cv act *PKIntegrated* V3.0

Administration Guide

Contents

1 Introduction

1.1 About cv act PKIntegrated

Thank you for choosing cv act PKIntegrated as your strategic platform for certificate management.

cv act PKIntegrated is an advanced PKI solution completely integrated into Novell eDirectory. It makes use of Novell Identity Manager as event system to trigger CA-relevant commands, and of Novell SecretStore Services to protect access to sensitive keys. Building on top of the extensible management framework of Novell iManager, cv act PKIntegrated provides role-based administration with fine-graded access control.

This makes cv act PKIntegrated a powerful and flexible, still lean and cost effective PKI solution overcoming the need to learn a new management interface, deploy and integrate another repository and manage a new security concept.

1.2 Deploying cv act PKIntegrated

Deploying an integrated product into a live system requires a good understanding not only about the product itself, but also about the existing infrastructure and technology.

cv cryptovision has deployed many enterprise-wide implementation of cv act PKIntegrated and has the experience to integrate 3rd party technologies and solutions.

Deploying cv act PKIntegrated without fully understanding the impact to your production environment can result in unplanned downtime, partial or complete loss of information and serious damage to your infrastructure, especially, but not limited, to your Novell eDirectory and Identity Management System.

We strongly recommend deploying cv act PKIntegrated in a testing environment and making extensive tests before installing into any production system.

1.3 How to use this Guide

This Administration Guide is designed to help you with administrative tasks of cv act PKIntegrated.

This guide gives detailed step-by-step instructions for an environment based on SLES10, Novell eDirectory 8.8, Novell iManager 2.7, Novell SecretStore 3.3.3 and Novell Identity Manager 3.0.1. If you work in a different environment, some instructions may be obsolete or functions are named differently. Please visit www.novell.com/documentation for product documentation of Novell Software.

For a better understanding, we added examples and screenshots for many administrative steps. In-text examples are highlighted in grey color and will likely not match your environment. For security reasons we ask you kindly to not use any of the passwords given as examples.

If you have any feedback, please don't hesitate to contact us. Contact details are listed on our homepage, http://www.cryptovision.com.

1.4 cv act PKIntegrated Components

1.4.1 Overview

cv act PKIntegrated comes with 6 components:

- ca/server
- dir/connector
- admin/extension
- ocsp/responder
- scep/responder
- eDirectory Schema extension

1.4.2 ca/server

This is the core CA component which runs on Linux. The ca/server executes all CA related commands sent from dir/connector.

The base functions include:

- setup of a CA key pair and a corresponding root certificate
- generation of a key pair
- creation of a certificate
- prolongation and update of a certificate
- revocation of a certificate
- maintenance of a certificate revocation list (CRL)
- email notification of specific events

1.4.3 dir/connector

The dir/connector component is an IDM driver. It reacts on certain eDirectory events and calls the ca/server component. The events are triggered by modifying LDAP attributes using admin/extension or by any other LDAP utility. The following events are currently supported:

- CA Create
- CA Activate
- CA Update
- CA Cross Certification
- Key Generation
- Key Update
- Certificate Request
- Certificate Update
- Certificate Revocation
- Certificate Suspend
- CRL Update

1.4.4 Admin/extension

admin/extension defines the front-end user interface for the certificate management. It is realized as a plug-in for Novell's iManager.

1.4.5 ocsp/responder

Novell eDirectory has built-in LDAP (Lightweight Directory Access Protocol) support to access certificates and certificate revocation lists. Linux-based cv act PKIntegrated ocsp/responder enhances Novell eDirectory with OCSP (Online Certificate Status Protocol) functionality.

1.4.6 scep/responder

SCEP (Simple Certificate Enrollment Protocol) automatically issues, distributes, updates and blocks certificates for VPN-Routers. scep/responder receives a request from network devices, and responds with a generated IPSec-Certificate. cv act PKIntegrated supports SCEP via its scep/responder.

1.4.7 eDirectory Schema extension

cv act PKIntegrated makes use of the flexible schema provided by Novell eDirectory. The schema extension for cv act PKIntegrated follows LDAP attribute syntax and has been registered and carries a valid ASN.1 number: 1.3.6.1.4.1.6522.

The schema extension of cv act PKIntegrated follows the Development Guidelines of Novell.

1.5 What is new in cv act PKIntegrated

The following new features have been added to cv act PKIntegrated 3.0:

- JCE module support for Utimaco HSMs
- cv act PKIntegrated is now available for Windows platforms
- Bug fixes and Browser compatibility enhancements

Administration

1.6 admin/extension

1.6.1 iManager Role Description

The admin/extension consists of four pre-defined iManager roles:

Role	Description	Tasks
cv act PKIntegrated Certificate Management (Administrator)	Contains tasks for managing mandator and CAs The purpose of this role is to assign it to a PKIAdmin user in the tree.	Create CA Configure CA Activate CA Cross Certify CAs Create Cross Certificate Pair Export CA Request Import CA Certificates Update CA Certificate List CA Certificates Preselect CA Mandator Installation Info
cv act PKIntegrated Certificate Management (Operation)	Contains tasks for generating and revoking certificates for user objects. The purpose of this role is to assign it to an CA operator in the tree.	Administer Certificates Centralized Key Generation Certificate Signing Request (CSR) Certify Browser Generated Key Key Generation and Certification List CA Certificates List Certificates Local Key Generation (PKCS#12) Preselect CA Mandator Renew Certificates Renew Keys Revoke Certificates
cv act PKIntegrated Certificate Management (User)	Contains tasks for generating and revoking certificates for the own user object. The purpose of this role is to assign it to all users in the tree.	Administer Certificates Centralized Key Generation Certificate Signing Request (CSR) Certify Browser Generated Key Key Generation and Certification List CA Certificates List Certificates Local Key Generation (PKCS#12) Renew Certificates Renew Keys Revoke Certificates
cv act PKIntegrated Certificate Management (SCEP)	Contains tasks for managing SCEP requests. The purpose of this role is to assign it to a PKIAdmin user in the tree.	List CA Certificates List Certificates Manage SCEP Requests Revoke Certificates

Task assignments to roles and role assignments to users can be configured in iManager to match your custom needs.

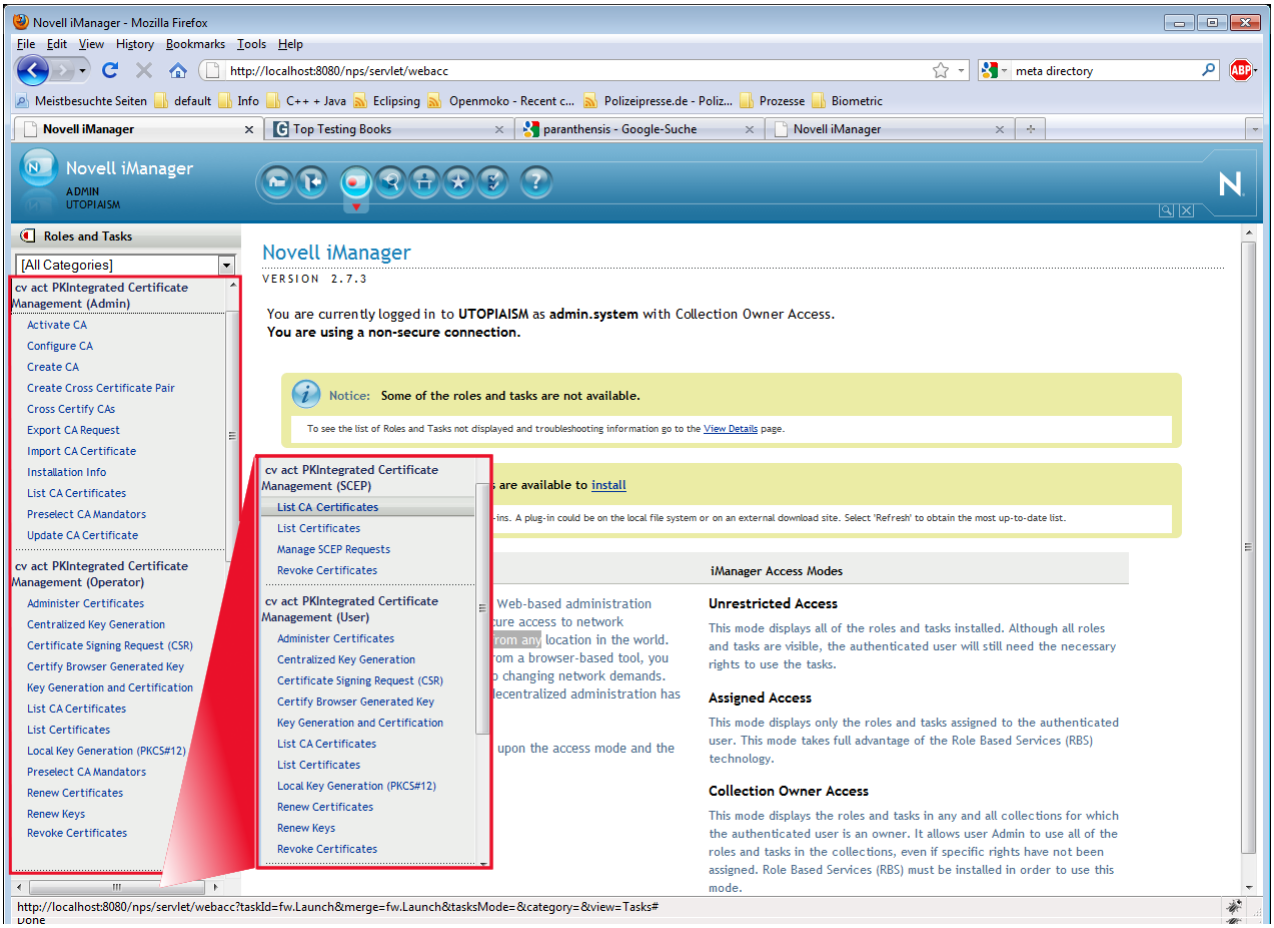

Figure 0.1: iManager Roles and Tasks

1.6.2 iManager Task Description

The admin/extension (rbs Module cv) consists of 19 iManager tasks:

Task **(rbsTask Object name)**	**Description**	**Role: cv act PKIntegrated Certificate Management**
Create CA (AdminCreateCA)	Task to create a new Certification Authority	Admin
Configure CA (AdminConfigCA)	Task to configure the attributes of a Certification Authority	Admin
Activate CA (AdminActivateCA)	Task to activate one of multiple Certificate Authorities	Admin
Cross Certify CAs (AdminCrossCertificationTask)	Task to cross certify internal CAs	Admin
Create Cross Certificate Pair (AdminCrossCertificate-PairTask)	Task to cross certify an external CA	Admin
Export CA Request (CARequestExport)	Task to export a PKCS#10 request	Admin

Task (rbsTask Object name)	Description	Role: cv act PKIntegrated Certificate Management
Import CA Certificates (CACertificateImport)	Task to import a CA certificate	Admin
Update CA Certificate (AdminUpdateCA)	Task to update a CA certificate	Admin
List CA Certificates (CACertificateList)	List of all mandator with all their CAs	Admin Operator User SCEP
Installation Info (InstallationInfo)	This task shows actual installation details.	Admin
Administer Certificates (AdminCertificateListTask, CertificateListTask)	Task to administrate certificates	Operator User
Centralized Key Generation (AdminCertificationGenerateTask, CertificationGenerateTask)	Task to generate a private key pair on the HSM	Operator User
Certificate Signing Request (CSR) (AdminCertificationPkcs10Task, CertificationPkcs10Task)	Task to certify a public key from a signing request	Operator User
Certify Browser Generated Key (AdminCertificationBrowserTask, CertificationBrowserTask)	Task to generate a browser key	Operator User
Key Generation and Certification (AdminCertificationTask, CertificationTask)	This task offers a selection of all certification options	Operator User
List Certificates (AdminCertificateListViewTask, CertificateListViewTask)	Task to list all certificates Operators may work on certificates of multiple objects	Operator User SCEP
Local Key Generation (PKCS#12) (AdminCertificationAppletTask, CertificationAppletTask)	Task to generate a private key pair on the client side	Operator User
Preselect CA Mandator (MandatorTask)	Task to select a mandator which will be used in the following tasks	Admin Operator
Renew Certificates (AdminCertificateListCertificateUpdateTask, CertificateListCertificateUpdateTask)	Task to renew a certificate Operators may work on certificates of multiple objects	Operator User
Renew Keys (AdminCertificateListKeyUpdateTask, CertificateListKeyUpdateTask)	Task to renew keys Operators may work on certificates of multiple objects	Operator User

Task (rbsTask Object name)	Description	Role: cv act PKIntegrated Certificate Management
Revoke Certificates (AdminCertificate-ListRevokeTask, Certificate-ListRevokeTask)	Task to revoke certificates Operators may work on certificates of multiple objects	Operator User SCEP
Manage SCEP Requests (AdminEnrollSCEPCert)	Task to approve SCEP Certificate Requests	SCEP

1.6.3 iManager Tasks and eDirectory Rights

When assigning an iManager Role to a user, additional eDirectory Rights need to be granted to the user to be able to process the involved tasks. The figures help you to distinguish which rights are required to process the corresponding task.

Every object using the CA needs read-rights on the CA list, CA objects, and the certificate repositories.

For generating certificates, the rights shown in the following figure have to be assigned.

