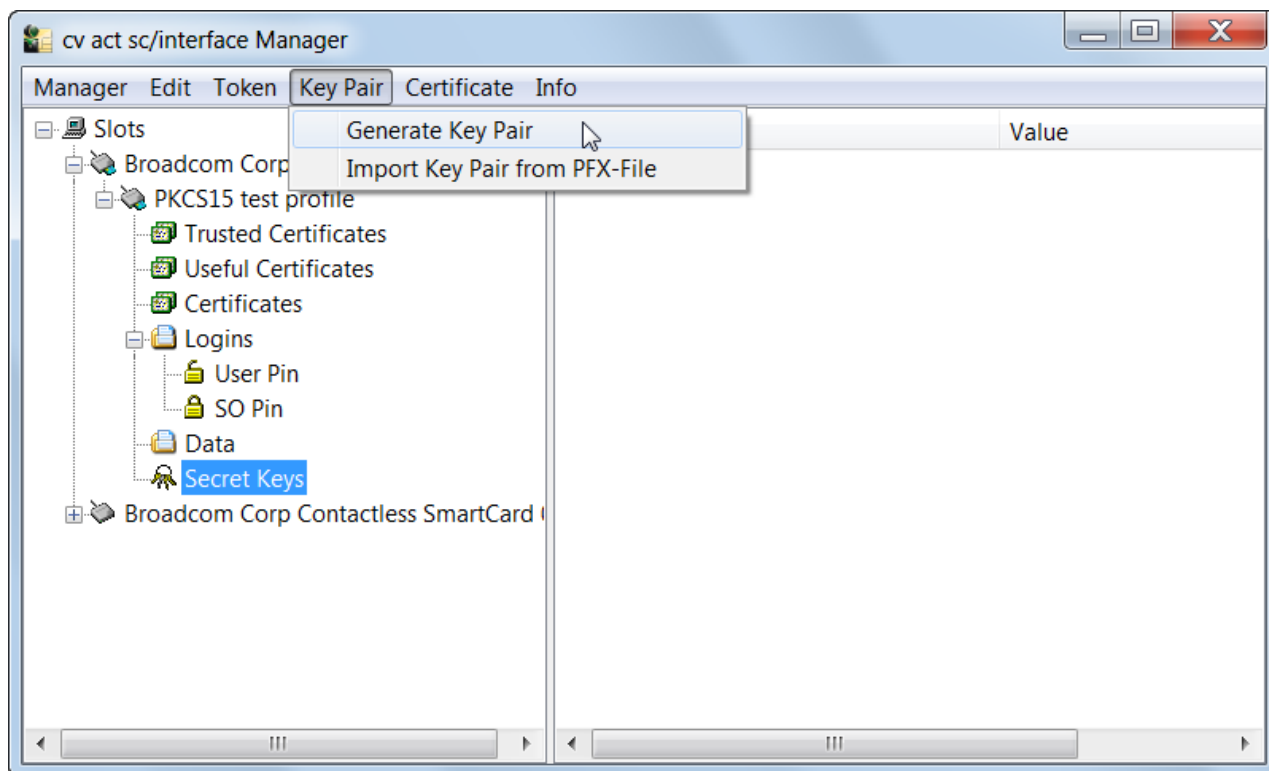


You can find all the functions to generate and import keys in the "Key Pair" menu and to import certificates in the "Certificates" menu, as shown in the following figure:



7.3.1 Generation of a Key Pair

The generation of a key pair (private and public key) can be performed with the menu item "Generate Key Pair." Upon successful completion, these keys can be viewed in corresponding container.



The available RSA key length depends on the smart card or the operating system, e.g.:

- G&D Sm@rtCafé Expert 3.1, G&D Sm@rtCafé Expert3.2 on StarSign Card Token 550 (USB), G&D Sm@rtCafé Expert64:

512, 1024, 1536 or 2048 bit

- CardOS M4.01, M4.01a or V4.20 smart card:

512 or 1024 bit

- CardOS V4.30 or V4.20 with additional package:

512, 1024, 1536 or 2048 bit

To generate RSA keys with 2048 bit on a CardOS V4.20 smart card you need a package that handles RSA keys with 2048 bit.

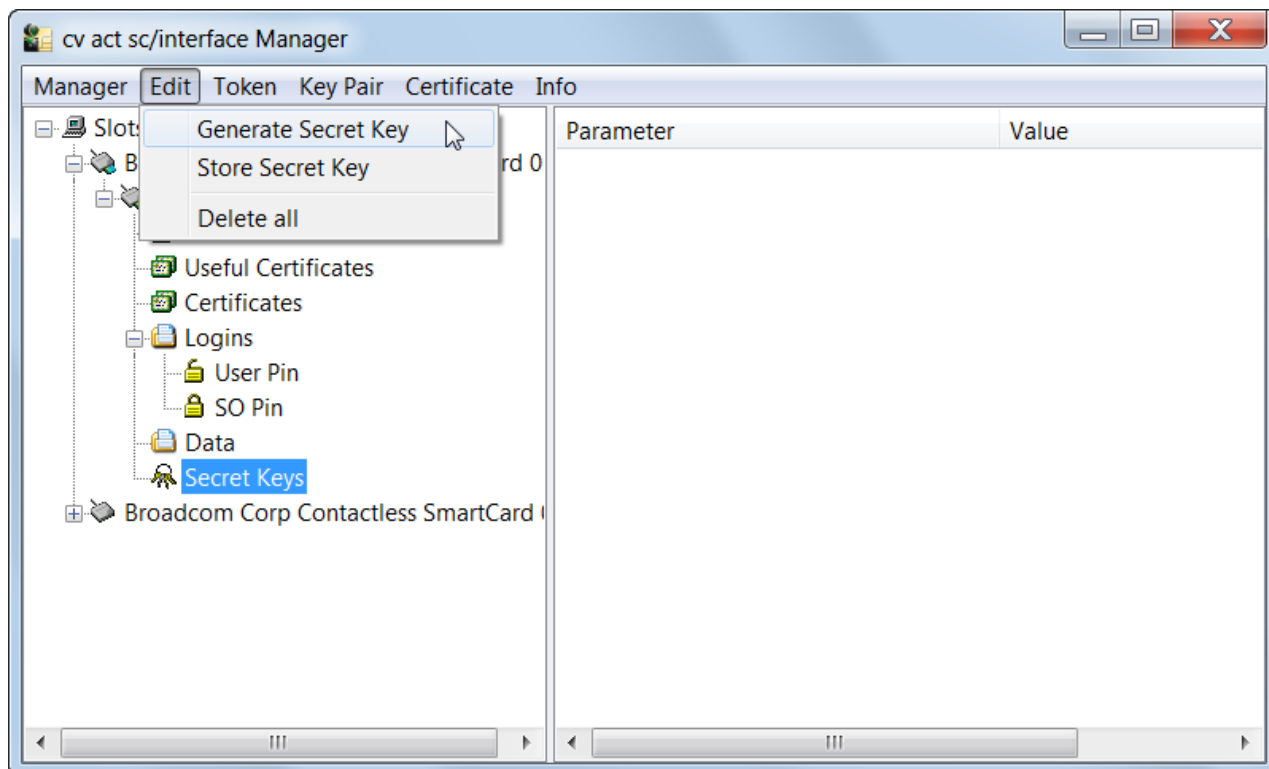
7.3.2 Importing a Key Pair

If you already own the key pair that you intend to employ, you can import it from the "Key Pair" menu with the item "Import Key Pair from PFX-File". This will prompt you for the path to the PFX file and the corresponding password used to secure the file.

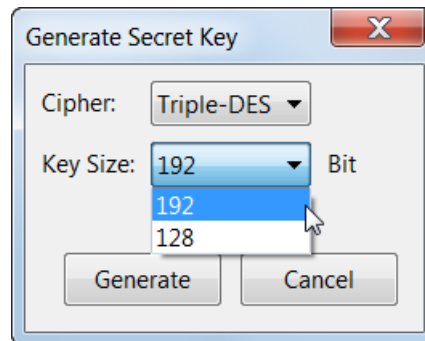
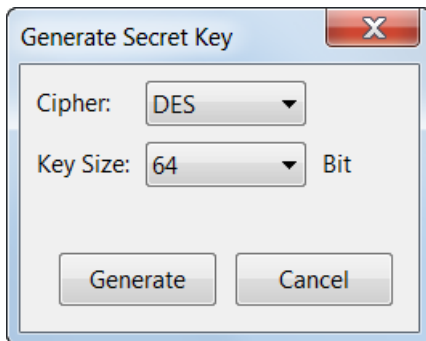
Note: The allowed key length for import of key pairs has to follow the same rules that are described above for the generation of a key pair. The password protected key file must be an RSA-key as pfx- or p12-file.

7.3.3 Generation of a Secret Key

To generate a secret key for encryption, select "Secret Keys" and select the item "Generate Secret Key" in the "Edit" menu.



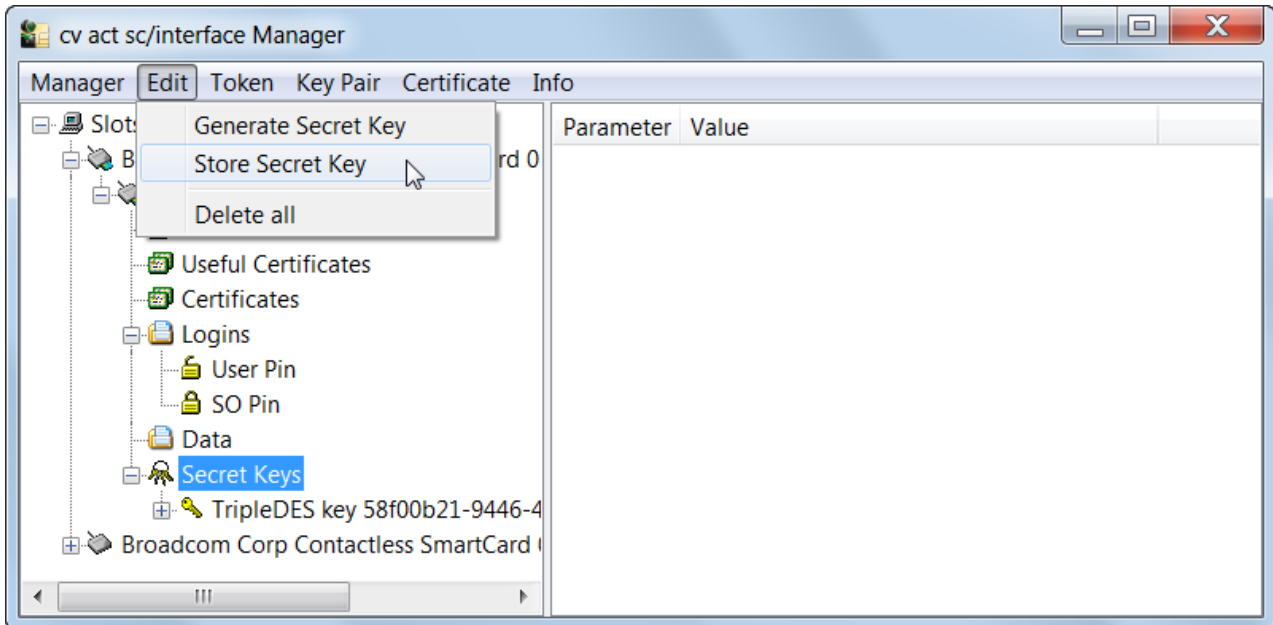
Here, you can generate a Triple-DES-key with 192 bits, a Triple-DES-key with 128 bits or a DES-key with 64 bits.



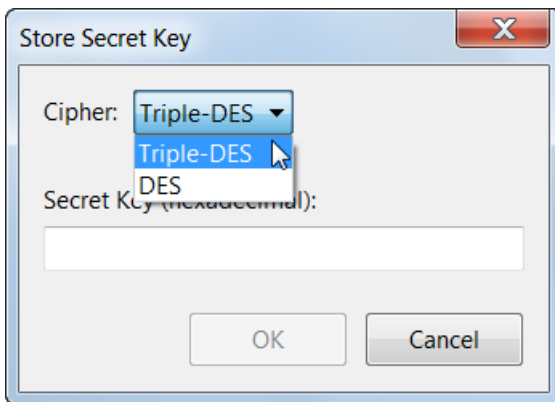
Note: Recommended key sizes are at least 128 bits (Triple-DES). According to present day standards, lesser key lengths cannot be considered as secure any more.

7.3.4 Importing a Secret Key

If you own a secret key, you can import it from the "Edit" menu with the item "Store Secret Key".



The Secret Key must be specified hexadecimal and be 192 or 128 bits in case of Triple-DES or 64 bits long in the case of DES. Importing takes place by inserting the bits into the field "Secret Key (hexadecimal)", e.g. copy and paste.



7.4 Changing PINs

There are 3 PINs on a smart card: the User PIN, the SO PIN (PIN of the system operator, i.e. security operator) and the Card-PIN. There are different functions to use with these 3 PINs:

The **User PIN** must be entered to write data on the card (e.g. key generation, storing a certificate), delete objects, or when the cryptographic functions (e.g. signing or decryption) are performed. The minimal length of the User PIN is four characters and the maximal length is ten characters. The Default-PIN is "11111111" (8 x 1).

IMPORTANT: After three incorrect entries the User PIN will be locked.

A locked User PIN can be unlocked by the **SO PIN** (sometimes called PUK). The minimal length of the SO PIN is eight characters and the maximum length is ten characters. The default SO PIN is "11111111" (10 x 1).

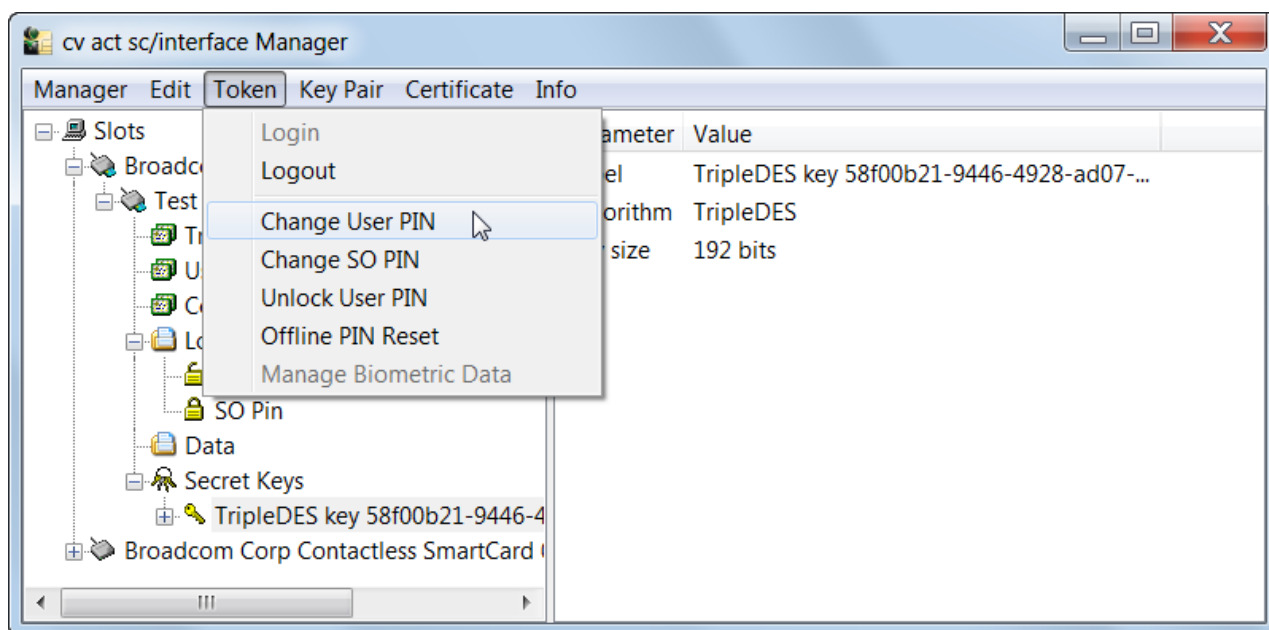
The SO PIN is used for unlocking the User PIN and for enrollment of biometric fingerprint data. There are no other functions like Create or Delete.

IMPORTANT: Depending on the card operating system, the SO PIN will be locked after a number of incorrect entries. Once this limit has been reached and the SO PIN locked, the profile will need to be deleted and recreated with the Card PIN.

With the **Card PIN** one can delete an existing profile on a card by setting up a new profile. The Card-PIN will be determined during the initialization and cannot be changed afterwards. The length of the Card-PIN is fixed to ten characters.

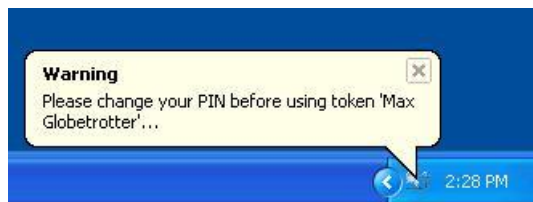
IMPORTANT: After a number of incorrect entries the Card PIN is locked and the card cannot be deleted anymore (e.g. StarCOS3.0 will be locked after three incorrect entries of the Card PIN). If the Card-PIN, the SO PIN, and the User PIN are locked, the card is useless.

You find all functions to change User and SO PIN in the "Token" menu, as shown in the following figure:



Note: With smart card using a PKCS#15 profile, the date of the last user PIN change is stored on the smart card. If the value "00000" is stored, a warning to change the user PIN is displayed by a pop up. If the user

PIN has yet not been changed following delivery, the pop up message reminds the user to do this, provided the smart card supports this functionality.

