

# Chapter 13

## Additional Graphical Applications

This chapter describes all additional graphical applications that can be accessed via Java clients and complement the functions of i-effect.

### Keystore Administration with the “i-effect Keymanager”

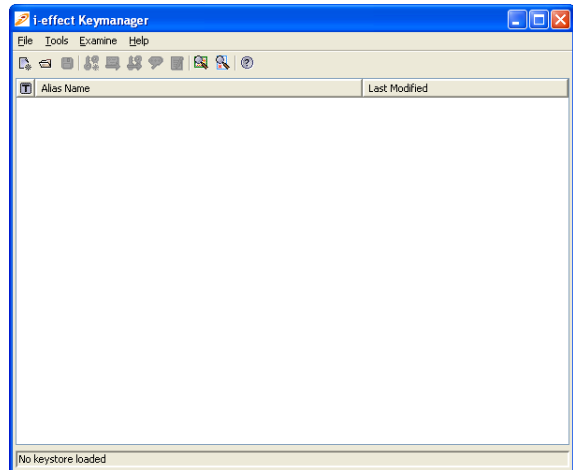
For key pair and partner certificates administration the program “i-effect Keymanager”, included in delivery, can be used. “i-effect Keymanager” is a program to generate, administrate and check keystores, keys, certificates, certificate requests and certificate responses. The program “i-effect Keymanager” is used to administrate key pairs and partner certificates. It is an Open Source program under GNU GENERAL PUBLIC LICENSE and can therefore be used unrestrictedly. After successful installation of i-effect, “i-effect Keymanager” is to be found under the name i-effect-Keymanager.jar in `/i-effect/<version>/CRYPT/tools`.

Version 1.1 offers the following functions:

- o generate, load, save, delete and convert keystores and keystore entries.
- o generate DSA and RSA key pairs with self-signed X.509 certificates
- o import X.509 certificates
- o import key pairs from PKCS#12 (Public Key Cryptography Standards) files
- o password assignment and administration for key pairs and keystores
- o detailed display of keystore certificates and key pairs
- o export function for keystore entries into several formats
- o generate certificate requests (CSRs)

## Starting the “i-effect Keymanager”

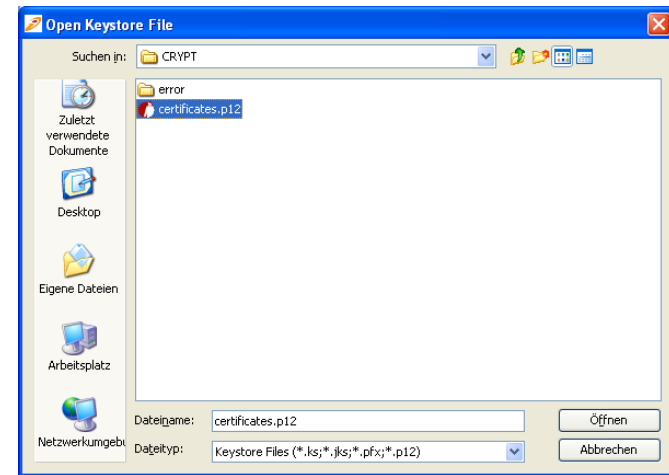
As “i-effect Keymanager” is a Java program, a Java Runtime Environment (JRE) version is needed on the system from where “i-effect Keymanager” is started. If there is no Java Runtime Environment (JRE) version on the system on which “i-effect Keymanager” will be run as keystore administration program, please download the latest Java Runtime Environment at <http://java.sun.com>. After successful installation, “i-effect Keymanager” should start after double click on file i-effect-Keymanager.jar.



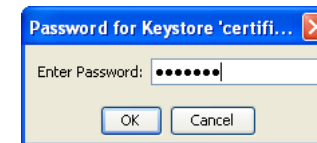
## Start “i-effect Keystore”

The standard keystore of i-effect is found under the name certificates.p12 in directory /i-effect/<version>/crypt. To open it with “i-effect Keymanager” proceed as follows:

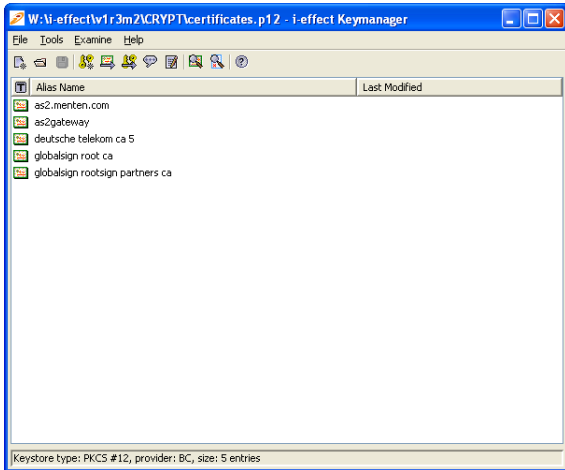
- 1) File -> Open Keystore File...(Ctrl+O)
- 2) In the appearing dialog window, change to directory /i-effect/<version>/crypt. Choose file certificates.p12 and confirm with “Open”.