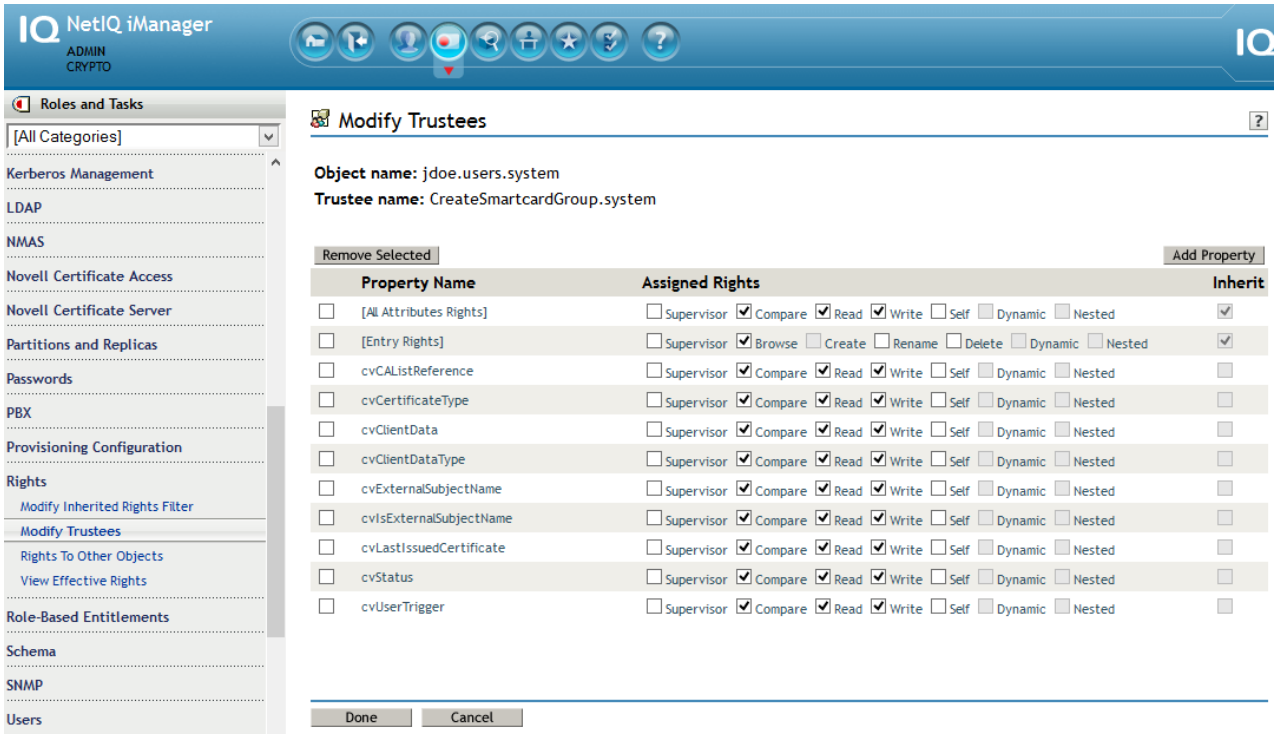

Figure 0.2: eDirectory rights for generating certificates

Write-rights to “Object class” are necessary, if admin/extension shall automatically extend the user with auxilliary class cvUserAttribAux. Otherwise this has to be done using a driver (e.g. during importing the user or workstation objects). cvExternalSubjectName is only needed, if a certificate template (e. g. for SSL server certificates) is used that requires this attribute.

Additionally it is necessary to assign write-rights to the attributes cvRepositoryTrigger and cvTriggerParamCRLReason in the certificate repository (class `cvIssuedCertificate`) if the PKI Admin wants to use the admin/extension task “Repository Tasks”.

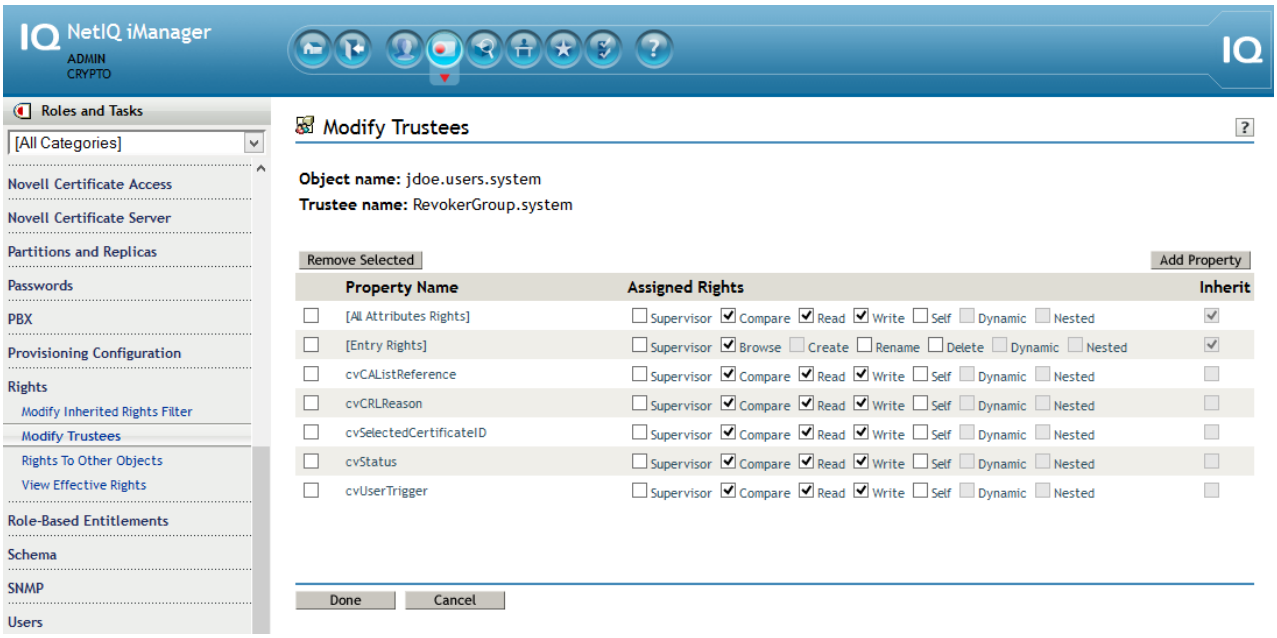

Figure 0.3: eDirectory rights for revoking certificates

For revoking certificates, the rights shown in figure 2.3 have to be assigned.

1.7 Certificate Types

cv act PKIntegrated provides several certificate templates, which can be used to generate certificates. These certificate templates are added as customized templates. New templates can be added as needed.

1.7.1 Standard Types

The following certificate types are supported by cv act PKIntegrated in the default configuration:

Certificate Type	Description
sign	Certificate used for email signing
encr	Certificate used for encrypting data or emails
auth	Certificate used for client authentication (SSL client)
ocsp	Certificate used by OCSP servers
serv	Certificate used by SSL servers (e.g. Web Server)
ipsc	IPSec certificates (client and server)
code	Certificate used for code signing (Authenticode)
scep	Certificate used by SCEP servers
scauth	SmartCard authentication certificates for use with an Active Directory server. Please note that the attribute “cvUniversalPrincipalName” has to be set. The setting of this attribute could be done with an IDM policy in an AD-driver.
ecdsa-prime256v1 ecdsa-secp384r1	Suite B ECDSA algorithms.
domaincontroller	Domain Controller Certificate.
efsrecovery	EFS Recovery Certificate.
subca	Sub CA Certificate.
xxxx-temp	To support a 1-day valid certificate, add the postfix -temp to the certificate type (e.g. auth-temp)

All certificate types available on the product CD will be installed as customized templates. Here some additional information about some of these types:

- OCSP
  The OCSP template provides certificate types with the IDs “ocsp” and “ocsp-temp” (thus overriding the built-in type “ocsp”). The OCSP template provides an OCSP server certificate with the “OCSP noCheck” flag set.
- SubCA
  The SubCA template provides the certificate type with the ID “subca”. If cv act PKIntegrated is used as root CA and a certificate for a subordinate CA (SubCA) should be issued, an appropriate template is needed. To create SubCA certificates you should create a proxy-user object which represents the SubCA and set cvNameOverwriteAllowed in the CA object to “subca”.

- ECDSA
  This template provides the certificate types with the IDs “ecdsa-prime256v1”, “ecdsa-prime256v1-temp”, “ecdsa- secp384r1” and “ecdsa- secp384r1-temp”. You can use these templates to create certificates for signing e-mails using ECC keys.

1.7.2 Customized certificate templates

The functionality of the ca/server can be extended using customized certificate templates.

1.7.2.1 Implementation and installation

Customized templates have to be stored in the folder “catemplates”, which has to be created in the folder with the IDM driver’s Java files (e.g. `/opt/novell/eDirectory/lib/dirxml/classes` on SLES10 with IDM3.6). After a restart of dir/connector and remote loader (if applicable) certificates can be created based on these customized templates.

The customized templates have to be implemented in Java. The source code of a sample implementation is available upon request from support@cryptovision.com.

1.7.2.2 iManager Configuration

After a customized template is implemented and installed, it is available in iManager. If the name of the template that is displayed in iManager should differ from the internal name of the template, the file …/iManager/nps/portal/modules/cv/configuration/certificateTypes.xml has to be extended or modified.

If the internal name (the id) of the customized template is “cust” and the name “customized template” should be displayed in iManager, the following lines have to be added:

```
<CertificateType id="cust" enableKeyLength="true">
      <Label locale="en" default="true">Customized Template</Label>
      <Label locale="de" >Angepasste Vorlage</Label>
</CertificateType>
```

1.7.2.3 JAVA Requirements

All customized certificate templates have to be compiled using the same Java version that is used by IDM.

The Unlimited Strength Java Cryptography Extension Policy Files have to be installed in the Java installation of IDM (e.g. in folder `/opt/novell/eDirectory/lib/nds-modules/jre/lib/security` on SLES10 with IDM3.6). These policy files are available from website of Sun (http://www.oracle.com/technetwork/java/index.html). Please make sure to use the correct version depending on the JAVA version that is used by IDM.

In addition the Unlimited Strength Java Cryptography files have to be installed in the Java installation of iManager.

1.8 CA Monitor

If you did not include the passphrase for your CAs in the CA configuration file you have to start a monitor client on the CA server. This can be done by executing the script startMonitor located in the /opt/cryptovision/bin directory. If a passphrase is needed the CA operator will be prompted on the monitor to enter the passphrase.

1.9 Mandator Selection

As described in the Installation Guide there must be defined at least one mandator. Each iManager task contains a mandator selection box at the top of the page.

A mandator usually has one active CA which is used if a request is executed. If you want to operate a Root CA and a Sub CA you have create two mandator container objects. After that you can create the Root CA selecting the Root CA mandator and the Sub CA selecting the Sub CA mandator in the Create CA task. Then configure each CA and activate these CAs. From now on the CAs are ready for operation.

Usually your users should not select the mandator. You can constrain the mandator selection for the user tasks by editing the iManager.xml (see Installation Guide) configuration file of the admin/extension.

```
<Configuration>
        <Parameter>
              (…)
              <MandatorsPath>Mandators.PKIntegrated.system</MandatorsPath>
              <EnableUserMandators>true</EnableUserMandators>
              <UserMandator>Sub.Mandators.PKIntegrated.system</UserMandator>
        </Parameter>
        (…)
</Configuration>
```

1.10 CA Management

1.10.1 Create Certification Authority

When you create a Certification Authority with iManager, you will create a CA key pair, a CA root certificate, an empty CRL and an empty DeltaCRL.

Before the CA can be used to sign certificates, it must be configured and activated.

- In iManager, select Role "cv act PKIntegrated Certificate Management (Administrator)", Task "Create CA"
    - Fill in the Object Attribute Values

Object Attribute (eDirectory Attribute name)	Description	Example
Manadator	Mandator of this CA.	
Name (CN)	Unique Name of the Certification Authority object (CA object).	TEST-CA
Context	eDirectory Context of CA object.	ou=PKIntegrated. ou=IT.o=system
Repository Name (cvRepository ListReference)	Unique Name of the repository of this CA (Repository object). All Certificates are stored as cvIssuedCertificate objects in the Certificate Repository of the CA.	TEST-CA Repository
Certificate Repository Context	eDirectory Context of Repository object. This might be a sub-container of the CA Context or any other container within your tree structure.	ou=PKIntegrated. ou=IT.o=system
CA SubjectName (cvCADN)	Subject name of Certification Authority in X.500 naming format. A maximum of 64 characters is allowed! Important note: if you want to include “State” as part of the CA DN please use “ST=” (not “S=”).	cn=TEST-CA.cv.com, o=cv cryptovision gmbh,c=de
Key Algorithm (cvCAAlgorithm)	Key Algorithm of CA keys. Value could be RSA or ECDSA. ECDSA is not available, if PKCS#11 is used to store the private root key (see cv act PKIntegrated V2.6 Installation Guide, chapter 2.4.4)	RSA
Named Curve (cvCAAlgorithm Parameter)	Key curve of CA keys. Will be shown if Key Algorithm ECDSA selected.	Prime256v1
Key Length (cvCAKeyLength)	Key length of CA keys. Will be shown if Key Algorithm RSA selected. Numeric value between 512 and 8192 bit in 256 bit steps. Default: 2048 Not all third party products are certified for a key length >2048 bit. Please verify that your environment is capable of handling key lengths >2048 bit.	2048

Object Attribute (eDirectory Attribute name)	Description	Example
Hash Algorithm (cvCAHash Algorithm)	Hash Algorithm of CA certificate.	SHA1
Validity (days) (cvCAValidity Period)	Validity of Certification Authority Numeric value between 1 and 65535 days (~180 years). Default: 2914 (8 years) Expiration date = <current date and time of server> + validity in days	2914
CRL validity (seconds) (cvCRLValidity Period)	CRL validity period in seconds. Numeric value between 1 and 2147483648 seconds (~68 years). Default: 86400 Some reference values: 3600 sec = 1 h 86400 sec = 24 h 604800 sec = 7 days	86400