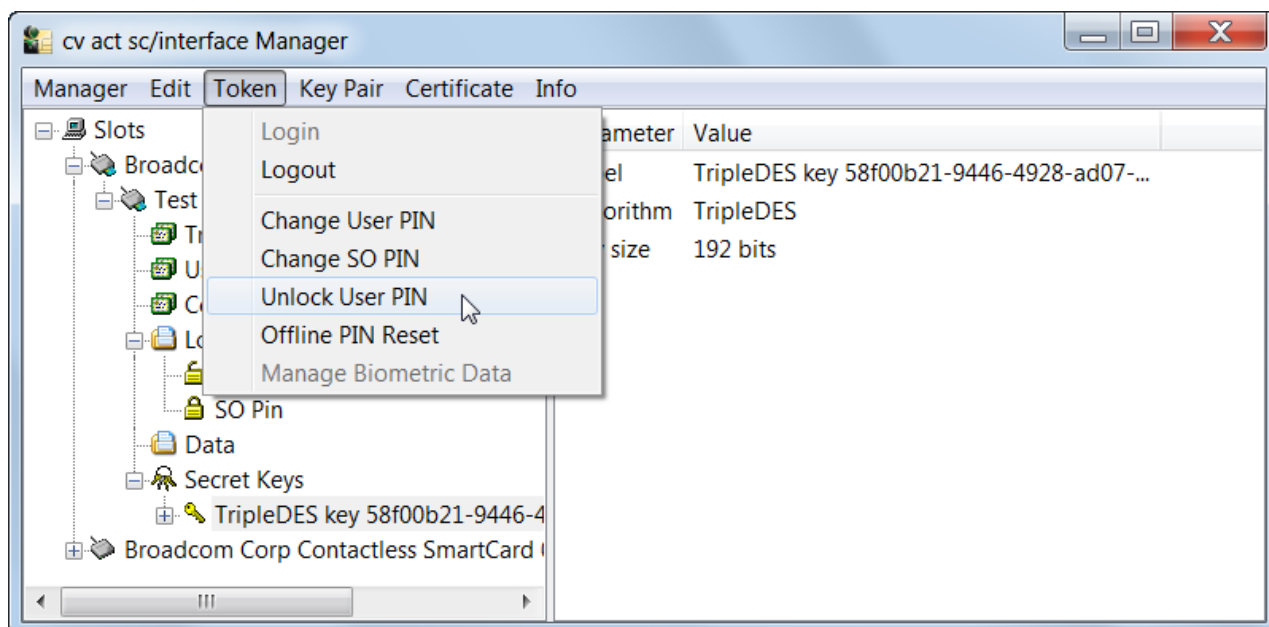


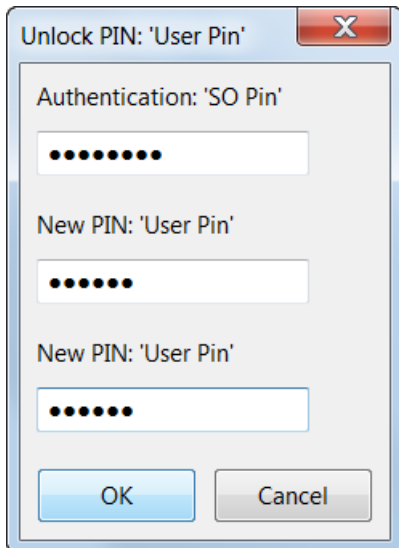
7.5 Unlocking Smart Cards / Unlock User PIN

As a security measure, a smart card will be locked if a user incorrectly enters a PIN three times. This provides security protecting against brute force PINs attacks if a smart card or it has been stolen.

User-PIN and SO-PIN are stored per virtual slot, this is why "unlocking smart cards" equates to "unlocking of a slot" in case of smart cards with multiple applications.

You need the SO PIN to unlock a User PIN. You find the function "Unlock User PIN" in the menu "Token", as shown in the following figure:





A screenshot of a Windows-style dialog box titled "Unlock PIN: 'User Pin'". The dialog has a standard title bar with a close button (X). Inside, there are three input fields, each preceded by a label: "Authentication: 'SO Pin'", "New PIN: 'User Pin'", and "New PIN: 'User Pin'". Each input field contains seven black dots, indicating a masked PIN. At the bottom of the dialog are two buttons: "OK" and "Cancel".

Note: for more information about PINs please read the chapter 8.2 “Creating Profile”.

8 Sample Configuration of a Smart Card for First Use

The following example is a representation a typical smart card production process. It starts with a blank smart card on which a profile will be created with cv act *sc/interface* manager user interface from a Windows desktop PC with standard smart card reader. Following the initial profile creation, a private key pair is generated on the smart card and a PKCS#10 request for the public key created and exported. The PKCS#10 request will be sent to a CA operator who issues a certificate after approving the request. The certificate will then be sent back and imported onto the smart card. Once this process is completed the smart card can then be handed over to the designated personnel.

8.1 Preparing a Smart Card (Initialization and Personalization)

Prior to first use, the smart card must be initialized and personalized. Typically, you must setup a profile on the smart card before keys and certificates can be stored on the smart card.

First Step: Creating a Profile (Initialization)

As a first step you must setup a profile on an empty smart card. This procedure is described in [section 4.6](#) "Creating Profiles".

Second Step: Creating Keys and Certificates (Personalization)

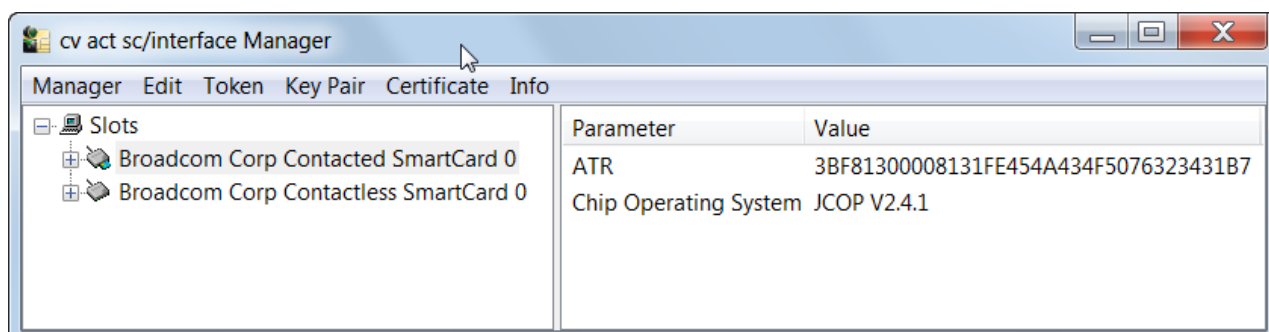
As a second step you must setup for a user key and certificate on the smart card. You have the possibility to either generate keys and certificates or to import them. Descriptions of these procedures are found in [section 4.4](#) "Generating and Importing Keys" and in [section 4.5](#) "Generation and Import of Certificates".

8.2 Create Card Profile

In order to create a card profile, start cv act *sc/interface* Manager from the Windows Start Menu located in All Programs → cv cryptovision → cv act sc interface folder or from the command shell with the command

```
C:\Program Files (x86)\cv cryptovision\cv act sc interface\scManager.exe
```

If a smart card reader is detected with smart card inserted an application window opens similar to the following where the smart card reader is named Broadcom Corp Contacted SmartCard0 and the detected smart card is a JCOP 2.4.1:



Before we can generate a key pair on the smart card the smart card needs to be initialized. This step is called profiling. To open the dialog for creating the card profile highlight the smart card reader and select the menu items Manager → Create Token Profile.

If the smart card is not empty, a warning will appear that all data on the card will be erased when you go on. Please be careful because any existing card profile data will be erased immediately if you go on.

In the Create Token Profile dialog you have to fill out the following fields:

Field	Value	Remarks
Profile	PKCS#15 profile	We recommend using the standardized PKCS#15 profile.
Token Label	A name or identifier for this smart card	The token label will be used by a lot of applications as an identifier for the card. The label can be the name of the owner, a staff id, a short description of the purpose of this card or any other identifying label.
Card PIN	Disabled	
SO PIN	Security Officer PIN	<p>If the user locked his card the SO PIN enables the security officer to unlock the card. The SO PIN must not be handed out to the user. It is good practice to store this PIN in a secure place with access control.</p> <p>We recommend using a numeric SO PIN so that the smart card can be used on smart card readers with PIN integrated PIN pad reader that usually only allows to enter numeric PINs.</p> <p>The SO PIN length limits are shown on the right side of the dialog.</p> <p>The SO PIN must be confirmed in the entry field below labeled "Confirm SO PIN"</p>
User PIN	User PIN	<p>We recommend using a numeric user PIN so that the smart card can be used on smart card readers with integrated PIN pad reader.</p> <p>A default PIN like "11111111" can be chosen at this moment. When the process is completed and the smart</p>

		<p>card will be handed out to the staff the staff can change the PIN when he receives the smart card. No PIN letter is needed proceeding this way.</p> <p>The length of the user PIN must be in the interval shown on the right side of the dialog.</p> <p>If the user fails to enter the correct PIN up to a defined limit the card will be locked by the card operating system. Usually the user is allowed to enter the user PIN up to three times. Only a security officer with access to the SO PIN is able to unlock the card.</p>
Serial Number	A free defined serial number	<p>Please check the “Use Hardware SN” box if this box is enabled.</p> <p>If a hardware serial number is not available choose a serial number from a managed pool. For example combine the staff id with the number of cards issued to the person.</p>
Challenge Response PIN	Uncheck this box	A challenge response PIN will not be used in this example.
Minidriver compatible	Disabled	This option is only available with the challenge response PIN which is not used here.
Session PIN support	Disabled	This option is only available if the card supports it.

Create Token Profile

Profile: PKCS#15 profile

Token Label: Test Profile

Card PIN:

SO PIN:

Confirm SO PIN:

User PIN:

Confirm User PIN:

Serial Number: ☐ Use Hardware SN

Challenge Response PIN: ☒ 11111111111111111111

Minidriver compatible: ☐

Session PIN support: ☒

The Card PIN is defined to consist of 10.

✓ The SO-PIN has to consist of at least 4.

✓ The SO-PIN shall not exceed 10.

✓ The SO-PIN was correctly verified.

✓ The user PIN has to consist of at least 4.

✓ The user PIN shall not exceed 10.

✓ The user PIN was correctly verified.

The serial number shall have not more than 16 and at least one alpha-numeric digits.

✓ The challenge response PIN must have exactly 48 hexadecimal digits.

OK Cancel

When all is green push the OK button to create a new card profile on the card. This process will take some minutes until the profile is successfully created.

The main application window should look similar to the following screenshot: On the left you will see some empty containers like "Trusted Certificates", "Secret Keys" and other containers and on the right you will see some basic information of the profiled card.

8.3 Generation and Import of Certificates

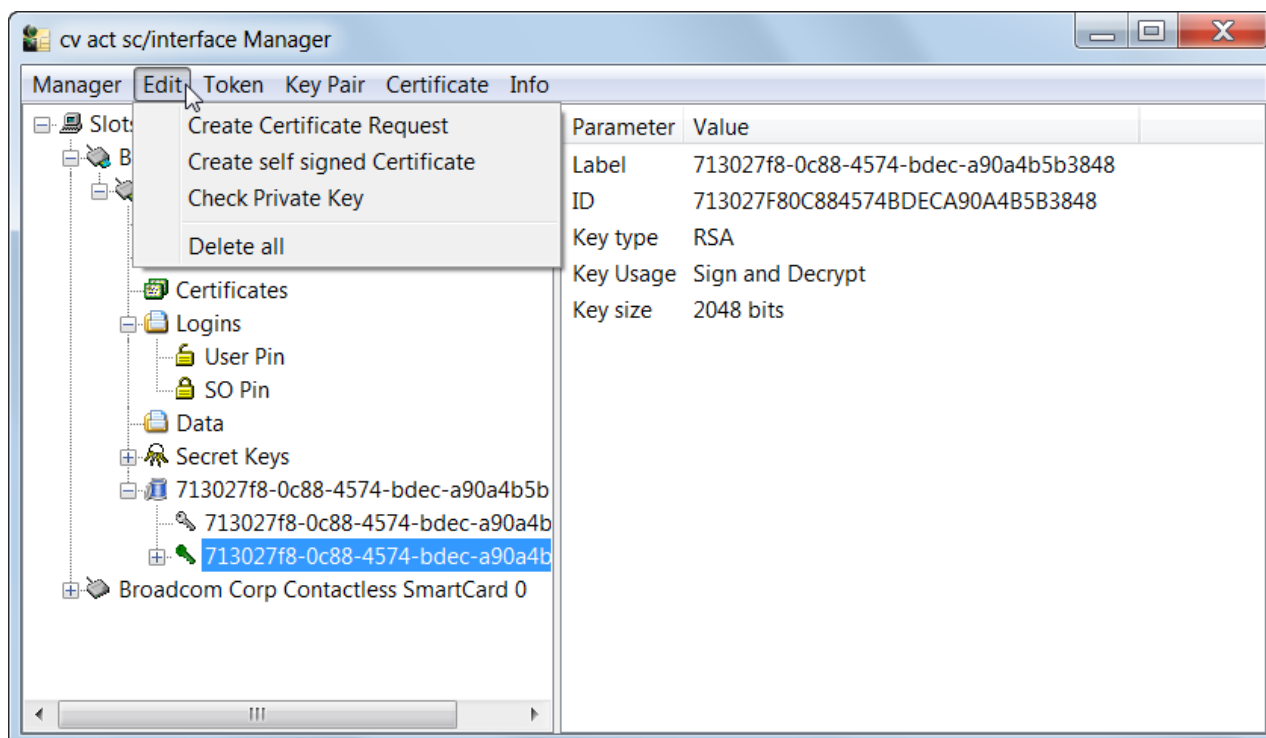
In order to use the smart card for digital signatures or encryption you need a key pair comprising a private key and a public key. The public key should be accessible to communication partners by a certificate. These certificates can be generated and managed by the administration tool.

In principle there are two possibilities:

1. You can self-sign the certificate corresponding to a public key or make a certificate signing request to another entity e.g. a trust center which will authenticate the public key.
2. You already have a key and/or certificates. Then, you can import the needed certificates together with the corresponding key.

8.3.1 Generating self-signed Certificates and Certificate Requests

You can generate the certificate belonging to a public key by signing it yourself or make a certificate request, so that another entity, e.g. a trust center authenticates the public key. To perform this, select the Private Key and use the "Edit" menu item "Create Certificate Request" to generate a certificate request or the item "Create Self Signed Certificate" to self-sign the certificate.



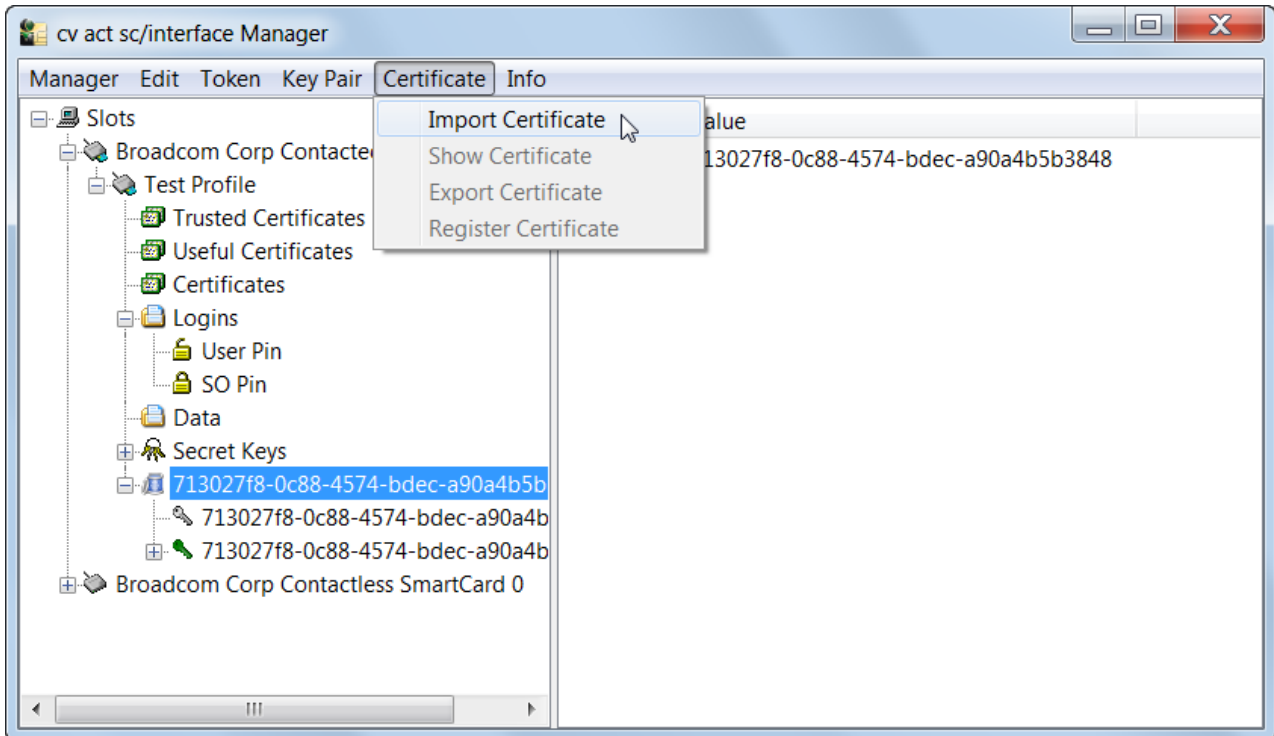
In order to generate the certificate request response, you must enter the data into the corresponding fields. Typically, a certificate request is done by generating file to send it to the authority that should sign the certificate (e.g. trust center). Therefore you store the request as a p10-file in a directory and attach it to the email which is sent to the corresponding authority intended to sign the certificate. You have to observe the specifications of the issuing authority (e.g. trust center).

If the certificate of the issuer has been returned, you have to import the certificate over the menu item "Import Certificate".

Note: There is an explanation of the certificate attributes and how to employ the keys in the appendix C of this manual.