- 3) To open the keystore enter the requested password. The standard password of the keystore certificates.p12 is “ieffect”.



- 4) After entering the password and confirming with “OK,” the entries in the keystore certificates.p12 file will be shown:



The entries are certificates of:

- o Cybertrust  
(keystore alias: gte cybertrust global root )
- o Deutsche Telekom  
(keystore alias: deutsche telekom ca 4 (gte cybertrust global root)
- o menten GmbH  
(keystore alias: as2.menten.com)

These three certificates form a certification chain (see definition Chapter 7a “Encryption and Advanced Electronic Signature”).

## Generate a Key Pair

- 1) To generate a key pair in a keystore please use  
Tools -> Generate Key Pair...(Ctrl + G)

The following dialog window will appear:



- 2) Select RSA.

RSA The RSA crypto system is an asymmetric crypto system, i.e. it uses different keys for en- and decryption. It is named after its inventors Ronald L. Rivest, Adi Shamir and Leonard Adleman.

DSA The Digital Signature Algorithm is a US government standard for digital signature.

After confirming with “OK,” the following dialog window will appear:



- 3) Enter the certificate details and confirm with "OK". A description of the input fields can be found below. Leave the setting "Signature Algorithm" on "SHA1 with RSA". In the next display, choose an alias name for the generated key pair.



- 4) After successful generation of a key pair, a confirmation dialog window will appear. Click on "OK" to return to the main menu.




---

We suggest a key length of 2048 bits and a validity period of 730 days for the key pair.

---

Description of input fields:

#### Signature Algorithm

The algorithm that is used to calculate the signature.

#### Validity (days)

Validity period of the generated certificate.

#### Common Name (CN)

When importing, the common name will be proposed as alias name. Many Certificate Authorities require the domain name to be used as CN as it is unambiguous. If you want your certificate to be signed by a Certificate Authority, please familiarize yourself with the specifications before generating certificates.

#### Organization Unit (OU)

The unit (e.g. branch, office ) of the organization (e.g. company, agency). This field should only be filled in if your organization has more than one unit.

#### Organization Name (O)

Name of the organization.

#### Locality Name (L)

Locality of the organization.

#### State Name (ST)

Name of the state where the organization is located.

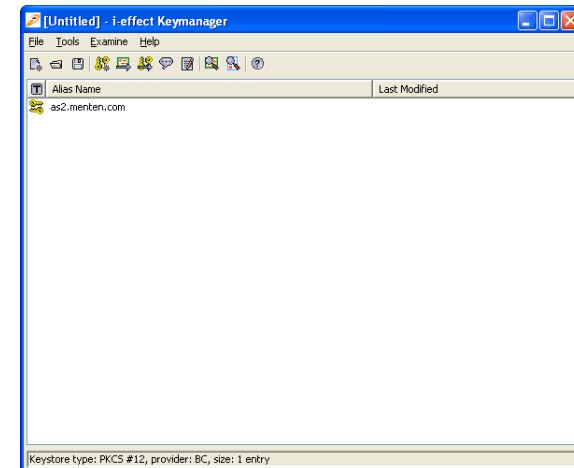
#### Country (C)

Country where the organization is located.

#### Email (E)

Email address of the organization.

The keystore now contains a valid key pair:



- 5) Because of adding a key pair and, therefore, changing the keystore, it is necessary to save the modifications with

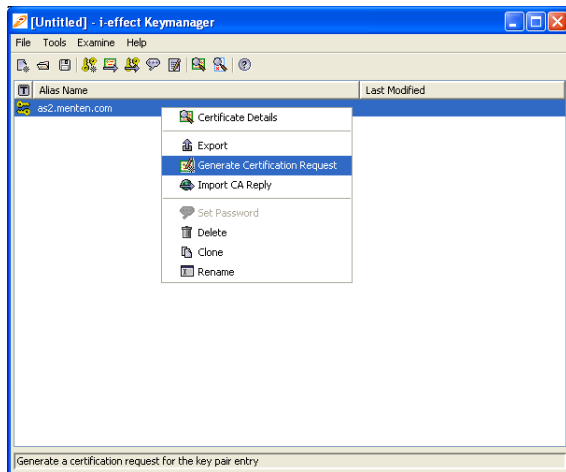
*File -> Save Keystore (Ctrl+S)*

The generated key pair contains both a private and a public key. In order to exchange encrypted AS2 messages with a partner, it is necessary to send the public key. The public key can be exported in the form of a certificate. This will be further explained in the course of this documentation.

