cryptovision

cv act *PKIntegrated* V3.0

Installation Guide

Content

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 Installation Guide is designed to help you with the installation of cv act PKIntegrated.

This guide gives detailed step-by-step instructions for an environment based on SLES11, Novell eDirectory 8.8, Novell iManager 2.7, Novell SecretStore 3.3.3 and Novell Identity Manager 4.0. 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 configuration examples and variables for each configuration step. They 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. 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
- suspension of a certificate
- maintenance of a certificate revocation list (CRL)
- email notification of specific events
- all relevant CA activities are logged into the syslog

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 Export Request
- CA Import Certificate
- CA Cross Certification
- Key Generation
- Key Update
- Certificate Request

- Certificate Update
- Certificate Revocation
- Certificate Suspension
- CRL Update

1.4.4 admin/extension

admin/extension defines the front-end user interface for the certificate management. It is implemented 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 SCEP-enabled 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

The following new features have been added to cv act PKIntegrated 2.8:

- New flag controls if SecretStore will be cleaned up before installing a new roaming key
- Private key blob can be provided as plain PKCS#8 blob instead of storing it as a roaming key in SecretStore

2 Installation

2.1 Before You Install

To install, configure and use cv act PKIntegrated successfully, several pre-requirements must be met.

Subject	Specification	Recommendation
Hardware	Standard Server Hardware with no additional requirements. In case a PCI-based HSM is used: 1 free PCI slot. In case a LAN-based HSM is used:ensure network connection	Restriction of Physical Access.
Operating System	SuSE Linux Enterprise Server 9 or higher Novell Open Enterprise Server Any other Linux Distribution supported by Novell eDirectory and Novell Identity Manager installed xinetd for scep/responder and ocsp/responder	SLES11/OES Linux.
Novell eDirectory	Any of your servers must run Novell eDirectory 8.7 or higher	8.8.x or newer. For a sophisticated tree design please contact cryptovision.
Novell eDirectory Replica Placement	The eDirectory server running the Metadirectory Engine, must hold at least a Read-Write replica of any Partition that contains managed user and machine objects, as well as a replica of the Partition that contains the cv act PKIntegrated objects and the Identity Manager Driver Set Object. The SecretStore Server must hold a replica of any Partition that contains managed user and machine objects.	Make use of Filtered Replicas to limit database size and synchronization traffic in a large, distributed tree design.
Novell eDirectory Rights	For most configuration tasks, advanced eDirectory rights are required. For Novell eDirectory Schema extensions, Supervisor entry right to [ROOT] is required.	Do not use the eDirectory admin account for administrative or proxy purpose, but assign rights to dedicated Users, Groups or Organizational Roles.
Novell SecretStore Service	At least one eDirectory Server within your Tree must run Novell Secret Store Services 3.3.3 or newer. SecretStore must be accessible via LDAP. cv act PKIntegrated stores the PKCS#12 file(s) into the entity’s secret store, where it can only be accessed by the entity itself. Optionally private keys can be stored into the SecretStore of a Recovery Administrator.	

Subject	Specification	Recommendation
Novell Identity Manager	The ca/server must run the Metadirectory Engine or the Remote Loader from Novell Identity Manager 3.5 or newer to be able to run the Java Driver Shim.	Version 4.0 or higher.
Designer for Novell Identity Manager	Only if you want to design, deploy and manage dir/connector offline, you need to install Designer for Novell Identity Manager 3.0 or higher	Version 3.0 or higher.
Novell iManager	One Server within the network must run Novell iManager 2.6 or higher	Version 2.7 or higher.
Novell iManager Role Assignment	If you have configured Role Based Services for iManager, please make sure that you operate as Collection Owner or as a user that has all required Roles assigned.	
HTML Browser	An HTML Browser compatible with your version of Novell iManager needs to be installed on the ca/server or a Management Workstation	Microsoft Internet Explorer 10 or higher Mozilla Firefox 28 or higher.

In addition the Unlimited Strength Java Cryptography files have to be installed into the Java Runtime Environments used by the Novell Identity Manager and the Novell iManager.

For support on other software versions, please contact cv cryptovision GmbH. Contact details are listed on our Homepage, http://www.cryptovision.com.

The following chapters will guide you step by step through the installation of cv act PKIntegrated.

2.2 Novell eDirectory Schema Extension

cv act PKIntegrated requires an eDirectory schema extension to store new objects and attribute values in eDirectory. For this purpose, cv act PKIntegrated provides an LDIF file that can be imported to an eDirectory LDAP server by a user with Supervisor Object Rights to the [ROOT] of your eDirectory tree. To simplify the schema extension process, we recommend importing the schema extension to an eDirectory LDAP server which holds a replica of the partition [ROOT].

We strongly recommend importing the LDIF file using LDAP over TLS to avoid sending your admin user CN and password in clear text to the LDAP server. LDAP over TLS is enforced by default for any new eDirectory server. To disable LDAP over TLS, modify the LDAP Server's LDAP Group object and disable "Require TLS for simple bind".

You can choose from one of several methods to extend your eDirectory Schema, based on your environment and experience. The use of iManager requires you to export the Trusted Root Certificate of you LDAP server's certificate. The examples assume that this was done to the user's home directory (~/PKI-TREE_CA.der). Refer to chapter 2.4.4 for a description on how to export the Trusted Root Certificate.

The following options are described in this chapter:

- LDIF import with Novell ICE
- LDIF import with Novell iManager
- LDIF import with ldapmodify tool

Because of its simplicity we recommend to use the command line utility Novell ICE (Novell Import Convert Export) or Novell’s Designer for Identity Management.

For LDAP Troubleshooting, enable LDAP Trace options at the LDAP server object and use the DSTrace Utility in iMonitor or the platform specific DS Trace utility.

Make sure that the screen log or log file show the following result after the process is completed:

```
Total entries processed: 78
Total entries failed: 0
```

The number of processed entries may vary. If you are upgrading the system the number of processed some failures may occur which can be ignored. Chapter 3 will give you an overview of classes and attributes that are included in the schema extension.

2.2.1 LDIF import with Novell ICE

ICE is the command line version of the Novell Import Convert Export Utility installed with Novell eDirectory, iManager or ConsoleOne.

ICE uses General and Schema Options (for more information, run ice -h options) and Source and Destination handlers with their options (for more information, run ice -h <handler name> where <hander name> is one of the following: LDAP, LDIF, DELIM, LOAD, SCH) for configuration.

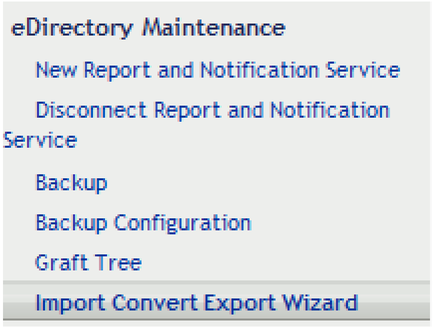
For the schema import of cv act PKIntegrated, run the following command (all in one line, options are case sensitive):

- ice
  ```
   -l <path of a new log file>
   -o
   -v
   -S LDIF
     -f <Path to the LDIF file on your installation medium>
     -a
   -D LDAP
     -s <Your Server's DNS name/IP address>
     -p <Your LDAP server's Secure port>
     -k
     -d <FqDN in LDAP format, needs Supervisor Rights of [ROOT]>
     -w <password>
  ```
- on a Linux Workstation or Server:
  ```
  ice -l ~/cvschema.log -o -v -S LDIF -f /media/cdrom/
  cvschema.ldif -a -D LDAP -s cv1.cryptovision.com -p 636 -k -d cn=Admin,ou=IT,o=CV
  -w cvpass
  ```
- on a Windows Workstation or Server:
  ```
  ice -l cvschema.log -o -v -S LDIF -f d:\cvschema.ldif -a -D LDAP -s
  cv1.cryptovision.com -p 636 -k -d cn=Admin,ou=IT,o=CV
  -w cvpass
  ```

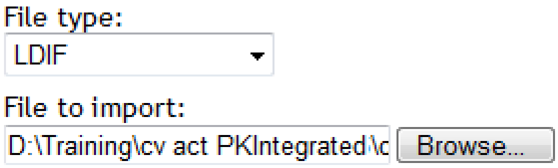
2.2.2 LDIF import with Novell iManager

Novell iManager is a role-based Management Framework for Novell eDirectory. iManager runs either as a web application on a server within your network or on your local workstation (called Mobile iManager). There is only 1 instance of iManager required within your network, even if you have to manage multiple eDirectory trees. iManager makes use of Novell ICE.

- In iManager select Role "eDirectory Maintenance", Task "Import Convert Export Wizard"


eDirectory Maintenance
New Report and Notification Service
Disconnect Report and Notification Service
Backup
Backup Configuration
Graft Tree
Import Convert Export Wizard

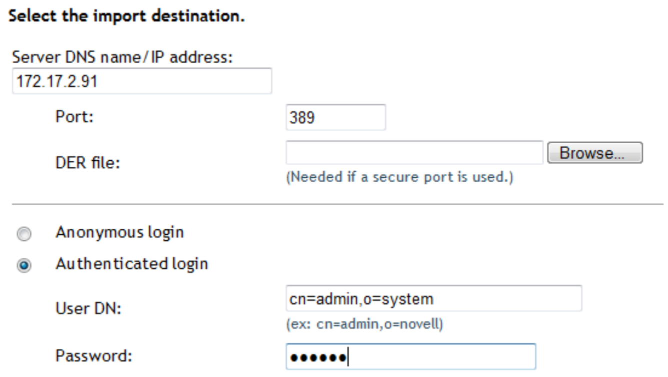
  - Select the task you would like to perform
    - Select "Import data from file on disk"
  - Click Next
  - Select the file you wish to import
    - File Type: LDIF
    - File to import: browse to the LDIF file on your installation medium
      on a Linux Workstation or Server: /media/cdrom/cvschema.ldif
      on a Windows Workstation or Server: d:\cvschema.ldif


File type: LDIF

File to import: D:\Training\cv act PKIntegrated\c

  - Click Next

- Select the import destination
  - Server DNS name/IP address: Type your Server's DNS name/IP address
    cv1.cryptovision.com
  - Port: Type your LDAP server's secure port. By default eDirectory is configured to not accept any LDAP request on the clear-text port 389.
    636
  - DER file: <Path to trusted root certificate>
    ~/PKI-TREE_CA.der
  - Authenticated Login
  - User DN: <FqDN in LDAP format of a user with Supervisor Rights of [ROOT] >
    cn=Admin,ou=IT,o=CV
  - Password: <Password>
    cvpass



Select the import destination.

Server DNS name/IP address: 172.17.2.91

Port: 389

DER file:
(Needed if a secure port is used.)