- Click OK

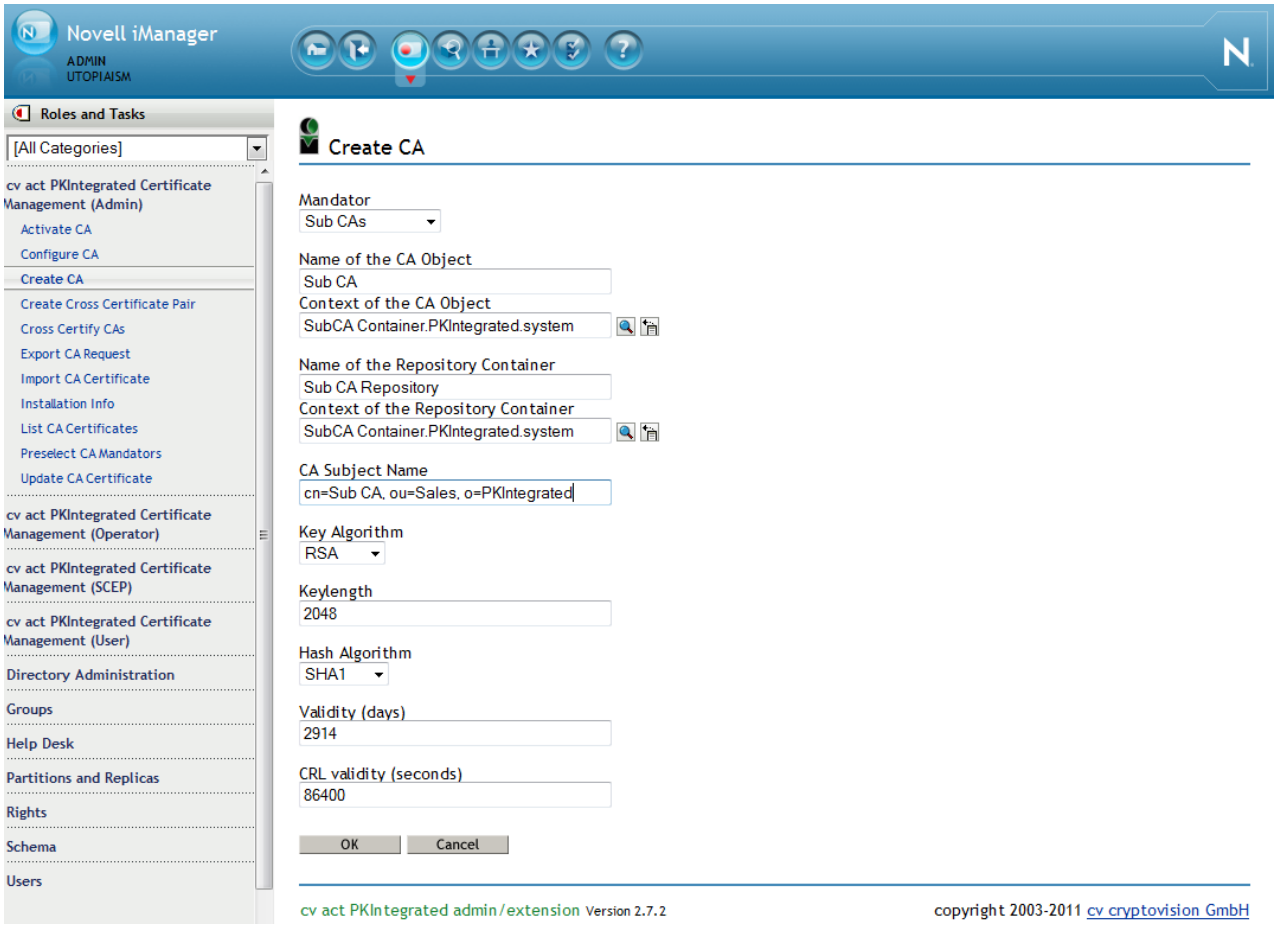

Figure 0.4: Create CA

- After a few seconds you should receive a success message like this. If the CA could not be created within 60 seconds, you receive an error message informing you about a

time-out.

- From the success message screen, you can continue to configure or activate your new CA.

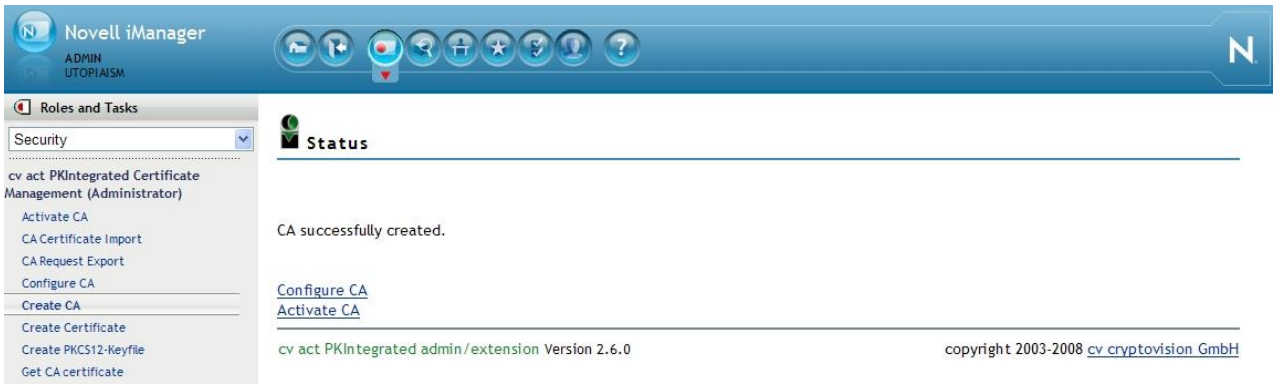

Figure 0.5: Create CA was successful

1.10.2 Configure Certification Authority

After CA Creation you have to configure the CA before it can be used.

Before the CA can be used to sign certificates, it must be configured and activated.

- In iManager, select Role "cv act PKIntegrated Certificate Management (Administrator)", Task "Configure CA"
  - Select the CA you want to configure in the selection box
  - Fill in the Object Attribute Values

Attribute (eDirectory Attribute name)	Description	Example
Certificate Types for decentralized Key Generation (cvRequestType)[1, 2]	Defines the certificate types available for key generation on the client. In this case, private keys are stored only on the client, ca/server only creates a certificate, key recovery is not possible.	auth, sign
Certificate Types for central Key Generation (cvGenerateType)[1, 2]	Defines the certificate types available for key generation by ca/server, private keys are stored in secret store, key recovery is possible, private keys and certificates can be downloaded via iManager.	encr
Multiple Allowed Certificate Types (cvMultipleCertificates Allowed)[1, 2]	Defines the certificates types the user can request more than once without revocation of the existing certificate of the same type.	auth
Certificate Types With Arbitrary DN (cvNameOverwriteAllowed) [2]	Defines the certificate types for which the distingued name must be defined manually during request.	serv, ocsp, scep
Certificate Types for Key Recovery (cvKeyRecoveryType)[1, 2]	Defines the certificate types for which the keys are stored within the secret store. All certificate types mentioned here have to be defined in cvGenerateType and must not be included in cvRequestType.	encr
Certificate Types for pki/roamer Keys (cvRoamerType) [1, 2]	Defines the certificate types for which a key in pki/roamer format should be created. All certificate types mentioned here have to be defined in cvGenerateType and must not be included in cvRequestType. This setting is only relevant if you have cv act pki/roamer installed.	encr
Validity period for User certificates (days) (cvValidityPeriod) [1, 2]	Default validity of Certificates Numeric value between 1 and 65535 days (~180 years). Default: 2914 (8 years) Expiration date = <current date and time of server> + validity in days	730

Attribute (eDirectory Attribute name)	Description	Example
CRL validity (seconds) (cvCRLValidityPeriod) [2]	CRL validity period in seconds. Numeric value between 1 and 2147483648 seconds (~68 years). Default: 86400 Some reference values for your convenience: 3600 sec = 1 h 86400 sec = 24 h 604800 sec = 7 days 2419200 sec = 4 weeks	86400
CRL distribution point (cvCRLLDAPUrl) [2]	Full qualified URL of the address where the CRL is available.	ldap://ldap.-cryptovision.-com/cn=testCA,o=cryptovision?-certificate-RevocationList
Delta CRL distribution point (cvDeltaLDAPUrl) [2]	Full qualified URL of the address where the Delta CRL is available.	ldap://ldap.-cryptovision.-com/cn=testCA,o=cryptovision?-deltaCertificate-RevocationList
OCSP Server URL (cvOCSPHTTPUrl) [2]	Full qualified URL of the address where the OCSP server is available.	http://ocsp.-crypto-vision.com/ocsp-responder
Key Length (cvKeyLength) [2]	Defines the default key length in bits for certificates with central key generation. Numeric value between 512 and 8192 bit in 256 bit steps. Default: 1024	1024
Minimal Key Length (cvMinKeyLength)[2]	Defines the minimum key length in bits for client-generated user certificates. Numeric value between 512 and 8192 bit in 256 bit steps. Default: 1023	1023
Maximal Key Length (cvMaxKeyLength)[2]	Defines the maximum key length in bits for client-generated user certificates. Numeric value between 512 and 8192 bit in 256 bit steps. Default: 2048	2048
Hash Algorithm (cvHashAlgorithm)	Defines the hash algorithm for user certificates	SHA1
Preferred Certificate (cvCACertificate)	The default certificate of the CA. If a PKCS#12 key file is created the default certificate of the CA will be included.	

[1] This attribute can also be set for a user object to overwrite CA settings.

[2] Any change will only be reflected in new certificates or CRLs.

- Click OK

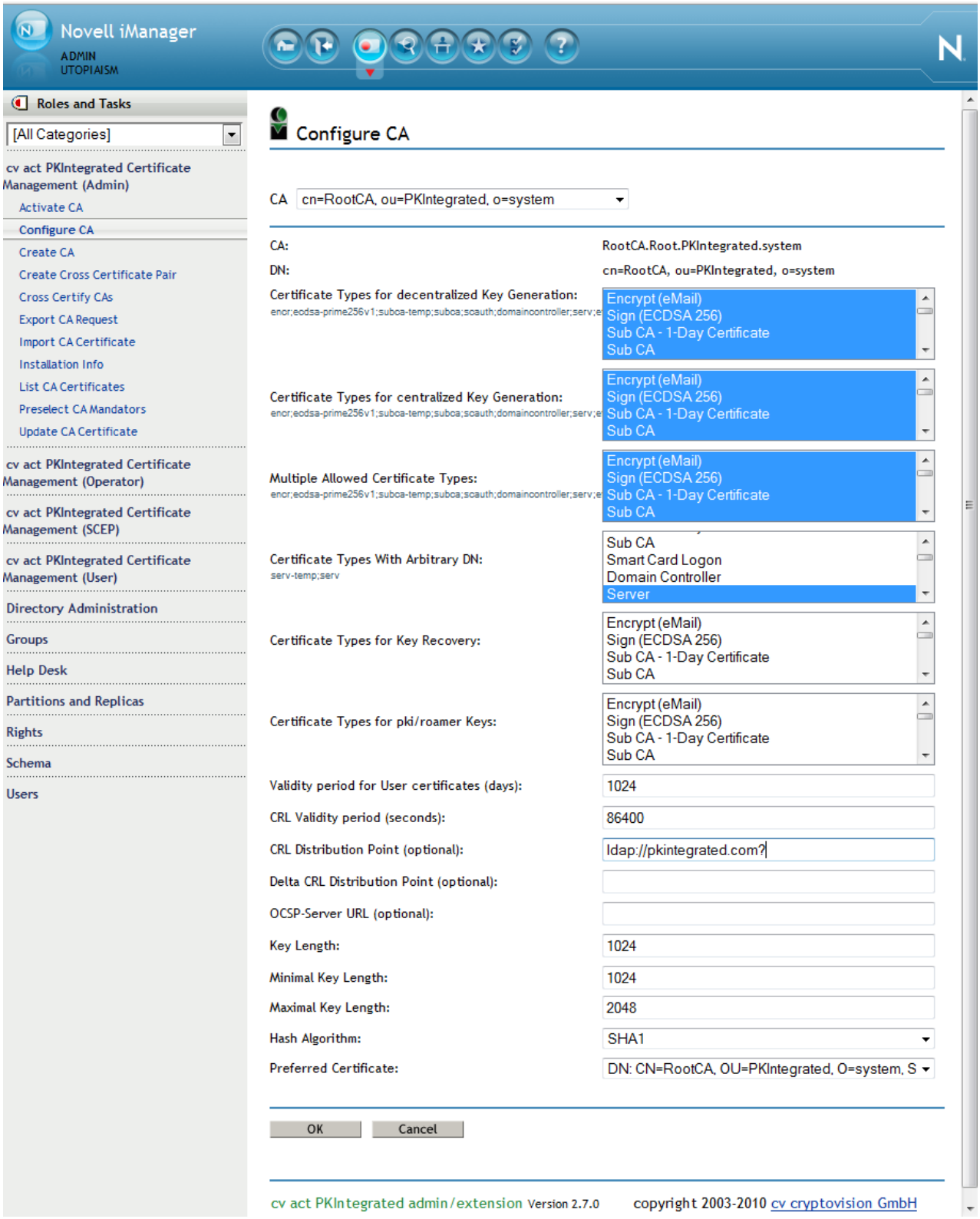

Figure 0.6: Configure Certification Authority

1.10.3 Manually updating the CRL

Usually all CRLs will be updated automatically within the defined period (see Installation Guide). To enforce an update of the CRL you have to set the attribute cvCRLTrigger in the appropriate CA object to “CRLUpdate”.

eDirectory Attribute name	Description	Example
cvCRLTrigger	This attribute is used to trigger events for the Certification Authority. Valid entries for manual CRL update is: CRLUpdate	CRLUpdate

1.10.4 Optional LDAP attribute mappings

You could consider an LDAP mapping for the following eDirectory attributes. External applications require this change to enable certificate validation and CRL checking for cv act PKIntegrated certificates and CRLs. By default, these mappings are configured for the Novell Certification Server. Changing these mappings will most likely disable validation of certificates issued by the Novell Certificate Server via LDAP.

