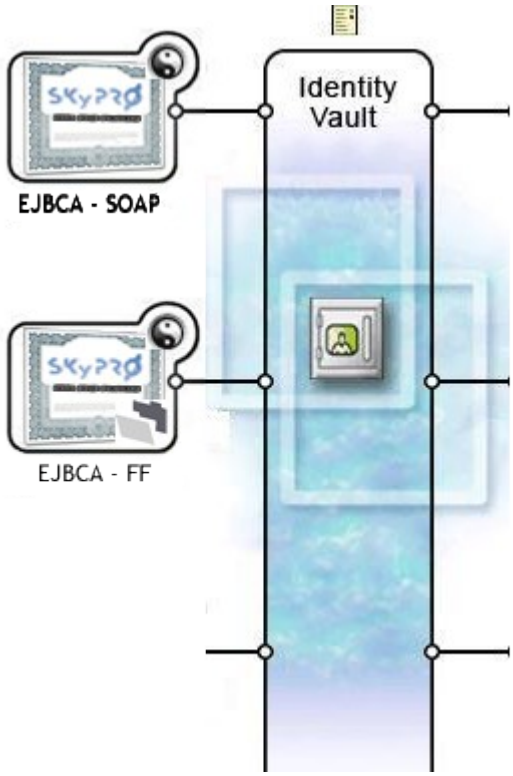
SKySec Certificate Driver

for Novell Identity Manager

Installation and Configuration Manual



Version:	1.0
last updated:	05/27/2008
issue date:	10/24/2007
filename:	SKySec Installation & Configuration v1.0

SKYPRO AG, Gewerbestrasse 7, CH-6330 Cham, Tel. 041 741 47 70, Fax 041 741 50 71, www.skypro.ch

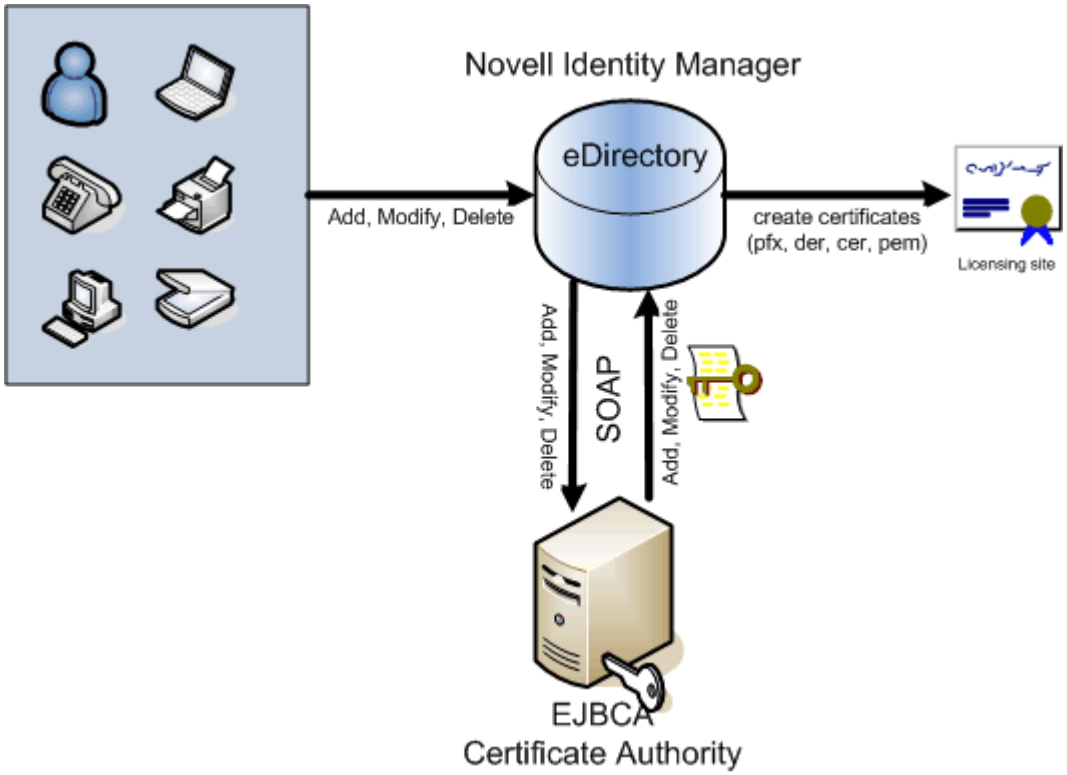
Table of Contents

1 Abstract

Based on the open source Certificate Authority (CA) EJBCA (ejbca.sourceforce.net), the SKySec driver is able to create certificates for any eDirectory object (e.g. users, workstations, devices like VoIP phones, printers and much more). Based on J2EE technology EJBCA constitutes a robust, high performance and component based CA. Actually EJBCA is an enterprise class PKI you can use to build a complete PKI infrastructure for your organization.

The EJBCA driver for Novell Identity Manager consists of two drivers.

1. a SOAP driver, that communicates with the EJBCA infrastructure
2. a loopback driver, which exports and renews certificates



The SOAP driver synchronizes objects from eDirectory to the EJBCA PKI infrastructure. It creates, modifies and deletes „end entities" in the EJBCA PKI infrastructure. EJBCA itself generates the specified certificates for the entities. The certificates, including public and private key material, are stored in eDirectory by the SOAP driver. Since all eDirectory object classes can by synchronized with EJBCA, you can create certificate for any eDirectory object.