8.3.2 Import of Certificates

In case you already own the certificates that you intend to employ, you can import them with the "Certificate" menu item "Import Certificate". Certificates which correspond to key pairs, are directly assigned to the associated "container" after the import. Certificates without keys, for example CA certificates, are assigned to the node "Certificates".



9 Using Biometrics

cv act *sc/interface* enables secure biometric access to Windows environments with smart cards with PIN and a fingerprint.

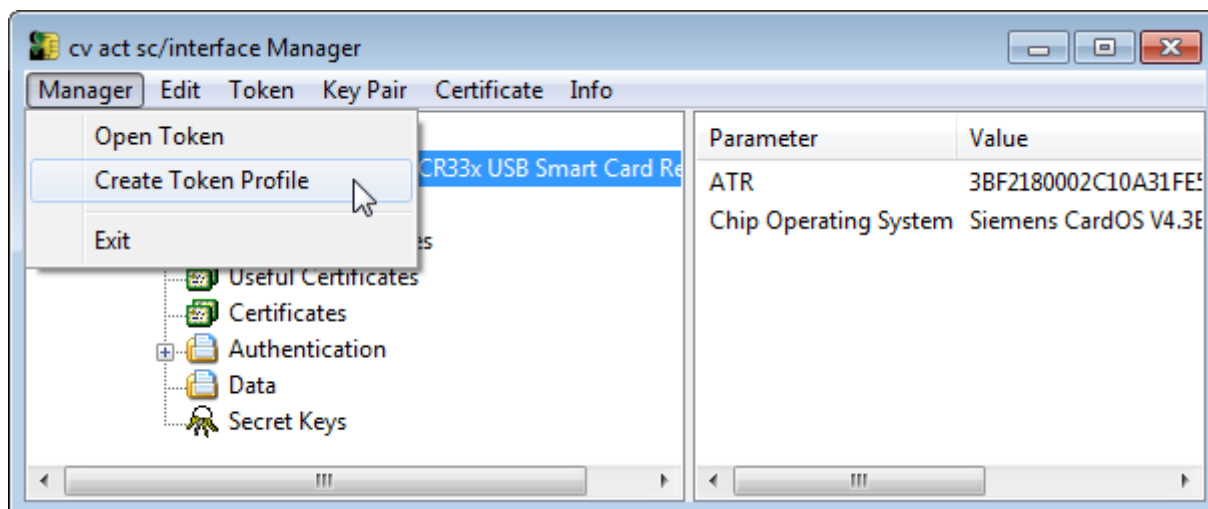
9.1 Supported Smart Cards and Reader

cv act *sc/interface* currently supports only Precise Biometrics fingerprint readers. A list of the tested readers you find in the section [“Smart card reader with fingerprint sensor”](#) in the chapter about the installation.

- CardOS 4.01a
- All listed Java Cards with the Precise Match-on-Card™ package (see [section 3.3 “Supported Smart Cards”](#))

9.2 Profile

If you want to use a smart card with biometry, the card needs to be equipped with a biometric profile. If a non-biometric profile is already installed, it must be deleted before a biometric profile can be created. To do so, the card PIN needs to be inserted. If you have created the existing profile yourself, you have to use the card PIN chosen by yourself. If the profile is a standard cryptovision profile, the card PIN is "0987654321". In order to create a biometric profile, choose (after having chosen a reader) "Create token profile" in the menu "Manager".



When you create a new profile (initialization) the following window opens:

Create Token Profile

Profile: PKCS#15 biometric profile

Token Label: Biometric test profile

Card PIN:

SO PIN:

Confirm SO PIN:

User PIN:

Confirm User PIN:

Serial Number: ☐ Use Hardware SN

Challenge Response PIN: ☐

Minidriver compatible: ☐

Session PIN support: ☒

The Card PIN is defined to consist of 10.

- ✓ The SO-PIN has to consist of at least 4.
- ✓ The SO-PIN shall not exceed 10.
- ✓ The SO-PIN was correctly verified.
- ✓ The user PIN has to consist of at least 4.
- ✓ The user PIN shall not exceed 10.
- ✓ The user PIN was correctly verified.

The serial number shall have not more than 16 and at least one alpha-numeric digits.

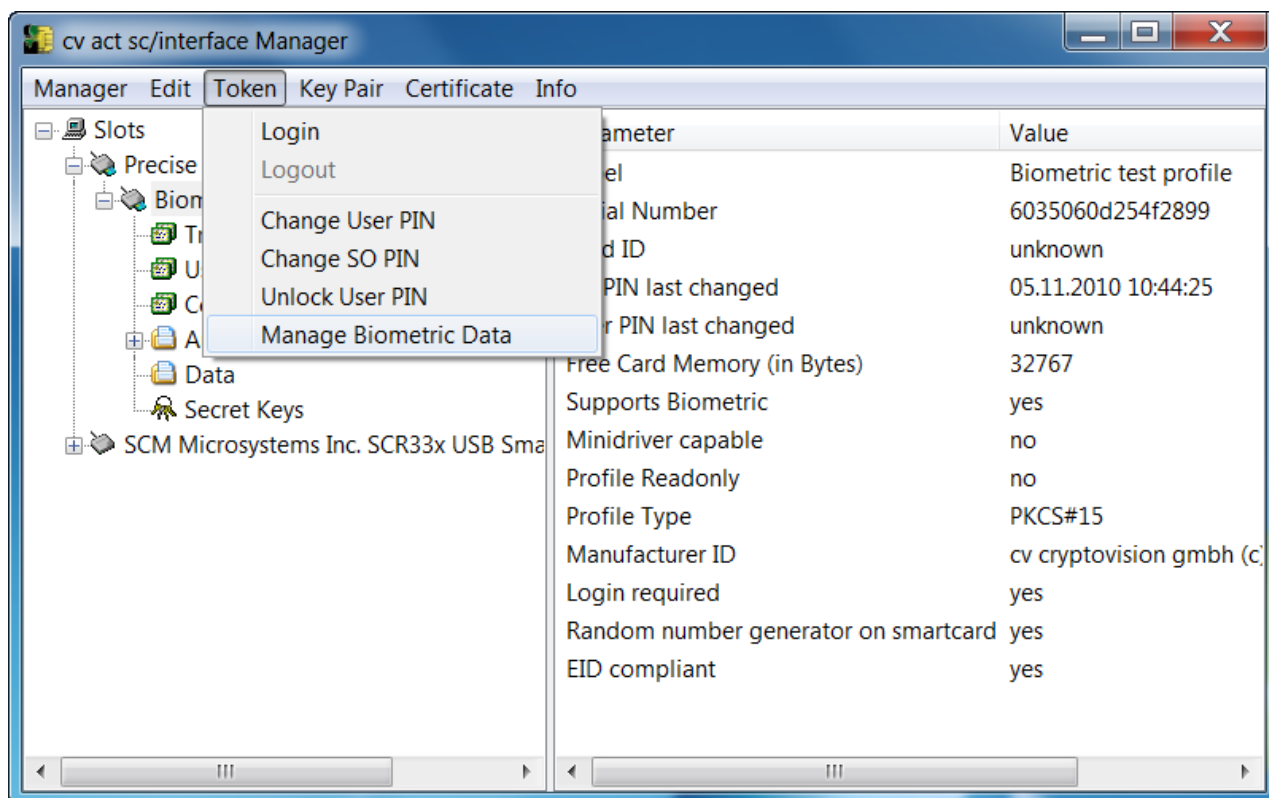
- ✓ No challenge response PIN needed.

OK Cancel

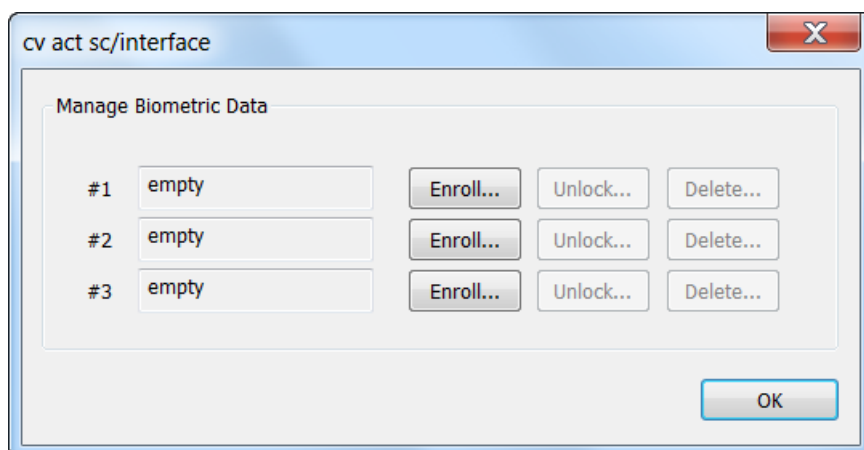
Choose "PKCS#15 biometric profile" and insert the card PIN, the SO PIN and a serial number. The right part of the window displays the requirements the values need to fulfill and the current state of the current state of each requirement. A green checkmark stands for "okay", a red checkmark for "requirements not fulfilled".

9.3 Enroll Fingerprint

Now you can enroll your fingerprint and store it on the smart card. Select "Manage Biometric Data" in the menu "Token":

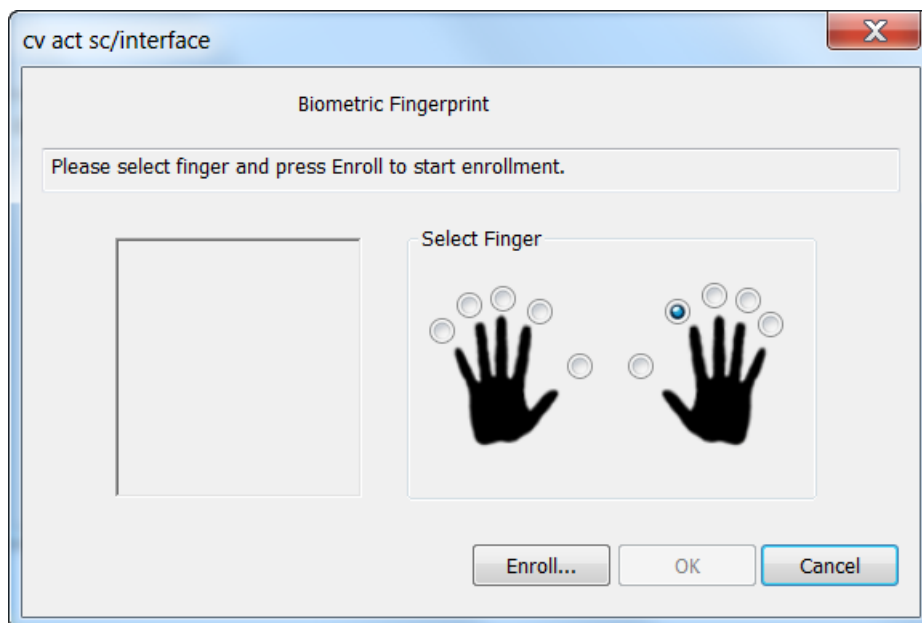


This function is only available when the inserted smart card contains a biometric profile. After selecting this menu item a window appears where you can store up to three fingerprints:



Now you can enroll a new fingerprint (each finger can only be enrolled once) unlock a fingerprint or delete a fingerprint. To enroll a fingerprint click on the button "Enroll..." in one of the slot which is marked "empty". If you don't have an empty slot you will have to delete one of the enrolled fingerprints first. After

clicking on "Enroll..." a new window opens. If you already have enrolled other fingerprints these fingers are not selectable. If you want to overwrite one of these fingerprints you first have to delete this fingerprint:



Select the finger to be enrolled and click on Enroll. Follow the instructions in the line under "Biometric Fingerprint Enrollment". When the fingerprint has been read and the quality check was successful, click on "OK". Enter the SO PIN to store the biometric data on the smart card.

After entering the SO-PIN the data can be stored on the smart card.

9.4 Unlock and Delete a Fingerprint

Swiping the finger more than three times incorrectly over the biometric scanner will lock the smart card, equal to the PIN usage of a smart card. You can unlock the smart card using the SO PIN. As seen above use the unlock button. There will be a popup asking for the SO PIN.

For deleting a fingerprint press the delete button behind the enrolled finger. There will be a popup prompting for the SO PIN.

9.5 Smart Card Reader Mapping

cv act *sc/interface* can be configured to use a biometric device that does not include an integrated card reader via the use of a logical mapping to link the two different devices. For example, a laptop with an integrated fingerprint swipe sensor can be used in conjunction with USB smart card reader. Even some biometric devices that contain integrated card readers require this reader mapping

Create the folder "reader mapping" in "HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface"

1. Create a String Value (RegSZ) with the name of the card reader which does not support biometric (e.g. "SCM Microsystems Inc. SPRx32 USB Smart Card Reader 0")

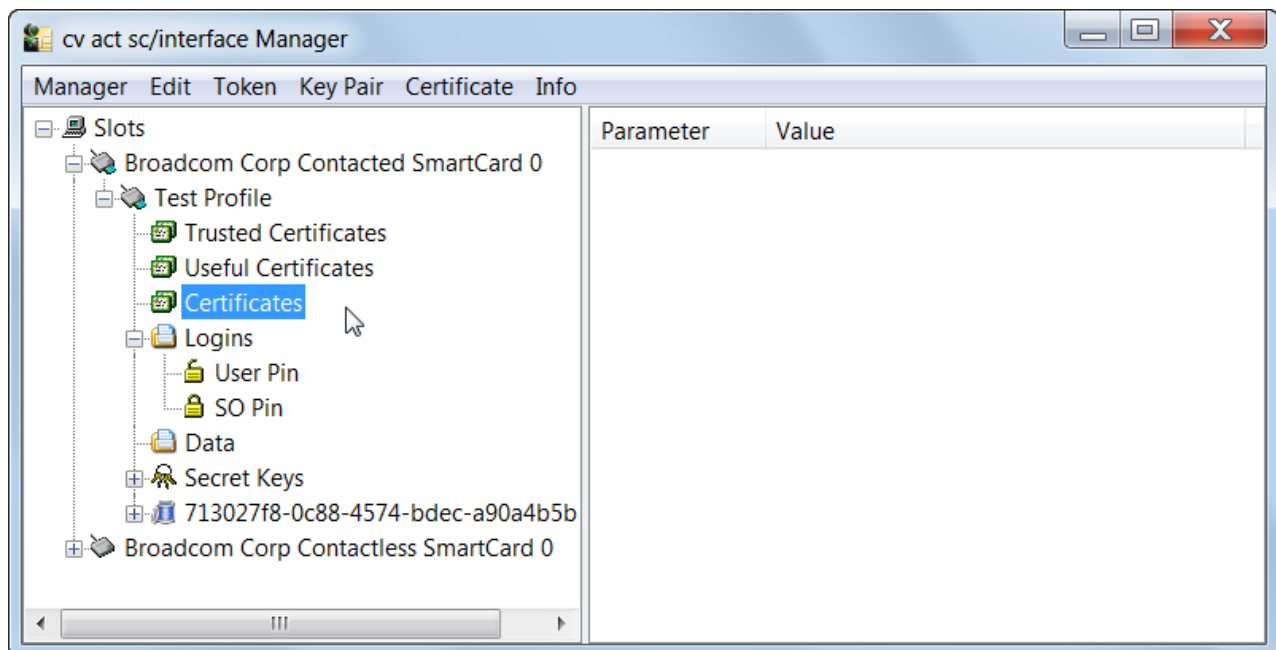
2. Set the String Value to the name of the biometric card reader (e.g. "Precise Biometrics Precise 250 MC 0").

The cv act sc/manager will show you the name of all connected card readers. This option is not available for CardOS 4.01a.

10 Advanced Functions

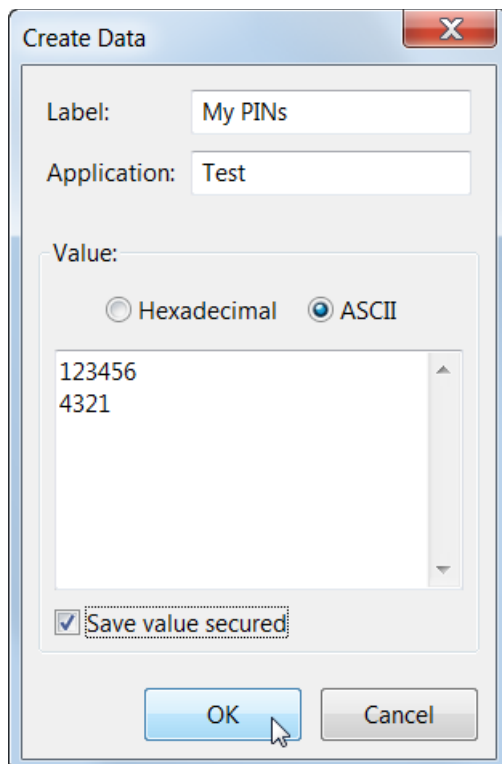
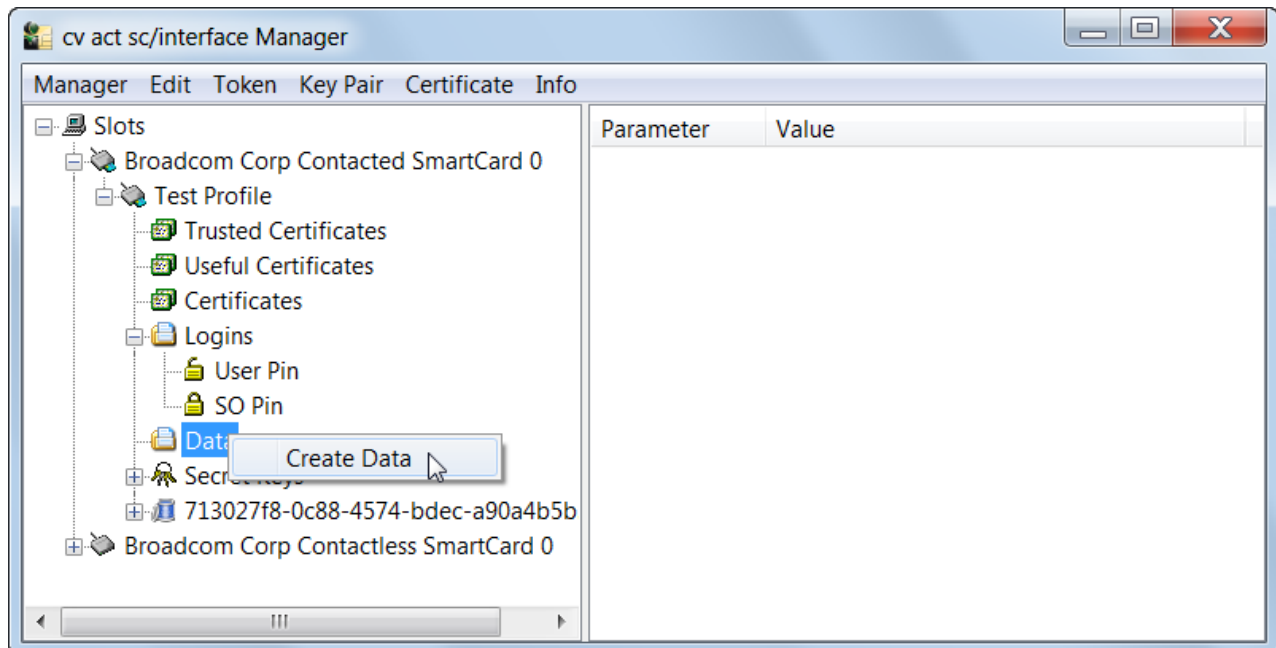
10.1 "Trusted Certificates", "Useful Certificates", "Certificates"

There are the nodes for "Trusted Certificates", "Useful Certificates" and "Certificates" for all certificates that are not directly corresponding to a key. These are root or intermediate certificates that have to be imported into these directories. For this purpose select the item "Import Certificate" in the menu "Certificate" or choose the context menu over the right mouse button.