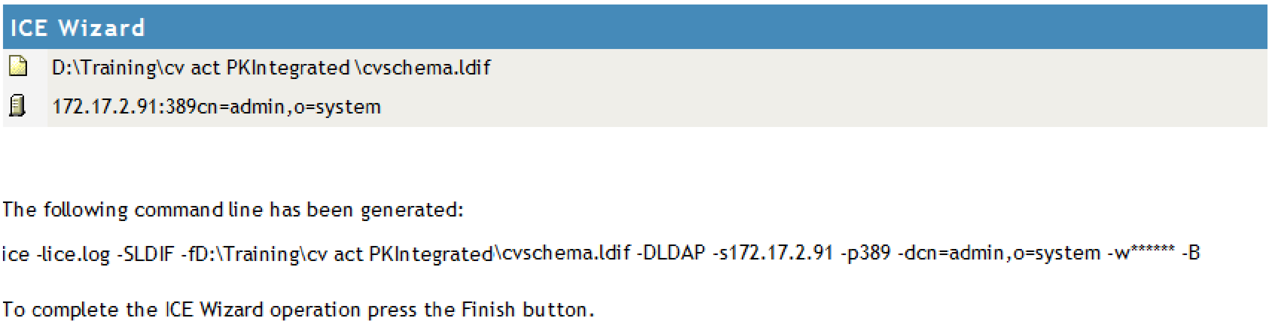
○ Anonymous login

● Authenticated login

User DN: cn=admin,o=system
(ex: cn=admin,o=novell)

Password: ••••••

- Click Next



ICE Wizard

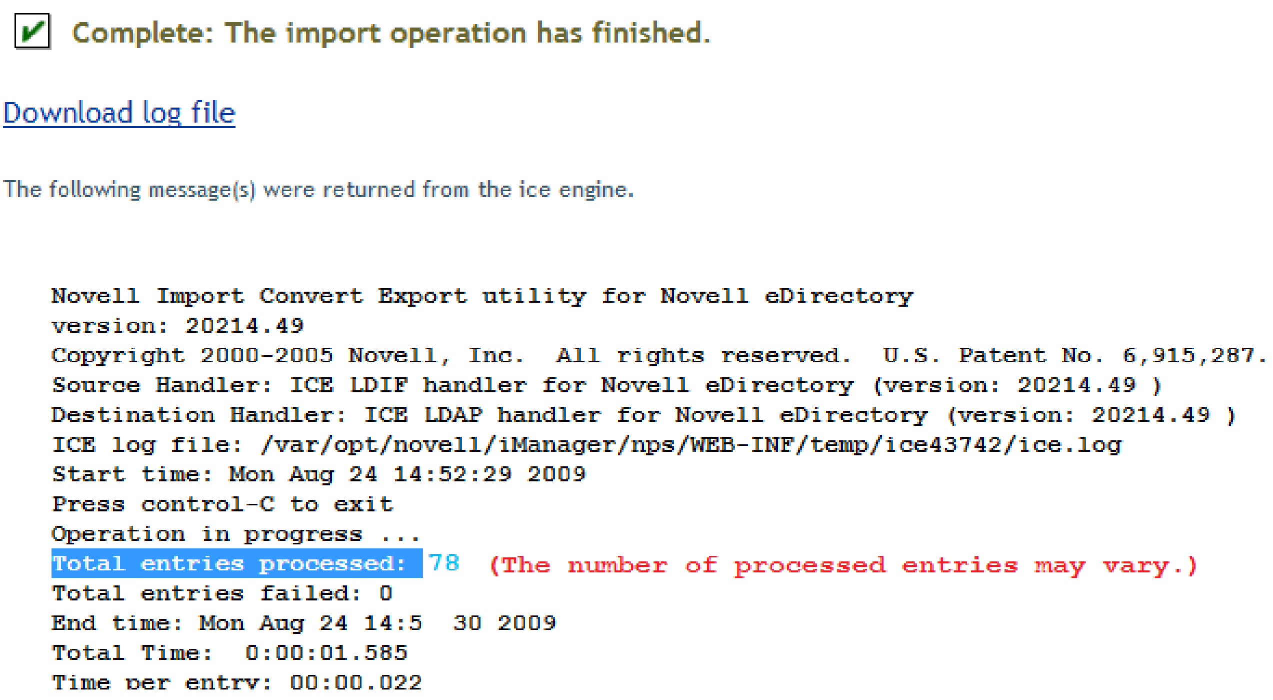
D:\Training\cv act PKIntegrated\cvschema.ldif

172.17.2.91:389cn=admin,o=system

The following command line has been generated:

```
ice -lice.log -SLDIF -fD:\Training\cv act PKIntegrated\cvschema.ldif -DLDAP -s172.17.2.91 -p389 -dcn=admin,o=system -w****** -B
```

To complete the ICE Wizard operation press the Finish button.



☑ Complete: The import operation has finished.

Download log file

The following message(s) were returned from the ice engine.

```
Novell Import Convert Export utility for Novell eDirectory
version: 20214.49
Copyright 2000-2005 Novell, Inc.  All rights reserved.  U.S. Patent No. 6,915,287.
Source Handler: ICE LDIF handler for Novell eDirectory (version: 20214.49 )
Destination Handler: ICE LDAP handler for Novell eDirectory (version: 20214.49 )
ICE log file: /var/opt/novell/iManager/nps/WEB-INF/temp/ice43742/ice.log
Start time: Mon Aug 24 14:52:29 2009
Press control-C to exit
Operation in progress ...
Total entries processed: 78  (The number of processed entries may vary.)
Total entries failed: 0
End time: Mon Aug 24 14:5   30 2009
Total Time:  0:00:01.585
Time per entry: 00:00.022
```

- Click Finish

2.2.3 *LDIF import with ldapmodify tool*

The ldapmodify tool edits the contents of a Lightweight Directory Access Protocol (LDAP) directory, either by adding new entries or modifying existing ones.

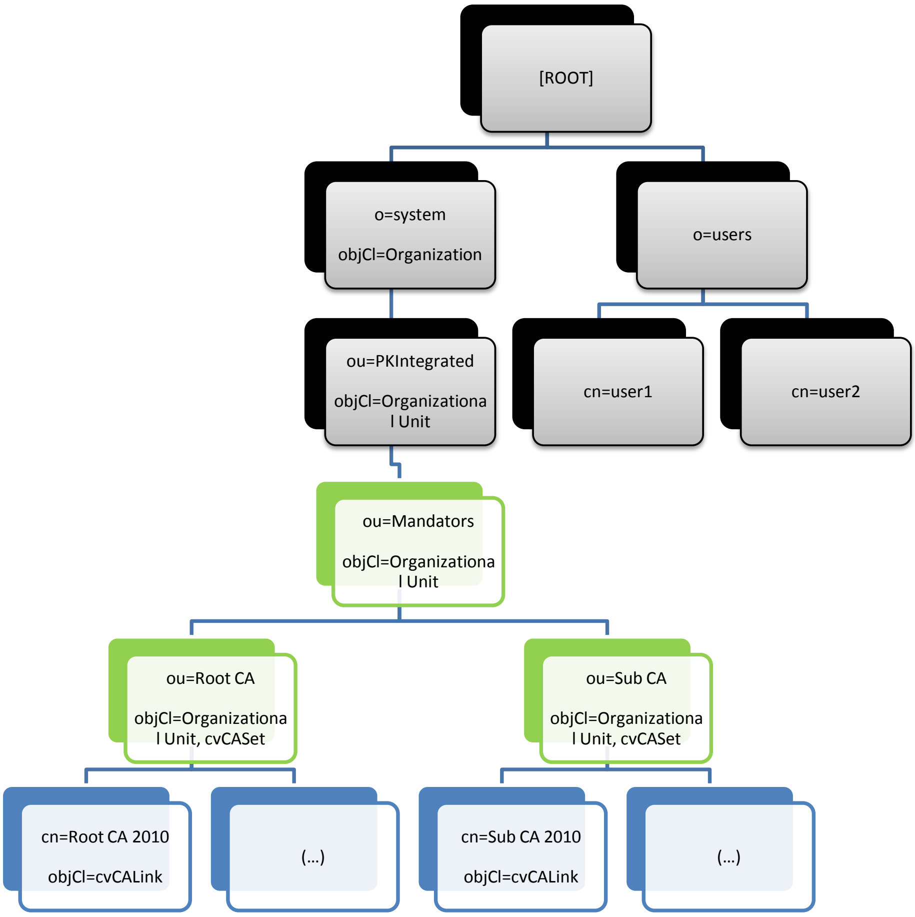
For the schema import of cv act PKIntegrated, run the following command in the shell on the eDirectory Server (all in one line):

- ldapmodify –x –Z –f cvschema.lif –D “cn=Admin,ou=IT,o=CV” –w cvpass
- The parameters have following meaning
  - -x Use simple authentication instead of SASL.
  - –Z Issue StartTLS (Transport Layer Security) extended operation.
  - –f Read the entry information from file instead of from standard input.
  - –D Use the Distinguished Name to bind to the LDAP directory.
  - –w Use password for simple authentication.
- Make sure that the shell shows the following result:

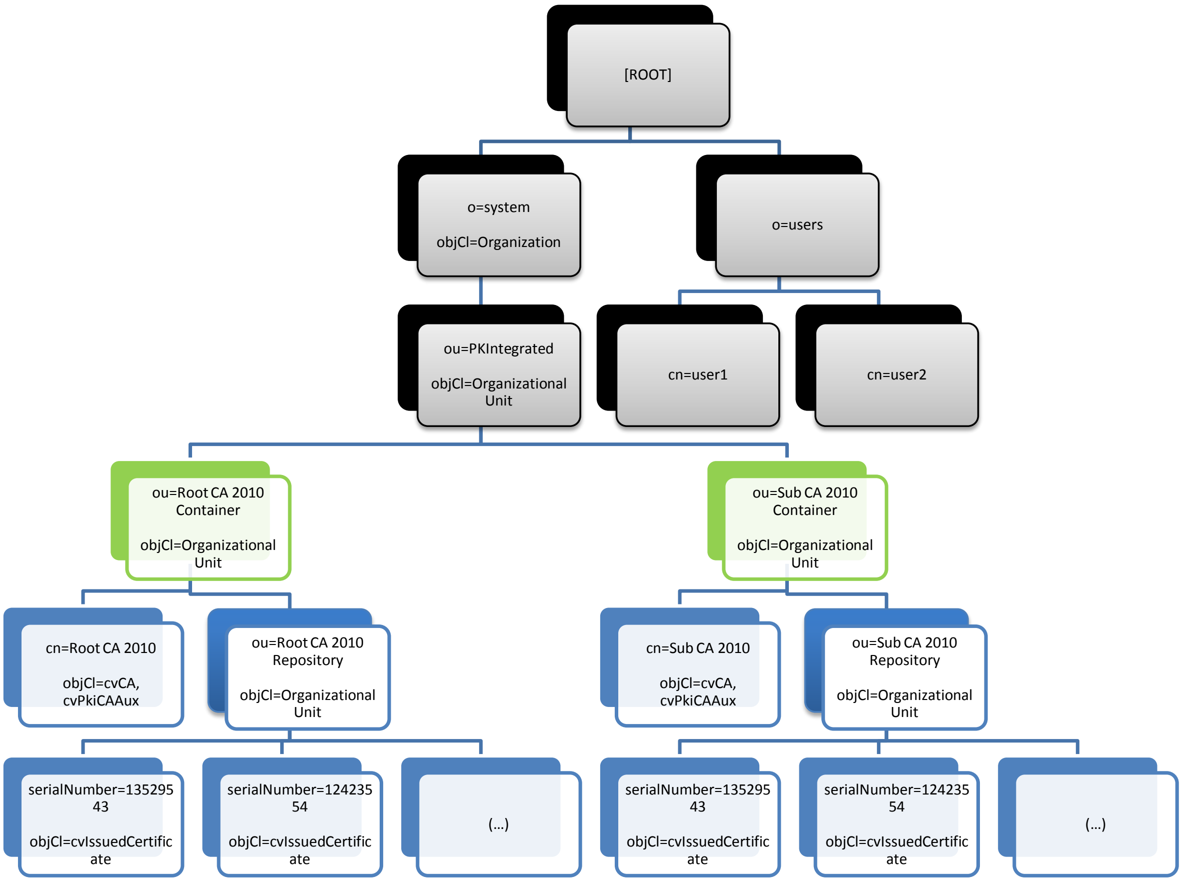
  Modifying entry “cn=schema”

2.3 Novell eDirectory Objects

Before you can use cv act PKIntegrated it is necessary to prepare the CA structure and to create a few objects. The diagram shows a sample of the structure. All objects with a green border in the diagram must be created using the iManager's Create Object task. The blue colored subordinary objects will be created when you use the cv act PKIntegrated tasks.



Mandators View


[ROOT]
o=system
objCl=Organization
o=users
ou=PKIntegrated
objCl=Organizational Unit
cn=user1
cn=user2
ou=Root CA 2010 Container
objCl=Organizational Unit
ou=Sub CA 2010 Container
objCl=Organizational Unit
cn=Root CA 2010
objCl=cvCA, cvPkiCAAux
ou=Root CA 2010 Repository
objCl=Organizational Unit
cn=Sub CA 2010
objCl=cvCA, cvPkiCAAux
ou=Sub CA 2010 Repository
objCl=Organizational Unit
serialNumber=13529543
objCl=cvIssuedCertificate
serialNumber=12423554
objCl=cvIssuedCertificate
(...)
serialNumber=13529543
objCl=cvIssuedCertificate
serialNumber=12423554
objCl=cvIssuedCertificate
(...)

CA View

2.3.1 Container for cv act PKIntegrated related Objects

For Administrative purpose, you might want to create a dedicated OU (or any other container object that can store other container objects) in a local or global container within your eDirectory tree. It will hold all objects related to ca/server.

ou=PKIntegrated,o=system

objCl=Organizational Unit (or similar)

2.3.2 Container for Mandators

The ca/server stores the information about the mandators and their CAs in a dedicated OU. It is recommended to create this container below the OU created for all ca/server objects.