The loopback driver exports the certificate into a PFX, CER, DER or PEM file for further distribution. In case of a PFX file you can define a standard password, which is exported in a separate password file. The loopback driver automatically renews certificates before they expire.

2 Installation

The installation of the EJBCA driver is done in the steps listed below:

1. extract downloaded zip
2. extend the eDirectory schema
3. replace existing java classes
4. copy new java classes
5. generate java keystore file
6. import the driver (either with iManager or Designer)
7. start the drivers

2.1 extract downloaded zip file

Please extract the EJBCA ZIP or RAR file, you have downloaded, in a working directory. The driver package contains the following files:

- this manual (EJBCA Driver Installation & Configuration)
- MakeKeystore.bat
- Drivers\EJBCA_LB.xml
- Drivers\EJBCA_Soap.xml
- Drivers\ejbcaSchema.sch
- Libraries\bcprov-jdk14-137.jar
- Libraries\SKyPRO-EJBCASoapUtils.jar
- Licenses\SKyPRO-EJBCASoap.jar
- sun\jce_policy-1_4_2.zip
- sun\jce_policy-1_5_0.zip

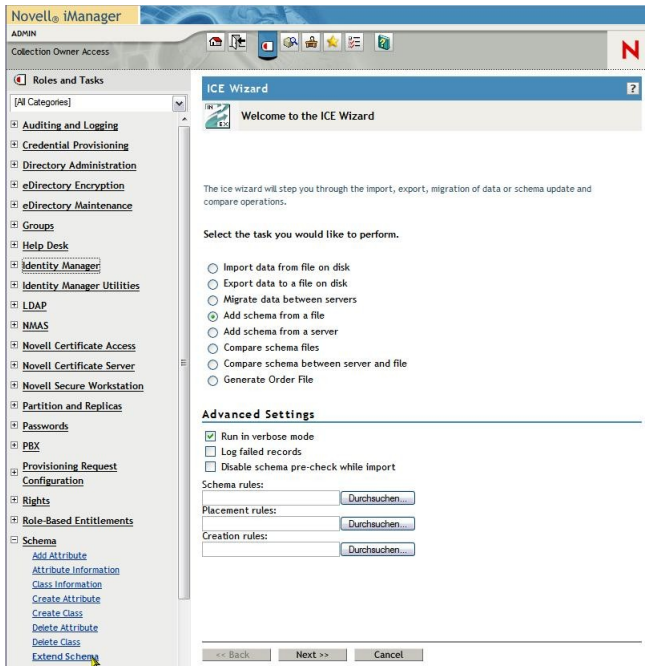
Please verify, that you have received all files before continuing.

2.2 extend eDirectory schema

The EJBCA driver uses an auxiliary class which is linked to any object, which is synchronized with EJBCA. To extend the eDirectory schema you can either use Novell's iManager or Designer for Identity Manager.

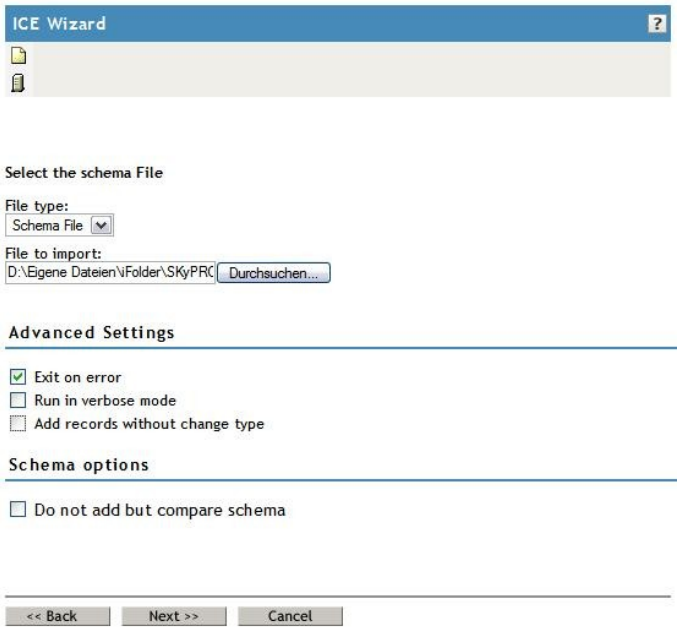
2.2.1 extend eDirectory schema using iManager

iManager uses the ICE wizard to extend the eDirectory schema. Start iManager and expand „Schema" under „Roles and Tasks".



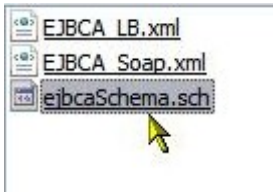
Select „extend schema" and choose „Add schema from a file".