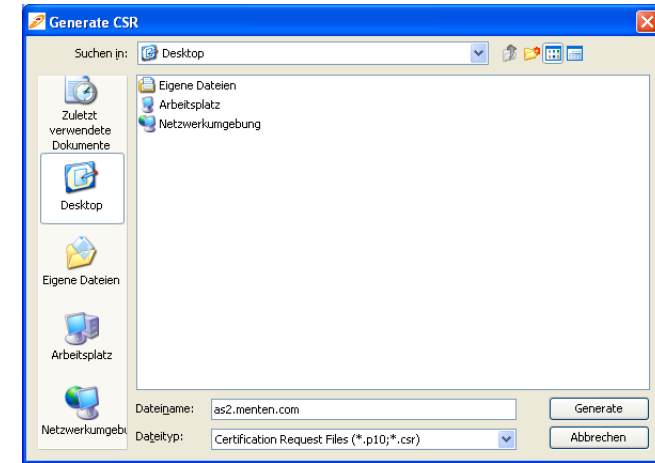
## Generate a certificate request

To have your certificate signed by a higher authority (Certificate Authority), i.e. to have the certificate authenticated, a certificate request must be generated. Proceed as follows:

- 1) Move the mouse pointer to the entry of the key pair and right-click. In the displayed context menu select menu item "Generate Certification Request".

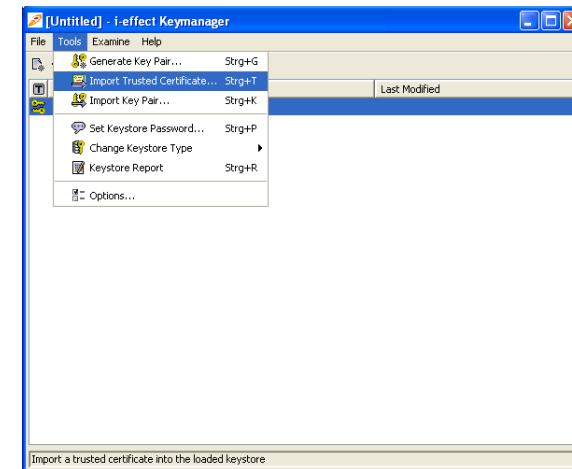


A "Save As" dialog window will open as pictured below.



- 2) Select a folder and type in the title of the certificate request. Please note: One of the two file extensions (\*.p10 ; \*.csr), which are proposed under file type, must be added. Confirm with "Generate".

To have your certificate signed by a Certificate Authority, the generated certificate request file must be transmitted to a Certificate Authority. After successful validation, the Certificate Authority will send back a signed certificate, which has to be reimported into the keystore.



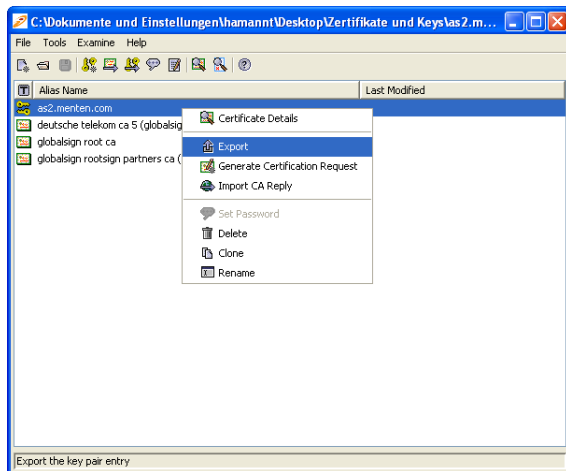
The newly generated certificate and the certificate signed by the Certificate Authority can now be sent to your partners. The generated certificate is considered trustworthy. However, the certificate signed by a Certificate Authority must exist in your partner's keystore before your certificate is imported.

## Export a Certificate

In order to successfully exchange files with a partner using all security mechanism that are available in AS2, it is necessary to send your partner the public key in the form of a certificate. To generate a certificate that contains the public key, it has to be exported from the key pairs as follows:

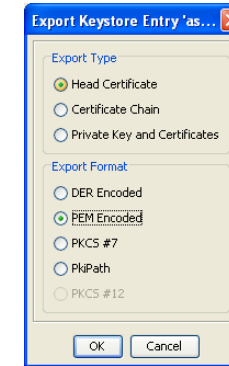
- 1) Move the mouse pointer to the entry of the key pair and right-click.