10.2 Directory "Data"

A smart card is the safest environment for the private key. Additionally, other sensitive data can be secured on the smart card. To write data to the card, login to the token with the user PIN select the "Data" directory and chose "Create Data" in the "Edit" menu. A window is displayed for you, where you can create your data:



Here you can enter either hexadecimal or text data values. Optionally, data can be secured by selecting "Save value secured" check box. Once saved, your existing data can be deleted, updated or exported.

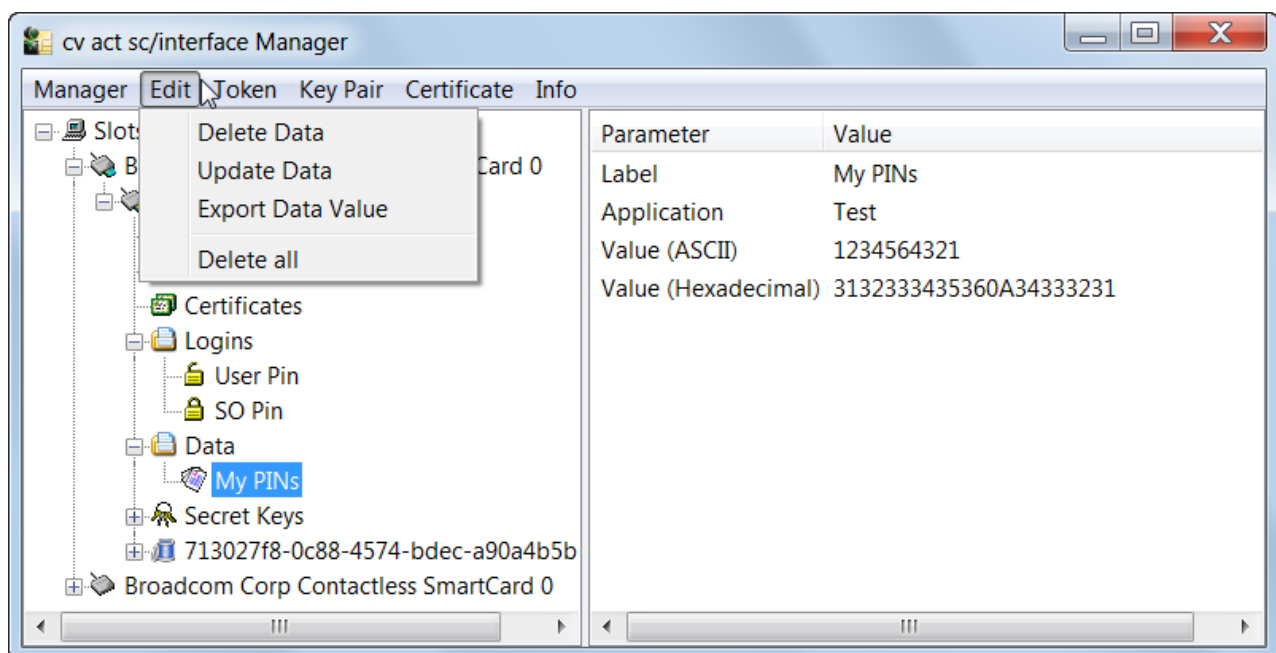
10.3 Function "Open Token"

The function "Open Token" of the menu "Manager" transfers data from the smart card to the user interface. This is recommended, if you work with many different cards or card readers.

10.4 Function "Delete all" and "Delete Certificate" / "Delete Data" / "Delete Secret key" / "Delete Container"

You can delete all objects, such as keys and certificates with the function "Delete all" of the "Edit" menu. Other "delete" functions offer the possibility to remove specific objects, keys, or certificates. These functions are available by a context menu. Select the object to be deleted, right-click and chose the item "Delete Certificate" or "Delete Data" or "Delete Secret Key" or "Delete Container".

Note: The "Delete All" function relies on functionality specific to the smart card operating system. Typically, native card operating systems support this feature, whereas Java Card based operating systems do not. If the smart card does not support this function, the context sensitive command is and can't be selected.

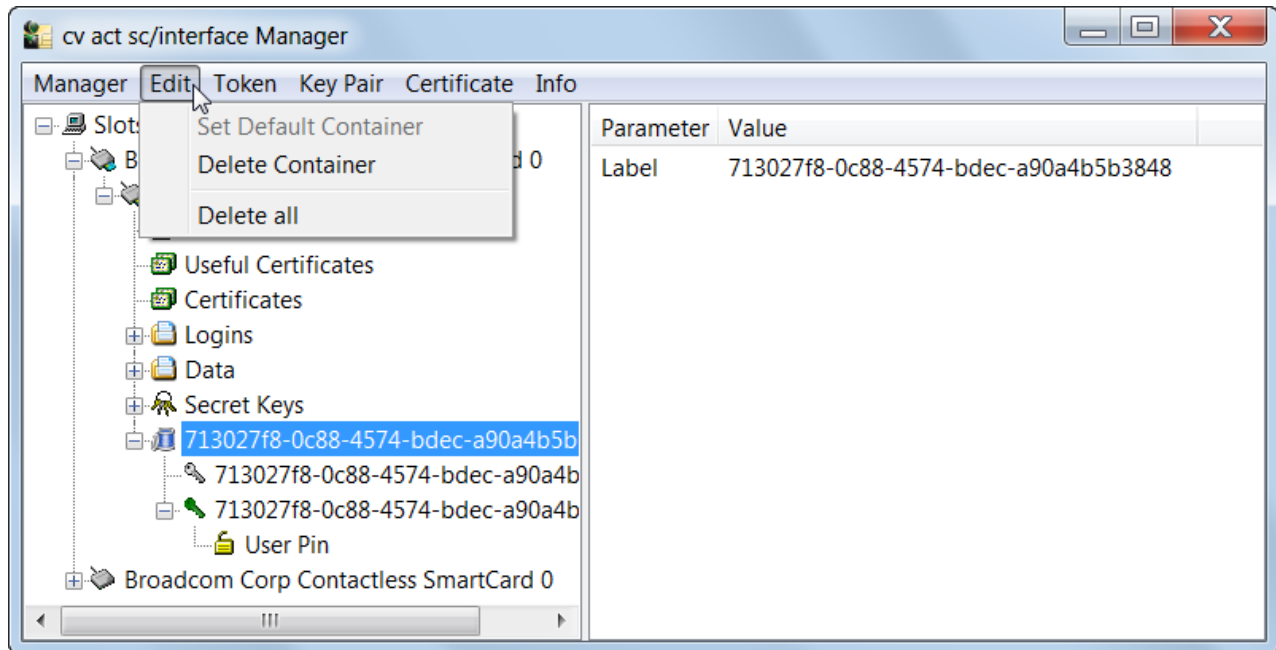


10.5 Function "Set Default Container"

The function "Set Default Container" of the menu "Edit" menu is relevant to smart card login to a Windows-2000/2003/2008 domain via CSP.

If you do not specify a container as Default Container, Windows will take the first key from the list for the login to a Windows-2000/2003/2008 domain via CSP. Smart cards containing multiple user containers may need to change this by selecting the desired container and the right click context menu option "Set Default Container".

The Default Container is displayed in bold face in the interface of the administration tool:

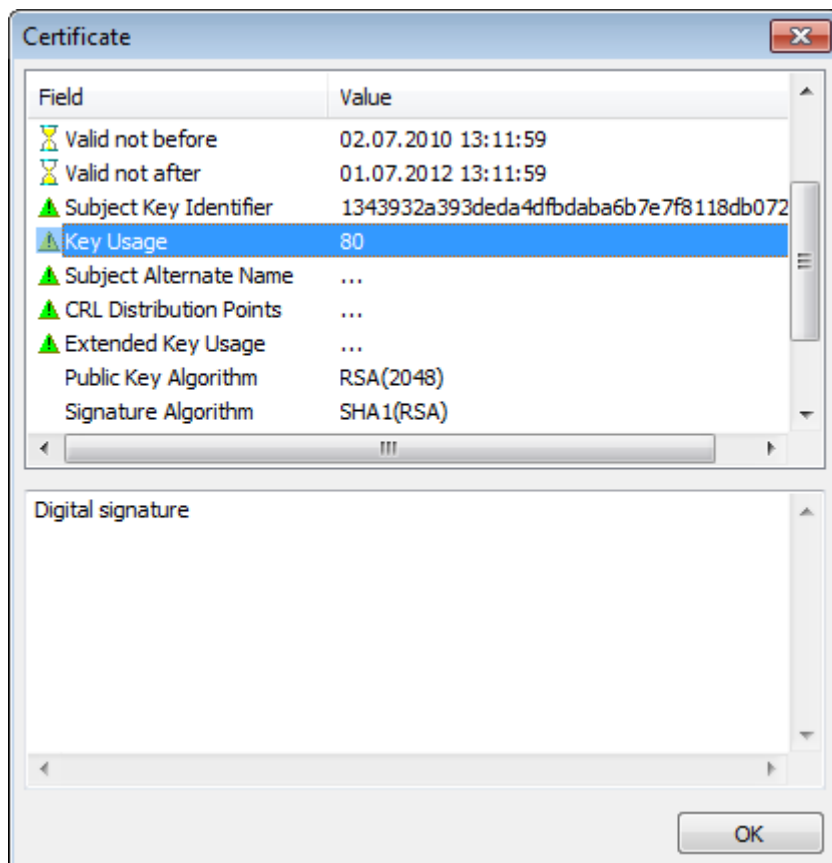


If one uses CSP for certificate enrollment, this creates a container on the smart card, in which the certificate is stored. If the certificate is suitable for smart card logon, the container becomes the standard container automatically.

10.6 Function "Show Certificate"

To display certificates, use the function "Show Certificate" from the "Certificate" menu.

This function is performed via a context menu; select the certificate that you intend to display, right click and choose the "Show Certificate" item. This displays the information contained in the certificate:



10.7 Function "Export Certificate"

If you want to employ a certificate for other applications, you can export it from the smart card with the function "Export Certificate" from the "Certificate" menu. This function can also be performed from a context menu; select the certificate to export, right click and choose the item "Export Certificate".

10.8 Function "Register Certificate"

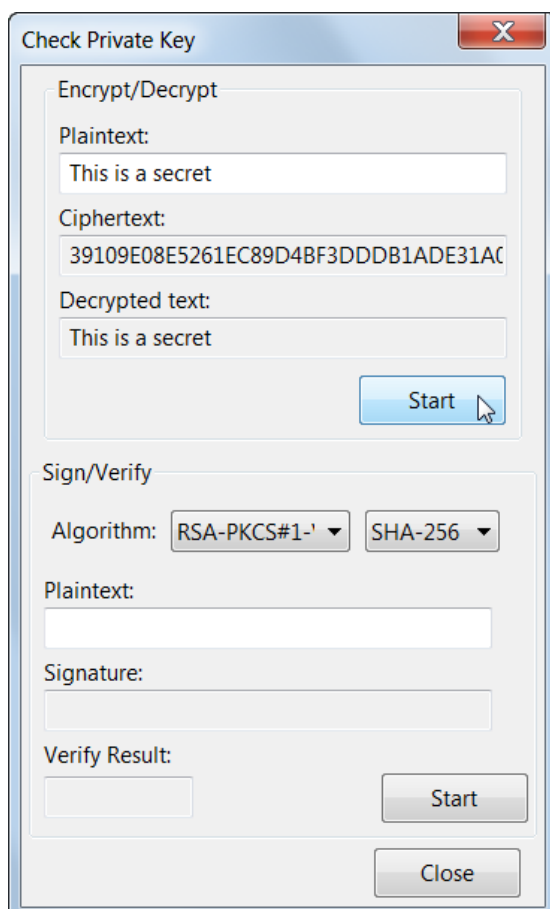
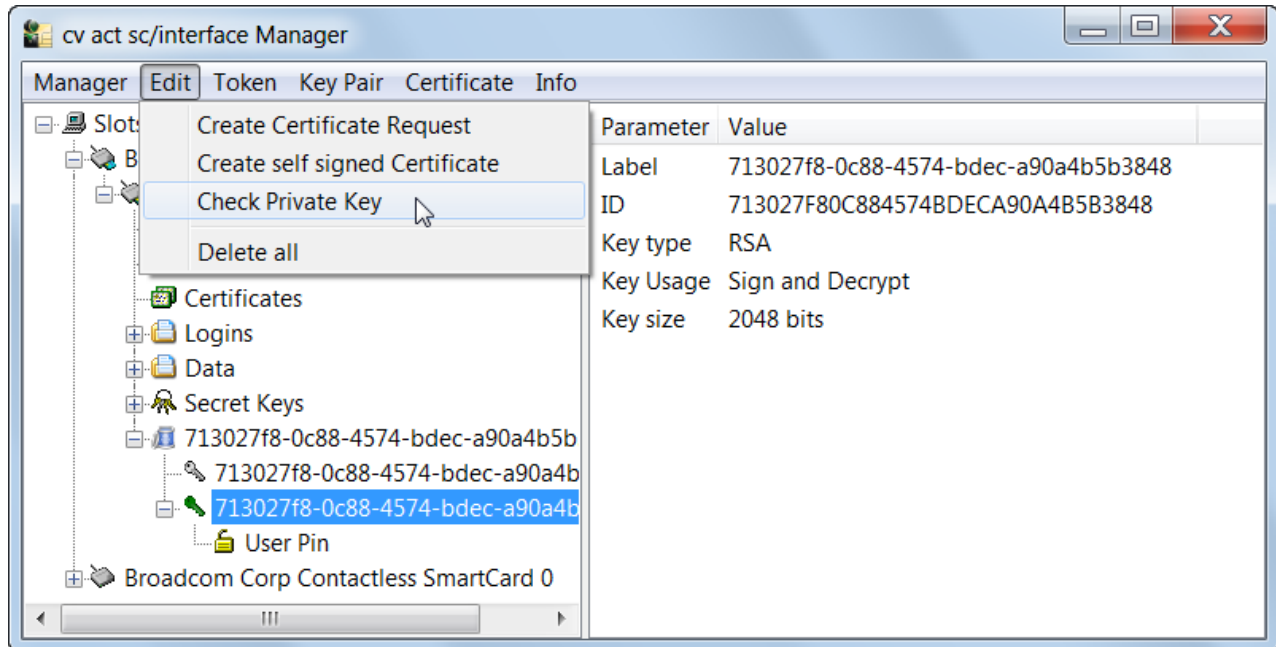
The function "Register Certificate" from the "Certificate" menu installs the certificate into the Windows Certificate Store to make it accessible for Windows applications, like Internet Explorer or Outlook Express.

This function is also available from a context menu; select the certificate to register, right click and then choose the item "Register Certificate".

Additionally, this certificate registration can be automated with the Register Tool. This configuration and settings relevant to registration are described in Chapter 6: [Register Tool](#).

10.9 Function "Check Private Key"

With this function you can test generated keys, e. g. for signing or decryption. To perform this, login to the token, select the private key you wish to test, and chose the function "Check Private Key" from the "Edit" menu. This function is also available from a context menu; select the private key to test, right click and then choose the item "Check Private Key".



To test the decryption key, enter text in the field "Plaintext" and click on the "Start" button. If the Decrypted text field matches the Plaintext, the decryption key successfully works.