Press „*Next*"



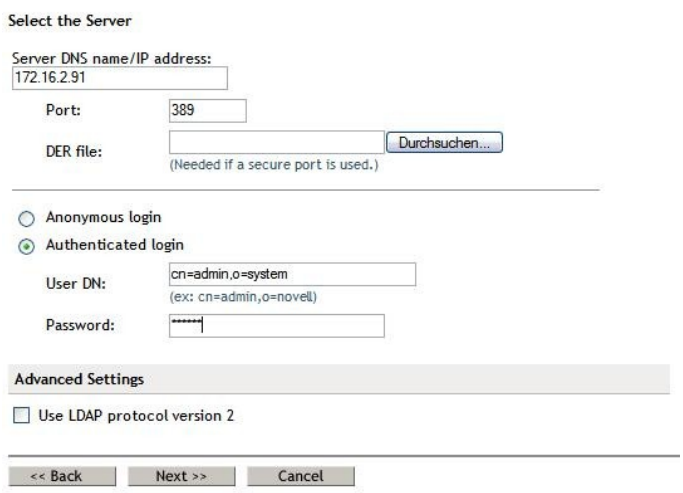
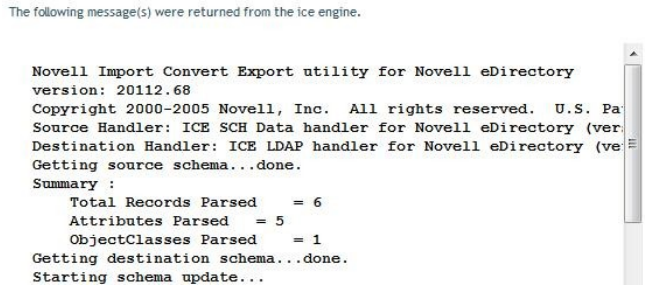
Choose „Schema File" as file type. Click „search" to look for the schema file.

Choose file „ejbcaSchema.sch" from the driver pack.



Press „*Open*".

Press „*Next*"

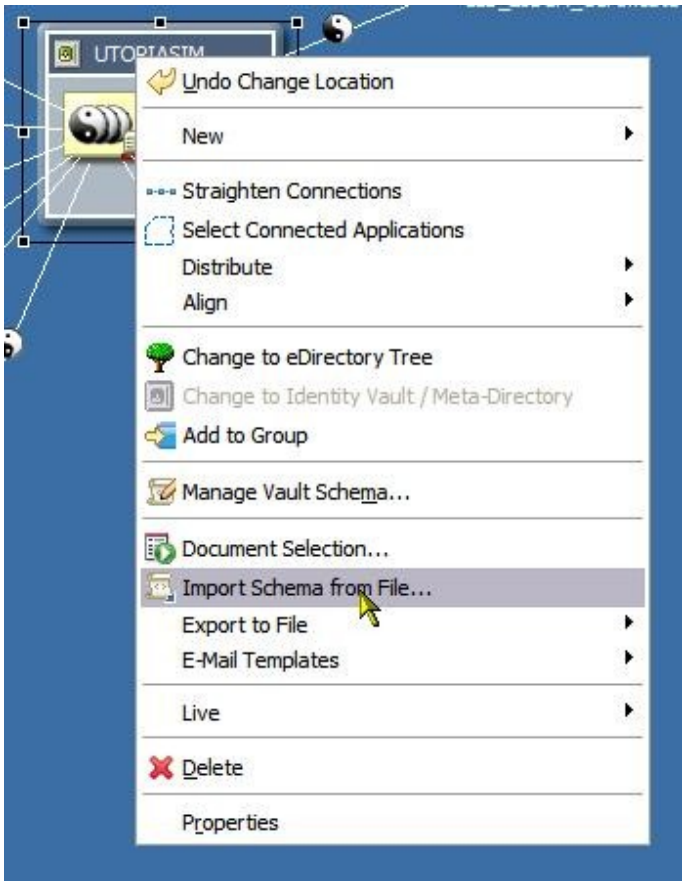
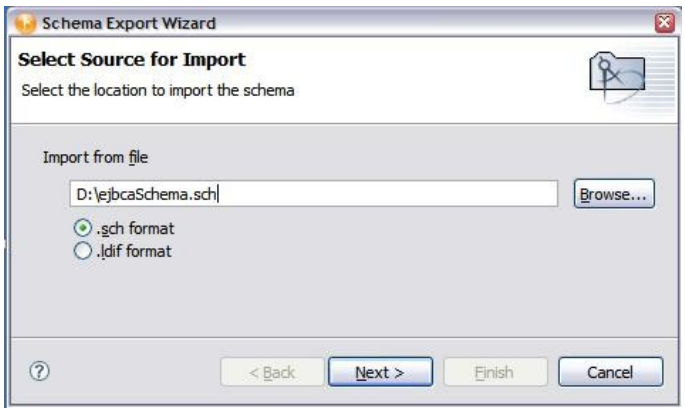
	Select DNS name or IP address of server, which is going to process the import. Provide port 389 for standard LDAP operation. Provide userID and password of the admin account or any other user, who has sufficient rights to expand the eDirectory schema.
	The ICE will process your request and expand the schema accordingly.