- In iManager, select Role "LDAP", Task "LDAP Options"
    - Select the LDAP Group the LDAP server(s) belong to cn=LDAP Group. o=system

Figure 0.7: LDAP Options

- For the LDAP attribute certificateRevocationList (and certificateRevocationList;binary), change the eDirectory attribute to cvCertificateRevocationList
- For the LDAP attribute deltaRevocationList (and deltaRevocationList;binary), change the eDirectory attribute to cvDeltaRevocationList

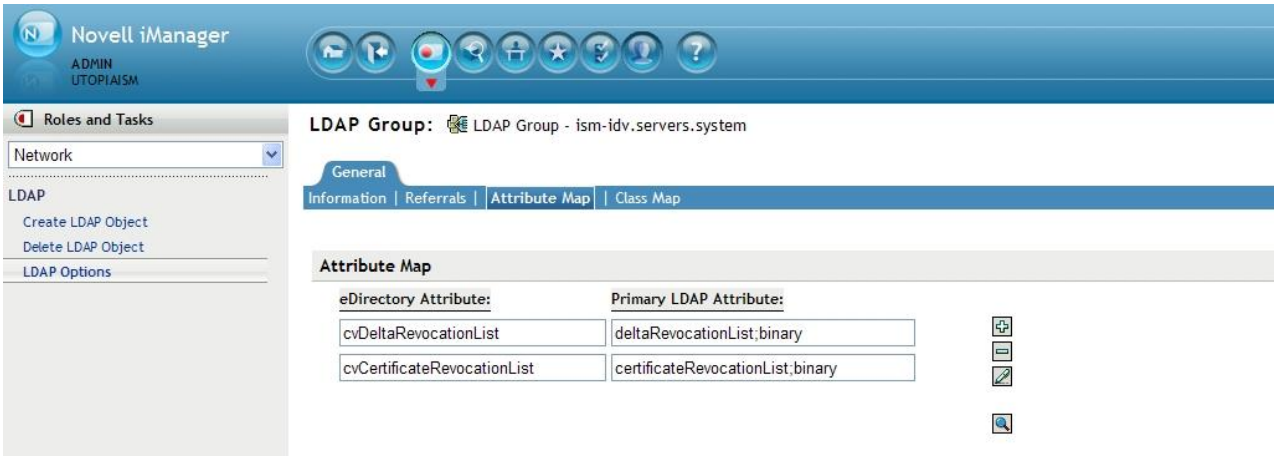

Figure 0.8 LDAP attribute mappings

1.10.5 Download CA Certificates and CRLs

You can use iManager to download the certificates and CRLs of the CA. For your environment it could be useful to publish the CA Certificate via the corporate Intranet, a public web server or distribute it automatically to all clients.

- In iManager, select Role "cv act PKIntegrated Certificate Management (Administrator)", Task "List CA Certificates"
    - This will display a list of all cv act PKIntegrated mandator with their CAs within your eDirectory tree, both active (green dot) and inactive (red dot). Make sure to select the appropriate certificate or CRL and click on the icon.

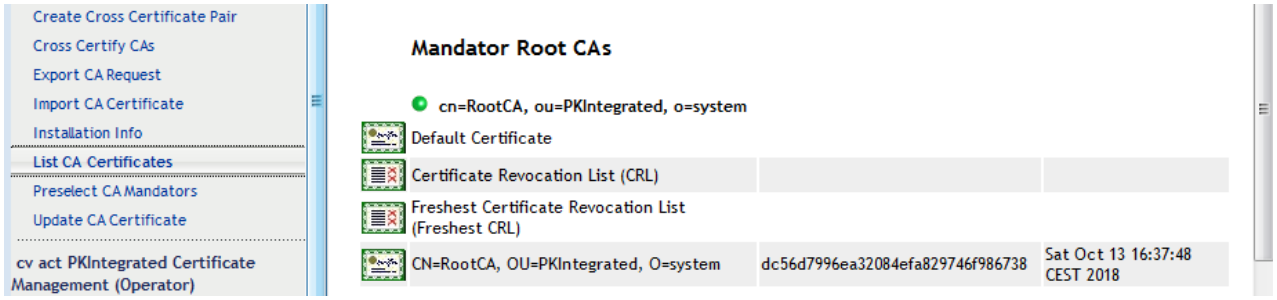

Figure 0.9: Download CA Certificate

1.10.6 Activate Certification Authority

A CA needs to be activated before it can be used. Only one CA of a mandator can be activated. CRLs will be created even if a CA is inactive.

- In iManager, select Role "cv act PKIntegrated Certificate Management (Administrator)", Task "Activate CA"
    - Each mandator can have not more than one active CA. Review the current active CAs (green and latin font).
    - Select the CA you want to activate cn=TEST-CA.ou=PKIntegrated,o=system
    - OK

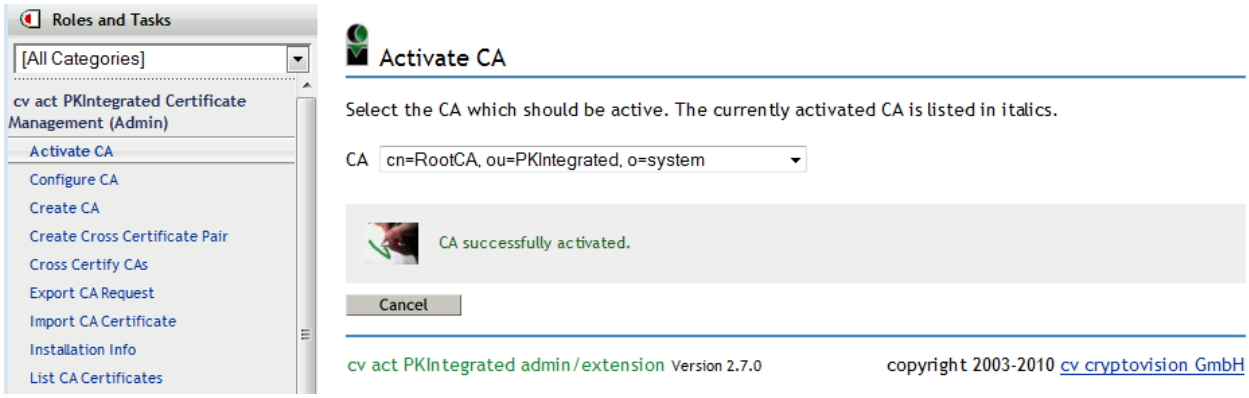

Figure 0.10: Activate CA was successful

1.10.7 Update CA

A CA needs to be updated when the CA certificate is about to expire. Updating a CA will create a new CA certificate with an extended "valid to" date (current date + cvCAValidityPeriod). The “valid from” date is not changed in the new CA certificate.

- In iManager, select Role "cv act PKIntegrated Certificate Management (Administrator)", Task "Update CA Certificate"
    - Select the CA you want to update cn=TEST-CA.ou=PKIntegrated,o=system
    - OK

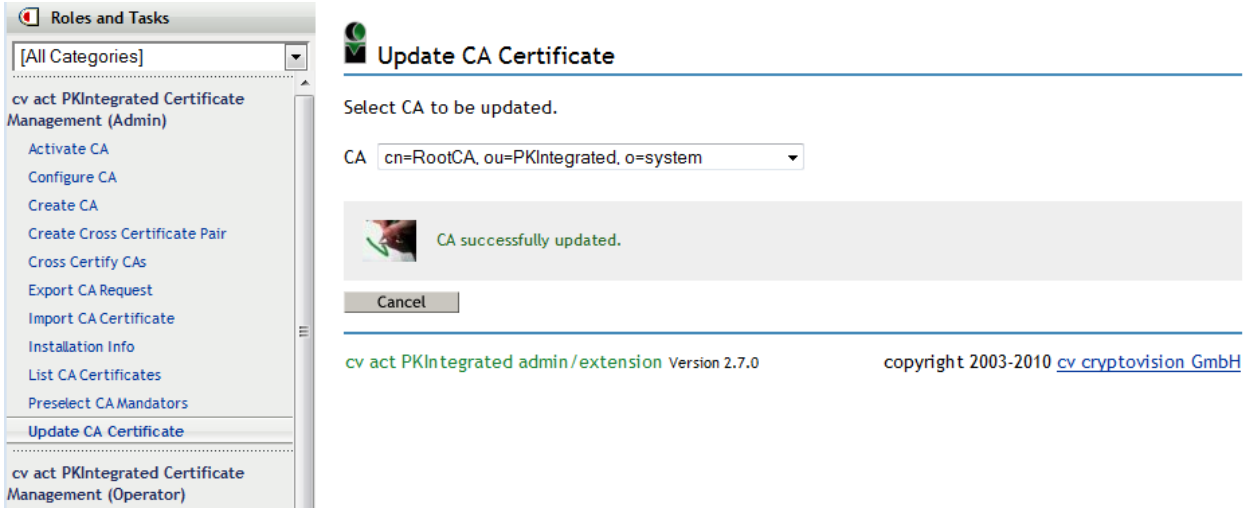

Figure 0.11: Update of CA was successful

1.10.8 PKIntegrated as a Subordinate CA

cv act PKIntegrated CA can be configured as a subordinate CA. Upon the creation of the CA a PKCS#10 request for the key pair of the root certificate is generated. This request can be downloaded and stored in a file by using the iManager task “Export CA Request”. This file has to be sent to the CA that should serve as root CA. The root CA will compute the request and generate a certificate for a subordinate CA. This certificate has to be stored in the CA object by using the iManager task “Import CA Certificate”. If this certificate should become the default certificate of the CA then check the box “Replace default certificate” or open the “Configure CA” task and select the certificate there.

1.10.9 Cross Certify CAs

If you want cross certify two internal CAs, e.g. to build CA chains of the internal CAs on your own, then you can do this by running the “Cross Certify CAs” task. Select two CAs to cross certify them. After you clicked on ‘Okay’, two new certificates will be created.

You can use the cross certificates for building certificate chains: Select the CA which shall become the subordinate CA in the “Configure CA” task and select the certificate which is signed by the CA who is assigned to become the Root CA as the default certificate. From now on this certificate will be included in every new created PKCS#12-file (Task “Local Key Generation (PKCS#12)“).

1.10.10 Create Cross Certificate Pair

Select the “Create Cross Certificate Pair” task und upload the certificate or put the Base64 encoding of the certificate into the text filed. Starting the task will create a cross certificate pair that will be stored in the crossCertificatePair attribute of the CA object. Download the cross certificate pair by clicking on the icon.

1.10.11 Backup CA

Because most references are stored in Novell eDirectory, the backup procedure of cv act PKIntegrated is a combination of eDirectory and file backup.

- For eDirectory backup, please consult your Novell System Administrator or Integrator on how to make a successful backup of the eDirectory database including Novell SecretStore. For dir/connector, a file export of the driver configuration or a backup of your Designer project is helpful.
- For file backup, please consult your Linux OS Administrator or Integrator on how to make a successful backup of the directory /opt/cryptovision. This directory contains all the configuration files including the private key of the CA. Protection of your backup files is strongly recommended.
- If a hardware security module (HSM) is used contact the vendor of the HSM for information on backup of HSM contents (especially private key of the root certificate of the cv act PKIntegrated).

1.10.12 Restore CA

If you need to restore your CA from a disaster or you want to migrate your CA to a new hardware, proceed as follows:

- Restore your eDirectory tree from backup if all replicas of any partition are not available. Check eDirectory tree health before continuing.
- Restore your Novell SecretStore Server, if affected
- Restore your Novell iManager Server, if affected
- Restore your IDM server including driver configuration, if affected
- Re-Install cv act PKIntegrated 2.6 (see Installation Guide)
- Restore /opt/cryptovision on your Linux System

1.11 Certificate Policies

1.11.1 User Certificate Policy

The CA object defines the default certificate policy for certificate type, validity period and key length. Whereas it is useful to restrict network users from requesting non-user certificates, different settings might be useful for the PKI Administrator of the Network.

These instructions will explain how to overwrite the default Certificate policy settings on a user level. These policy settings are not available for group or container objects. User policy settings always override CA policy settings.

The admin/extension currently has no plug-in for managing certificate policy settings for user objects. Therefore you may have to extend the user object with the Auxiliary Class cvUserAttribAux first, before you can configure the attribute values.

- In iManager, select Role "eDirectory Schema", Task "Object Extensions"
    - Browse to and Select the user object cn=PKIAdmin.ou=PKIntegrated,o=system
    - Add cvUserAttribAux from available auxiliary class extensions (might already exist)

Figure 0.12: User Extension with Auxiliary Class cvUserAttribAux

    - close

- In iManager, select Role "eDirectory Administration", Task "Modify Object"

- Browse to and Select the user object
  cn=PKIAdmin.ou=PKIntegrated,o=system

- Select Tab "Other"
- If one of the attribute you want to configure is listed in the unvalued attribute list, add the attribute before you configure the attribute value.
  add cvAllowedCertificateType
  add cvGenerateType
  add cvKeyRecoveryType
  add cvMultipleCertificates Allowed
  add cvRequestType

Attribute (eDirectory Attribute name)	Description	Example
Certificate Types for decentralized Key Generation (cvRequestType)	Defines the certificate types available for local key generation.	auth, sign, serv, ocsp, scep
Certificate Types for central Key Generation (cvGenerateType)	Defines the certificate types available for CA key generation.	encr
Multiple Allowed Certificate Types (cvMultipleCertificates Allowed)	Defines the certificates the user can request more than once.	auth, serv, ocsp, scep
Certificate Types for Key Recovery (cvKeyRecoveryType)	Defines the certificate types for which the keys are stored within the secret store. This must be the same list as defined in attribute cvGenerateType and must not contain the same elements as defined within attribute cvRequestType	encr
Certificate Types for pki/roamer Keys (cvRoamerType)	Defines the certificate types for which a key in pki/roamer format should be created This setting is only relevant if you have cv act pki/roamer installed.	encr
Validity period for User certificates (days) (cvValidityPeriod)	Defines the validity period in days for certificate.	730
Key Length (cvKeyLength)	Defines the default key length in bits for certificates with central key generation.	1024

2 Certificate Management

2.1 CA Operator and User Role

There are four roles defined in the default installation: The administrator role, operator role, the user role and the SCEP administrator role.