The mandator container is the place where all mandators have to be created.

ou=Mandators,ou=PKIntegrated, o=system

objCl=Organizational Unit (or similar)

2.3.3 Mandator

For each mandator, e.g. Root CA or Sub CA, a container must be created. It must be a sub container of the mandator container.

A mandator equates to a CA instance, e.g. the Root CA. Usually the mandator holds only one CA object that represents the active CA at the same time. If a rekeying is needed then a new CA has to be created for the mandator of the CA which will be replaced. After the new CA is configured it can be activated. The replaced CA will only be used for revocation of issued certificates and CRL signing.

For a Sub CA another mandator must be created in the mandator container. The number of CAs in a mandator is not limited.

The CA objects themselves will be created by cv act PKIntegrated using the “CA Create” task.

After you created a mandator use the ‘Modify Object’ task of the iManager and go to the tabulator ‘Other’. Extend the ‘objectClass’ attribute by adding the auxiliary class ‘cvCASet’. Save your changes. Now you are able to set the attribute ‘cvMandatorDescription’ which holds the description of the mandator. The value of the attribute ‘cvMandatorDescription’ will be shown in the mandator selection area of the cv act PKIntegrated iManager tasks.

ou=Root CA,ou=Mandators,ou=PKIntegrated, o=system

objCl=Organizational Unit (or similar)

objCl=cvCASet

2.3.4 Container for Certificate Repository

All issued certificates of a CA are stored as cvIssuedCertificate objects in its Certificate Repository. It is recommended to create this container below the OU created for all ca/server objects or in a container object for this CA Each CA must have its own repository.

ou=Root CA 2010 Repository,ou=Root CA 2010 Container, ou=PKIntegrated, o=system

objCl=Organizational Unit (or similar)

2.3.5 Container for SCEP Requests

All SCEP requests from VPN routers are stored as cvSCEPRequest objects in the SCEP Repository.. It is recommended to create this container below the OU created for all ca/server objects.

ou=SCEPRepository,ou=PKIntegrated,o=system

objCl=Organizational Unit (or similar)

2.3.6 User or Role for ca/server administration

This role is designed to create, configure and activate a Certification Authority and manage the dir/connector component (IDM Driver). In addition, this user may request and revoke certificates. This role can be implemented as a group object in eDirectory. For more details on iManager Task Assignment and eDirectory Right Requirements, please consult the cv act PKIntegrated Administration Guide.

cn=PKIAdmin,ou=PKIntegrated,o=system

objClass=User

2.3.7 User or Role for ca/server operation

This role is designed to create, modify and revoke certificates. This role can be implemented as a group object in eDirectory. For more details on iManager Task Assignment and eDirectory Right

Requirements, please consult the cv act PKIntegrated Administration Guide.

cn=PKIOperator,ou=PKIntegrated,o=system

objCl=User

2.3.8 User for ca/server recovery

The recovery keys of the Certification Authority will be stored in the SecretStore of this user. For security reasons, we recommend to choose a dedicated user for this role. Use a strong password or multifactor authentication via NMAS and IP address restrictions to protect the SecretStore from unauthorized access.

This user does not need any additional eDirectory rights.

cn=PKIRecovery,ou=PKIntegrated,o=system

objCl=User

2.3.9 User for dir/connector authentication

The dir/connector needs a valid user account to authenticate to the LDAP server and read information from the certificate repository container. This user account will be used to trigger publisher events like ‘CRL update’ too.

For security reasons, we recommend restricting the user's rights and configure network address restriction. The user needs read-rights in the PKI relevant containers and write access on the attribute ‘cvPublisherTrigger’ of the CA objects (objectClass ‘cvPkiCAAux’).

cn=PKIProxy,ou=PKIntegrated,o=system

objCl=User

2.3.10 User for dir/connector modifications

The IDM Driver Object needs to have certain rights within eDirectory to manage objects and attributes. In particular, the driver needs to update user and workstation objects, create certificate objects and manage the Certification Authority.

Security equivalence is not limited to user objects. We recommend to assign the driver administrative rights to your tree or branch. For security reasons, consider to use an object that is not able to authenticate to the tree, or a user object that has login disabled.

cn=IDMProxy,ou=IDM,o=system

objCl=User

2.3.11 User for ocsp/responder

The ocsp/responder needs a valid user account to authenticate to the LDAP server and read information from the certificate repository container. It is possible to use the same user account as the user for dir/connector authentication.

cn=PKIProxy,ou=PKIntegrated,o=system

objCl=User

2.3.12 User for scep/responder

The scep/responder needs to log in to the LDAP directory to create cvSCEPRequest objects. Therefore this user needs write access in the container for SCEP requests.

cn=SCEPProxy,ou=PKIntegrated,o=system

objCl=User

2.3.13 User or Role for Administrative Tasks

These roles are designed to process specific administrative tasks for cv act PKIntegrated. They can be implemented as a group object in eDirectory. For more details on iManager Task Assignment and eDirectory Right Requirements, please consult the cv act PKIntegrated Administration Guide. Occupant of roles could be system administrators, help desk users or power user.

cn=PKIAdmin,ou=PKIntegrated,o=system
objCl=User

cn=SCEPAdmin,ou=PKIntegrated,o=system
objCl=User

cn=CertRequest,ou=PKIntegrated,o=system
objCl=Group

cn=CertRevoke,ou=PKIntegrated,o=system
objCl=Group

2.4 ca/server

The ca/server can be installed on a server running the IDM Metadirectory services or the Remote Loader.

The ca/server component includes the IDM driver shim and the CA application software.

2.4.1 Install rpm Package

cv act PKIntegrated ships with four different rpm packages. Which one to use depends on the eDirectory version (8.7 or 8.8), on the Linux version (SLES10 or SLES 11) and on the IDM version (if remote loader is used). To choose the right rpm package, refer to the following table:

	Java 1.6 / Java 1.7
eDirectory 8.7 / no remote loader	caserver-3.x-x.x_eDir8.7.i586_ java1.6.rpm
eDirectory 8.8 / no remote loader	caserver-3.x-x.x_eDir8.8.i586_ java1.6.rpm
IDM 3.5 with remote loader	caserver-3.x-x.x_eDir8.7.i586_ java1.6.rpm
IDM 3.6 / IDM 4.0 with remote loader	caserver-3.x-x.x_eDir8.8.i586_ java1.6.rpm

The rpm package is installed with the following command:

```
rpm -i <path to rpm file>/<name of rpm file>
```

2.4.2 3rd Party Packages

After the rpm installation several 3rd party packages must be installed if they do not exist. Please check that there are not several versions of the same package installed. Usually the latest version should be kept.

The target folder depends on the eDirectory version and on whether the Remote Loader is used:

eDirectory 8.7 or IDM3.5 with Remote Loader: /usr/lib/dirxml/classes/

eDirectory 8.8 or IDM3.6 / 4.0 with Remote Loader: /opt/novell/eDirectory/lib/dirxml/classes/

The following packages from the 3rd party folder must be copied into the target folder:

- Bouncy Castle

  The appropriate Bouncy Castle JAR file and jsso.jar file need to be copied from the CD (3rd party folder) into the target folder. Which Bouncy Castle JAR file to use depends on the IDM version:

  - bcprov-jdk16-nnn.jar
- SecretStore Package jsso.jar

  Do not copy the jsso.jar file if it is already present in the target folder. You only have to replace the jsso.jar if there are exceptions found in the ndstrace when the driver tries to access the SecretStore.
- Apache Commons Logging commons-logging.jar

This package is needed by the driver and the monitor. It is important that the package does not contain the version number in its name otherwise the CA monitor will fail. Rename the package if necessary.

- Apache log4j.jar
- Syslog syslog4j.jar

2.4.3 SecretStore

To access the SecretStore the DirXML driver needs the storePKCS12.jar package and access to the java keystore containing the root certificate of the eDirectory.

The package ‘storePKCS12.jar’ must only be copied if you are using remote loader. Otherwise the package was copied by the rpm install into the correct location.

If remote loader is used the file storePKCS12.jar needs to be copied to the eDirectory server:

	SLES
eDirectory 8.7	/usr/lib/dirxml/classes/
eDirectory 8.8	/opt/novell/eDirectory/lib/dirxml/classes/

The file is installed by the rpm on the server where the remote loader runs. In addition the java keystore (sslkey.keystore) containing the root certificate of eDirectory has to be copied to the eDirectory server (see chapter 2.4.4 for further information on the keystore) and the new path to the file on the eDirectory server has to be configured in dir/connector settings (see chapter 2.6.3.2).

2.4.4 Export Trusted Root Certificate of the LDAP server Certificate

For secure communication between eDirectory LDAP server and ca/server component, the Trusted Root Certificates of the LDAP server's certificate needs to be exported and stored on the ca/server system. By default, this is the same certificate as the Selfsigned Certificate of the eDirectory CA.

It is recommended to verify the CA which has signed the LDAP server's certificate first, and then export the Trusted Root Certificate from the LDAP server's certificate object in eDirectory.

- In iManager, select Role "LDAP", Task "LDAP Options", Tab "View LDAP Servers"
  - Select your LDAP Server and the Tab "Connections"
    - Make a note of the Server Certificate Name. You need to access this eDirectory object with the next task.
      SSL CertificateDNS
  - Click Cancel
- Select Role "eDirectory Administration", Task "Modify Object"
  - Object Name: Browse to the KMO Object referenced by the LDAP Server. You can find this object in the NCP Server Context. Keep in mind that the eDirectory Object name has the NCP Server name attached.

SSL Certificate DNS - CV1.IT.CV

- Click Next
- Select Tab "Trusted Root Certificate"
- Click Export
    - Do you want to export the private key with the certificate?
        - Select No
    - Click Next
    - Select an Output Format
        - Select File in binary DER format
    - Click Next
    - Click on the HTML Link "Save the exported certificate to a file"
        - Select Save to Disk
        - Save file locally and transfer it to the PKI Server as /opt/cryptovision/etc/<Name of CA>.der
          /opt/cryptovision/etc/PKI-TREE_CA.der
    - Click Close
    - Click OK

For configurations where the LDAP server and the SecretStore Service Server have certificates signed by different CAs, the steps described above have to be repeated for the SecretStore service server.

2.4.5 Import Trusted Root Certificate into Certificate Store

The ca/server connects to Novell eDirectory via LDAP over TLS. To be able to establish the TLS handshake, the Trusted Root Certificate of the LDAP server's Certificate needs to be imported into the certificate store of the ca/server. The java utility keytool can be used to manage the Certificate Store.

The following command simplifies the certificate import and creates the Certificate Store /opt/cryptovision/sslkey.keystore after you provide an initial password.