Check Private Key

Encrypt/Decrypt

Plaintext:

Ciphertext:

Decrypted text:

Start

Sign/Verify

Algorithm: RSA-PKCS#1- SHA-256

Plaintext: This is a secret

Signature: 54E701450FCA1B634D45261E7416B5FEEB1

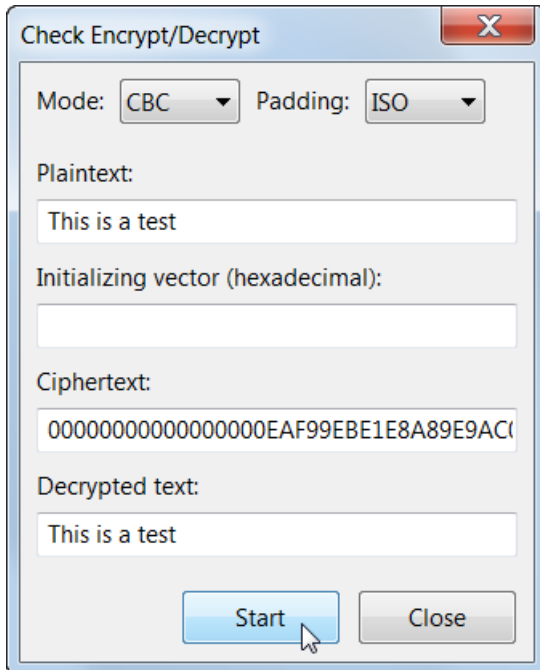
Verify Result: True

Start

Close

To test the signing key you specify a hash algorithm enter text in the field "Plaintext" and click on the "Start" button. If the Verify Result field is displayed as "True" the signing key is valid.

10.10 Function "Check Secret Key"



Check Encrypt/Decrypt

Mode: Padding:

Plaintext:

Initializing vector (hexadecimal):

Ciphertext:

Decrypted text:

With this function you can test generated keys for encryption. First login to the token, then highlight the private key you intend to test and chose the function "Check Secret Key" from the "Edit" menu.

Many cryptographic modes can be used for testing the key. The different methods are: Cipher Block Chaining (CBC), Electronic Code Book (ECB). Either ISO or PKCS5 can be specified as Padding.

To test the encryption key enter text in the field "Plaintext" and click on the button "Start". If you know the initializing vector you can insert it; otherwise it will be filled with zero. If the decrypted text the same as the Plaintext, the encryption key works successfully.

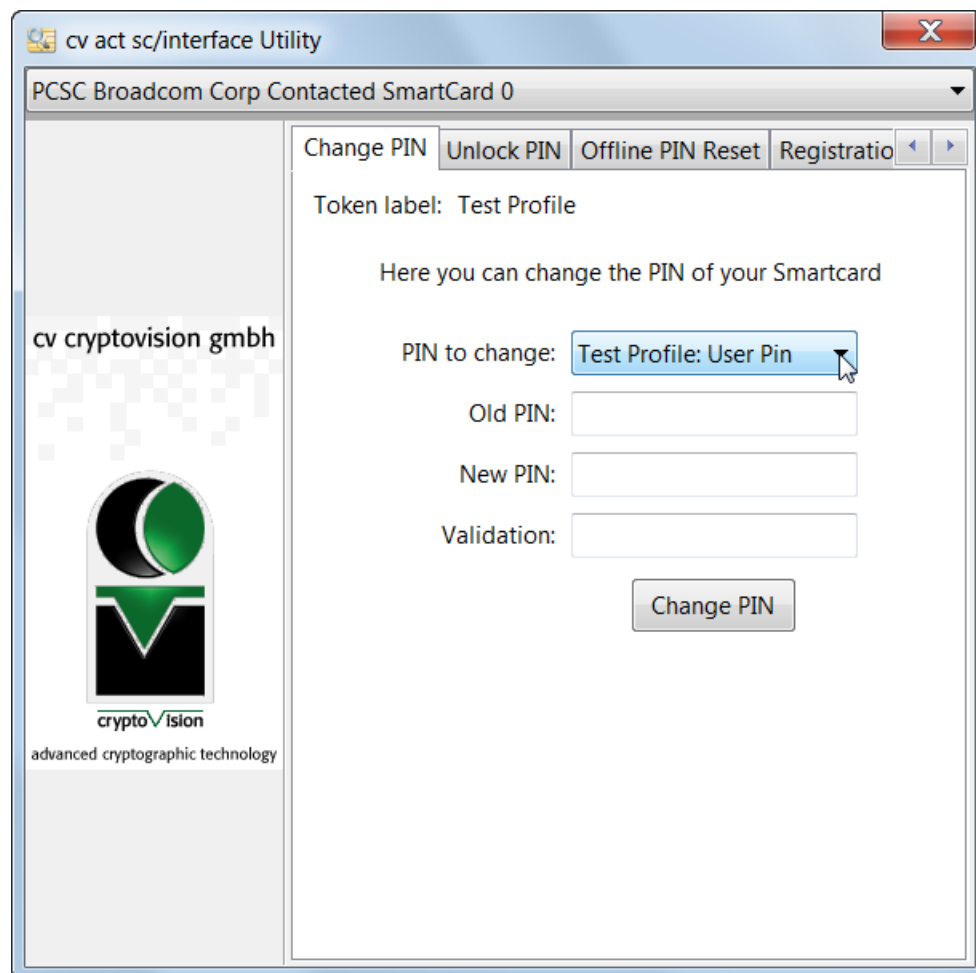
11 User Tool

cv act *sc/interface* supports user self-service with the User Tool. In the user edition, all relevant user functions are available: like changing your PIN, the registration of your key/certificates of the smart card or the export of certificates.

11.1 Changing PINs

Insert your smart card in the reader and open cv act *sc/interface* Utility: click on "Start" and then follow the path "Programs"->"cv cryptovision"->"cv act sc/interface"->"cv act sc/interface Utility".

To change your PIN, first enter the old PIN. Then enter a new PIN, followed by a second identical entry as confirmation. The minimal length of the User PIN is four characters and the maximal length is ten characters.



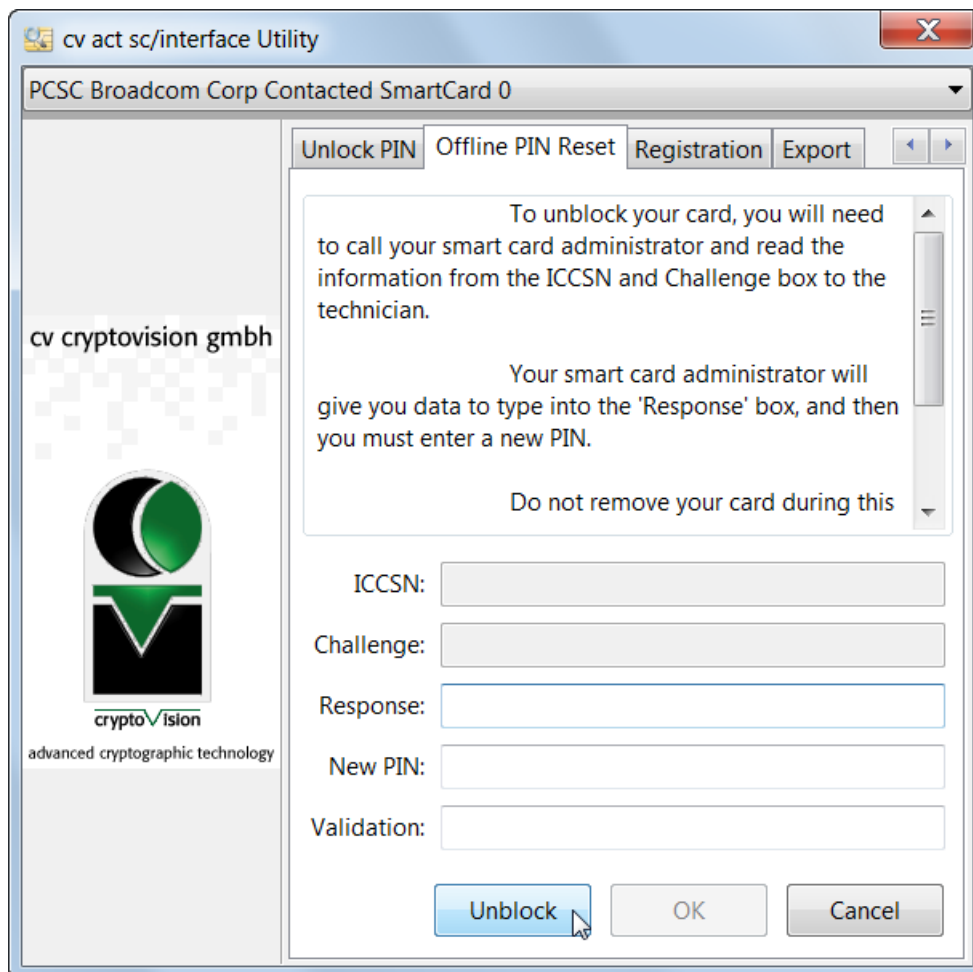
Click on the button "Change PIN" and you receive a window with the confirmation. When a user changes the User PIN the current date and time is stored.

IMPORTANT: After three incorrect entries the User PIN will be locked. Please choose a PIN, which you can remember well, but which cannot be easily guessed. Avoid e.g. birthdays or simple sequences of numbers like 1234 or 1111.

Note: With smart card using a PKCS#15 profile, the date of the last user PIN change is stored on the smart card. If the value "00000" is stored, a warning to change the user PIN is displayed by a pop up. If the user PIN has yet not been changed following delivery, a pop up message reminds the user to do this, provided the smart card supports this functionality.

11.2 Offline PIN Reset

If your smart card is blocked, your administrator can unblock it. Please follow the instructions on the tab "Offline PIN Reset" in your cv act sc/interface Utility:



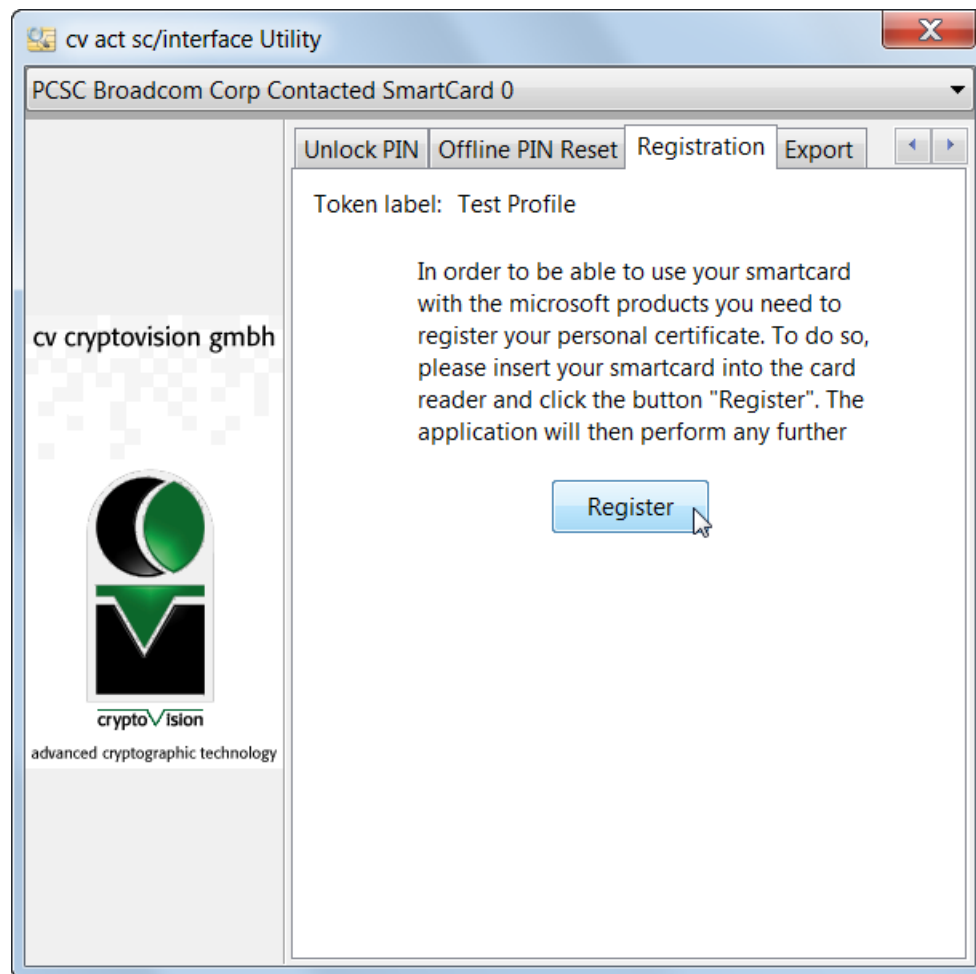
11.3 Smart Card Registration

Your smart card contains certificates and keys. These certificates must be registered so that applications can use these. Registration of the certificate/keys in the Microsoft Windows Certificate store allows them to be used by applications like Internet Explorer, Outlook, and many more.

IMPORTANT: The registration has to be accomplished only once.

Insert your smart card in the reader and open cv act *sc/interface* Utility: click on "Start" and then follow the path "Programs -> cv cryptovision -> cv act sc interface -> cv act sc/interface Utility".

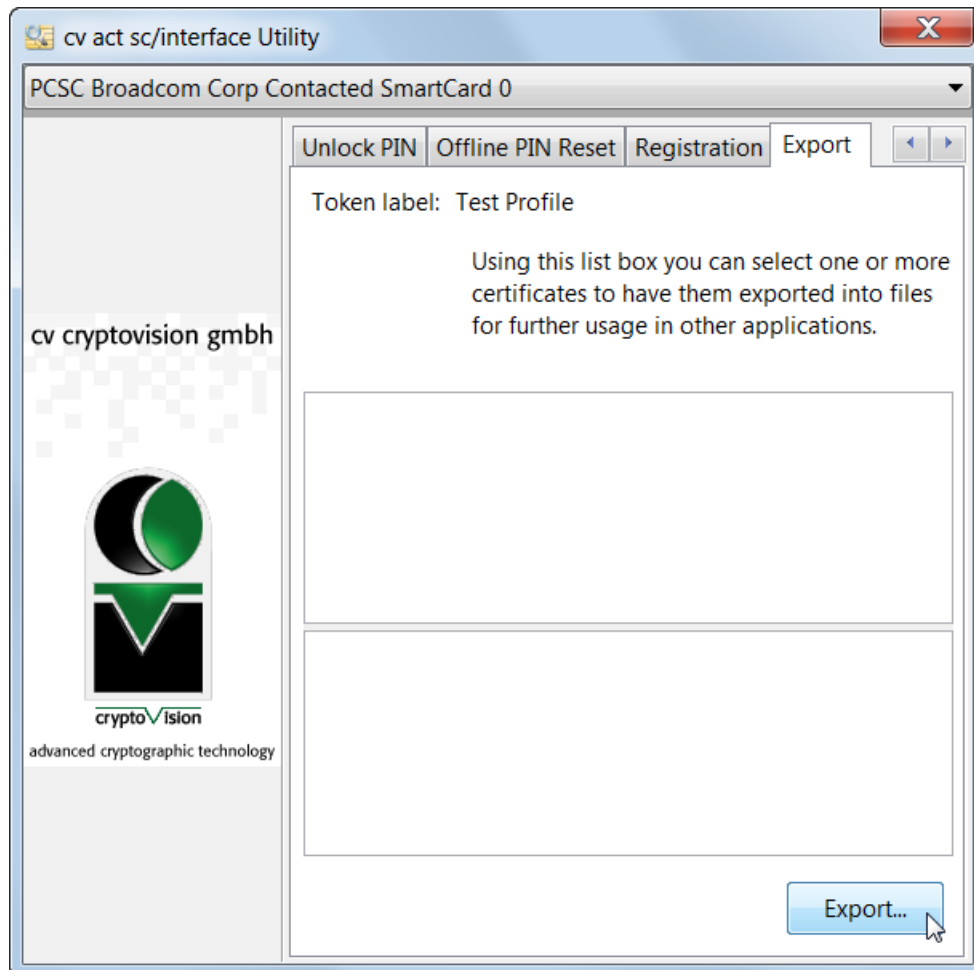
Now click on the tab "Registration" and on this window on the button "Register now". Follow the instructions on the monitor.



Registration is confirmed.

11.4 Export Certificates

"Export" provides a function to export certificates into files that are stored in one of the certificate container on the smart card. One certificate has to be marked in order to export such certificate, the export is initiated by the button "Export ..."

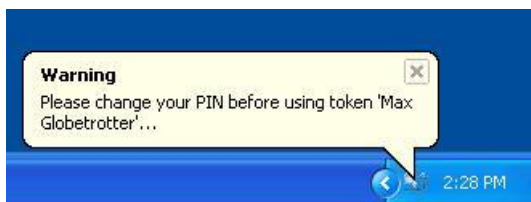


12 Register Tool

If you use cv act *sc/interface* in the Admin or in the User Windows edition, the Register Tool can automate functions for you.

To make certificates accessible for Windows applications like Internet Explorer or Outlook Express, you can automatically register the certificates from your smart card in the Windows certificate store via the Register Tool.

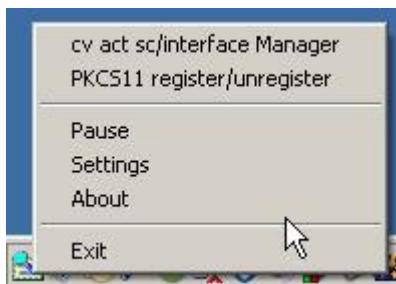
The default behavior is to automatically register certificates when a smart card is inserted into the card reader when the Register Tool is active. Optionally, upon smart card removal, the certificates can also be automatically unregistered. If this is desired, you can enable or disable this via the "Settings" right click options.



You can start the cv act *sc/interface* Register Tool using the start menu. It can also be loaded at startup. The user interface is available via the system tray icon:



By right clicking this system tray icon, the following possibilities are presented: starting the Administration Tool *sc/interface* Manager or the User Tool *sc/interface* Utility, register the PKCS#11 module in Netscape browsers, to pause the Register Tool, to configure Settings, to read information or to Exit the Register Tool.



Or



12.1 Start cv act *sc/interface* Manager and Start cv act *sc/interface* Utility

If you have the Admin Edition the function "Start cv act *sc/interface* Manager" gives you the possibility to start the Administration Tool "cv act *sc/interface* Manager". If you have the User Edition with the function "Start cv act *sc/interface* Utility" you can start the User Tool cv act *sc/interface* Utility. Further explanations concerning this Administration Tool you find in the relevant sections of this document.