The certificate management is done by the operator and the user role. The difference between these roles is that the operator always have to select an entity (usually a user or workstation) before he can execute the selected task. A user always works in its user object context.

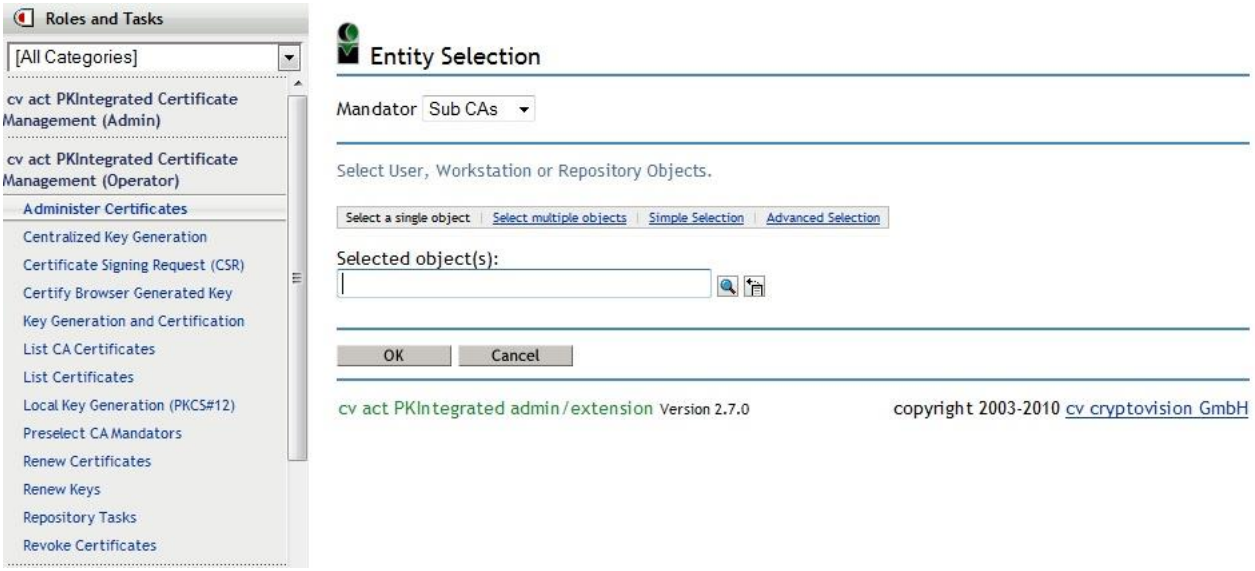

Figure 3.1: Entity selection dialog of a task using the operator role

2.2 Certificate Management

2.2.1 View Certificate

If a certificate icon appears on a page you can click on it and a detail view of the certificate content will be shown.

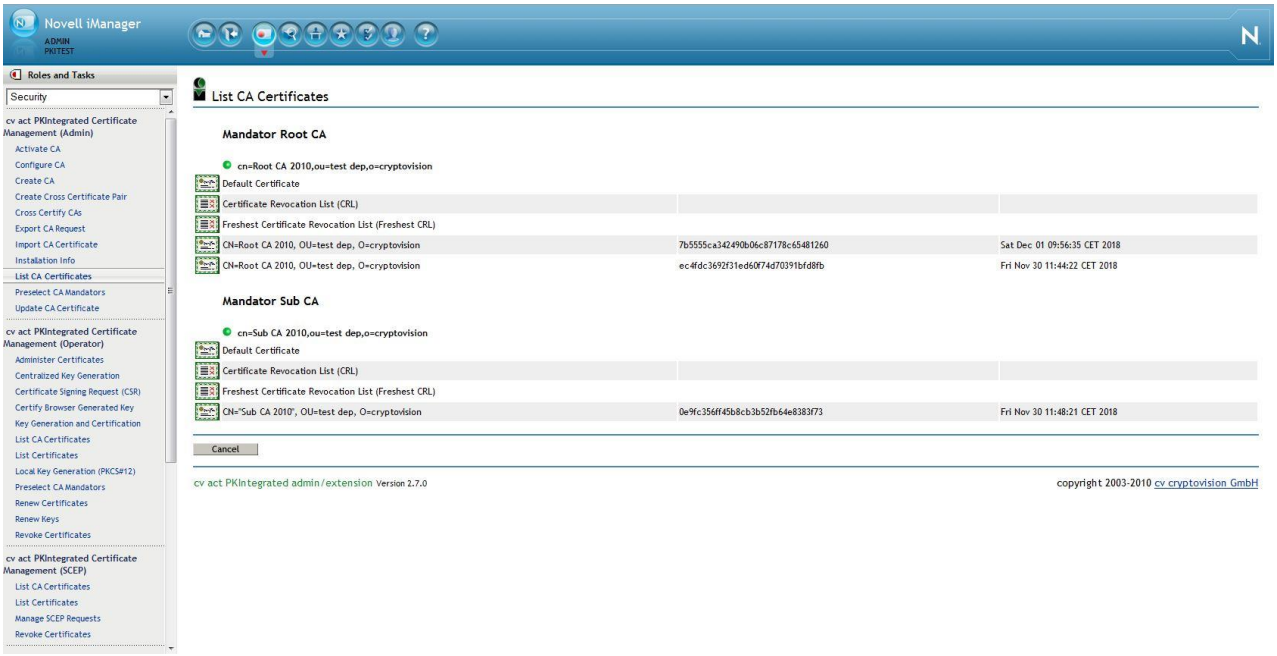

Figure 3.2: List of CA Certificates

You can always download the certificate by selecting the download button.

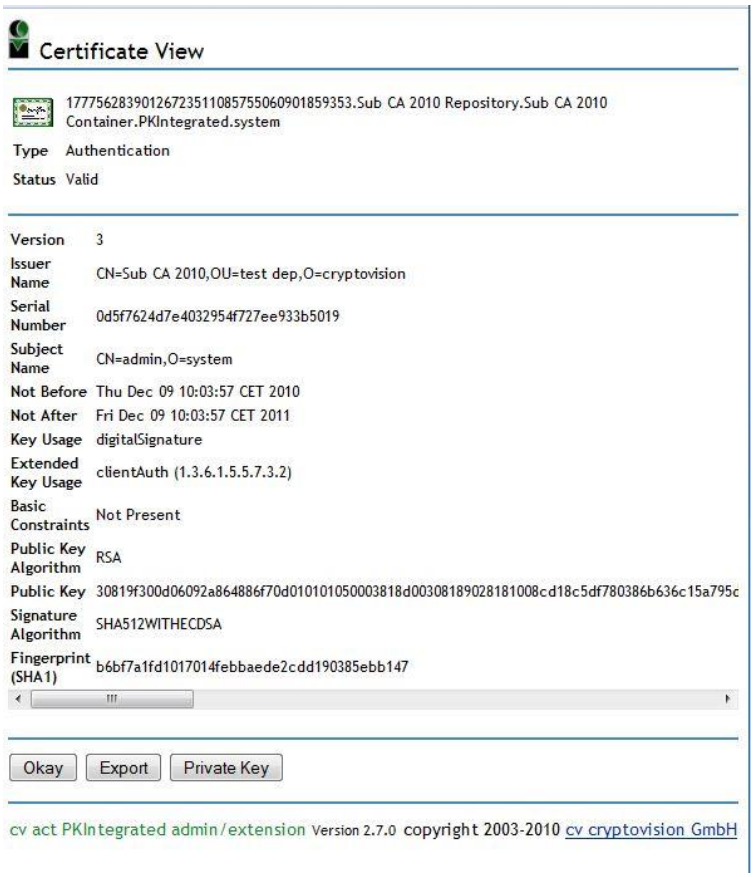

Figure 3.3: Certificate Deteil View

There may be other actions shown on the dialog depending on the context:

- Download Private Key

  This button will be shown if the private key is available. This is usually the case if a user is logged in who has access to its SecretStore or a recovery admin is logged and a recovery key exists in its SecretStore.

- Import Certificate

  This button will not be shown in Internet Explorer. If you believe that this certificate belongs to a browser generated key pair then you can click on this button and the certificate will be installed in the browser. Otherwise an error message will be shown by the browser.

The dialog will be closed by selecting the okay button.

2.2.2 Centralized Key Generation

Centralized key generation means that the private key pair will be generated on the server side. Centralized key generation allows key recovery if key recovery is configured for the selected certificate type within the CA configuration.

The configuration settings of the Certification Authority and the certificate template will be used for key generation. No further attributes need to be defined.

Figure 3.3: Centralized Key Generation

The CA will generate the private key pair and certifies the public key. The certificate and an encrypted PKCS#12 container with the private key and the certificate will be returned.

The PKCS#12 container will be stored in the SecretStore of the entity that was used to trigger the generate request. This ensures that only the (functional) user has access to the private key. Optionally key recovery might be activated for the certificate template so that the SecretStore administrator is able to access the private key when needed.

The passphrase of the PKCS#12 container will be printed by the configured print service of the CA.

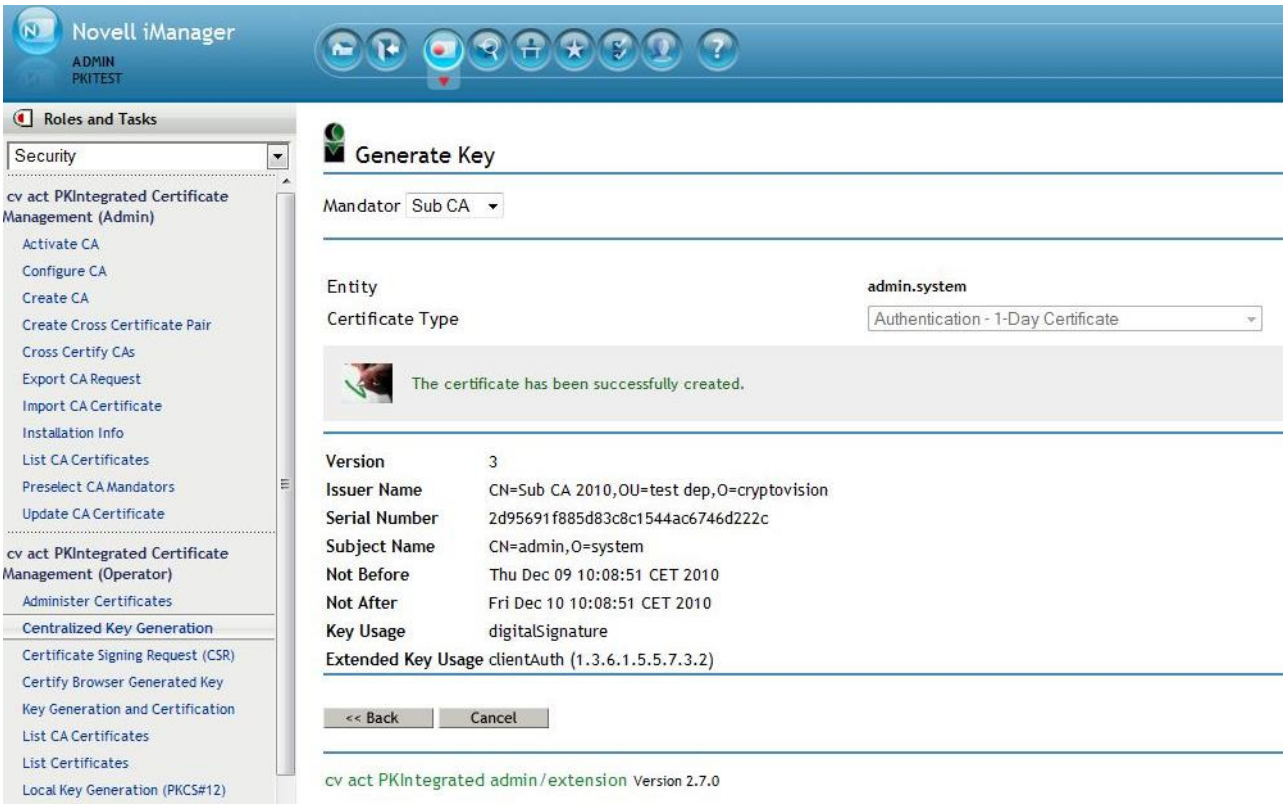

Figure 3.4: Successfully Centralized Key Generation

When the task finished a status message will be shown and on success the certificate details. The private key and the certificate can be downloaded with the “List Certificates” or the “Administer Certificates” task.

2.2.3 Certificate Signing Request (CSR)

Two Certificate Signing Requests are supported: PKCS#10 and SPKAC (Signed Public Key and Challenge). Select the appropriate type of the request and copy the Base64 encoded request data into the text field.