- /opt/cryptovision/bin/createKeyStore /opt/cryptovision/etc/<Name of CA>.der
  /opt/cryptovision/bin/createKeyStore /opt/cryptovision/etc/PKI-TREE_CA.der
    - Password: <Password>
      Password: cvpass
    - Trust this certificate? yes
      Trust this certificate: yes

2.4.6 Configure ca/server

The ca/server sample configuration file is /opt/cryptovision/etc/caconfig.xml. Which configuration file will be used is set in the driver configuration. The following table lists the configuration parameters:

Tag	Parent	Description	Example
Configuration		Root tag.	
CATemplateInterface	Configuration	CA template for all CAs whose Subject DN ends with the DN in the id. This tag may occur multiple times. Attributes: id partial DN	<CATemplateInterface id="o=system">
className	CATemplateInterface	Class name of the CA template to be used. The value must be com.cryptovision.pkintegrated.catemplates.BaseCATemplate. Change this value only if you got a modified template that fits your needs.	<className>com.cryptovision.pkintegrated.catemplates.BaseCATemplate</className>
pinprintcommand	CATemplateInterface	Pathname of the PIN print command shell. A simple sample will be found here: /opt/cryptovision/bin/pinprint.	<pinprintcommand>/opt/cryptovision/bin/pinprint</pinprintcommand>
HSMInterface	Configuration	Determines the HSM configuration. Which configuration will be used depends on the id attribute: If the Subject DN ends with the partial DN of the configuration then this configuration will be used. This tag may occur multiple times. Attributes: id partialDN	<HSMInterface id="cn=HSM CA,ou=pkintegrated,o=system">
className	HSMInterface	The value must be set to com.cryptovision.pkintegrated.hsminterface.implementation.PKCS11HSM if a hardware HSM should be used. If the value is set to com.cryptovision.pkintegrated.hsminterface.implementation.SoftkeyHSM a software crypto provider will be used. Please contact cryptovision If a special HSM must be integrated without using the PKCS#11 interface.	<className>com.cryptovision.pkintegrated.hsminterface.implementation.SoftkeyHSM</className>
keystorepath	HSMInterface	Pathname of the keystore which will be used to store the keys. Only com.cryptovision.pkintegrated.hsminterface.implementation.SoftkeyHSM	<keystorepath>/opt/cryptovision/keys/pkintegratedSystem.keystore</keystorepath>
passphrase	HSMInterface	Optional: If set this passphrase will be used to protect the keystore. If no passphrase is given you will be prompted by a monitor application to set the passphrase. You can start the monitor from the commandline: /opt/cryptovision/bin/startmonitor.	<passphrase>an other passphrase</passphrase>
usePincache	HSMInterface	If set tor true the password must only be set once for a CA. This feature is not available if a PIN Pad Reader is used.	<usePincache>true</usePincache>
monitorId	HSMInterface	This value must be set to RMIMonitor. This value is a reference to the monitor tag described below.	<monitorId>RMIMonitor</monitorId>

Tag	Parent	Description	Example
pkcs11filename	HSMInterface	Pathname of the PKCS#11 library to be used. Only com.cryptovision.pkintegrated.hsminterface.implementation.PKCS11HSM	<pkcs11filename>/usr/local/lib/libcs2_pkcs11-1.4.6.so</pkcs11filename>
slotname	HSMInterface	Name of the slot to be used. Only com.cryptovision.pkintegrated.hsminterface.implementation.PKCS11HSM	<slotname>CryptoServer Device '3001@172.17.2.1' - Slot No: 0</slotname>
fipsMode	HSMInterface	If set to true and the HSM supports the FIPS mode then this mode will be used. Only com.cryptovision.pkintegrated.hsminterface.implementation.PKCS11HSM	<fipsMode>false</fipsMode>
generateExportableKeyOnHSM	HSMInterface	Set this value to true if the HSM supports the export of keypairs. Not all HSMs support this feature. Only end entity keys will be exported. Only com.cryptovision.pkintegrated.hsminterface.implementation.PKCS11HSM	<generateExportableKeyOnHSM>true</generateExportableKeyOnHSM>
Monitor	Configuration	The monitor enables the CA operator to enter the PIN. The monitor client must be started on the server where the CA with its driver runs. Just execute the script startmonitor located in /opt/cryptovision/bin. The client will be registered by the server and you will be prompted if any input is needed. This tag may occur multiple times. Attributes: id Identifier of the monitor interface, Referenced by the HSM interfaces.	<Monitor id="RMIMonitor">
className	Monitor	Class name of the monitor to be used. The value should not be modified.	<className>com.cryptovision.pkintegrated.monitor.server.MonitorRMIServer</className>
port	Monitor	MonitorRMIServer: Port number of the monitor service.	<port>1099</port>
serverTimeout	Monitor	MonitorRMIServer: Timeout value of the server in milliseconds. If no monitor client is started within the defined time the CA server interrupts the actual task with an error message.	<serverTimeout>10000</serverTimeout>
clientTimeout	Monitor	MonitorRMIServer: Timeout value of the client in milliseconds. If no input from the monitor client is received within the defined time the CA server interrupts the actual task with an error message.	<clientTimeout>10000</clientTimeout>
Syslog	Configuration	All CA relevant events are logged into the system log. For this purpose the system log must support the tcp protocol. Please enable the tcp protocol in the syslog configuration on the server where the logfile entries should be written. cv act PKIntegrated was tested with syslog-ng.	<Syslog>

Tag	Parent	Description	Example
clazz	Syslog	Optional. Only on Windows systems: If clazz is set to ‘com.cryptovision.pkintegrated.logging.SyslogWindows’ all other Syslog parameters will be ignored and the log4j logging configuration will be used instead.	
host	Syslog	Hostname of the logging server.	<host>172.17.2.91</host>
port	Syslog	Port to be used on the logging server.	<port>514</port>
useSSL	Syslog	Set this value to true if SSL is to be used.	<useSSL>false</useSSL>
trustStore	Syslog	Set this value to the pathname of the trust store if SSL is to be used.	<trustStore/>
trustStorePassword	Syslog	Set this value to the password of the trust store.	<trustStorePassword/>
keyStore	Syslog	Set this value to the pathname of the key store if SSL with client authentication is to be used.	<keyStore/>
keyStorePassword	Syslog	Set this value to the password of the key store.	<keyStorePassword/>
facility	Syslog	Set the facility to a value that corresponds with your syslog configuration. Refer to the syslog manual. Valid facilities: kern, user, mail, daemon, auth, syslog, lpr, news, uucp, cron, authpriv, ftp, local0 - local7	<facility>user</facility>
appendLineNumber	Syslog	If set to true then a line number will be appended at the end of each log entry. On each driver start the line number will be reset to a value of 1.	<appendLin-Lin-eNumber>true</appendLineNumber>
appendHash	Syslog	If set to true then a hash value will be appended at the end of each log entry.	<appendHash>true</appendHash>
hashAlgorithm	Syslog	Valid algorithms: MD5, SHA1, SHA160, SHA256, SHA384, SHA512	<hashAlgorithm>SHA1</hashAlgorithm>
appendChecksum	Syslog	If set to true then a checksum will be appended at the end of each log entry.	<appendChecksum>true</appendChecksum>
checksumAlgorithm	Syslog	Valid algorithms: CRC32, ADLER32	<checksumAlgorithm>CRC32</checksumAlgorithm>
continuousChecksum	Syslog	If set to true then the previous checksum will be considered when the actual checksum is calculated.	<continuousChecksum>true</continuousChecksum>
EmailNotification	Configuration	Email notification configuration for all CAs whose Subject DN ends with the DN in the id. This tag may occur multiple times. Attributes: id partial DN	<EmailNotification id="o=system">
className	EmailNotification	Class name of the notification implementation.	<className>com.cryptovision.pkintegrated.notification.EMailNotification</className>
smtpHost	EmailNotification	Hostname of the email server.	<smtpHost>127.0.0.1</smtpHost>
from	EmailNotification	Sender email address.	<from>admin@domain.sample</from>

Tag	Parent	Description	Example
cc	EmailNotification	CC email address.	<cc>cc@domain.sample</cc>
sendAllToAdmin	EmailNotification	If this value is set to true then the email address of the entity (user) will be ignored and all notifications will be sent to the admin email address.	<sendAllToAdmin>true</sendAllToAdmin>
adminAddress	EmailNotification	Admin email address.	<adminAddress>admin@domain.sample</adminAddress>
firstNotificationSubject	EmailNotification	Subject of the first notification message. The following placeholders can be used: {certificate}, {issuer}, {subject}, {serNum}, {notBefore}, {notAfter}, {certType}, {repositoryDn}, {ownerReference}, {eMail}, {firstName}, {lastName}, {notificationCount}, {present}	<firstNotificationSubject>First Notification</ firstNotificationSubject>
firstNotificationBody	EmailNotification	Body of the first notification message. For placeholders see above.	<firstNotificationBody>Dear {lastName}, please update your certificate. </ firstNotificationBody>
secondNotificationSubject	EmailNotification	Subject of the second notification message. For placeholders see above.	<secondNotificationSubject>Second Notification</ secondNotificationSubject>
secondNotificationBody	EmailNotification	Body of the second notification message. For placeholders see above.	<secondNotificationBody>Dear {lastName}, please update your certificate.</ secondNotificationBody>

Please assure that the configuration file is stored as a valid XML file. If you are using umlauts or any other special characters keep in mind that these characters must be UTF-8 encoded.

Make a copy of the sample file and modify it so it fits your need. Each driver can have its own configuration file.

With this version of cv act PKIntegrated one driver can handle several mandators. And each mandator can hold several CAs, but only one active. Each CA may have its own CA, HSM and email notification configuration.

The configuration for a CA is found using the subject DN of the CA. The distinguished name will be compared with the id attribute of the configuration tag. If the ending of the distinguished name equals to the id attribute this configuration will be used. If there is more than one configuration where the distinguished name matches then the configuration where the largest correlation occurred will be used.

2.4.7 Syslog Configuration

The CA uses for its logging the syslog mechanism of linux. The advantage is that there are solutions available which are able to sign or encrypt the logging entries.

Usually syslog-ng is installed on a linux system. The syslog-ng configuration must be modified in order to match the ca/server configuration. At least you have to enable the tcp protocol.

Please consult the syslog-ng manual.

Only on Windows systems: If you prefer to use log4j for logging purposes set the clazz attribute in the syslog section of the configuration to `com.cryptovision.pkintegrated.logging.SyslogWindows`.

```
      <Syslog>
            <clazz>com.cryptovision.pkintegrated.logging.SyslogWindows</clazz>
      </Syslog>
```

Do not set this tag on a Linux system wherewe recommend using a syslog configuration which is the default configuration.

2.4.7.1 Sample Syslog Configuration

To enable the tcp protocol and log all ca/server messages into the the /var/log/pkintegrated file you have to add the following lines into the syslog-ng configuration file /etc/syslog-ng/syslog-ng.conf:

```
(…)

source network {
      tcp(ip("0.0.0.0") port(514));
};

(…)

filter f_user       { facility(user); };

(…)

# pkintegrated
destination pkintegrated { file("/var/log/pkintegrated"); };
log { source(network); filter(f_user); destination(pkintegrated); };

(…)
```

In the syslog section of the ca/server configuration (see chapter 2.4.6) the host must be set to the ip address of the local server, the port must be set to ‘514’ and the facility to ‘user’ to meet this sample syslog-ng configuration.

2.4.8 Configure e-mail notification

cv act PKIntegrated supports an e-mail notification function. The purpose of this function is to inform the certificate owner (or some other party), when his certificate is going to expire.

The notification templates are configured in the configuration file of the ca/server. There are several predefined placeholders that you can use. The sample configuration file contains a notification configuration where several placeholders are used. A list of all valid placeholders can be found in the previous chapter.

If you do not want to notify your users but the CA administrator then activate the sendAllToAdmin feature in the configuration file.

E-mail notification behavior is configured with the driver parameters Notification Interval, Notification Delay, Notification Sizelimit, First Notification, and Second Notification (see chapter 2.6.3.1). If you want to turn off e-mail notification, set Notification Interval to 0.

The number of notifications that have been sent for a certain certificate is stored in the eDirectory attribute cvNotificationCount of the corresponding object. If no e-mail address is available, cvNotificationCount is automatically set to 100. If an e-mail address is entered at a later point of time, the attribute cvNotificationCount should be deleted, which enables an e-mail notification. A notification will be sent immediately after the deletion, if the number of days before certificate expiry is smaller than the value in the notification interval.

2.4.9 Configure Remote Loader Instance

This configuration is only required, if the dir/connector IDM driver shim is connecting via remote loader to the Metadirectory server. rdxml is the executable on Linux that enables the Metadirectory engine to communicate with the Identity Manager drivers.

The Remote Loader configuration file for the ca/server driver shim is /opt/cryptovision/etc/caserverRemote.conf. The following list displays the configuration file parameters:

Parameter	Description	Example
-class -cl	Java class name of driver shim	-class com.cryptovision.pkintegrated.driver.ca.caserverDriverShim
-commandport -cp	Command port a remote loader is listening for commands such as Start, Stop and Change Trace Level. Each instance of the remote loader must have a unique command port number. The default command port is 8000.	-commandport "port=8001"
-connection -conn	Connection port a remote loader is listening for connections from the Metadirectory server to exchange data. Each instance of the remote loader must have a unique connection port number. The default connection port is 8090. For SSL keystore and storepass have to be added.	-connection "port=8091" "port=8091 keystore='/opt/cryptovision/sslkey.keystore' storepass=<password of the keystore>"

Parameter	Description	Example
-password -p	Specifies the password for command authentication. This password must be the same as the first password specified with setpasswords for the loader instance being commanded. If a command option (for example, unload or tracechange) is specified and the password option isn't specified, the user is prompted to enter the password for the loader that is the target of the command.	-password cvremote
-tracefile -tf	File name of driver trace messages	-tracefile /var/log/cryptovision/caserver_DirXML.log
-trace -t	Drivers trace level. The higher the number, the more details will be added to the trace file. To disable trace, set a value of 0 A trace level beyond 5 is reserved for driver debugging by the developer.	-trace 3

For a complete list of Parameters, see IDM online documentation, chapter "Configuring the Remote Loader by Using Command Line Options"

The Remote Loader configuration file for setting new passwords is /opt/cryptovision/etc/caserverRemoteInit.conf, storing basically the same configuration information as /opt/cryptovision/etc/caserverRemote.conf. The following list shows the additional configuration file parameters not supported in /opt/cryptovision/etc/caserverRemote.conf:

Parameter	Description	Example
-setpasswords -sp	The Remote Loader Password is used to authenticate the Driver to the Remote Loader. The Driver Object Password is used to authenticate the Remote Loader to the Metadirectory Server. These passwords need to match the Remote Loader Password and the Driver Object Password of your Driver configuration.	-setpassword cvremote cvobject

For the current version, both files need to be updated when making configuration changes.