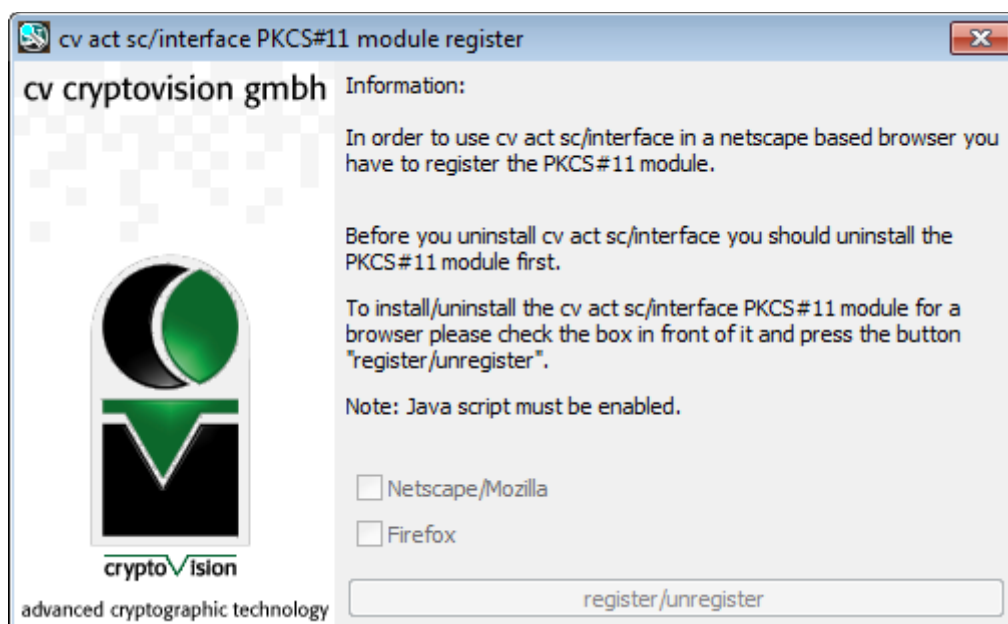
12.2 PKCS#11 register/unregister

Firefox supports PKCS#11 and can be configured to use the cv act *sc/interface* PKCS#11 libraries for use with the smart card and browser based use cases. It concerns applications and functionalities, like mutual SSL and E-Mail security with Netscape. Further explanations concerning the PKCS#11 Module you find in the PKCS#11 Module section of this document.

NOTE: Mozilla Firefox Version 3.6.6 or higher has changed their guidelines and doesn't allow a HTML page to register or unregister PKCS11 modules. Please make a manual registration.

Via the function "PKCS11 register/unregister" you can start the module. A window for registering and the deregistering will be displayed.

If you would like to register the cv-PKCS#11 in the Firefox Browser, select the browser and click on the button "register/unregister".



A window appears to ask whether you want to install the PKCS#11 module. The name and path of the module are indicated.

For the confirmation of the action a further dialog appears.


After you have clicked on the button "register/unregister", the browser opens. When all actions are finished, the confirmation text or an error message appears also in this browser window:

If you would like to deregister the cv-PKCS#11 in Firefox, deselect the appropriate Browser and click on the button "register/unregister".

12.3 Pause / Continue

If it is not desired that the certificates of the smart card are registered automatically, you can pause the Register Tool. To do so, select the item "Pause" in the pop-up menu of the tray icon.



Note that the tray icon changes . It now displays that the Register Tool is set to pause. In order to continue with the automatic registration, you select the now appeared point "Continue" in the pop-up menu of the tray icons:



12.4 Settings



If certificates stored on smart card should be registered automatically in the Windows Certificate Store the first checkbox has to be selected (default).

The so called "Friendly Name" will be set during registration of the certificates stored on smart card if the relevant checkbox is selected.

In order to unregister certificates from windows certificate store upon smart card removal the last checkbox has to be selected.

12.4.1 Configuration via registry

In addition to the adjusting these settings via the Register Tool user interface, these can be manually configured via registry settings. The following registry key governs the register tool settings

```
[HKEY_LOCAL_USER\SOFTWARE\cv cryptovision\sc interface]
```

Each Register Tool setting has a corresponding binary value.

```
[DeactivateRegister]
```

Supported values 00, 01 (default).

A value of 00 will prevent the Register Tool from automatically importing the certificates on the smart card to the Windows Certificate Store upon insertion.

```
[SetFriendlyName]
```

Supported values 00, 01 (default)

A value of 00 will disable the use of the “Friendly Name” during certificate registration.

```
[DeactivateUnregister]
```

Supported values 00, 01 (default)

A value of 00 will prevent the Register Tool from automatically removing the certificates from the Windows Certificate Store upon removal.

12.5 Exit

With "Exit" in the menu of the tray icon you can end the Register Tool.

13 Advanced Cryptographic Interface Configuration

13.1 CSP Module

The Windows operating system supports cryptographic functions like encryption and digital signature with the Crypto-API. Furthermore, CSPs (Cryptographic Service Providers) enable programs to support smart cards. During the installation of cv act *sc/interface* the cryptovision-CSP (abbreviated cv-CSP) will be added.

IMPORTANT: The cv-CSP is a DLL. The name of this dll is "cvcsp.dll" and it is stored after installation in the system folder, e. g. in WINDOWS\system32.

With this cv-CSP you can use certain programs and functionality delivered with Windows 2000, XP, 2003, Vista, Windows 7 and 2008, like Outlook Express, Internet Explorer, network login and VPN-login with the smart card.

NOTE: *It is outside the scope of this manual to describe how to configure your Microsoft environment for the use of smart cards. Please consult the help files for each relevant program for the desired use cases. To configure the network login and the VPN-login for smart cards please consult the documentation on the Microsoft websites.*

If you need support for the Windows configuration or implementation, cryptovision's consulting team can help you. Feel free to contact your account manager.

13.1.1 General Notes

One minimum requirement is the installation of the cv-CSP, which the cv act *sc/interface* installation performs automatically. Here are some general notes on employing the cv-CSP:

- > If you want to use a Microsoft product in connection with the CSP for the first time on a certain computer, you must register the certificate that you want to use. Please read in [chapter 6](#) "Register Tool" or in [section 4.9.8](#) "Register Certificate", if you need to know how to register your own certificate.
- > You as a user need keys and certificates on the smart card. There are several different possibilities. The most popular are:
 - Generation of key pair and corresponding certificate directly on the smart card with the functions of standard browsers, like Internet Explorer or Netscape. This ensures an access on the modules of cv act *sc/interface*, i.e. correspondingly on cv-CSP or cv-PKCS#11.

- Generation of key pair and corresponding certificate directly on the smart card with cv act PKIntegrated or Microsoft Certificate Server (in "Enterprise CA" and in "Stand Alone" mode).
- Import of existing keys and certificates on the smart card that were generated by other CAs or trust centers, resp. request of certificates from a trust center.
- Generation of key pair and corresponding self-signed certificate directly on the smart card by the administration tool cv act *sc/interface*. Please observe that the employment of self-signed certificates makes sense only in environments without a PKI or for testing.

Note: *If you request a certificate from a trust center, you might be requested to choose a security module, e.g. a token. In this case choose the cv Profile, the cv-CSP or the cv-PKCS#11. Furthermore, your smart card has to be inserted in the card reader, so that certificates can be written on it.*

- > Programs must be configured, so that they work with your smart card.
- > The programs must be configured, so that they work with your keys and certificates. There you have to take into account the preconditions of the programs that need certain input. E.g. some programs need root-certificates that must be in certain directories or for other programs you must register your certificate.

13.1.2 Smart Card Login to a Windows 2000 or 2003 Domain

Users enabling this function should have a fundamental understanding of Windows Server administration. Follow the procedure specified in the following steps:

Setup of Active Directory Services. Please observe the correct configuration of the DNS-Server.

1. either: use cv act PKIntegrated.

Upon creation of this manual further steps are described on the following website:
<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q281245>

2. or: Installation of the EnterpriseCA and at least the templates "Enrollment Agent", "Smart card Logon" and "Smart card User".

Furthermore, observe that "Set Default Private Key" must set the private key on the client (see in [section 4.8](#)).

13.2 Biometric-Login GINA-integration

The GINA-integration, which is part of cv act *sc/interface*, provides support for biometrics-based smart card logon with Windows XP. It supports all cv act *sc/interface* biometric smart card profiles in conjunction with the cv-CSP.

If no smart card is used for login the standard dialog with username and password appears.



If a smart card without biometric data is used for login the PIN entry dialog appears when you insert the smart card.



When the smart card contains biometric data the login dialog with fingerprint authentication is shown. You can also enter the PIN for login.



Note: The Biometric Login does not work with the Card Minidriver.

Note: GINA-integration will only be supported in the 32-Bit-Windows XP versions.

13.3 Configuration Parameters

13.3.1 PIN Cache Mode (Disable PIN-Cache)

The CSP with the default configuration applies PIN caching. After entering the PIN once, it is stored in the RAM. In this case the user does not have to enter the PIN every time the private key on the smart card is accessed. If the smart card is removed out of the smart card reader, the PIN is deleted in RAM, i.

PIN caching can be disabled with a registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface]
"CSP_Disable_PIN_Cache"=dword:00000000
```

If the registry key is configured with the value "0", PIN caching is active. The value "1" disables PIN caching. If the registry key is absent the default value "0" is used and PIN caching active.

Note: Certain Microsoft Windows behaviors rely on a CSP where PIN caching is supported. This means, if "CSP_Disable_PIN_Cache" is set, the CSP doesn't work as specified according to Microsoft specifications.

This means that applications which use their own PIN entry dialogue with disabled PIN caching may lead to unwanted results. For instance an application might present only one PIN entry dialogue, and afterwards it might pass the PIN to the CSP and expect the CSP to cache the PIN. In such a case it might be impossible to use the CSP with the application without enabling PIN caching.

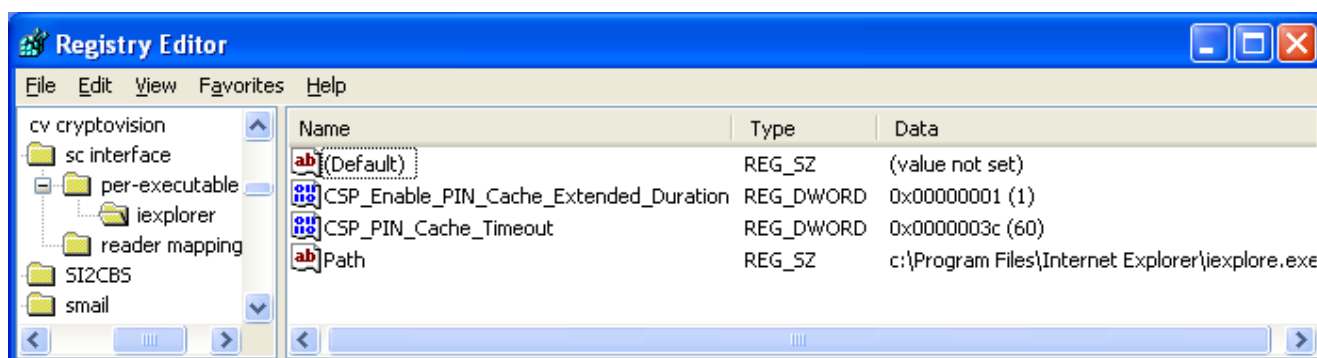
This is the case e. g. if the smart card is accessed multiple times during authentication procedure and therefore the PIN is needed multiple times. For Windows smart card logon and Windows VPN this behavior was tested. It is not possible to logon to a Windows domain or to Windows VPN via smart card, if PIN cache is disabled (registry key set to "1"). In an environment where smart card logon to a Windows domain or Windows VPN with smart cards is used, PIN cache mode should therefore not be disabled.

13.3.2 Extended PIN-Caching

You have the possibility to extend the duration of PIN-caching in the cv-CSP. As changing the PIN caching behavior is not in accordance with Microsoft specification and lowers the overall security level, it must be manually configured for use via the registry.

Configuration of the PIN-Cache "per-executable" (<Program name> is a variable for the set of adjustment):

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface\per-executable\<Program name>]
```



Parameter of applications like Internet Explorer:

CSP_Enable_PIN_Cache_Extended_Duration <REG_DWORD> (0|1)

- Switch-on (1) or switch-off (0) of the extended PIN-Cache

CSP_PIN_Cache_Timeout <REG_DWORD> (1-N)

- Timeout of the PIN-Cache in seconds.

Path <REG_SZ>

- Path of the application:

- 1.either complete, i.e.: c:\Program Files\Internet Explorer\iexplore.exe
- 2.or only the name of the application, i.e.: iexplore.exe

Note: You can activate Extended PIN-caching in general for all applications (and not only per executable). Thereby you must note: If settings for a single program are found, the settings in the main tree are overridden. This means, you may need to copy all registry entries from the main tree in the per-executable settings if you want to use them for the respective program.

13.3.3 Session Mode

The CSP can be used by multiple applications concurrently. If this is not desired and the CSP should be used by one application exclusively, this can be configured by a registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface]
```