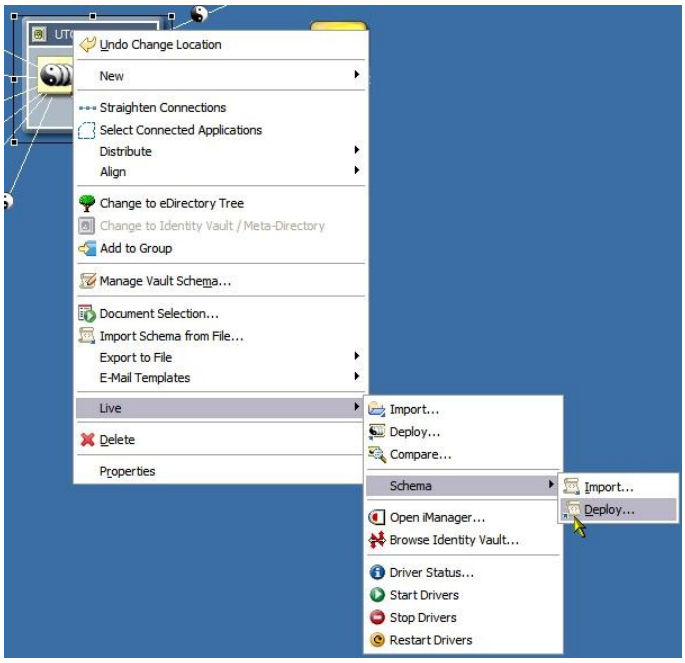
2.2.2 extend eDirectory schema user Designer

Instead of using the iManager to expand the eDirectory schema you can use Novell's Designer for Identity Manager.

Before you start expanding the schema in Designer, make sure you have the current schema of your eDirectory imported into your designer project. To import the schema into your designer project select *live/schema/import* on your identity vault.

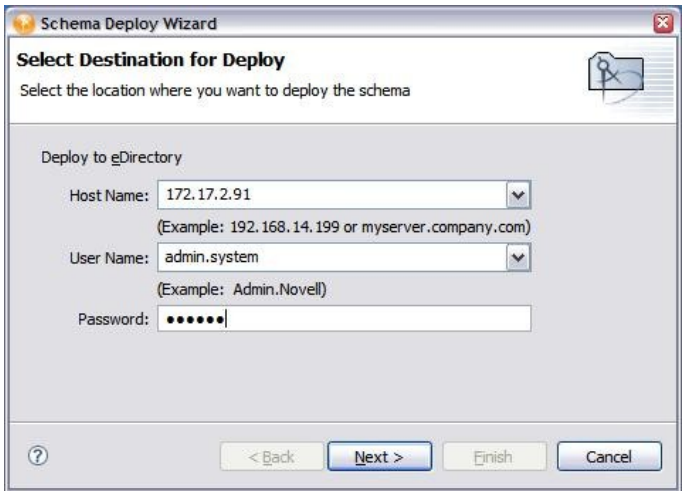
Please start Designer and right click on the identity vault you're going to extend.

	Select „*Import schema from File..*" from the context menu.
	Select the schema file *ejbcaSchema.sch* provided with the driver pack. You'll find the schema file in the subdirectory */drivers* in the directory, where you unpacked the downloaded file. Press *Next* and *Finish*.



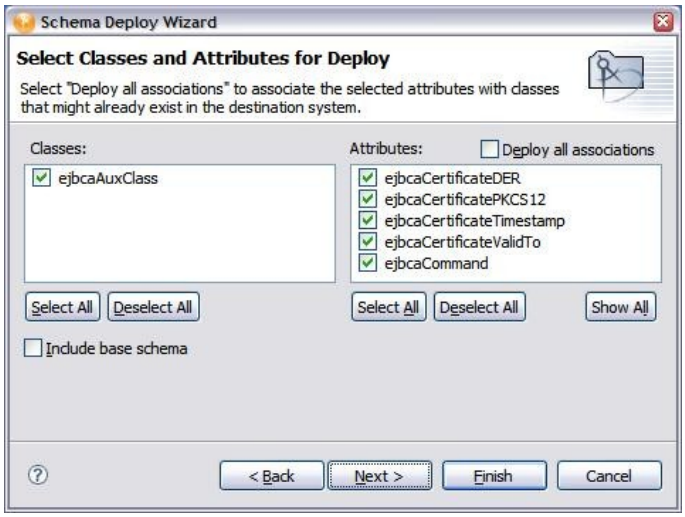
After you have extended the schema in Designer you have to deploy the schema to your identity vault.