2.4.10 Start/Stop Remote Loader

With the installation of the ca/server, the script /etc/init.d/caserver_remoteloader is added to run the remote loader. To start the Remote Loader each time the server is rebooted, add it to runlevel 3 and 5. For SLES, this can be done using YaST or using `chkconfig`.

The script expects the following paths to be valid:

Description	eDirectory 8.7.x	eDirectory 8.8.x
rdxml binary file	/usr/bin/rdxml	/usr/bin/rdxml
rdxml configuration file	/opt/cryptovision/etc/ caserverRemote.conf	/opt/cryptovision/etc/ caserverRemote.conf
rdxml init configuration file	/opt/cryptovision/etc/ caserverRemoteInit.conf	/opt/cryptovision/etc/ caserverRemoteInit.conf
ca/server binary files	/opt/cryptovision/bin	/opt/cryptovision/bin
Working directory	/var/log/cryptovision	/var/log/cryptovision

The script supports the following arguments:

Parameter	Description	Example
start	start the remote loader and dir/connector	/etc/init.d/caserver_ remoteloader start
stop	stop the remote loader and dir/connector	/etc/init.d/caserver_ remoteloader stop
status	displays the current status of the remote loader and dir/connector	/etc/init.d/caserver_ remoteloader status

2.4.11 Customized certificate templates

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

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.

CA certificate templates can be customized too. If you need a customized CA certificate template please contact cryptovision to get advice.

Further information on this topic is available in cv act PKIntegrated Administration Guide, chapter 2.2.

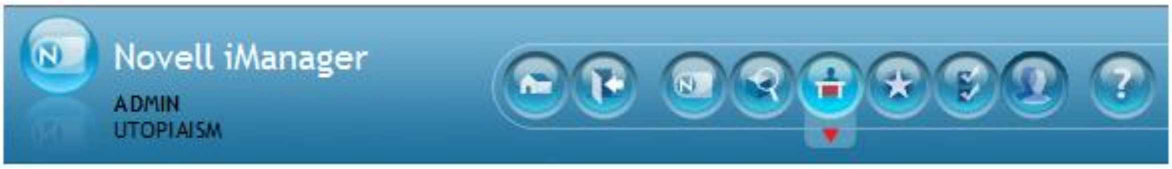
2.5 admin/extension

2.5.1 Installation of NPM package in Novell iManager

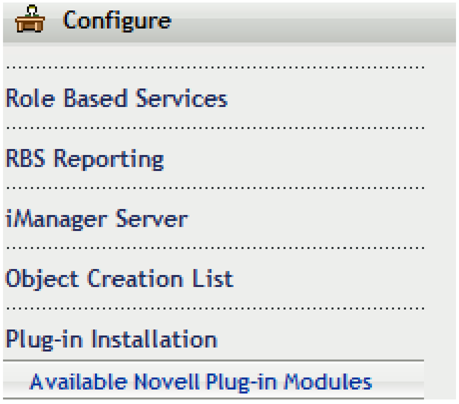
The admin/extension is provided as an NPM package that can be installed in Novell iManager:

- admin_extension_3.0.0_java1.6.npm

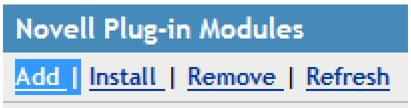
- In iManager, select Configure on Top Navigation bar



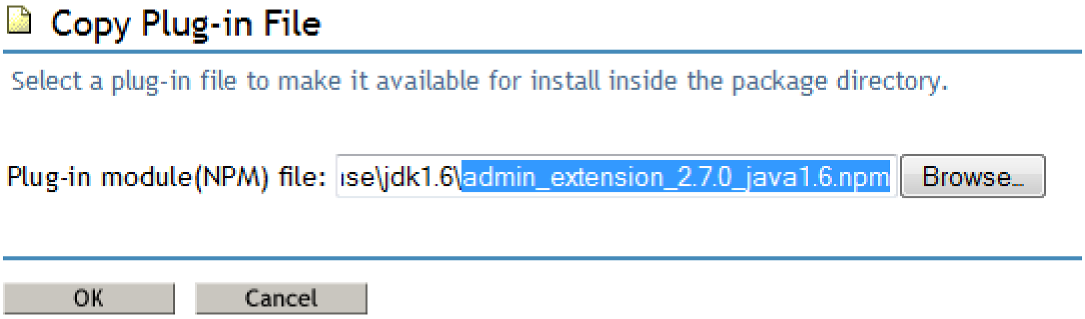
- Select Role Plug-in Installation, Task Available Novell Plug-in Modules



  - Click on Add



    - Browse to and open <Path to npm file on your installation medium> (please choose the appropriate package according to the version of your iManager installation).
      On a Linux Workstation or Server:
      /media/cdrom/jdk1.x/admin_extension_3.x.x_java1.x.npm
      On a Windows Workstation or Server:
      d:\jdk1.x\admin_extension_3.x.x_java1.x.npm

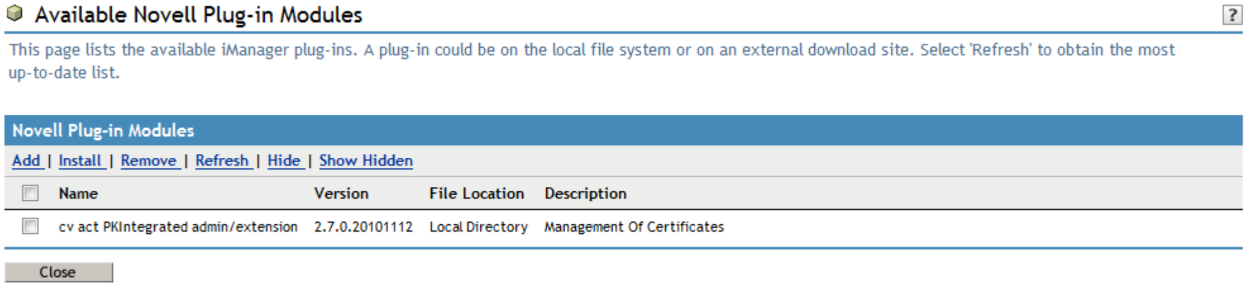


Copy Plug-in File

Select a plug-in file to make it available for install inside the package directory.

Plug-in module(NPM) file: ıse\jdk1.6\admin_extension_2.7.0_java1.6.npm

- Click OK
- Mark the cv act PKIntegrated admin/extension Plug-in
- Click on Install

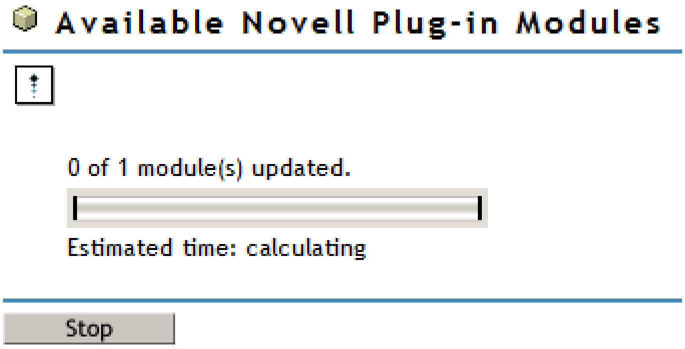


Available Novell Plug-in Modules

This page lists the available iManager plug-ins. A plug-in could be on the local file system or on an external download site. Select 'Refresh' to obtain the most up-to-date list.

Novell Plug-in Modules

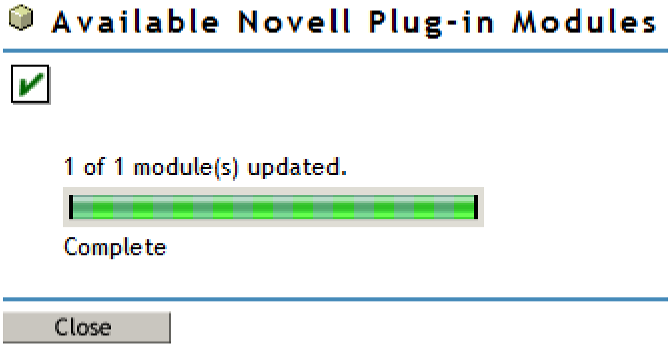
□	Name	Version	File Location	Description
□	cv act PKIntegrated admin/extension	2.7.0.20101112	Local Directory	Management Of Certificates



Available Novell Plug-in Modules

0 of 1 module(s) updated.

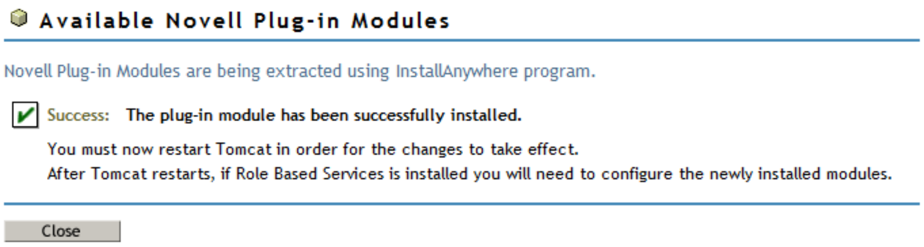
Estimated time: calculating



Available Novell Plug-in Modules

1 of 1 module(s) updated.

Complete



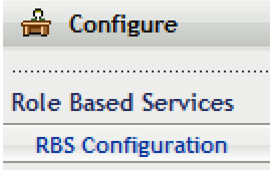
Available Novell Plug-in Modules

Novell Plug-in Modules are being extracted using InstallAnywhere program.

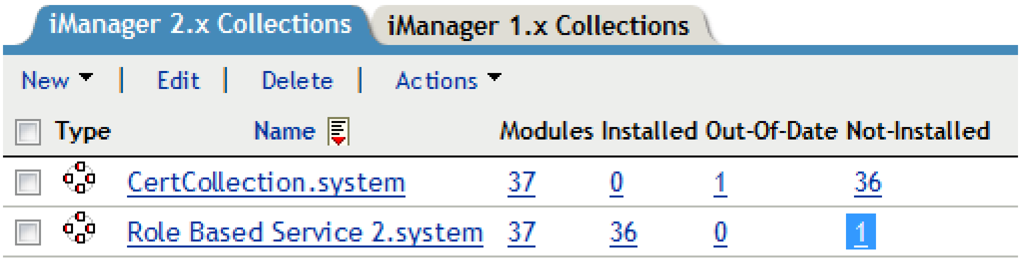
☑ Success: The plug-in module has been successfully installed.

You must now restart Tomcat in order for the changes to take effect.
After Tomcat restarts, if Role Based Services is installed you will need to configure the newly installed modules.