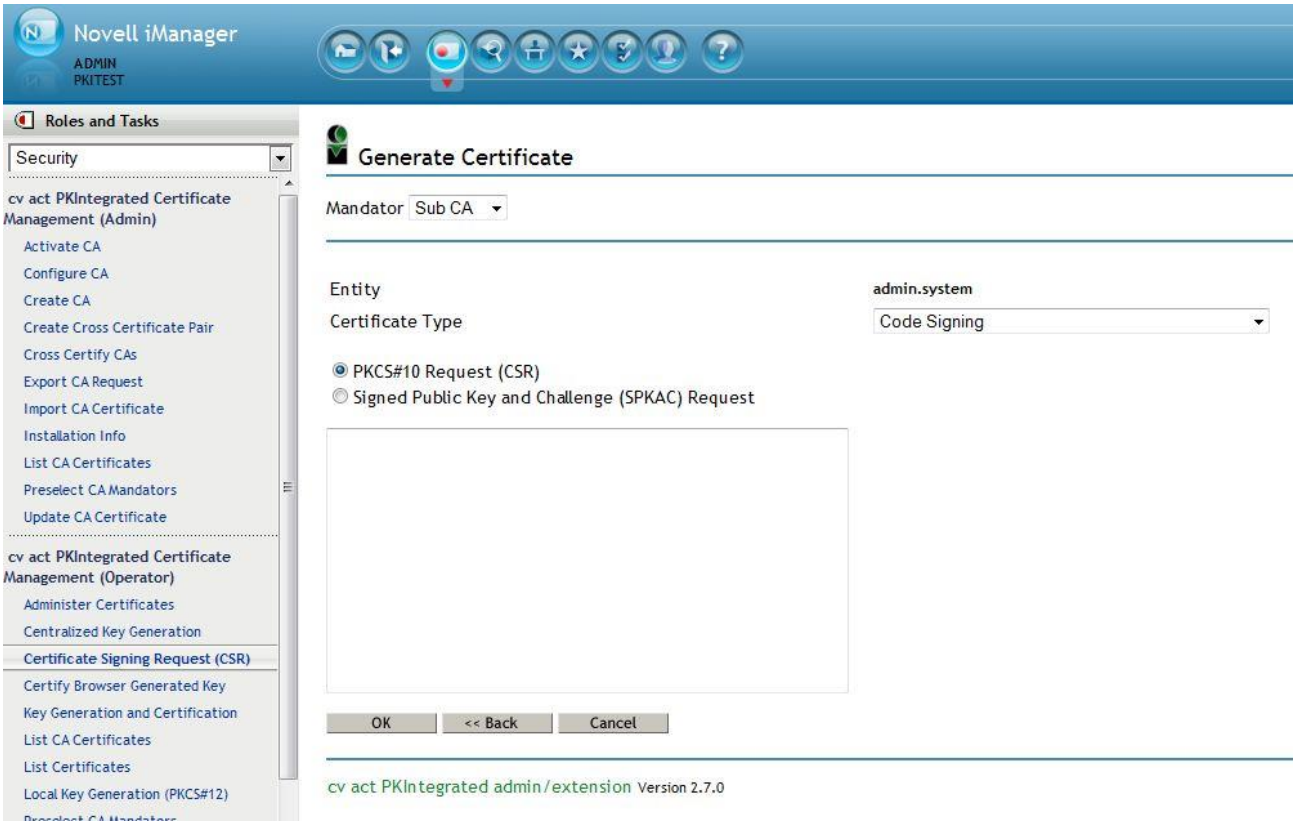

Figure 3.5: Certificate Signing Request

When the task finished the certificate details will be shown. The certificate can be downloaded with the “List Certificates” or the “Administer Certificates” task.

2.2.4 Certify Browser Generated Key

Internet Explorer and Firefox are using different mechanism to generate the private key pair and requesting the certificate. The Internet Explorer is using an ActiveX Control and supports different CSPs. Firefox supports different Cryptographic Modules too.

2.2.4.1 Internet Explorer

For Certificate Types configured for decentralized key generation, the local browser will create the key pair. The certificate is imported into the local browser and stored in eDirectory.

The Key generation request has multiple attributes that can be set for configuration. The following is a complete list of attribute values:

Attribute (eDirectory Attribute name)	Description	Example
CSP	Select one of the available Cryptographic Service Providers to generate the key pair. The list depends on installed CSP modules on your PC.	Microsoft Base Cryptographic Provider v1.0
Key size	depends on policy setting and CSP support	1024
Private key protection	This setting specifies how the private key is protected in IE. There are three options available: no additional protection, medium protection (user is informed by message box that the private key is used) and high protection (user has to provide a password for the private key, that has to be defined during key generation).	False
Private key exportable	This setting specifies if the private key can be exported from IE.	False
Use local machine store	This setting specifies if the certificate is stored in the registry under HKCU or HKLM.	False

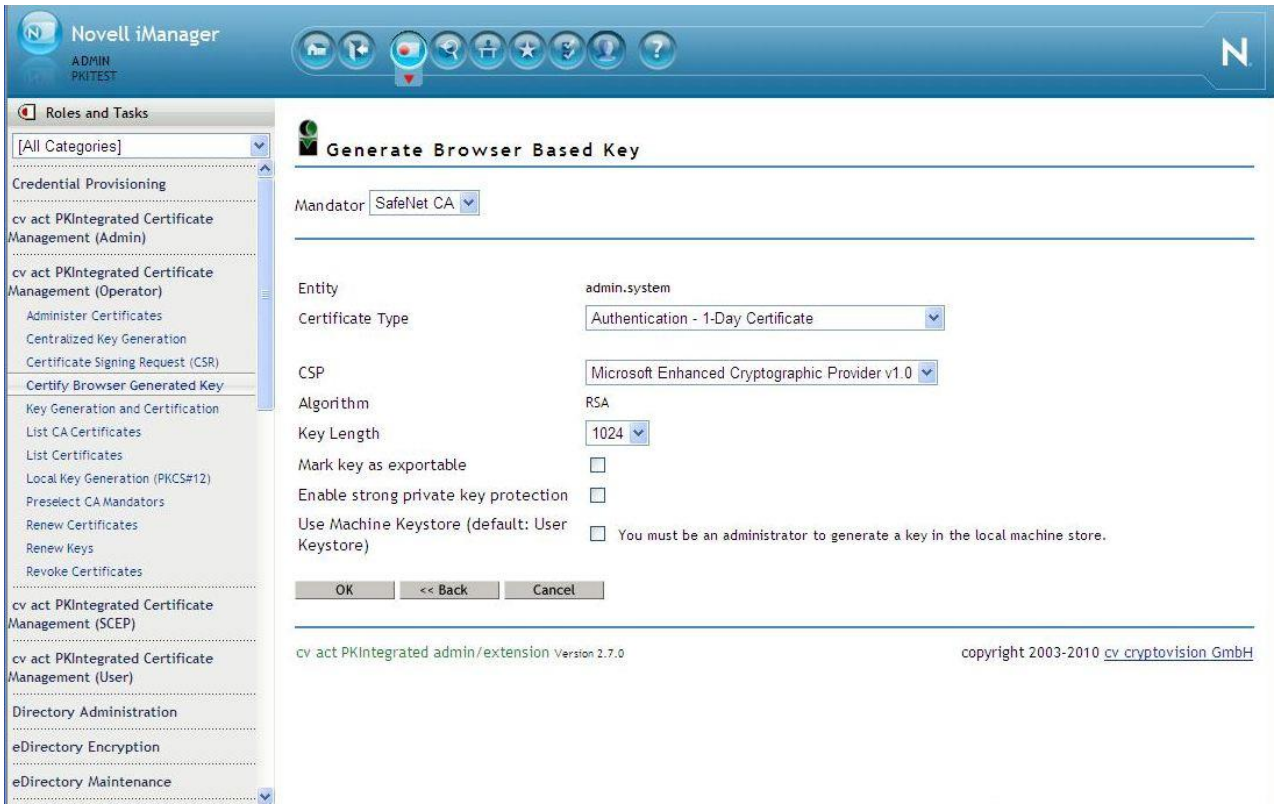

Figure 2.6: Internet Explorer Certification import

- Allow Certificate to be created (Yes)

Figure 2.7: Internet Explorer Warning for Certificate request

- Allow Certificate to be added (Yes)

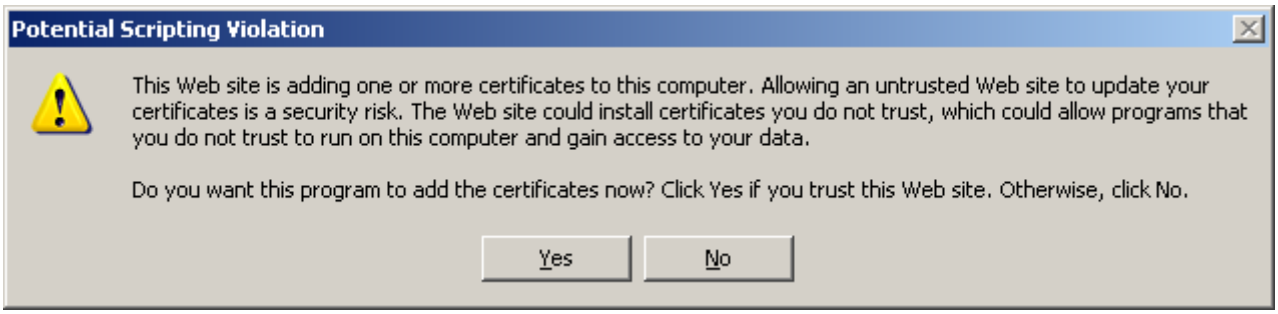

Figure 2.8: Internet Explorer Warning for Certificate Import

2.2.4.2 Firefox

For Certificate Types configured for decentralized key generation, the local browser will create the key pair. The certificate is imported into the local browser and stored in eDirectory.

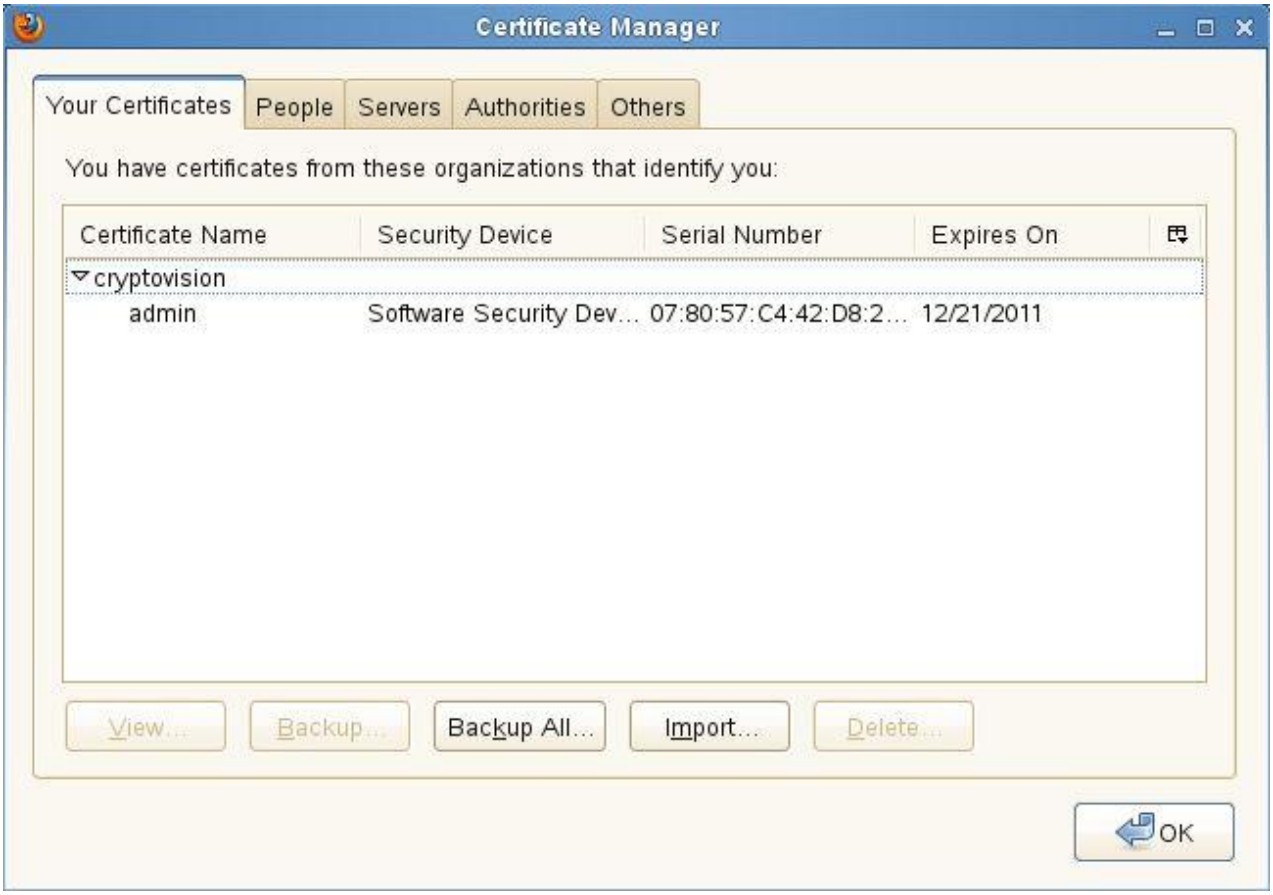

Figure 2.9: Firefox Certification import

The Key generation request has a single attribute that can be set for configuration:

Attribute (eDirectory Attribute name)	Description	Example
Key size	choice depends on local machine. Two options are available: High grade (2048 bit key length), medium grade (1024 bit key length).	High Grade

- Provide current Software Security Device password or define a new password.

Figure 2.10: Software Security Device Password

- After the task completed the certificate will be imported without prompting the user.

2.2.5 Local Key Generation (PKCS#12)

All decentralized certificate types can be used to generate the private key pair on the client. This is done by an applet which generates the key pair and sends a certificate signing request to the CA. When the certificate is returned by the CA the applet creates an encrypted PKCS#12 file that can be stored on the local hard disk.