Right click on your identity vault and select *„Live/Schema/Deploy..“*



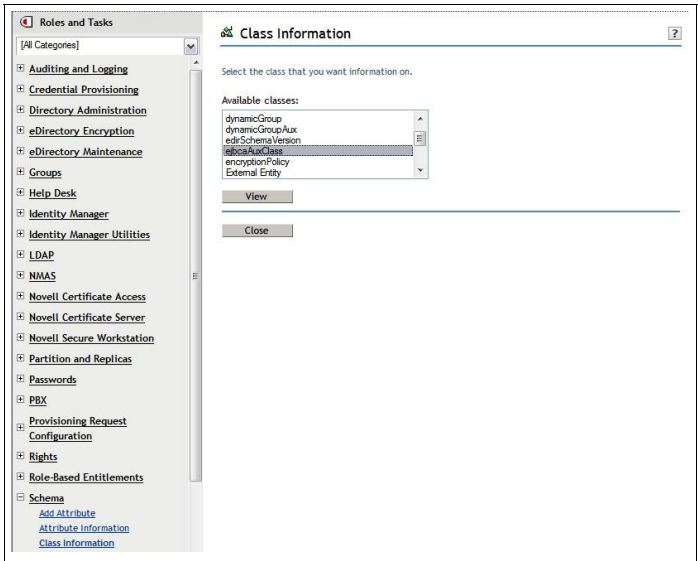
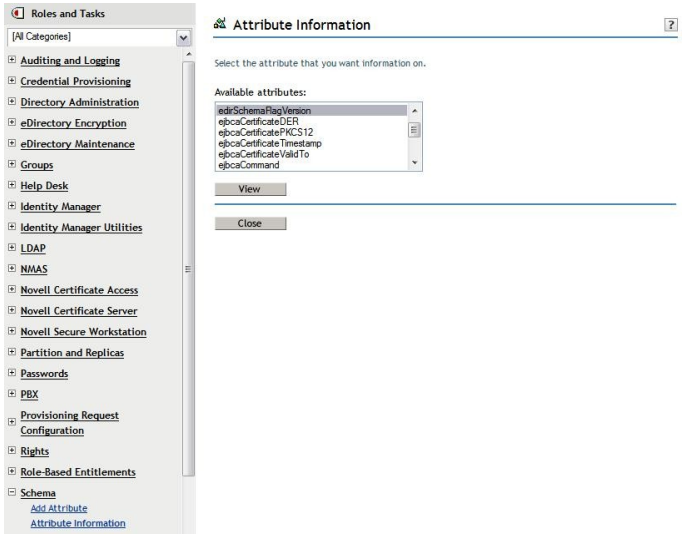
Provide the name or IP address or your host holding the eDirectory identity vault.

You also have to provide the user name and password for your admin account or any other user, who has sufficient rights to extend the eDirectory schema.

	Only select „ejbcaAuxClass" and its appropriate attributes to deploy to your identity vault. Press „*Next*"
	View summary and press „*Finish*".

2.2.3 Check schema extension

Before you continue installing the EJBCA driver, check the schema extension you've made. Logout of iManager and login again. From „*Roles and Tasks*" expand „*Schema*" and select „*Class Information*".

	Search for „*ejbcaAuxClass*“ and select „*View*“
	Make sure you see all attributes as listed aside: *ejbcaCertificateDER* *ejbcaCommand* *ejbcaCertificateTimestamp* *ejbcaCertificatePKCS12* *ejbcaCertificateValidTo*

2.3 replace existing java classes

The EJBCA driver needs to replace a couple of existing java classes on the server, where identity manager is installed.

1. bcprov-jdk14-137.jar (may not be installed already)
2. local_policy.jar and US_export_policy.jar (from java cryptograhpy extension jce)

2.3.1 bcprov class

This java class is needed for higher encryption than 48-bit. Please locate **all** instances of the bcprov-jdk14-137.jar file and replace them with the version provided with the driver. The correct version of the file is:

name	**date**	**size**
bcprov-jdk14-137.jar	August 20th 2007	1,437kB

The file is located in the subdirectory *libraries* in the extracted driver packages.

If you do not find any instance of this file, copy it into the subdirectory, where all Identity Manager java classes are installed. By default this is one of the following directories:

- Windows: \novell\RemoteLoader\lib
- Linux: /opt/novell/eDirectory/lib/dirxml/classes
- NetWare: sys:\system\lib

2.3.2 java cryptography extension (jce)

To use more then 7 characters in a certificate password you need to download and install the java cryptography extension (jce). For your convenience we have provided the jce for java 1.4.2 und java 1.5 (found in subdirectory *sun*). To see which version of java you have installed, you can use the command *java -version*. If you have installed other java version,please go to www.sun.com/download to look for the appropriate version.

If you unzip the jce policy file, you will find two java classes:

US_export_policy_jar
local_policy.jar

Please locate all instances of these files on your server and replace them with the correct versions from your jce download. Please restart java.

2.4 install new java classes

The EJBCA driver needs two additional java classes:

1. SKyPRO-EJBCASoapUtils.jar (found in subdirectory *Libraries*)
2. SKyPRO-EJBCASoap.jar (found in subdirectory *Licenses*)

These java classes have to be copied to the directory where you have installed your Identity Manager SOAP driver. Look for the files SOAPshim.jar and SOAPUtil.jar. These files are installed by default in the following directories:

- Windows: \novell\RemoteLoader\lib
- Linux: /opt/novell/eDirectory/lib/dirxml/classes
- NetWare: sys:\system\lib

Copy both files, SKyPRO-EJBCASoapUtil.jar and SKyPRO-EJBCASoap.jar, into this directory. Please restart java to take effect of the new installed java classes.