- Click on continue to configure RBS (if RBS is in use within your environment)

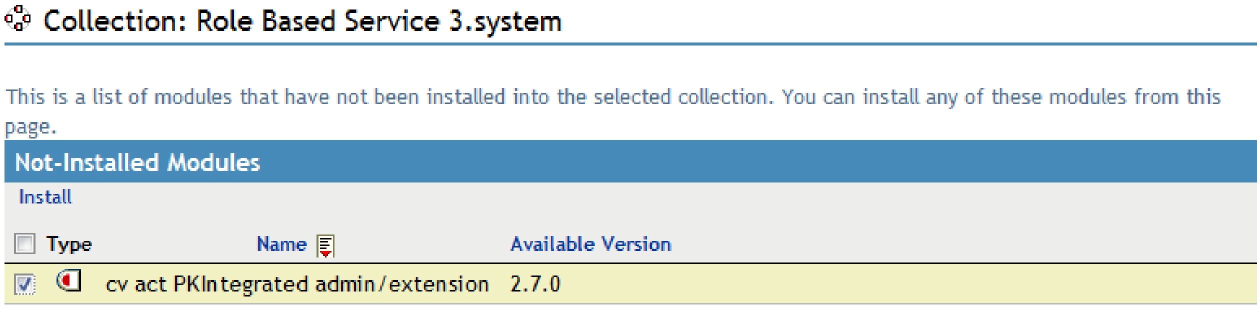


- Click on the 1 Not-Installed Modules



□ Type	Name	Modules	Installed	Out-Of-Date	Not-Installed
□	CertCollection.system	37	0	1	36
□	Role Based Service 2.system	37	36	0	1

- Mark the module cv
- Click on Install



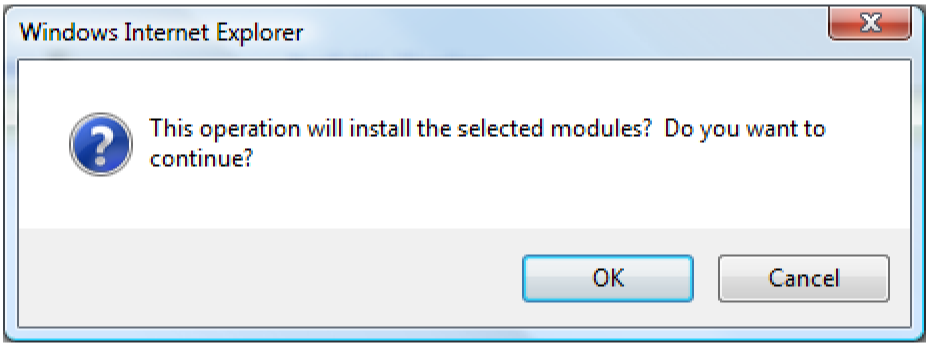
Collection: Role Based Service 3.system

This is a list of modules that have not been installed into the selected collection. You can install any of these modules from this page.

Not-Installed Modules

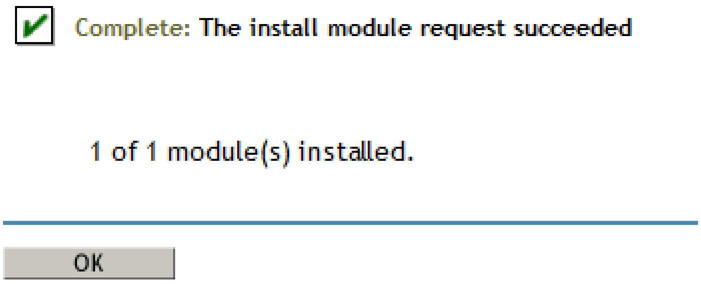
□ Type	Name	Available Version
☑	cv act PKIntegrated admin/extension	2.7.0

- Click OK



Windows Internet Explorer

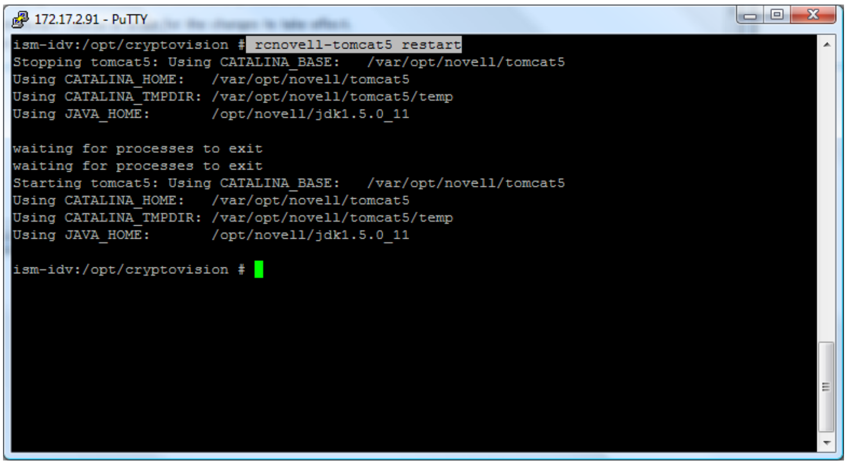
This operation will install the selected modules? Do you want to continue?



☑ Complete: The install module request succeeded

1 of 1 module(s) installed.

- Log out of iManager
- Restart Tomcat



```
ism-idv:/opt/cryptovision # rcnovell-tomcat5 restart
Stopping tomcat5: Using CATALINA_BASE:   /var/opt/novell/tomcat5
Using CATALINA_HOME:   /var/opt/novell/tomcat5
Using CATALINA_TMPDIR: /var/opt/novell/tomcat5/temp
Using JAVA_HOME:       /opt/novell/jdk1.5.0_11

waiting for processes to exit
waiting for processes to exit
Starting tomcat5: Using CATALINA_BASE:   /var/opt/novell/tomcat5
Using CATALINA_HOME:   /var/opt/novell/tomcat5
Using CATALINA_TMPDIR: /var/opt/novell/tomcat5/temp
Using JAVA_HOME:       /opt/novell/jdk1.5.0_11

ism-idv:/opt/cryptovision # 
```

2.5.2 3rd Party Packages

Additionally, the following files are needed for iManager:

- bcprov-jdk16-137.jar
- AXIS axis.jar
- Apache-Commons File Upload commons-fileupload.jar
- dom4j.jar
- Apache Commons Logging commons-logging.jar
- jsso.jar

Copy the JAR files from the “3rd Party” folder from the CD to the following target folder:

SLES10/ SLES11: /var/opt/novell/iManager/nps/WEB-INF/lib/

Notice: If you update your older admin/extension version, please check that in the named target folder above only the cvPKIntegratedAdminExt.jar exists. cv*.jar files from earlier versions of admin/extension have to be deleted. Also check that there are not different versions of the same package. Usually it is a got choice to keep the latest version.

Restart the instance of tomcat that is running iManager using the appropriate command on your iManager system.
For Linux, this is `rcnovell-tomcat4 restart` resp. `rcnovell-tomcat5 restart`.

2.5.3 Configuration of admin/extension

You can configure the behavior of the admin/extension by setting the appropriate parameters in the configuration file `iManager.xml` in the folder `$TOMCAT_HOME/webapps/nps/portal /modules/cv/configuration`

You have to restart the instance of tomcat that is running iManager after changing the configuration.

2.5.3.1 Section "Parameter"

- CertificateAutoInstall (boolean)
  Should the certificate be installed automatically in the browser or on the smartcard? (default: true)
- AutoRequest (boolean)
  If only one certificate type is available for request, should the selection of the certificate be skipped? (default: true)
- AutoRequestDisabledType (String)
  List of certificate type where the AutoRequest should not be true (for server certificates (serv, ocsp, etc.)
- AutoExtensionAssignment (boolean)
  Should the admin/extension add the object extension `cvUserAttribAux` automatically when a certificate is requested for a user or workstation? (default: true)
  Note: write-rights to the class "Object class" are necessary to extend the object with the `cvUserAttribAux` class.
- MandatorsPath (String)
  Path to the mandator list. If this is not set, the admin/extension will search for the mandator list. This can take a long time in large trees. It is recommended to set the path to the mandator list to speed up the search.
- EnableUserMandators (boolean)
  It is recommended setting this value to true if you have more than one mandator and your users are only allowed to use one or a subset of the mandator.
- UserMandator (String)
  List of all mandators users are allowed to use. The value of each entry must contain a valid path to a mandator in dot notation.

2.5.3.2 Section "SecretStore"

For providing the keys from the SecretStore (Recovery Key or key of the user) for download the admin/extension has to have access to the server where the SecretStore is installed. Additionally (because the connection to SecretStore uses LDAP/SSL) the Trusted Root Certificate (see 2.4.4) from the eDirectory server has to be imported into the Java keystore of the iManager server (i.e. tomcat). Use the Java `keytool` to import the certificate into the keystore `$JAVA_HOME/jre/lib/security/cacerts`" by this command line:

```
/opt/novell/eDirectory/lib/nds-modules/jre/bin/keytool -import -file <filename>
-keystore $JAVA_HOME/jre/lib/security/cacerts
```

$JAVA_HOME depends on your iManager/tomcat installation. The password of the keystore is most likely "changeit".

- Host (String)
  Hostname of the server with the SecretStore server

- Port (Integer)
  LDAP/S portnumber of the server with the SecretStore server

2.5.3.3 Section „SCEP“

In this section you can define the default container of your requests and the certificate template that should be used.

- DefaultContainer (String)
  Dot notated path to the default container containing the SCEP requests.
- CertificateType (String)
  Name of the certificate template to be used certifying the SCEP requests.

2.5.3.4 Example configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
      <!--
            Restart iManager after changing configuration!
      -->
       <Parameter>
            <CertificateAutoInstall>true</CertificateAutoInstall>
            <AutoRequest>true</AutoRequest>
            <AutoRequestDisabledType>ocsp</AutoRequestDisabledType>
            <AutoRequestDisabledType>serv</AutoRequestDisabledType>
            <AutoExtensionAssignment>true</AutoExtensionAssignment>
            <MandatorsPath>Mandators.PKIntegrated.cryptovision</MandatorsPath>
            <EnableUserMandators>true</EnableUserMandators>
            <UserMandator>Sub1.Mandators.PKIntegrated.system</UserMandator>
            <UserMandator>Sub2.Mandators.PKIntegrated.system</UserMandator>
</Parameter>
      <SecretStore>
      <!--
            Use this section to configure LDAP access to Secret Store Server.
            Important: You need to have the server's SSL certificate installed
            into the keystore.
      -->
            <Host>localhost</Host>
            <Port>636</Port>
       </SecretStore>
      <SCEP>
            <DefaultContainer>SCEPRequests.PKIntegrated.system</DefaultContainer>
            <CertificateType>scep</CertificateType>
      </SCEP>
</Configuration>
```

2.5.3.5 Customized certificate templates in iManager

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

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 certificate 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>
```

Further information on this topic is available in cv act PKIntegrated Administration Guide, chapter 2.2.

2.6 dir/connector

cv act PKIntegrated commands are triggered by events in eDirectory. These events are detected by the eDirectory Interface and processed by the Meta Directory Engine of Novell Identity Manager. The purpose of the dir/connector is to connect the ca/server application software to the Meta Directory Engine. In Novell IDM-terms, the dir/connector is the Identity Manager Connected Application Driver.

You have 2 options to import the cv act PKIntegrated preconfigured Driver Template CADriver.xml into an Identity Manager Driver Set:

- Use Novell iManager to install and configure the Identity Manager Driver for cv act PKIntegrated.
- Use Novell Designer for Identity Manager to design and deploy the Identity Manager Driver for cv act PKIntegrated.

2.6.1 Create Driver with Novell iManager

Novell iManager might be the preferred utility for IDM Administrators to add new drivers to their driver set. iManager will immediately create all necessary eDirectory objects and allows modifications on these objects in the live system.

- In iManager, select Role "Identity Manager Utilities", Task "New Driver"
  - Decide and select the proper driver placement
    - Most likely you need to select "In an existing driver set" if you have Identity Manager already deployed and connected to several other systems and you plan to use the Remote Loader to run the Driver Shim.
    - Most likely you need to select "In a new driver set" and configure a new driver set if you use an exclusive Meta Directory Engine to run the Driver Shim.
  - Import a driver configuration from the client (.XML file)
    - Browse to <Path to the CADriver.xml file on your installation medium>
      On a Linux Server or Workstation: /media/cdrom/CADriver.xml
      On a Windows Server or Workstation: d:\CADriver.xml
    - Define Driver Settings
    - Finish

2.6.2 *Create Driver with Novell Designer for Identity Manager*

Novell Designer for Identity Manager might be the preferred utility for IDM Consultants to design and model new drivers. Designer allows project management functionality, offline modeling and testing before deploying into a live system.