```
"CSP_Enable_Session_Mode"=dword:00000000
```

If this key is configured with the value "0", the CSP can be used by multiple applications concurrently. If the value "1" is used, the smart card is available exclusively to the first application which accesses the smart card by the CSP. After this application is stopped, the smart card can be accessed by another application. If the registry key is not available, the value "0" is used.

The exclusive usage of the smart card prevents other applications from accessing the smart card. No other application can access the smart card by CSP, PKCS#11 or direct (without the two standard interfaces). In such a case e. g. the register tool is not usable, while the smart card is used by internet explorer for SSL client authentication. In case of Windows smart card logon or Windows VPN this effect appears. Other applications can access the smart card only after the removal of the smart card and inserting the smart card again in the card reader. This should be considered if session mode is configured.

14 Minidriver

The card minidriver is designed for use with Microsoft Identity Lifecycle Management (ILM) / Certificate Lifecycle Management (CLM) and other applications supporting it.

14.1 CMCK Certification

The Card Minidriver Certification Kit performs functional, stress, performance, and reliability testing on a smart card minidriver. This kit calls the Microsoft BaseCSP and the Microsoft Smart Card Key Storage Provider and accesses the card minidriver methods directly to test the correctness of operation of the card minidriver and the associated card. The kit also uses the Smart Card Resource Manager to access the card directly.

The 32 and 64 bits versions of the minidriver, which is a component of cv act *sc/interface*, passed all CMCK tests successfully. For the tests under Windows XP (32 and 64 bits version) the version 6.0.6001.17031 of the test suite was used. For Windows Vista (32 and 64 bits version) CMCK version 6.1.7000.0 served as the basis of the test.

Further information on this topic is available at the time of the production of this manual under [http://msdn.microsoft.com/de-de/library/dd327365\(en-us\).aspx](http://msdn.microsoft.com/de-de/library/dd327365(en-us).aspx).

14.2 Minidriver Support with PACE

To increase the level of security, you can use PACE (Password Authenticated Connection Establishment). PACE has the advantage that even with a not so secure PIN (ie with low entropy) the data on the chip of the smart card and during the transmission is highly protected (secure messaging) by using a secure channel from cv act *sc/interface* to the smart card and a PIN will not be stored anywhere outside of the card intermediately.

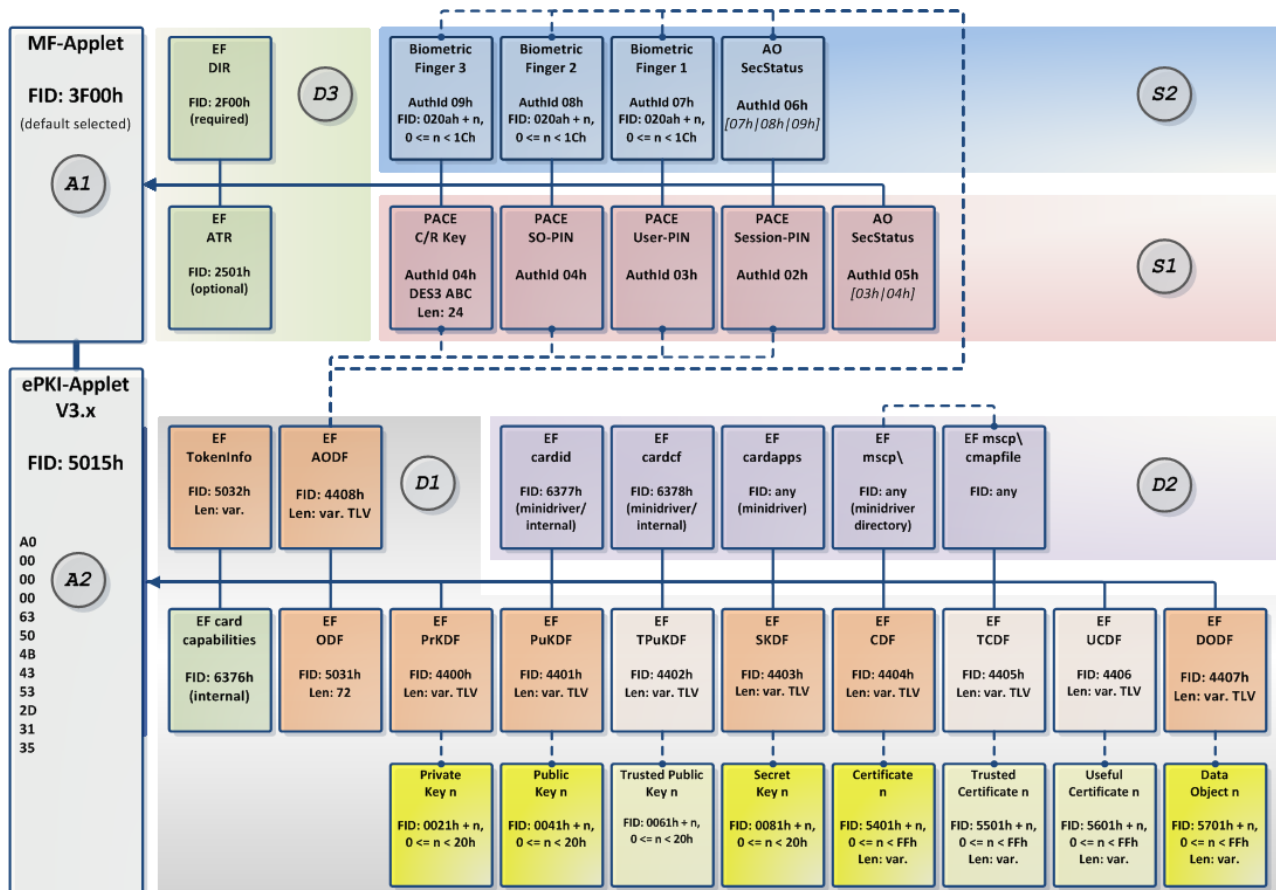
The User-PIN is from type PIN-PACE, the SO-PIN from type PUK-PACE and with using the Minidriver there are the Admin Key and a Session PIN.

cv act *sc/interface* now supports Sessions PINS with PACE Session PINs according the Minidriver Spec. Version 7.06. A session PIN is a temporary PIN that is generated from the card and will be deleted after the session. If a session PIN is used, the current PIN is not passed and the minidriver must use the Session PIN to authenticate the card. The smart card must support the generation of session PINs.

Note: Because of the additional feature to increase the security level the use of the card takes about 2 seconds longer (at session key a little longer).

Next point is that the PACE PIN is used instead of the Challenge Response Key during challenge response. PACE can be used with cv act *sc/interface* with the cards, which supports PACE (eg NXP JCOP V2.4.2R3 PACE profile), and which have loaded the cv act ePasslet suite Aplet, exactly from the cv act ePasslet Suite 1.1, 1.2 and 2.0.

To illustrate, for more technical details, see now the following diagram:



15 PKCS#11 Module

If you use an application of software supporting PKCS#11, you can employ cv act *sc/interface*-**PKCS#11** – abbreviated cv-PKCS#11 – with the smart card. Many applications and functionalities which employ this standard can be used with smart cards like: network login, SSL, email security with Netscape, and others,

NOTE: *No description how to configure specific applications for use with smart cards is given here. For this purpose please consult the documentation of the programs you intend to use.*

IMPORTANT: *cv-PKCS#11 is a DLL with the name "cvP11.dll" and is after the installation in the system directory, e.g. in \WINDOWS\system32.*

15.1 General Methodology

In the following, some general notes are given concerning the employment of cv-PKCS#11. A minimum requirement is the installation of cv-PKCS#11. This module is installed automatically by cv act *sc/interface*.

If the application does not support card initialization (card initialization is usually only available in a card management system) a profile has to be created on the smart card prior to using the smart card with an application.

As a user, you need keys and certificates on the smart card. There are several different possibilities. The most prevalent are mentioned below:

- Generation of key pairs and corresponding certificate directly on the smart card with the functions of standard browsers, like Firefox or Netscape. This ensures access to the modules of cv act *sc/interface*, i.e. to cv-PKCS#11.
- Generation of key pair and corresponding certificate directly on the smart card with cv act PKIntegrated.
- Import of existing keys and certificates on the smart card that were generated by other CAs or trust centers, resp. requesting a certificate from a trust center.
- Generation of key pair and corresponding self-signed certificate directly on the smart card with the delivered administration tool of cv act *sc/interface*. Please observe that the employment of self-signed certificates makes sense only in environments without PKI or for the purpose of testing.

NOTE: *If you use a browser or cv act PKIntegrated or request a certificate from a public CA, you might have to choose a security module. Please choose in this case the cv Profile, the cv-PKCS#11. Furthermore, your smart card has to be inserted in the card reader, so that certificates can be written on it.*

As in [section 6.2](#) described, the possibility is offered to install cryptovision's PKCS#11-Module easily in Netscape. Please use the Register Tool.

The programs must be configured, so that they can work with your smart card.

The programs must be configured, so that you can work with keys and certificates. Here you must take into account the prerequisites of the programs that have certain inputs. E.g. some programs need root certificates, that must be in certain directories or for other programs you have to register your certificate.

15.2 Smart Card Login to a Novell eDirectory

Users intending to enable this function you should have a very good command of administration of Novell servers and observe the installation preconditions. Additionally, to leverage smart card login to an eDirectory, you specifically need the product NMAS and the corresponding Enhanced Smart card Login Method. These features are also included in the Identity Assurance client.

15.3 SSL- Authentication with Smart Card over Firefox / Safari

If you want to use your smart card with one of the above named browsers please register the cv-PKCS#11 in that browser. The necessary functions are provided by the register tool of cv act *sc/interface* and a description is included in [chapter 6.2](#).

15.4 E-mail-Security with Smart Cards for Netscape Messenger

The notes for the employment of Netscape and screen shots to manage certificates and modules are available in the example of version 7 in the previous section.

Normally, there pull-down menus in the email windows, where you can tick whether an email should be encrypted and/or signed, resp. a function for verification of received signed emails, to employ the security functionality.

15.5 cryptovision products, for Instance cv act *s/mail* or cv act PKIntegrated

With all cryptovision products you can address your smart card directly or employ the cv-PKCS#11. We recommend the second possibility.

Detailed configuration information for all cryptovision products is available in the corresponding product manuals.

15.6 Support of Elliptic Curve Cryptography (ECC)

The PKCS#11 Module of cv act *sc/interface* supports cryptographic algorithms is based on elliptic curves (ECC). All methods that are integrated in the PKCS#11 Module can be used with appropriate parameters, if such usage is required.

It is required that the application, which calls the PKCS#11 module, is capable of ECC. Furthermore, smart cards have to be used that support ECC. Further information on such smart cards is available on request from support@cryptovision.com.

15.7 Initialization by PKCS#11

Some applications, for instance card management systems, can be used to initialize a smart card by using PKCS#11. In such case there is no need for using the same function of the administration tool.

15.7.1 Supported PIN-lengths

The following minimum and maximum PIN-lengths are available during initialization of a smart card by the PKCS#11 Module:

	User	SO	Admin/Card
ACOS	4/8	8/8	8/8
CardOS	4/10	8/10	10/10
JavaCard	4/10	8/10	10/10 (only cvProfile)
StarCOS	4/8	8/8	8/8

15.7.2 Default values

Challenge response key (crkey) and minidriver compatibility can also be configured if the profile is generating by PKCS#11. There are two options for this configuration:

- cvP11.ini:

This file is part of cv act *sc/interface* and has to be modified. The file has to be stored in the same folder as scManager.exe

- Configuration via Registry:

The following registry keys can be used. They are subject to modification according to special requirements:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface\keys]
"crkey"="0000000000000000000000000000000000000000000000000000000000000000"

[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface\profile]
"userpin"="11111111"
"#cardpin"="sopin"
"cardpin"="0987654321"
"remindpinchange"="true"
"minidriver"="true"
"virtual_slots"=dword:00000002
```

The configuration set via the registry overrides the file scManager.ini.

The file cvP11.ini contains the following values by default. If one of the fields is left blank, these values are used.

		Default value
Challenge Response Key		00
Minidriver compatible		TRUE
Card PIN	ACOS	87654321
	StarCOS	87654321
	Other	0987654321
User PIN		11111111
Remind PIN change		TRUE
Card PIN = SO PIN		Deactivated
Virtual Slots		Deactivated

Some requirements need to be kept in mind:

- Unlike the possible initialization with the administration tool, the SO-PIN is not used. This value has to be provided by the application which launches the PKCS#11 Module.
- The Card PIN and the SO PIN have to be chosen within the smart card specific boundaries.
- If the method C_InitToken is called the option cardpin=sopin leads to using the SO PIN, which is transferred by the calling application, also as card PIN. This option is deactivated by default. It should only be activated, if this absolutely necessary and if notes on the SO PINs are available. Such notes are typically generated by card management systems for smart card management.
- The Challenge Response Key is a two-key TripleDES (ABA) key in case of ACOS smart cards. For all other smart cards it is three-key (ABC) TripleDES key. In both cases the key consists of three times eight hex-bytes, for ACOS the first and the last eight bytes have to be the same. For more information please read <http://msdn.microsoft.com/en-us/library/windows/desktop/bb468064%28v=vs.100%29.aspx>
- "Remind PIN change" can be set to false. In this case no warning will be displayed by the register tool after profile creation if the user does not change his user PIN.
- Starting with cv act *sc/interface* 4.0.1 a profile can be created on smart cards with Java-operating systems which don't use Visa Fixed Keys. In such case the relevant keys have to be included in the file cvP11.ini. One set of keys has to look like this:

```
[javacard]
# VISA-Fixed Keyset
#      enc      mac      kek
keyset=4041...4e4f,4041...4e4f,4041...4e4f
keyset=...
```

15.8 Identification of Smart Card

Every time smart cards and certificates are used, the certificate to be used has to be identified. There is no standardized method for such identification, different application use different identification data.

Beside the data that is provided by the standard configuration of cv act *sc/interface* (among others ATR, label, certificate), the parameter model or modelID can be used in addition by the following configuration. This is necessary for supported applications of Secude if cv act *sc/interface* is used together with smart cards not profiled by cv act *sc/interface*.

The configuration can be applied either by registry keys or by entries in the file cvP11.ini. Below the configuration for smart cards with the operating system StarCOS 3.0 is listed as an example.

15.8.1 Configuration File

In the file cvP11.ini the following entries have to be added:

```
[pkcs11]
# cardid (historical bytes) fixed CK_TOKEN_INFO.model mapping
# model=cardid,model[1-16]
  model=80670412b003030000,3384110107000000      # G&D STARCOS 3.0 contactless
  model=80670412b0030300008105,3384110107000000  # G&D STARCOS 3.0
  model=c808,disabled                             # Disable standard CardOS 4.3B
```

In this case "cardid" must not be mixed up with the value that is displayed as Card ID of a smart card with mini driver compatible profile in scManager. The latter is the GUID which is mentioned in the Microsoft Minidriver specification.

15.8.2 Using Registry

The following registry entries lead to the same result:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface\pkcs11\model]
"80670412b003030000"="3384110107000000"
"80670412b0030300008105"="3384110107000000"
```

Glossary

AES	The AES (Advanced Encryption Standard) has been defined as a standard for symmetric data encryption. It is a block cipher with a block length of 128 bit and key lengths of 128, 192 and 256 bit.
asymmetric Cipher	Encryption procedures employing two different keys (in contrast to a symmetric cipher): one publicly known (public key) for data encryption and one key only known to the message receiver (private key) for decryption.
authentication	Authentication defines a procedure that verifies the identity of the communication partner. The most elegant method is based on the use of so called digital signatures.
Block cipher	An algorithm processing the plaintext in bit groups (blocks). Its alternative is called stream cipher.
CA	see Certification Authority
certificate	see digital certificate
certificate revocation list	A list of revoked certificates issued by a certificate authority
certification authority	An entity responsible for registering and issuing, revoking and generally managing digital certificates
CHAP	(Challenge Handshake Protocol) simple challenge-response protocol for authentication in computer networks
Chip Authentication	<p>Chip authentication (CA) has two functionalities:</p> <ul style="list-style-type: none"> • authenticate the chip and prove that the chip is genuine (not cloned); • establishes a strongly secured communication channel. <p>This protocol and its interaction are described in the technical guideline BSI-TR-03110.</p>
CMC	Certificate Management over CMS. The Certificate Management over CMS (CMC) is an internet standard by the IETF, defining transport mechanisms for the Cryptographic Message Syntax (CMS). It is defined in RFC 5272, its transport mechanisms in RFC 5273.
CRAP	(Challenge Response Authentication Protocol) simple challenge-response protocol for authentication in computer networks
CRL	see Certificate Revocation List
cryptography	In the classical sense, the science of encrypting messages. Today, this notion comprises a larger field and also includes problems like authentication or digital signatures.
CSP	(Cryptographic Service Provider) Cryptographic module used by the Microsoft Crypto API
DES	(Data Encryption Standard) symmetric 64 bit block cipher, which was developed (first under the name Lucifer) by IBM. The key length is 64 bit of which 8 bit serve for a parity check. DES is the classic among the encryption algorithms, which nevertheless is no longer secure due to its insufficient key length. Alternatives are Triple-DES or the successor AES.
digital certificate	A data set that identifies the certification authority issuing it, identifies its owner, contains the owner's public key, identifies its operational period, and is

	digitally signed by the certification authority issuing it.
digital signature	The counterpart of a handwritten signature for documents in digital format. A digital signature grants authentication, integrity, and non-repudiation. These features are achieved by using asymmetric procedures.
DSA	(Digital Signature Algorithm) algorithm for the generation of digital signatures.
ECC	(Elliptic Curve Cryptography) class of procedures providing an attractive alternative for the probably most popular asymmetric procedure, the RSA algorithm.
elliptic curves	A mathematical construction, in which a part of the usual operations applies, and which has been employed successfully in cryptography since 1985.
fingerprint (digital)	Checksum that can be used to easily determine the correctness of a key without having to compare the entire key. This is often done by comparing the hash values after application of a hash function.
hash function	A function which forms the fixed-size result (the hash value) from an arbitrary amount of data (which is the input). These functions are used to generate the electronic equivalent of a fingerprint. The significant factor is that it must be impossible to generate two entries which lead to the same hash value (so called collisions) or even to generate a matching message for a defined hash value. Common hash functions are RIPEMD-160 and SHA-1, each having hash values with a length of 160 bit as well as the MD5, which is still often used today having a hash value length of 128 bit.
HSM	Hardware Security Module.
ICAO	International Civil Aviation Organization.
integrity	The test on the integrity of data is carried out by checking messages for changes during the transmission by the receiver. Common test procedures employ Hash functions, MACs (Message Authentication Codes) or – with additional functionality – digital signatures.
IPsec	A collective name for the architecture and a set of protocols designed to provide encryption and other cryptographic services for Internet Protocol communication.
Kerckhoff's principle	An important principle in the evaluation of cryptographic algorithms. The security of the procedures should not be based on keeping the procedure secret, but only on the used key.
key exchange	The use of symmetric cipher procedures requires that two communication partners decide on one joint key only known to them. The difficulty is that for the exchange of such information usually only partially secure channels exist. Additionally, protocols for key exchange must be prepared in such a way that only those pieces of information are exchanged which do not lead to knowledge of the real secret (the key). The most popular protocol of that type is Diffie-Hellman, whose presentation in 1976 can be regarded as the birth of public key cryptography.
key recovery	This defines the depositing of a key, i.e. the possibility for a superior authority to gain access to the private key of a user. This is usually not desired for the private area, but is useful for the use of cryptography in companies.
LDAP	Lightweight Directory Access Protocol
LDS	Logical data structure. The collection of groupings of data elements stored in the optional capacity expansion technology, defined in [ICAODoc].
MAC	Algorithm that expands the message by means of a secret key by special redundant pieces of information, which are stored or transmitted together with the message. To prevent an attacker from targeted modification of the attached redundancy, requires its protection in a suitable way.

non-repudiation	One of the objectives in the employment of digital signatures. It describes the fact that the sender of a message is prevented from denying the preparation of the message. The problem cannot be simply solved with cryptographic routines, but the entire environment needs to be considered and respective framework conditions need to be provided by pertinent laws.
OCSP	(Online Certificate Status Protocol) Protocol for online inquiries concerning the status of a particular certificate and for responses to such inquiries.
One Time Pad	The only proved secure method to encrypt data. But there is a big disadvantage: The key used must have the same length as the text to be encrypted, and may only be used once. On the other hand, the basic encryption is very simple: The bits of the plaintext are XOR-connected to the corresponding bits of the key receiving the cipher text; repeated XOR-connection reverses this process and leads to decryption.
Passphrase	A long, but memorable character sequence (e.g. short sentences with punctuation) which should replace passwords as they offer more security.
Password	A secret character sequence whose knowledge is to serve as a replacement for the authentication of a participant. A password is usually short to really ensure that an attacker cannot guess the password by trial and error.
PGP	(Pretty Good Privacy) Pretty Good Privacy is a program developed by Phil Zimmermann for encryption and signing of e-mails. It was especially a result of this program that the use of public key procedures became popular after 1994.
PKCS	(Public Key Cryptography Standard) A family of standards issued and supported by RSA Laboratories and is a company standard meant to support the initially difficult problem of product compatibility. The expression comprises a range of different documents, examples are PKCS#1 (for the RSA algorithm), PKCS#7 (for the formats used within cryptography) or PKCS#11 (for a generic interface to cryptographic tokens like e.g. Smartcards).
PKI	refer to Public Key Infrastructure
PKIX	(Public Key Infrastructure X.509) Comprehensive family of standards for PKIs.
Private key	Secret key only known to the receiver of a message, which is used in asymmetric ciphers for encryption or generation of digital signatures.
Pseudo random number	Many cryptographic mechanisms require random numbers (e.g. in key generation). The problem, however, is that it is difficult to implement true random numbers in software. Therefore, so called pseudo-random number generators are used, which then should be initialized with a real random element (the so called <i>seed</i>).
public key	Publicly known key in an asymmetric cipher which is used for encryption and verification of digital signatures.
public key infrastructure (PKI)	Combination of hardware and software components, policies, and different procedures used to manage digital certificates.
random numbers	Many cryptographic algorithms or protocols require a random element, mostly in form of a random number, which is newly generated in each case. In these cases, the security of the procedure depends in part on the suitability of these random numbers. As the generation of real random numbers within computers still imposes a problem (a source for real random events can in fact only be gained by exact observation of physical events, which is not easy to realize for software), so called pseudo random numbers are used instead.
signature (digital)	The counterpart of a handwritten signature for documents in digital format. A digital signature grants authentication, integrity, and non-repudiation. These features are achieved by using asymmetric cryptographic procedures like RSA or ECDSA.

smart card	A smart card is a chip card which contains an internal micro controller with CPU, volatile (RAM) and non-volatile (ROM, EEPROM) memory, i.e. which can carry out its own calculations in contrast to a simple storage card. Sometimes a smart card has a numerical coprocessor (NPU) to execute public key algorithms efficiently. Smart cards have all of their functionality comprised on a single chip (in contrast to chip cards, which contain several chips wired to each other). Therefore, such a smart card is ideal for use in cryptography as it is almost impossible to manipulate its internal processes.
SOD	Document Security Object (stored in EF.SOD). A RFC3369 CMS Signed Data Structure, signed by the Document Signer (DS). Carries the hash values of the LDS Data Groups. It is stored in the MRTD's chip. It may carry the Document Signer Certificate (CDS).
stream cipher	Symmetric encryption algorithm which processes the plaintext bit-by-bit or byte-by-byte. The other usually employed class of procedures comprises so called block cipher.
symmetric cipher	Encryption procedure using the same key for enciphering and deciphering (or, in which these two keys can simply be derived from each other). One distinguishes between block ciphers processing plaintext in blocks of fixed length (mostly 64 or 128 bit) and stream ciphers working on the basis of single characters.
S/MIME	(Secure Multipurpose Internet Mail Extension) Extension of MIME (Multipurpose Internet Mail Extensions) by the functionalities of encryption and message signing, mainly by using PKCS specifications.
terminal authentication	Terminal authentication (TA) is used to determine whether the Inspection System (IS) is allowed to read the sensitive data from an eID document. The mechanism is based on digital certificates. The certificate format is CV (card verifiable) certificates, not standard X.509.
Trust Center	<p>The protocol and its interaction are described in the technical guideline BSI-TR-03110.</p> <p>Trust Center (sometimes also called Trusted Third Party) is an expression for a designated part of a PKI. Normally the task fields of a Trust Center are divided in three different areas:</p> <ul style="list-style-type: none">• The actual certification authority (CA) carries out the actual approval of the information on hand.• The registration authority (RA) is responsible for identification of the participant and the issuance of certificates. <p>The directory service provides the necessary information required for preparation and examination of issued certificates and signatures.</p>
USB-Token	Small device with a USB port. Most of them have the form of a key fob and provide similar functionality to a smart card. Since most equipment today has a USB port, they can save the cost of a card reader.
VPN	(Virtual Private Network) Simulation of a private network by utilizing a public network. In this network, all computer links are encrypted so that every communication is carried out confidentially (privately).
X.509	Standard for certificates, CRLs and authentication services. It is part of the X.500 standard of the ITU-T for realization of a worldwide distributed directory service realized with open system.

Appendix A: Reference for Developers

This appendix contains detailed specifications regarding the supported functions of the PKCS#11-standard, a synopsis of particular functions, and a list of objects and mechanisms. This information is useful and necessary for application developers who want to develop their own applications utilizing the cryptovision PKCS#11 libraries.

Functions according to PKCS#11-Standard

cv act *sc/interface* supports PKCS#11-standard version 2.20. In the following, we list the functions that are not supported at all and the functions supported with deviations from the standard.

These Functions are not supported