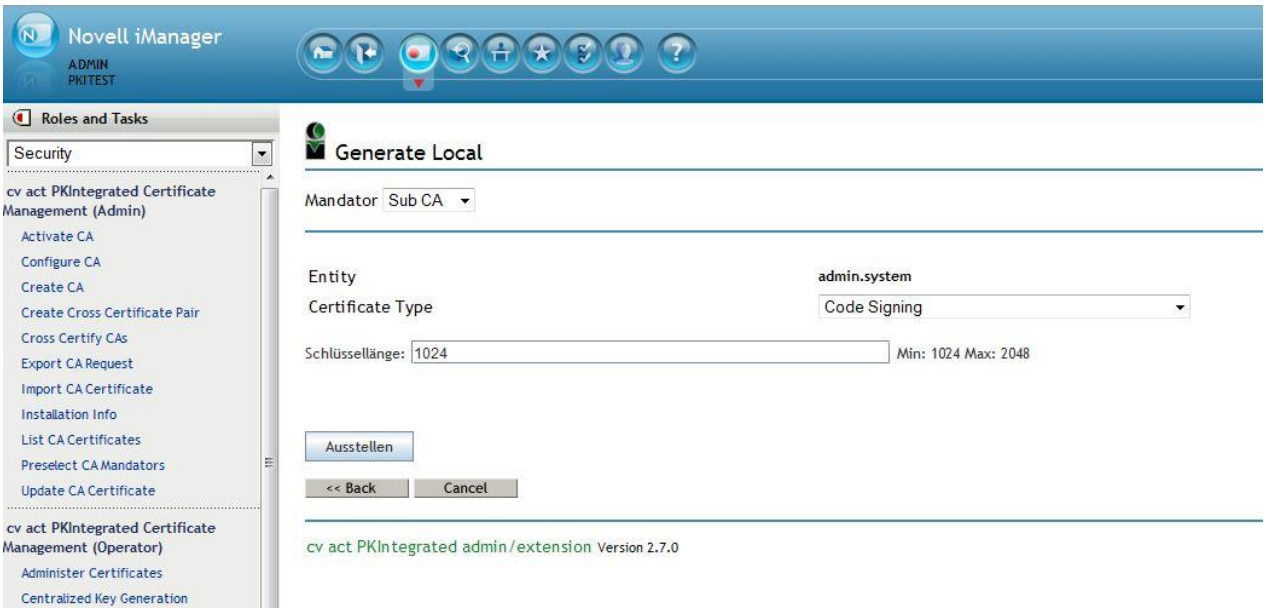

Figure 3.11: Local Key Generation

The passphrase of the PKCS#12 file should be down written immediately. Without the passphrase the PKCS#12 file is useless. Keep the passphrase in a secure place.

To copy the passphrase into the clipboard mark the text as usual and then press <Ctrl>-C. A context menu is not yet available.

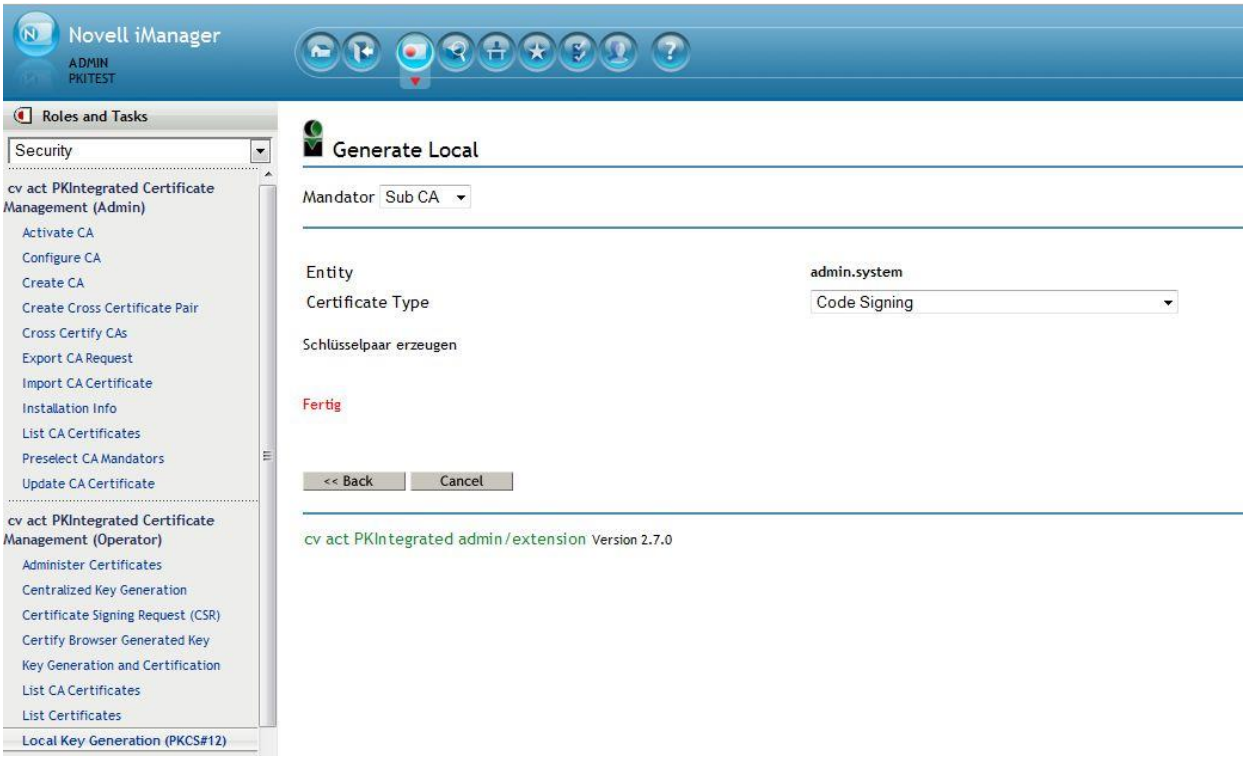

Figure 3.12: Local Key Generation succeeded

2.2.6 *Key Generation and Certification*

The task “Key Generation and Certification” starts with a selection of one of the four operator tasks in a clearly arranged dialog:

- Centralized Key Generation
- Certificate Signing Request (CSR)
- Certify Browser Generated Key
- Local Key Generation (PKCS#12)

This task accumulates all tasks which will generate new key pairs or certify new public keys.

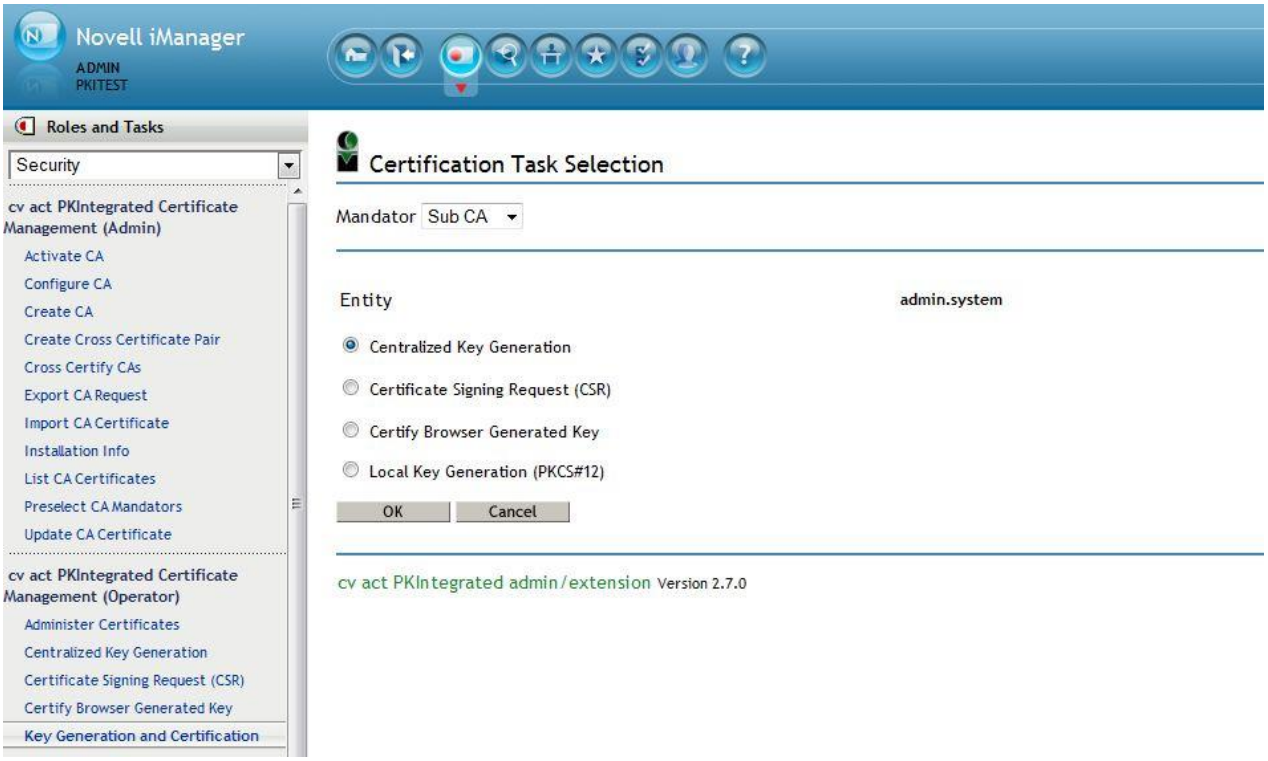

Figure 3.13: Key Gereration and Certification

The system operator may decide which tasks will be shown for the different roles using the role based services.

2.2.7 List Certificates

This task lists certificates. An operator may select the certificates by different selection criteria and a user gets all his certificates shown.

Sorting can be down by clicking on a column header.

By clicking on the icon the details dialog opens. The details dialog allows to export the certificate, import the certificate into the browser or to download the private key.

Figure 3.14: Certificate List

No further actions are available. The following tasks allow updating the key or the certificate if needed. Or a certificate may be revoked.

2.2.8 Renew Keys

If keys should be renewed then use this task.

Select all certificates for which the key has to be renewed. New keys will be generated with the same certificate types the selected certificates had. The certificate type must be enabled for centralized key generation.

Figure 3.15: Renew Keys

2.2.9 Renew Certificates

This task allows renewing of the selected certificates.

Select all certificates for which the certificate has to be renewed. New certificates will be created with the same certificate types the selected certificates had. The attribute validNotBefore of each certificate will be used for the new certificate. The certificate type may be enabled for centralized key generation or request certificate.

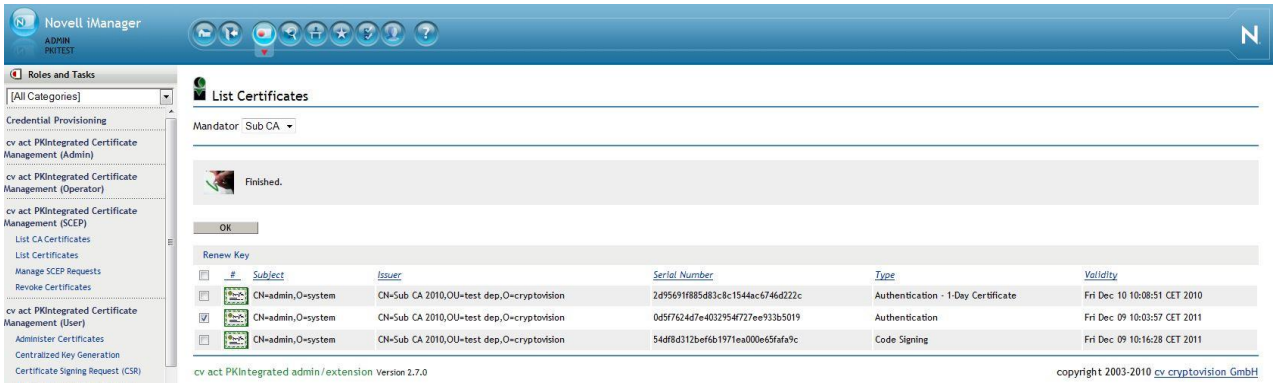

Figure 3.16: Renew Certificate

2.2.10 Revoke Certificates

To revoke certificates select this task.

The operator role allows suspending certificates too. The user role will not have this action.

Select all certificates that have to be revoked. Then go to the menu and open the ‘Revoke’ menu and select one of the reasons. If you can’t specify a reason then select the ‘Unspecified’ entry.

Figure 3.17: Revoke Certificates

The revocation task starts immediately and the actual state will be shown at the top of the window. If the revocation of a certificate fails, e.g. it is already revoked, a dialog is shown and you can decide to resume or to cancel the task.

When the task has finished press on the ‘Okay’ button and the list of certificates will be updated.

Attribute (eDirectory Attribute name)	Description	Example
cvReason	The reason why the certificate is being revoked. This is added to the Certificate Revocation List (CRL) entry for the re-voked certificate. The selection is: Unspecified Key Compromise Affiliation Changed Superseded Cessation of Operation Suspend selection: Certificate on Hold Remove from CRL	Superseded

2.2.11 Suspend Certificates

When a certificate has to be suspended the action is similar to the revocation task described above.

The certificates have to be selected as usual and then the action ‘Certificate Hold’ from the ‘Suspend’ menu has to be chosen. The selected certificates must not be revoked.

To remove certificates from the CRL select these certificate and choose the ‘Remove from CRL’ action of the ‘Suspend’ menu. This action fails on certificates that were revoked for some over reason.