2.5 Generate Java keystore

The SOAP connector needs a certificate to connect to the EJBCA SOAP server. This certificate has to be importet into a java keystore file. This keystore file is used by the EJBCA driver to read the certificate and to authenticate to the EJBCA soap server. The keystore file is create in three steps:

1. create end entity
2. export certificate

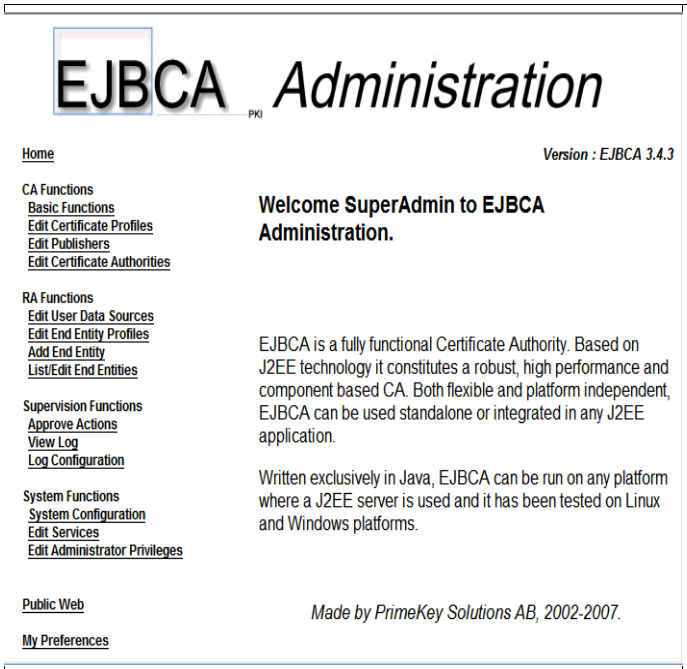
3. create java keystore

2.5.1 create end entity

Contact your EJBCA administrator to provide you with an end entity within EJBCA, that has the appropriate rights to create entities and generate certificates. In this documentation we assume, that you have an end entity within EJBCA with these rights. Otherwise please refer to the EJBCA manuals.

2.5.2 export certificate

In our example we export the certificate of the end entity *superadmin*, who has all needed privileges.

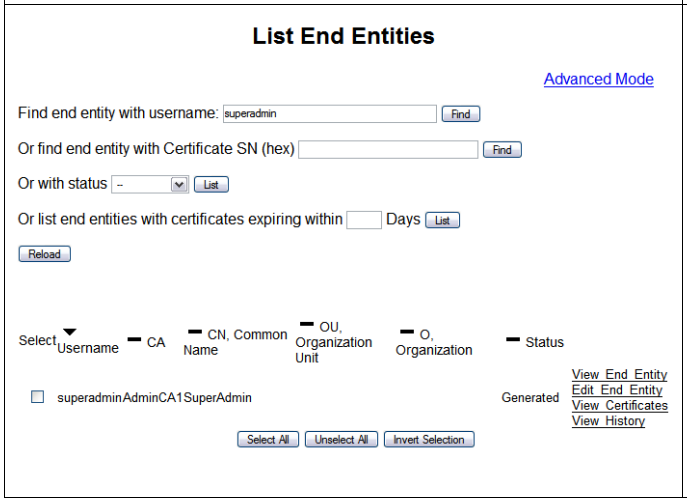


Start your web browser an go to the EJBCA administrative site. As you call the administrative web site of EJBCA you have to have installed a certificate in your web browser, that allows you to administer the EJBCA infrastructure. Please refer to the EJBCA manual. The EJBCA Administration site will open up.

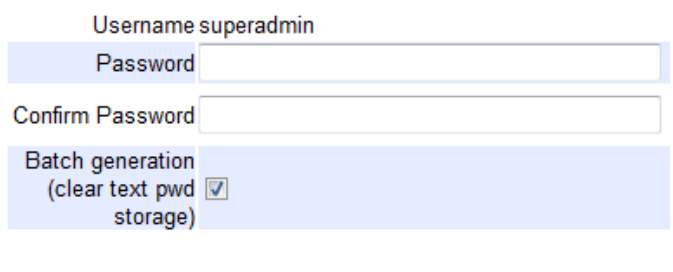
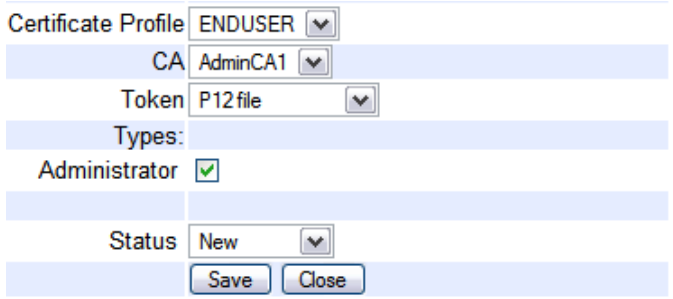
By default the URL of the admin web page of EJBCA is:

https://<server>:8443/ejbca/adminweb/

You have to use a DNS name. EJBCA does not work with ip addresse



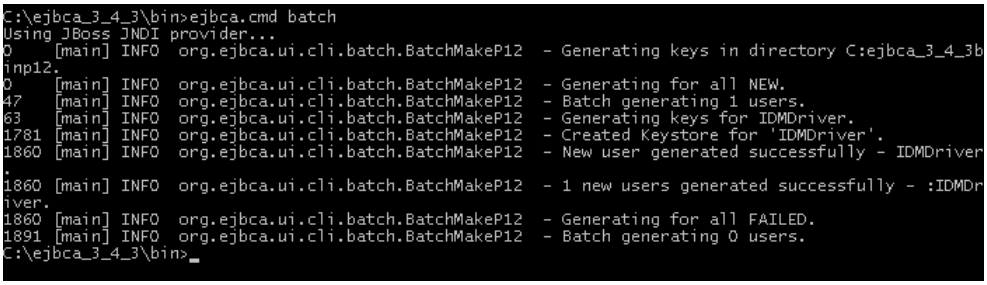
Select *List/Edit End Entity* in the left menu. In our example the end entity *superadmin* has the appropriate rights. List the user and select *Edit End Entity.*

	Set and confirm the password and activate the check box *batch generation.*
	At the end of the form select *„P12 file"* as Token and change the status to *new.* Save the form.