- Start Novell Designer for IDM
  - Open the IDM project you want to install the cv act PKIntegrated dir/connector
  - Open the Modeler in Developer mode
  - From the Palette, drag and drop the Generic App Driver under Tools into an empty space close to the Identity Vault.
    - Browse to <Path to the CADriver.xml file on your installation medium> and press Run to launch the Driver Import Wizard.
      On a Linux Workstation or Server: /media/cdrom/CADriver.xml
      On a Windows Workstation or Server: d:\CADriver.xml
    - Define Driver Settings
    - Finish

2.6.3 *Driver Settings*

2.6.3.1 Driver Configuration

The Driver Template requests information for these Driver Configuration Parameters

Parameter	Description	Example
Driver Module	Java Use this setting to run the dir/connector on the Metadirectory Engine Native not supported for this java driver Connect to Remote Loader Use this setting to run the dir/connector on a different server than the Metadirectory Engine or to connect via RemoteLoader to the local host.	Java
Driver Module name	Java class of the Driver Shim	com.cryptovision.pkintegrat-ed.driver.ca.caserverDriverShim
Driver object password	Password used by the Remote Loader to authenticate itself to the Metadirectory Server	cvobject
Authentication ID	DN in LDAP format of a user with read-rights on the PKI container and with write-rights to the attribute ‘cvPublisherTrigger’ of the CA objects (objectClass ‘cvCA’).	cn=PKIProxy, ou=PKIntegrated, ou=IT,o=CV
Authentication context	Hostname or IP address of the server with the certificate repository	cv1.cryptovision.com

Parameter	Description	Example

Remote loader connection parameters	only if Driver Module is configured to Connect to Remote Loader.
	hostname
	DNS name or IP Address of the host running Remote loader service. Default: localhost
	port
	Port where the Remote loader accepts connections from the remote interface shim. Default: 8091
	kmo object
	Key name of the KMO object used for SSL
		hostname= cv1.cryptovision.com
		port=8090
		kmo='RemoteCert'

Driver cache limit (kb)	Limits the size in kb of the driver cache. A value of 0 defines unlimited cache size.	0
Authentication password	Password of the User specified in parameter Authentication ID	cvpass
Remote loader password	Password to control access to the remote loader instance.	cvremote
Startup option	Auto start The driver starts automatically when the Metadirectory Server starts. Manual The driver must be started manually Disabled The driver does not run. No changes are stored in the event cache.	Auto start
CA configuration	Path of the CA configuration file.	/opt/cryptovision/etc/caconfig.xml
Mandator	Path of the mandator container in eDirectory in slash format (see example)	system\PKIntegrated\Mandator
LDAP port	Secure LDAP port number of the host specified in the parameter Authentication context	636
Heartbeat interval	For monitoring purposes: Publication Heartbeat Interval specified in seconds. If no documents are sent on the Publisher channel for this specified interval (duration of time), then a heartbeat document is sent by the driver. A value of 0 indicates that no heartbeat documents are to be sent.	300
Polling interval	Number of seconds the driver sleeps between two checks if the CRL has to be updated	10
CRL update delta	Number of seconds before the CRL needs to be updated. The update process is triggered at <CRL next update> - <CRL update delta>	120
Notification Interval	When this number of seconds has past, the CA checks in the repository, if there are certificates, whose remaining validity time is less than configured in the first notification or second notification parameter.	60
Notification delay	Number of seconds the e-mail notification process is interrupted in one round. The interruption lets the CA care about other tasks.	0
Notification sizelimit	maximum number of e-mails that are sent in one interval	100

Parameter	Description	Example
First notification	If there are less days remaining before the end of the validity period of the certificate, a first e-mail notification is sent.	60
Second notification	If there are less days remaining before the end of the validity period of the certificate, a second e-mail notification is sent.	30
Deploy plain private roamer key	We recommend storing roaming keys into SecretStore. Therefore disable this option by setting the attribute value to 'false' (default). Activating this option allows to customize the handling of the PKCS#8 encoded private key blob. Please be careful with this option and keep private keys protected all time.	false

2.6.3.2 Global Configuration Values (gcv)

The Driver Template requests information for these Global Configuration Values

Global Configuration Value	Description	Example
gcvRecoveryAdmin	DN of a user who gets the recovery keys stored in its SecretStore in LDAP syntax	cn=PKIRecovery, ou=PKIntegrated, ou=IT,ou=CV
gcvSecretStoreAdmin	DN of a user with write-rights to the SecretStore of managed users in LDAP syntax	cn=SSSAdmin, ou=SSS,ou=IT,o=CV
gcvLDAPSecStoreHost	Hostname or IP address of the server which is running the Novell SecretStore Service.	cv1.cryptovision.com
gcvLDAPSecStorePort	Secure LDAP port number of the LDAP server which is running the Novell SecretStore Service	636
gcvTrustedKeyStore	Path of the Java keystore with the trusted root certificate of LDAP server's certificate. The TrustedKeyStore can be created with the script /opt/cryptovision/bin/createKeyStore. If remote loader is used, this is the path of the Java keystore on the eDirectory server (see chapter 2.4.1)	/opt/cryptovision/ sslkey.keystore

2.6.3.3 Named Passwords

The Driver Template requests information for these Named Passwords

Named Password	Description	Example
SecretStoreAdminPass	Password of the User specified in Global Configuration Value gcvSecretStoreAdmin	ssspass

It is absolutely necessary to store a note of the SecretStoreAdminPass in a secure place.

The password has to be kept absolutely confidential, and there is no need to type in the password again during normal operation. But in some cases, e. g. during update of cv act PKIntegrated, it has to be typed in again. Therefore it has to be available in such a case.

2.6.3.4 Security Equivalent

The IDM Driver Object needs to have certain rights within eDirectory to manage objects and attributes. In particular, the driver needs to update user and workstation objects, create certificate objects and manage the Certification Authority.

Security equivalence is not limited to user objects. We recommend to assign the driver administrative rights to your tree or branch. For security reasons, consider to use an object that is not able to authenticate to the tree, or a user object that has login disabled (see 2.3.8 for further information).

2.6.3.5 Excluded Users

Users that should not be managed by dir/connector, can be added to the exclude list. This will prevent the driver from processing any request for these users.

Even if the PKIRecovery user is added to the excluded users list, the recovery keys will be stored in this user's SecretStore.

2.6.3.6 Remark on DirXML-Association

When an object is created modified or removed a driver event will be fired.

If it is a new object then a random number will be generated from the driver application and stored in the DirXML-Association attribute of the object. By this association the driver decides whether the add or the modify path has to be performed.

If you like to restrict the objects you can solve this by modifying the driver as needed. If you need assistance, please contact cryptovision.

cv act PKintegrated uses a lot of helper objects which also get an association. For any licensing questions concerning the DirXML associations please contact Novell.

2.7 ocsp/responder

2.7.1 Introduction

cv act PKIntegrated supports OCSP (Online Certificate Status Protocol) via the additional product cv act ocsp/responder. The cv act ocsp/responder is a gateway that forms an additional interface for PKI use.

The procedure to query the validity of a certificate with OCSP is conceivably simple: an OCSP-capable client initially sends an inquiry with a given certificate number over the network to cv act ocsp/responder. Subsequently, cv act ocsp/responder verifies the validity of the certificate by querying the certificate repository in eDirectory for the revocation status and sends a response back, that indicates whether the given certificate is revoked or not. The sent data only amounts to several hundred bytes, which cannot be compared with the size of a revocation list.

cv act ocsp/responder derives the data directly from the eDirectory, in which cv act PKIntegrated is integrated. All revocation information is therefore always up to date.

2.7.2 Requirements

cv act ocsp/responder accepts OCSP requests via http using xinetd. Therefore xinetd has to be installed and running on the server where cv act ocsp/responder runs.

cv act ocsp/responder queries eDirectory via LDAP, an applicable user account has to be available and LDAP connection between cv act ocsp/responder and eDirectory has to be possible.

2.7.3 Install rpm package

The ocsp/responder rpm package is installed with the following command:

- rpm -i <path to rpm file>/ocsp_responder-1.2-0.x.i386.rpm
  rpm -i /media/cdrom/ocsp_responder-1.2-0.2.i386.rpm

If necessary, install the LDAP/SSL libraries rpm package with the following command first.

- rpm -i <path to rpm file>/libldapssl-0.0-0.i386.rpm
  rpm -i /media/cdrom/libldapssl-0.0-0.rpm

The rpm installation creates the user cv_ocsp. This is the user as whom the server should run.

2.7.4 Register port and service

Port number and Service of the ocsp/responder need to be registered on your Linux server.

Add the following line to /etc/services:

-
```
ocsp_responder        <your ocsp server port>/tcp     #<comment>
ocsp_responder        40000/tcp            #cv act PKIntegrated ocsp/responder
```

2.7.5 xinetd configuration

The rpm installation creates a configuration file /etc/xinetd/ocsp_responder with the following entries:

-
  ```
  service ocsp_responder
  {
     disable        = no
     socket_type    = stream
     wait           = no
     user           = cv_ocsp
     server         = /opt/cryptovision/bin/ocsp_responder
     server_args    = /opt/cryptovision/etc/ocsp_responder.conf
  }
  ```

2.7.6 Configure ocsp/responder

The ocsp/server configuration file is /opt/cryptovision/etc/ocsp_responder.conf. The following list displays the configuration file parameters.

For every CA there is a section defining these parameters relevant for this CA. These sections are divided by [CA].

Parameter	Description	Example
log_filename	Filename of the logfile of the ocsp/responder. If this entry is empty no logfile will be written.	/var/log/cryptovision/ocsp.log
log_level	error: only logs error info: also logs further debug messages. If this entry is empty or invalid, no log file will be written.	error
key_filename	The ocsp/responder needs a private key file to sign the responses. This is the filename of the key file.	/opt/cryptovision/etc/ocsp.pfx
key_password	The passphrase of the key	cvpass
CRLsafe	ldap: the ocsp/responder should connect to the certificate repository via LDAP ldapssl: the ocsp/responder should connect to the certificate repository via LDAP/SSL.	ldapssl
LDAP_servername	IP-address or hostname of the server which hosts the certificate repository.	cv1.cryptovision.com
LDAP_port	Portnumber of the server which hosts the certificate repository (normally 389 if LDAP is used and 636 if LDAP/SSL is used)	636

LDAP_repository	Distinguished name of the certificate repository	ou=CertRepository,
		ou=PKIntegrated,
		ou=IT,o=CV

LDAP_user	Distinguished name of a user with permission the read the certificate repository	cn=OCSPProxy,
		ou=PKIntegrated,
		ou=IT,o=CV

LDAP_password	Password of the User specified in parameter LDAP_user	cvpass
LDAP_sslcertificate_filename	Trusted Root Certificate of LDAP server, if SSL is in use.	/opt/cryptovision/etc/PKI-TREE_CA.der
CAcertificate_filename	CA Certificate of issuing CA	/opt/cryptovision/etc/TEST-CA.cer

2.8 scep/responder

2.8.1 Introduction

The registration process, when generating a digital certificate, is essential and the most important process within a Public Key infrastructure.

The router registration takes place via a special protocol called SCEP (Simple Certificate Enrollment Protocol). cv act PKIntegrated supports SCEP over the additional component cv act scep/responder and is therefore in the position to automatically generate, distribute, update and if necessary revoke certificates.

cv act scep/responder operates between eDirectory and router.

2.8.2 Requirements

cv act scep/responder accepts SCEP requests via http using xinetd. Therefore xinetd has to be installed and running on the server where cv act scep/responder runs.

cv act scep/responder queries eDirectory via LDAP, an applicable user account has to be available and LDAP connection between cv act scep/responder and eDirectory has to be possible.

In addition the Novell libldapsdk package, which is included in the cv act scep/responder installation package has to be installed manually.

2.8.3 Install rpm package

The scep/responder rpm package is installed with the following command:

- rpm -i <path to rpm file>/scep_responder-1.1-0.x.i386.rpm
  rpm -i /media/cdrom/scep_responder-1.1-0.3.i386.rpm

If necessary, install the LDAP/SSL libraries rpm package with the following command first.

- rpm -i <path to rpm file>/libldapssl-0.0-0.i386.rpm
  rpm -i /media/cdrom/libldapssl-0.0-0.rpm

The rpm installation creates the user cv_scep. This is the user as whom the server should run.

2.8.4 Register port and service

Port number and Service of the ocsp/responder need to be registered on your Linux server.