The following context menu will appear:



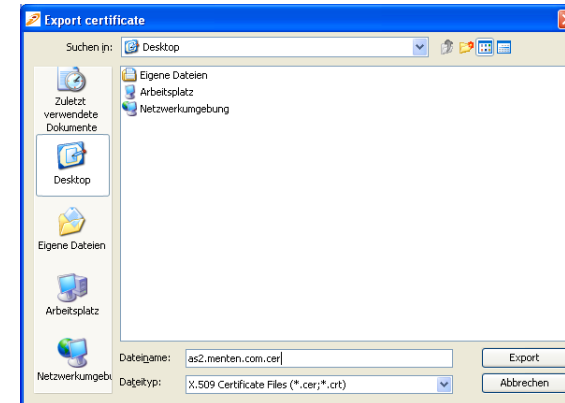
- 2) Please select the menu item "Export".

The following dialog window will appear:



- 3) Change the export format from DER Encoded to PEM Encoded and confirm with "OK" to export the public key as certificate.

A dialog window to save the certificate will open.



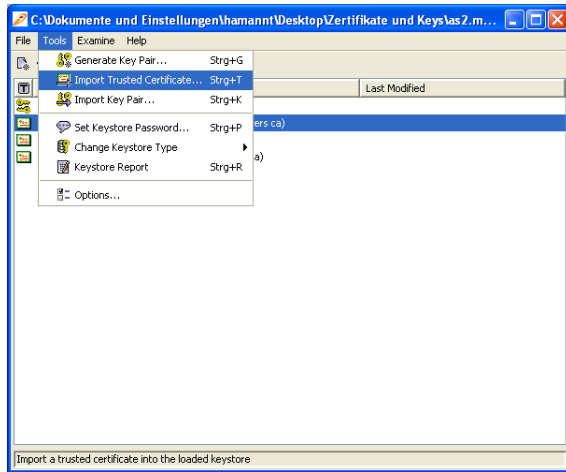
- 4) Choose a folder and type in a name for the certificate. Please note: One of the two file extensions (\*.cer ; \*.crt), which are proposed under file type, must be added. Confirm with "Export".

## Import of Partner Certificates

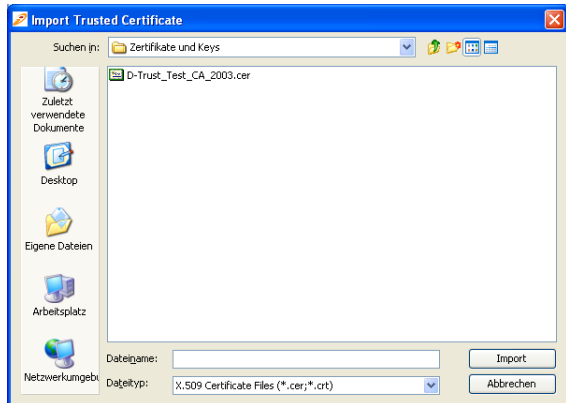
To encrypt files in AS2 messages, which will be sent to your partners, it is necessary that the certificates containing the partner's public key exist in the keystore. To import your partner's certificates to the keystore, proceed as follows:

- 1) *Tools -> Import Trusted Certificate...(Ctrl+T)*

The following context menu appears:



- 2) In the following dialog window please go to the directory in which your partner's certificate is filed. Select the certificate and confirm with "Import".

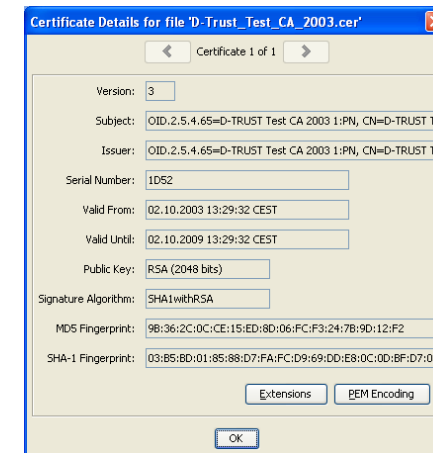


When importing certificates, two possible situations may occur.

First, it may happen that the imported certificate is part of a certification chain but the root certificate (see definition Chapter 7a "Encryption and Advanced Electronic Signature") or other certificates depending on it do not exist in the keystore. In this case, or if a self-signed certificate is concerned, the following "i-effect Keymanager" advice will occur:



This advice also explains the described situation. To validate your partner's certificate, after confirming the dialog window with "OK," the certificate details will be shown. The following screenshot shows such a detail display.