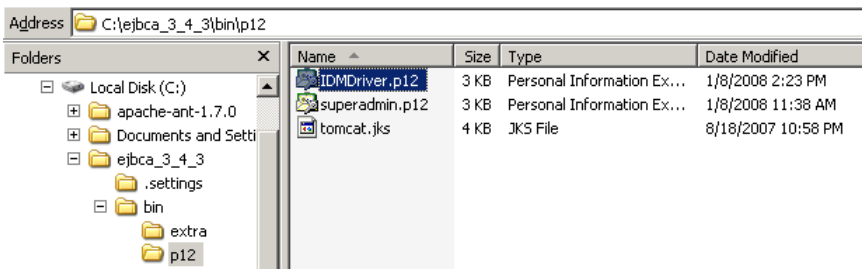
As you have edited the end entity, you have selected to batch generate the P12 certificate file. Now we're ready to export this certificate. (In this manual we assume you have installed EJBCA on a windows based server. Otherwise please refer to the EJBCA manual for instruction how to export the certificate.)

Go to the server console and open a command console. In the command console change to the EJBCA installation directory. Default installation directory is *c:\ejbca_3_4_3.* Change to the subdirectory *bin*. Enter the batch command to export certificates of new generated end entity:

ejbca.cmd batch



This script will export the superadmin certificate to the subdirectory *p12.*



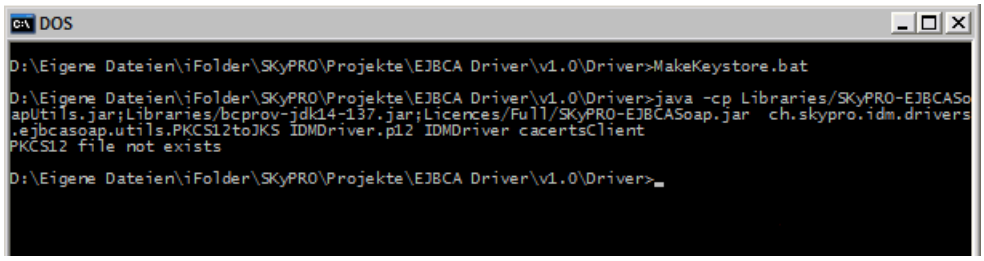
2.5.3 create java keystore

Copy the certificate you created to the directory, where you have extracted the EJBCA driver files. For your convenience we have provided a script *MakeKeystore.bat* to generate the java keystore file *cacertsClient the driver needs to authenticate to the soap server*.

Open the script *MakeKeystore.bat* and correct the name and the password (*superadmin.p12* and *ejbca*) with the certificate you have create.

```
java -cp Libraries/SKyPRO-EJBCASoapUtils.jar;Libraries/bcprov-jdk14-137.jar;Licences/SKyPRO-EJBCASoap.jar ch.skypro.idm.drivers.ejbcasoap.utils.PKCS12toJKS superadmin.p12 password cacertsClient
```

Start the script:



Copy the *cacertsClient* keystore file to the directory where you have copied the SKyPRO EJBCA java classes in chapter 2.4.

2.6 configuration parameters

Before we start to import the drivers, we have a look at the configuration parameters. Both drivers have some important parameters, which you must understand to define them correctly.

2.6.1 SOAP driver parameters

As explained above the SOAP driver creates an EJBCA end entity. Since we do not want to create a certificate for all the objects in our eDirectory, we have to filter out the objects we want to pass to the EJBCA. At the moment the end entity is created, the PKI generates the certificate. To create the correct certificate, we pass some parameters over to the EJBCA, so it knows what type of certificate it has to create. As soon as the certificate is created, it is passed back to the driver. The driver stores the certificate in eDirectory, so the loopback driver can export it.

URL of remote DSML Server
The communication between the IDM server and the soap server will always be SSL secured. So you always have to use „*https*". EJBCA compares the SSL server certificate that will be used for the encryption with the name of the URL you call. If the URL you call does not match with the server name in the SSL certificate, EJBCA will refuse the connection.

You can never use an ip address as the URL!

In the log file of the soap driver you will find the error message „ *No trusted certificate found*". A valid URL could be:

https://www.ejbcatest.local:8443/ejbca/adminweb/

You can check the URL of the server certificate with your browser when you are connected to the EJBCA Administration site. Double click on the closed security lock in internet explorer or firefox and display the certificate information. There you will find the common name (cn) of the server the certificate is issued for.

keystore file
EJBCA does not use userID and password to authenticate to the web server but it request a valid certificate. The certificate is stored and passed in the keystore file, which we have created and copied to the driver directory. This parameter contains the path and filename of the keystore file.