Add the following line to /etc/services:

-
  ```
  scep_responder     <your scep server port>/tcp     #<comment>
  scep_responder     40004/tcp             #cv act PKIntegrated scep/responder
  ```

2.8.5 xinetd configuration

The configuration file /etc/xinetd/scep_responder will be added *automatically* by the scep/responder installation package:

-
  ```
  service scep_responder
  {
      disable         = no
      socket_type     = stream
      wait      = no
      user      = cv_scep
  ```

```
        server          = /opt/cryptovision/bin/scep_responder
        server_args     = /opt/cryptovision/etc/scep_responder.conf
   }
```

2.8.6 Configure scep/responder

Note: Please also refer to section 2.5.3 (configuration of admin/extension) for further information about how to configure the iManager Plug-In for the usage of cv act scep/responder.

The scep/server configuration file is /opt/cryptovision/etc/scep_responder.conf. The following list displays the configuration file parameters:

Parameter	Description	Example
Log-file	Filename of the logfile of the scep/responder. If this entry is empty no logfile will be written.	/var/log/cryptovision/scep.log
Log-request	If this entry is set, also the request will be logged (for debugging purposes)	1
Pfx-file	The scep/responder needs a private key file to sign the responses. This is the filename of the key file.	/opt/cryptovision/ scep.pfx
Pfx-passwd	The passphrase of the key	cvpass
ldap-server	The hostname or the IP address of the server which hosts the SCEP requests repository	cv1.cryptovision.com
ldap-port	The port number of the server which hosts the SCEP requests repository	636
ldap-ssl-root	Because the connection to the eDirectory is via LDAP/SSL the server sends a certificate to authenticate itself. To verify this certificate the trusted root certificate which issued the server certificate is used to verify the server certificate. This parameter is the filename of the trusted root certificate.	/opt/cryptovision/etc/ PKI-TREE_CA
scep-dn	Distinguished name of the “SCEP admin” user	cn=SCEPAdmin, ou=PKIntegrated, ou=IT, o=CV
scep-passwd	Password of the User specified in parameter scep-dn	cvpass
scep-request-dn	Distinguished name of repository for the SCEP requests	ou=SCEPRequests, ou=PKIntegrated, ou=IT,o=CV
ca-dn	Distinguished name of the CA Object	cn=Test-CA, ou=PKIntegrated, ou=IT,o=CV
ca-repository-dn	Distinguished name of the CA Certificate Repository	ou=CertRepository, ou=PKIntegrated, ou=IT,o=CV

3 Novell eDirectory Object Classes and Attributes

3.1 Object Class cvCA

The object class cvCA is an effective class.

Superclass: Top

Naming attributes: CN

Mandatory attributes: ObjectClass, CN

Attribute Name	Type	Description
CN	String	Common Name.

3.2 Auxiliary Class cvPkiCAAux

The Auxiliary Class cvPkiCAAux is designed to extend cvCA objects

Naming attributes: -

Mandatory attributes: -

Attribute Name	Type	Description
CN	Sized String (1-64)	Common Name.
authorityRevocationList	Octet String	
cACertificate	Octet String	CA root certificate.
co	String	
crossCertificatePair	Octet String	
cvAllowedCertificateType	String Multivalued	Default certificate types user is allowed to request.
cvBridgeCAData	Octet String	From CA generated input data for the BCA in PKCS#10 format.
cvBridgeCADataType	String	Type of generated format. “PKCS#10” or “BCAFormat”
cvCADN	String	Distinguished name of CA .
cvCAKeyLength	Numeric String	Length of CA key.
cvCAListReference	DN	Link to CA list.
cvCAReference	DN	Link to CA object.
cvCAValidityPeriod	Numeric String	Validity period of CA root certificate. max. 65535 days (~ 180 years)
cvCRLLDAPUrl	String	Full qualified URL of the CRL distribution point.
cvCRLNextUpdate	Generalized Time	Date next scheduled CRL update.

Attribute Name	Type	Description
cvCRLReason	Numeric String	Revocation Reason (If missing default: 0).
cvCRLTrigger	String	Trigger for CA/CRL commands.
cvPublisherTrigger	String	Trigger used by Publisher Shim of the driver.
cvCRLValidityPeriod	Numeric String	Validity period of CRL. max. 2147483647 sec (~ 68 years)
cvCrossCertificatePairCert	Octet String	The appropriate cross certificate of the CA.
cvCertificateRevocationList	Stream	CRL.
cvCreateRequest	State	CA should generate BCA data for request to certify CA by the BCA.
cvDeltaLDAPUrl	String	Full qualified URL of the distribution point of the Delta CRL.
cvDeltaRevocationList	Octet String	Delta CRL.
cvGenerateType	String Multivalued	All certificate types for which a key generation is permitted.
cvKeyLength	Numeric String	Key length of key to be generated.
cvKeyRecoveryType	String Multivalued	Certificate types for which CA creates a recovery key.
cvLastIssuedCertificate	Octet String	The last certificate issued by the CA.
cvLastIssuedCrossCertificatePair	Octet String	The last cross certificate pair issued by the CA.
cvLastStatus	String	Status of previous request.
cvMaxKeyLength	Numeric string	Maximum allowed key length.
cvMinKeyLength	Numeric String	Minimum allowed key length.
cvMultipleCertificatesAllowed	String Multivalued	Default certificate types user is allowed to request more than once.
cvNameOverwriteAllowed	String	Certificate types where the name is not taken from the path in eDirectory.
cvOCSPHTTPUrl	String	Full qualified URL of the OCSP Responder.
cvRepositoryListReference	DN	Reference to the repository of this CA.
cvRepositoryReference	DN	References to all repository entries of this CA.
cvRequestType	String Multivalued	All certificate types for which a request is permitted.
cvRoamerType	Case Exact String	All certificate types that can be accessed by pki/roamer.
cvSelectedCAList	DN	Specifies the CA List of the issuing CA while processing a certificate request.
cvSelectedCertificateID	String	Certificate to revoke. Format: <IssuerDN>$<SerNum>
cvStatus	String	Status of request.
cvValidityPeriod	Numeric String	Validity period of certificates. max. 65535 days (~ 180 years)
cvHashAlgorithm	Case Exact String	Hash algorithm used in user certificates.
cvCAAlgorithm	Case Exact String	RSA or ECC.

Attribute Name	Type	Description
cvCAAlgorithmParameter	Case Exact String	Name of the elliptic curve.
cvCACertificate	Octed String	The CA Certificate set as default
cvCAHashAlgorithm	Case Exact String	Hash algorithm used in CA certificates.
crossCertificatePair	Octet String	Contains the cross certificate pair.

3.3 Object Class cvCALink

The object class cvCALink is an effective class.

Superclas: Top

Naming attributes: CN

Mandatory attributes: ObjectClass, CN, cvCADN, cvRepositoryListReference, cvActiveCA

Attribute Name	Type	Description
CN	String	Common Name of entry.
cvActiveCA	State	True if CA is activated. Only one CA is allowed to be active.
cvCADN	String	Distinguished Name of the CA.
cvCAReference	DN	Link to the CA object.
cvRepositoryListReference	DN	Link to the repository of the CA.

3.4 Object Class cvIssuedCertificate

The object class cvIssuedCertificate is an effective class.

Superclass: Top

Naming attributes: cvSerialNumber

Mandatory attributes: ObjectClass, cvSerialNumber

Attribute Name	Type	Description
cvCAListReference	DN	Link to CA list of the CA generating users's certificate.
cvCAReference	String	Reference to CA generating user's certificate by CA's distinguished name.
cvCRLEntryPresent	State	True, if this certificate is revoked.
cvCRLReason	Integer	Reason for revocation as stated in RFC3280.
cvCertificateType	String	Type of certificate.
cvHashAlgorithm	Case Exact String	Hash algorithm used in user certificates.
cvInvalidityDate	Generalized Time	Starting time when certificate was revoked.
cvIsExternalSubjectName	State	True, if the subject name has not been ta from the path in the eDirectory.
cvIssuerName	String	Name of the issuer.
cvLastStatus	Case Exact String	Status of Last operation.
cvOwnerReference	String	Reference to owner of this object by owners distinguished name.
cvRfc822mailbox	String	Internet mail address.
cvSerialNumber	Numeric String	Serial number of the certificate.

Attribute Name	Type	Description
cvStatus	Case Exact String	Status of current operation.
cvSubjectKeyIdentifier	Octet String	Key identifier.
cvSubjectName	String	Distinguished name of user.
cvUserCertificate	Octet string	User's certificate.
cvValidityNotAfter	Generalized Time	Expiration date of the certificate.
cvValidityNotBefore	Generalized Time	Earliest valid date of the certificate.
cvNotificationCount	Integer	Number of notifications sent to the address mentioned in cvRfc822mailbox.
cvParamSelectedCAList	DN	Distinguished name of the CA list.
cvRepositoryTrigger	String	Contains the CA operation.
cvTriggerParamCRLReason	Integer	Reason for revocation as stated in RFC3280.
cvSelectedCAList	String	Reference to the mandator whose active CA is triggered to handle a request.

3.5 Auxiliary Class cvUserAttribAux

The Auxiliary Class cvUserAttribAux is designed to extend User objects

Naming attributes: -

Mandatory attributes: -

Attribute Name	Type	Description
cvAllowedCertificateType	Case Exact String	Certificate types a user is allowed to request..
cvCRLReason	Integer	Reason code of revocation.
cvCertificateType	String	To define the Certificate Type.
cvClientData	Octet String	Certificate request (PKCS#10 or Netscape format).
cvClientDataType	String	Type of data from client.
cvCurrentSCEPRequestID	Case Ignore String	SCEP request ID.
cvDestDN	Case Exact String	Used by cv act workstation/cic.
cvExternalSubjectName	String	Subject name in the certificate (for server certificates, only for certificate types where cvNameOverwriteAllowed is set).
cvGenerateType	Case Exact String	All certificate types for which a key generation is permitted.
cvHashAlgorithm	Case Exact String	Hash algorithm used in user certificates.
cvKeyLength	Numeric String	Key length of key to be generated .
cvKeyRecoveryType	Case Exact String	Certificate types for which CA creates a recovery key.
cvLastIssuedCertificate	Octet String	Last issued certificates, deleted by client.

Attribute Name	Type	Description
cvLastStatus	Case Exact String	Status of last request.
cvMultipleCertificatesAllowed	String, Multi Valued	Certificate types a user is allowed to request more than once.
cvPKCS12	Octet String	Holds temporarily the generated encrypted private key pair.
cvRepositoryReference	Distinguished Name, Multi Valued	Reference for LDAP request, if valid certificate exist.
cvRequestType	Case Exact String	All certificate types for which a request is permitted.
cvRoamerType	Case Exact String	All certificate types that can be accessed by pki/roamer.
cvSelectedCertificateID	String	Identifier of the certificate for update and revocation requests
cvStatus	String	Status of request, deleted by client.
cvUserCertificate	Octet String	User certificate.
cvUniversalPrincipleName	Case Ignore String	Universal Principle Name, necessary for Windows login.
cvUserTrigger	String	Contains the CA operation.
cvValidityPeriod	Numeric String	Validity period of certificates.
userCertificate	Octet String	Certificate of the user or workstation.
cvSelectedCAList	String	Reference to the mandator whose active CA is triggered to handle a request.
cvLastIssuedPrivateKey	Octet String	Will be provided by driver if the option ‘Deploy plain private key’ is activated and the attribute is not blocked by the filter (default setting).

3.6 Object Class cvSCEPRequest

The object class cvSCSPRequest is an effective class.

Superclass: Top

Naming attributes: cvSCEPRequestID

Mandatory attributes: ObjectClass, cvSCEPRequestID

Attribute Name	Type	Description
cvClientData	Octet String	PKCS#10 request of the router.
cvExternalSubjectName	Case Exact String	Name of the router.
cvKeyLength	Numeric String	Key length of key to be generated.
cvSCEPRequestID	Case Ignore String	Unique ID, generated by the router.
cvSCEPRequestRejetion-Reason	Case Ignore String	Reason, why request was rejected.
cvUserCertificate	Octet String	User certificate.

3.7 Object Class cvCASet

The Auxiliary Class cvCASet is designed to extend the CA list object

Naming attributes: -

Mandatory attributes: -

Attribute Name	Type	Description
cvAvailableTypes	Case Exact String	All certificate types available for this instance of the CA driver (certificate template types). This value will be set automatically at the start of the driver.
cvMandatorDescription	Case Exact String	Description of the mandator.

4 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.

5 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 his identity. Normally a user enters his user-name, which might be known publicly, and then he identifies himself by his password, which should be only known to himself. 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, his 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.