With the help of this detail display, the partner's certificate can be verified in order to decide whether to trust the certificate or not, which needs to be confirmed in the following dialog window.



After confirming with “YES” and giving the certificate an alias name in the next dialog window, the certificate will be saved in the keystore. “i-effect Keymanager” will automatically suggest the certificate’s common name as alias name. The certificate can be saved under this name or a self-chosen name. The alias name can always be changed without great effort.



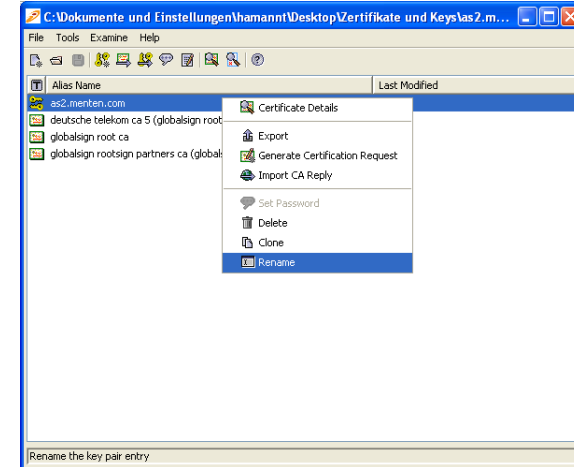
The second situation that may occur when importing certificates, is the following: If the certificate that is to be imported was signed by a Certificate Authority (and is therefore part of a certification chain) and if the root certificate of the Certificate Authority and all dependent certificates exist in the keystore, a validation will not be required. Nor will the confirmation dialog window (Import Trusted Certificate), which is shown above, occur, because the certification chain is complete with the dependent certificates. The certificate is automatically considered trustworthy.

## Change an Alias Name

- 1) To change an alias name of a certificate that is saved in the keystore, proceed as follows:

Move the mouse pointer to the certificate whose alias name is to be changed and right-click.

The following context menu will appear:



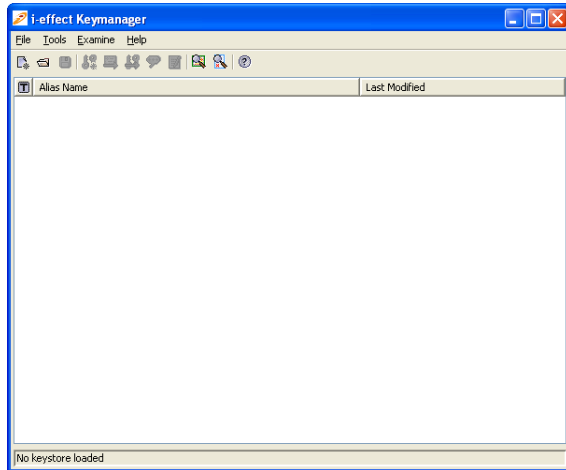
- 2) Select the menu item “Rename”. A dialog window as shown below will appear. Enter the new alias name and confirm with “OK”.



## Generate a new Keystore

- 1) Start "i-effect Keymanager".

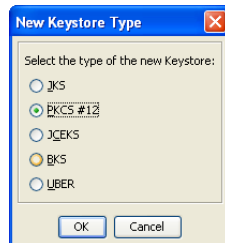
A dialog window as shown below will appear:



- 2) To generate a new keystore, proceed as follows:

*File* -> *New Keystore (Ctrl + N)*

The following dialog window will appear:



The following keystore formats are supported:

- o JKS: Java Keystore (Sun Keystore Format)
- o PKCS#12: Public Key Cryptography Standards #12 Keystore (RSA's Personal Information Exchange Syntax Standard)
- o JCEKS: Java Cryptography Extension Keystore (More secure version of JKS)
- o BKS: Bouncy Castle Keystore (Bouncy Castle's version of JKS)
- o UBER: Bouncy Castle UBER Keystore (More secure version of BKS)

- 3) Select the keystore format PKCS#12 and confirm with "OK" to return to the main menu. Now the newly generated keystore can be saved as follows:

*File* -> *Save Keystore / Save Keystore (Ctrl + S)*

The creation of a password for the newly generated keystore will be requested.



- 4) After entering and confirming the password, the generated keystore can be saved in the iSeries file system. (Please do not forget to add the file extension according to the chosen keystore format.) After saving, "i-effect Keymanager" returns automatically to the main menu.