```
C_VerifyRecoverInit
C_VerifyRecover
C_SeedRandom
```

These functions are supported with deviations

```
C_Initialize
C_OpenSession
C_GetObjectSize
C_GetTokenInfo
C_CreateObject
C_GenerateKeyPair
```

Synopsis of specific functions

C_Initialize

Parameter:	CK_VOID_PTR_PTR	CinitArg
Description:	Library will be initialized. Slots will be created. Inserted cards are read.	
Deviation:	CinitArg is expected in the format CK_C_INITIALIZE_ARGS. From these the flags are picked out, in particular CKF_LIBRARY_CANT_CREATE_OS_THREADS which decides over Multi-threading. The rest is ignored.	

If C_Initialize is called several times, CKR_CRYPTOKI_ALREADY_INITIALIZED is returned. The number is taken in account.

C_OpenSession

Parameter:	CK_SLOT_ID	slotID
	CK_FLAGS	flags
	CK_VOID_PTR	pApplication
	CK_NOTIFY	Notify
	CK_SESSION_HANDLE_PTR	phSession

Description: Opens a new session on the Slot.

Deviation: Notify and pApplication are ignored and should be set to NULL_PTR. Sessions can only be opened, if a card is inserted.

Special Feature: If a session is opened and then the card will be removed, all sessions on the Slot will return CKR_DEVICE_REMOVED. If there is an error with CKR_DEVICE_REMOVED, CKR_TOKEN_NOT_RECOGNIZED or CKR_TOKEN_NOT_PRESENT a pauseAllSessions is automatically produced on this Slot.
If a paused session is used again, this session will be reopened automatically.

If a card is inserted into or removed from a Slot, then this is an Event (see C_WaitForSlotEvent). If C_OpenSession is called, the Event will be finished, even if the card has been removed and C_OpenSession returned CKR_TOKEN_NOT_PRESENT.

C_GetObjectSize

Parameter:	CK_SESSION_HANDLE	hSession
	CK_OBJECT_HANDLE	hObject
	CK_ULONG_PTR	pulSize

Description: The size of an object will be returned.

Deviation: The returned size is the minimal size of the object that means it doesn't include the size of additional attributes as label or ID. The sizes of private objects are standard deviations.

C_GetTokenInfo

Parameter:	CK_SLOT_ID	slotID
	CK_TOKEN_INFO_PTR	pInfo

- Description:** Returns whether a card is inserted in a slot. If the card is not inserted, CKR_TOKEN_REMOVED will be returned.
- Special Feature:** Inserting or removing a card from a slot is an event (see C_WaitForSlotEvent). If C_GetTokenInfo will be called, the event will be finished, even if the card was removed and C_GetTokenInfo CKR_TOKEN_NOT_PRESENT has been returned.
- The serial number, which is returned by CK_TOKEN_INFO, is not the hardware serial number of the smart card. It is the serial number, which is generated during profile creation. For further information please contact support@cryptovision.com and request the documentation of the extensions of the PKCS#11 module (HW Serial NR).

C_CreateObject

- Parameter:**
- | | |
|----------------------|------------|
| CK_SESSION_HANDLE | hSession, |
| CK_ATTRIBUTE_PTR | pTemplate, |
| CK_ULONG | ulCount, |
| CK_OBJECT_HANDLE_PTR | phObject |

Description: Creates an object following the template.

Deviation: C_CreateObject for RSA keys ("import of RSA keys") heavily differs from the standard use of this function. The specific behavior of the cv act *sc/interface* implementation:

1. Creating only the public key is not supported. Calling C_GenerateObject for such a template (i.e. with entry CKA_CLASS = CKO_PUBLIC_KEY) results in CKR_FUNCTION_NOT_SUPPORTED.
2. Creating the private key requires that you specify prim_p, prime_q (CKA_PRIME_1 and CKA_PRIME_2) and the public exponent (CKA_PUBLIC_EXPONENT) in addition to the modulus and the private exponent. A call to C_CreateObject for a private key template not specifying those values will result in CKR_TEMPLATE_INCOMPLETE.
3. cv act *sc/interface* always creates both (private and public) keys. While the actual call to C_CreateObject uses the template for a private key (CKA_CLASS = CKO_PRIVATE_KEY), the corresponding public key will always be created simultaneously. phObject is a handle for the created private key, there's no handle directly available for the public key.
4. It is always required to specify the private key attributes: CKA_SIGN and CKA_DECRYPT.
5. To set any flags for the public key you have to manually search the key (C_FindObject using for example CKA_ID) and eventually set the values using C_SetAttributes.

Objects

cv act *sc/interface* supports a wide range of PKCS#11 objects. A few of the object attributes have derivations from the standard in their use. This is described in the following sections.

Note: Every PKCS#11 function which generates an object expects a parameter in terms of "CK_ATTRIBUTE_PTR pTemplate". This parameter is used to set the different attributes of the object which is to be generated. If in the following "a template" is mentioned, always this parameter is meant.

Note: For all objects: if in a template CKA_TOKEN is not set, the default value CK_FALSE will be used and only a session object is generated. If the object shall be written on the smart card, so CKA_TOKEN must be set in the template on CK_TRUE.

CKO_CERTIFICATE (CKC_X_509)

cv act *sc/interface* supports certificates of the X.509 format. Thereby are for a few attribute the following derivations:

- CKA_URL: This attribute is not supported. So CKA_VALUE must always be set.
- CKA_VALUE: Because CKA_URL is not supported, this attribute must be set, otherwise the concerning C_CreateObject call returns the error message CKR_TEMPLATE_INCOMPLETE.
- CKA_SUBJECT: If this attribute in the template is not set explicit, it will be generated automatically from the subject field of the certificate which is included in CKA_VALUE.
- CKA_ISSUER: If this attribute is not set explicit tin the template, it will be generated automatically from the issuer field of the certificate which is included in CKA_VALUE.
- CKA_SERIAL_NUMBER: If this attribute is not set explicit in the template, it will be generated automatically from the serial number of the certificate which is included in CKA_VALUE.

C_GenerateKeyPair

Parameter:	CK_SESSION_HANDLE	hSession,
	CK_MECHANISM_PTR	pMechanism,
	CK_ATTRIBUTE_PTR	pPublicKeyTemplate,
	CK_ULONG	ulPublicKeyAttributeCount,
	CK_ATTRIBUTE_PTR	pPrivateKeyTemplate,
	CK_ULONG	ulPrivateKeyAttributeCount,
	CK_OBJECT_HANDLE_PTR	phPublicKey,
	CK_OBJECT_HANDLE_PTR	phPrivateKey

Description:	Generates a public-key/private-key pair.
Deviation:	Unlike other implementations the following attributes must always be specified in addition to the mandatory attributes. <ol style="list-style-type: none">1. Public key template: CKA_ENCRYPT and CKA_VERIFY.2. Private key template: CKA_DECRYPT and CKA_SIGN.3. To create a signature only key, set CKA_ENCRYPT and CKA_DECRYPT to false but CKA_VERIFY and CKA_SIGN to true.

RSA Key Pair (CKO_PRIVATE_KEY and CKO_PUBLIC_KEY with CKK_RSA)

There are two different PKCS#11-functions to create an RSA key-pair: C_GenerateKeyPair ("key generation") and C_CreateObject ("importing a key pair"). The cv act *sc/interface* implementation of the C_CreateObject function differs heavily from the standard use when creating RSA keys. See the description of C_CreateObject in the above chapter [Synopsis of specific functions](#). In either case we will always generate the complete key pair, consisting of both keys (public and private). Creating only one key (be that the private or public key) alone is not supported by cv act *sc/interface*.

- **C_CreateObject:** Whether the key-pair is stored on the token or not is determined by the CKA_TOKEN setting of the private key. Setting CKA_TOKEN = CK_TRUE in the private key template will automatically force the corresponding public key to be written to the token as well.
- **C_GenerateKeyPair:** This function takes two templates as parameters, one for the public and one for the private key. The CKA_TOKEN settings in both templates must match (if present).

Mechanisms

cv act *sc/interface* supports several mechanisms as shown in the following table.

Note: Mechanisms not explicitly mentioned here are not supported. Especially, cv act *sc/interface* does not support any of the mechanisms introduced in the PKCS #11 v2.20 amendments (1-3).

Mechanism	Functions						
	Encrypt / Decrypt	Sign / Verify	SR / VR	Digest	Generate Key	Wrap / Unwrap	Derive
CKM_RSA_PKCS_KEY_PAIR_GEN					X		
CKM_RSA_PKCS	X	X				X	
CKM_RSA_X_509	X	X				X	
CKM_MD2_RSA_PKCS		X					
CKM_MD5_RSA_PKCS		X					
CKM_SHA1_RSA_PKCS		X					
CKM_SHA224_RSA_PKCS		X					
CKM_SHA256_RSA_PKCS		X					
CKM_SHA384_RSA_PKCS		X					
CKM_SHA512_RSA_PKCS		X					
CKM_RIPEMD128_RSA_PKCS		X					
CKM_RIPEMD160_RSA_PKCS		X					
CKM_EC_KEY_PAIR_GEN (CKM_ECDSA_KEY_PAIR_GEN)					X		
CKM_ECDSA		X					
CKM_ECDSA_SHA1		X					
CKM_DES_KEY_GEN					X		

CKM_DES_ECB	X						
CKM_DES_CBC	X						
CKM_DES_CBC_PAD	X						
CKM_DES3_KEY_GEN					X		
CKM_DES3_ECB	X						
CKM_DES3_CBC	X						
CKM_DES3_CBC_PAD	X						
CKM_MD2				X			
CKM_MD5				X			
CKM_SHA_1				X			
CKM_SHA224				X			
CKM_SHA256				X			
CKM_SHA384				X			
CKM_SHA512				X			
CKM_RIPEMD128				X			
CKM_RIPEMD160				X			

Appendix B: Debug

PKCS#11 Logger (win)

Description: Log all PKCS11 function calls to a file. One entry contains the function name, the parameter before and after the function call and the result of the function. Private information is hidden by a static string "[-----]", so only the length is readable.

Settings: The settings are written in the registry at HKEY_LOCAL_MACHINE, Software\cv cryptovision\sc interface. The PKCS11_LogFile_name is the name of the log file.

LogFile_mode can be 0 for off or 1 for on.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface]
    "PKCS11_LogFile_name"="c:\\cvPKCS11_log.txt"
    "LogFile_mode"=dword:00000064
```

Special Feature: A debug library logs private information as plain text.

CSP Logger (win)

Description: Log all CSP function calls to a file. One entry contains the function name, the parameter before and after the function call and the result of the function. Private information is hidden by a static string "[-----]", so only the length is readable.

Settings: The settings are written in the registry at HKEY_LOCAL_MACHINE, Software\cv cryptovision\sc interface. The CSP_LogFile_name is the name of the log file.

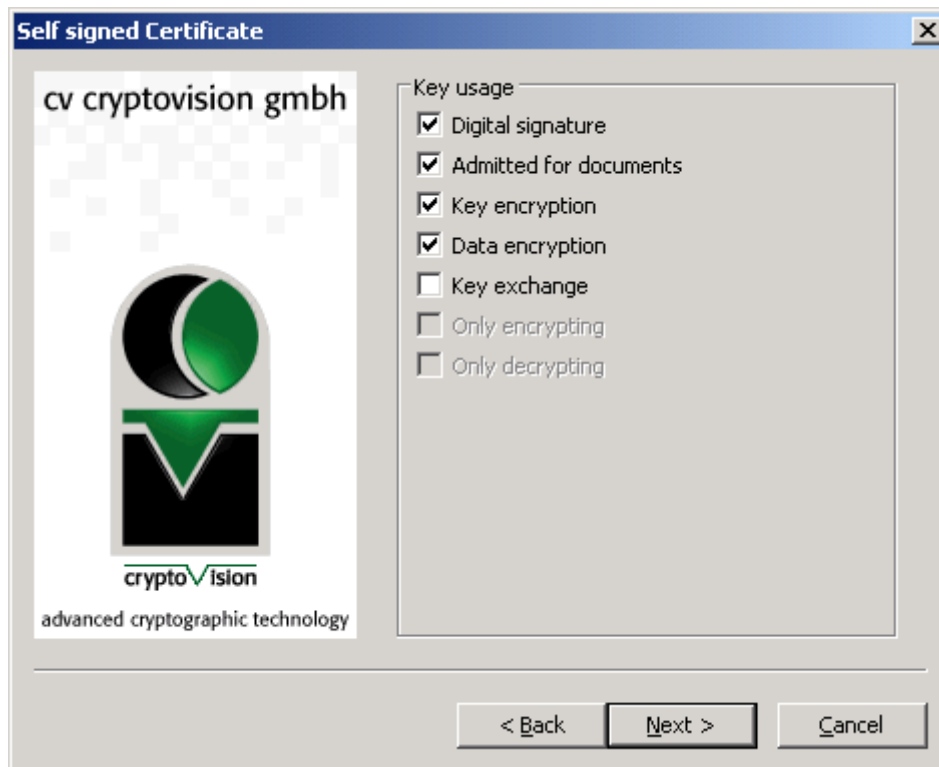
LogFile_mode can be 0 for off or 1 for on.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\cv cryptovision\sc interface]
    "CSP_LogFile_name"="c:\\cvPKCS11_log.txt"
    "LogFile_mode"=dword:00000064
```

Special Feature: A debug library logs private information as plain text.

Appendix C: Certificate Attributes (Key Usage)

The different uses of a key pair are shown by the example of a self-signed certificate:



Note: Please note that this functionality is only for testing purpose.

These are briefly explained in the following:

1. Digital Signature: Here you can verify digital signature, except those with explicitly named purposes e.g. authentication.
2. Authorized for Documents: Here you can verify signatures that check the liability and repudiation of documents (except signatures of certificates and CRLs of CA).
3. Key Encryption: Encryption of keys for the purpose of their transmission.
4. Data Encryption: Encryption of data for the purpose of transmission, but not of keys.
5. Key exchange: Employment of the key to agree on other keys, e.g. a Diffie-Hellman key.

Information / Export Restrictions

Please observe!

The product delivered to you is liable to export control. Please observe the legal requirements of specific countries. For export out of the EU an export approval is necessary. To obtain such approval contact

cv cryptovision GmbH

Munscheidstr. 14

45886 Gelsenkirchen

Germany

© Copyright cv cryptovision GmbH 2002-2013

All rights reserved. Without the express prior written consent of cryptovision you must not distribute, edit or translate copyrighted material.

Trade Mark

All mentioned software and hardware names are in most of the cases trademarks and are liable to legal requirements.