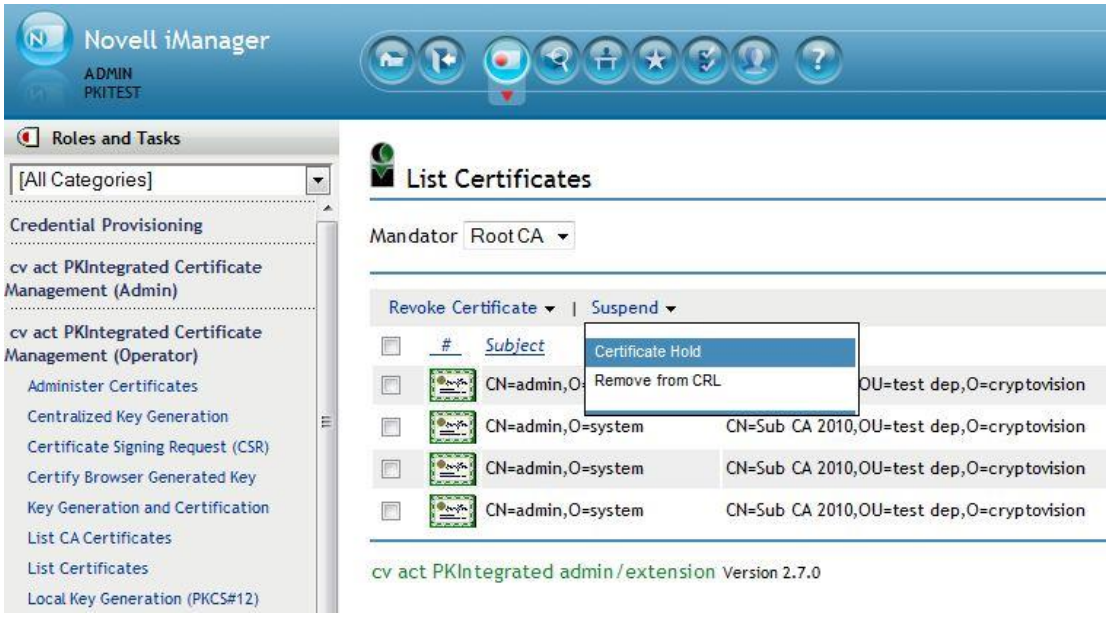

Figure 3.18: Suspend Certificates

2.2.12 Administer Certificates

This task offers all actions that are useful for certificate administration.

The ‘Administer Certificates’ task includes the following actions:

- Renew Certificate
  Renew the selected certificates
- Renew Key
  Create a certificate of the same type with a new - on the server generated - key
- Revoke Certificate
  Revoke selected certificates
- Suspend Certificate (if you are an operator)
  Suspends selected certificates or undo the action
- Export (if you are an operator)
  Exports the most important attributes from the repository

The last action, ‘Export Certificate Information’, is only available in this task and only visible for operators. The most important attributes from the repository entries of the selected certificates will be exported into a CSV file.

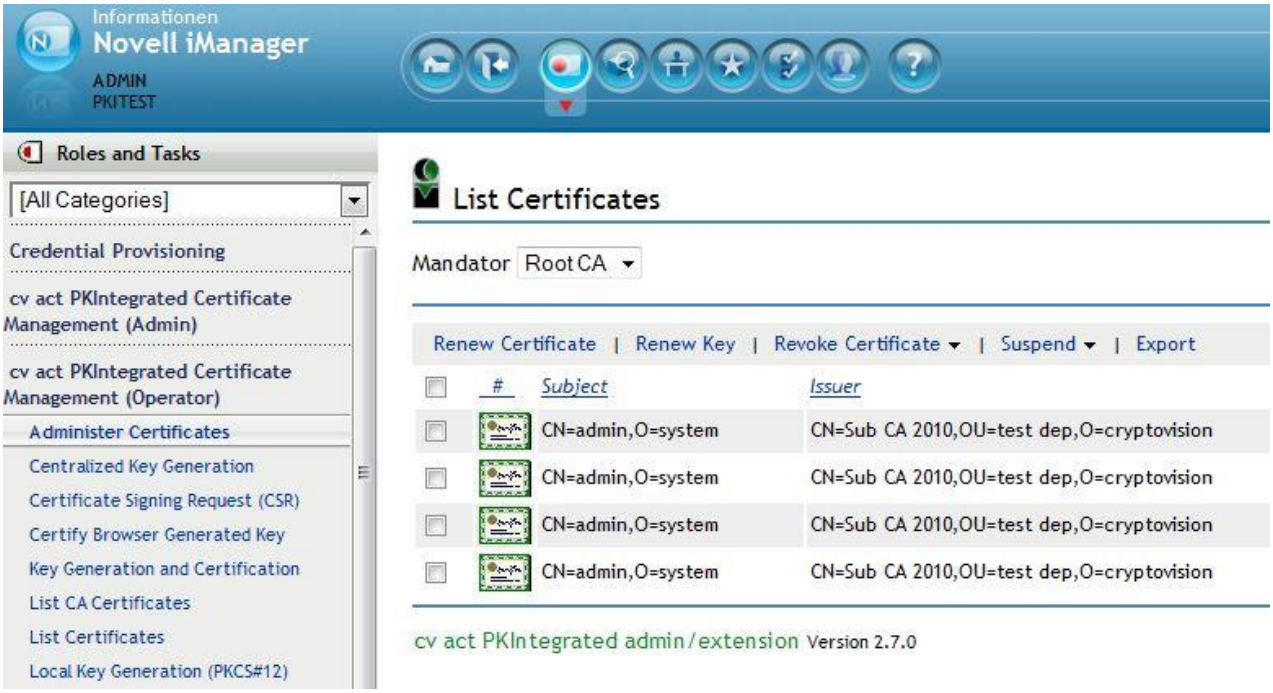

Figure 3.19: Administer Certificates

If an action has finished please press the ‘Okay’ button: The status of the certificate list will be updated.

2.3 Overwrite Subject Name

The generation of server certificates generally follows the same steps as with user certificates. By default, the context of an object in eDirectory is used as name of the owner of the certificate. cv act PKIntegrated provides a configuration option (set cvNameOverwriteAllowed) where the name of the owner has to be provided by a user or the operator during certificate creation. The following description is valid for certificate templates, for which this option is set. Most commonly this is the case for SSL, OCSP and SCEP certificates.

Create PKCS#12-Keyfile is the simplest way to create a server certificate. Other methods to request a certificate work as well, e.g. Certificate Signing Request (CSR).

To create a certificate with cvNameOverwriteAllowed set for the selected certificate type you have to notice the following:

- Select one of the following tasks:
    - Centralized Key Generation
    - Certificate Signing Request (CSR)
    - Certify Browser Generated Key
    - Local Key Generation (PKCS#12)
- Select the certificate type for which cvNameOverwriteAllowed is set.

On the page an additional input field appears:

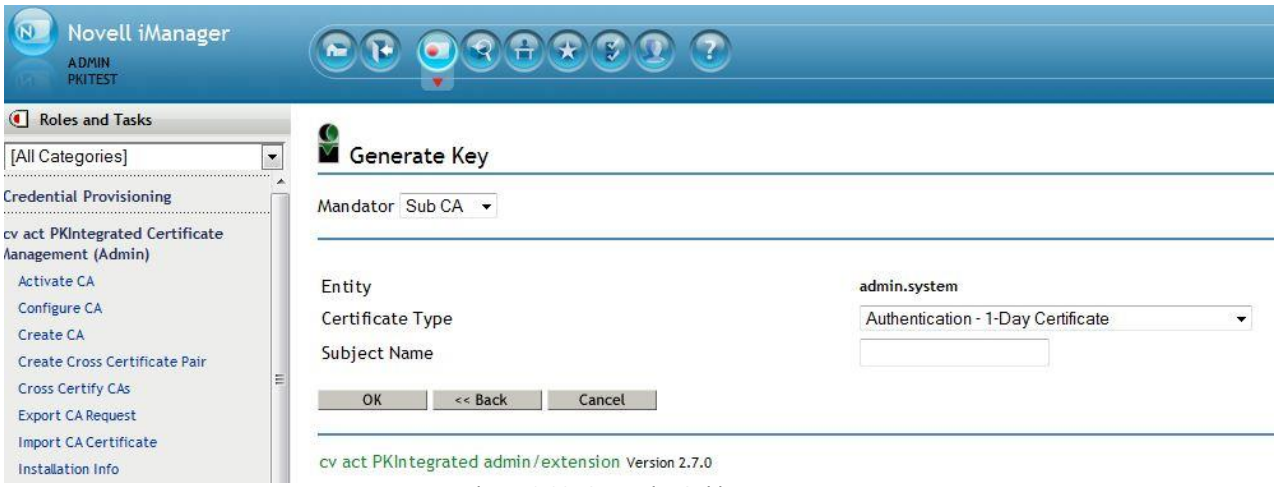

Figure 3.20: Overwrite Subject Name

- Specify the subject name for the certificate cn=servername,o=organsation,c=country
- Start the task by clicking on the ‘Okay’ button.

3 Information / Export Notice

cv cryptovision gmbh
Munscheidstr. 14
45886 Gelsenkirchen
Germany

Release: Mai 2011

Trademarks

All software and hardware names mentioned in this book are in most cases registered trademarks and are liable to the legal regulations.

Please note:

The product delivered to you is subject to export control. For shipping outside the EU export permission is required. Please observe the legal regulations of the country that applies to your case.

4 Glossary

ANSI

Abbreviation for American National Standards Institute, (http://www.ansi.org).

ASN.1

Abbreviation for Abstract Syntax Notation One. ASN.1 is a widely used standard for the decryption of abstract objects. In encoding (rules describing how such objects are to be produced as a string) it is distinguished between Basic Encoding Rules (BER) and Distinguished Encoding Rules (DER).

Asymmetric Cipher

Encryption procedure employing two different keys (in contrast to symmetric cipher): One publicly known key - the public key - for data encryption and one key only known to the message receiver - the private key - for decryption.

Authentication

By authentication an entity, e.g. a user, proves its identity. Normally a user enters its user-name, which might be known publicly, and then it identifies itself by its password, which should be only known to itself. Authentication types include: authentication by knowledge (password), possession (cryptographic token), or biometric characteristics (fingerprint, etc.). The most elegant method is based on the use of so called digital signatures.

Brute Force Attack

An attack on a cryptographic algorithm, in which the entire key space is systematically searched.

CA

See Certification Authority.

Certificate

A digital certificate is an electronic document, which is connected to a public key. A trustworthy authority (like a CA) verifies that the key belongs to a certain person and has not been modified. The advantages of such procedures are that only the public key of the so called root instance of the PKI (and not of every participant) will be required for complete verification.

Certification Authority (CA)

A CA is a trustworthy agency whose task is to certify cryptographic keys (see Certificate). It is part of a PKI. Some details: A CA issues certificates. It confirms the accuracy of the data of the certificate by its signature. The data contains the name of the key bearer, a set of identifying attributes, its public key, its period of validity and the name of the CA. The CA must have a CRL, where it publishes revoked certificates, which might have invalid data or compromised secret data.

Certificate Revocation List (CRL)

A list of certificates which are no longer valid. CRLs are defined in the X.509-standard.

Collision

Occurs in a hash function, if two different messages lead to one and the same hash value. If no such collisions can be generated by a given function, this is defined as collision-resistant.

CRL

See Certificate Revocation List.

Digital Signature

The counterpart of a handwritten signature for documents in digital format; this is to provide security concerning the following questions:

- Authentication, i.e. confidence about the identity of the sender of the document
- Maintenance of the document's integrity
- Non-repudiation, i.e. the sender shall not be able to deny the signature generation

These features can be achieved by using asymmetric procedures. Pieces of information are generated by using private keys by which a third person, who knows the appropriate public key, can verify its correctness.
For popular public key procedures like RSA, protocols exist for employment in the scope of digital signatures. For DL-based procedures, ElGamal-type procedures have established themselves.

ECC

The use of elliptic curves in cryptography is called ECC (*Elliptic Curve Cryptography*). This class of procedures provides an attractive alternative for the probably most popular asymmetric procedure, the RSA algorithm. The basic mathematical problem is - similar to the DSA algorithm - the calculation of the discrete logarithm in finite sets. The set of the elements considered here is a set of points, which solve a certain mathematical equation, that is, an elliptic curve.
The decisive advantage of this procedure is the fact that the fast algorithms known so far for solving the DL problem in finite fields cannot be applied in this case. As for the DL problem only very general procedures exist, in the group of points on elliptic curves significantly shorter key and parameter lengths are sufficient without reducing the security. This is especially effective when used in situations with limited storage or computing capacity, as e.g. in smartcards or other small devices.

Elliptic curves

A mathematical construction, in which a part of the usual operations applies and which has been employed successfully in cryptography since 1985.
If the base field is GF(p) (p prime), an element (or point) of an elliptic curve (with the parameters A, B) is e.g. defined by a tuple (x,y), which solves an equation of the following form:

$y^2 = x^3 + Ax + B$

If the *finite fields* has characteristic 2, the equation has the following form:

$y^2 + xy = x^3 + Ax^2 + B$

Elliptic curves can be defined over any field; but only curves over finite fields are used in cryptography. If the elliptic curve and field on which it is based meet certain conditions, the problem of discrete logarithms cannot be efficiently solved.

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 key point 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.

PKCS

Abbreviation for Public Key Cryptography Standard. It was issued and supported by RSA Laboratories and is a company standard meant to solve the 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. smart cards).

PKCS5 padding

A padding scheme often used for block ciphers, where padding assures that the input text length is a multiple of the cipher's block size.
As an example, our CBC modus BlowFish implementation (block size is 8 byte) of the cvactLibCore would pad a 10 byte input text with 6 byte(0x06). Even if the input length is a multiple of 8 byte, padding is added. In this case, PKCS5 padding would add 8 byte(0x08). Therefore the output of the complete encryption is generally longer than the input.

PKI

See Public Key Infrastructure

Private key

This is the key only known to the person who generated a key pair. A private key is used in asymmetric ciphers for decryption or the 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 number generators in software. Therefore, so-called pseudo-random number generators are used, which then should be initialized with a real random element (the so-called seed).

Public key

This is the publicly known key in an asymmetric cipher which is used for encryption and verification of digital signatures.

Public Key Infrastructure (PKI)

The biggest problem in the employment of public key procedures is the authenticity of keys. This imposes the question of how to ensure that the key on hand is really the key belonging to the communication partner. A PKI is a combination of hardware and software components, policies, and different procedures. It is based primarily on so called certificates. These are keys of communication partners which have been certified by digital signatures of trustworthy authorities.

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 a software), so-called pseudo random numbers are used instead.

Symmetric cipher

Encryption procedure using the same key for enciphering and deciphering (or, in which these two keys can be simply 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.

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.