Without a valid keystore file the driver can not establish a connection with the EJBCA web service. In the driver log file you'll find an error message „*AuthorizationDeniedException*".

If you use our *MakeKestore.bat* command file to create the keystore file the password for the keystore file will be „*changeit*".

Path to object container
The driver only processes objects residing within this container and its subcontainers. Please use the browse button to select the right context to be sure to use the correct syntax. Be aware: the syntax between importing the driver and modifying the value in iManager is different!

object class
While importing the driver with designer or iManager the import wizard will ask you for an object class to create certificates for. Certificates will only be created for this object class. If you want to create certificates for different object classes, you have to install a driver for each object class.

EJBCA CA Name
This parameter contains the name of your Certificate Authority (CA) issuing the certificates. You'll find the name your CA in the EJBCA Administration portal. See „EJBCA Administration".

EJBCA Certificate Profile Name
The driver needs to pass a Certificate profile name to EJBCA. EJBCA uses this profile to generate the certificates. The profile defines the key usage, available key length and much more. Be sure to use an existing Certificate profile name. Otherwise EJBCA can not generate a certificate. See „EJBCA Administration" to look for valid Certificate Profile names.

EJBCA End Entity Profile Name
An End Entity Profile works as a template for the end entity to create. EJBCA uses this profile to create the end entity. Be sure to use an existing End Entity profile name. Otherwise the EJBCA will not create the end entity. See „EJBCA Administration" to look for valid End Entity Profile names.

End Entity Password Prefix
EJBCA requires a password for each end entity. This password is only needed to pass to EJBCA to create the end entity. Enter any password that has at least 6 characters, upper and lower letters and one special character (e.g. EjbCA.).

Certificate Key Length
Defines the key length of the certificate. The available key lengths depend on the Certificate Profile that will be used.

Certificate key algorithm
The key algorithm will always be RSA. Please do not change this field.

2.6.2 Loopback driver parameters

The loopback driver exports the certificates to PFX, DER or CER files.

Path to object container
The driver only processes objects residing within this container and its subcontainers. Please use the browse button to select the right context to be sure to use the correct syntax. The syntax between importing the driver and modifying the value in iManager is different!

object class
While importing the driver with designer or iManager the import wizard will ask you for an object class. Certificates will only be created for this object class. If you want to create certificates for different object classes you have to install a driver for each object class.

default password
If you export a certificate to a PFX file you can protect the file with a password. You can provide any password. If you define a password, the driver will create a password file containing the password. The name of the password file corresponds to the name of the object with the extension .PWD. This is very convenient if you plan to deploy and install the certificates with a resource management system. So the resource management system is able to install the certificate automatically.

automatic renewal process
You can automatically renew a certificate before it expires. This parameters defines how many days before the expiration date is reached the certificate will be renewed.

certificate file formats (PFX, DER, CER)
The driver can export the certificate in different formats. Please select the file formats (but at least one) you want the certificate be exported to. The PFX file format corresponds to the PKCS12 certificate format. DER is a binary coded X.509 certificate. CER is a base64 coded X.509 certificate.

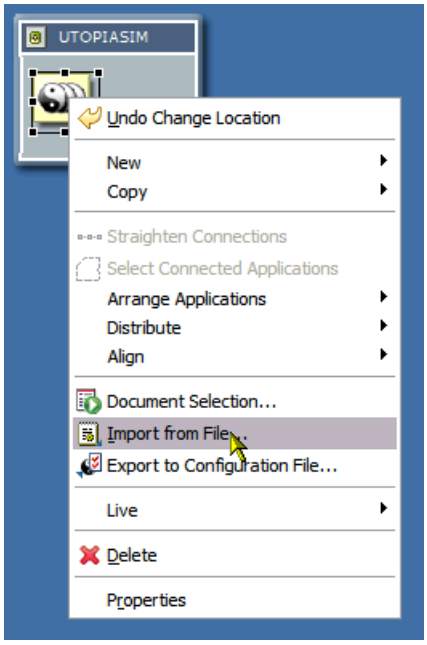
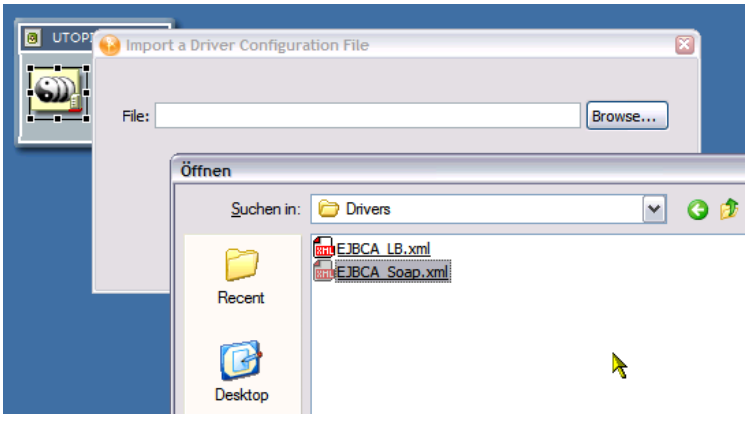
2.7 import and configure driver

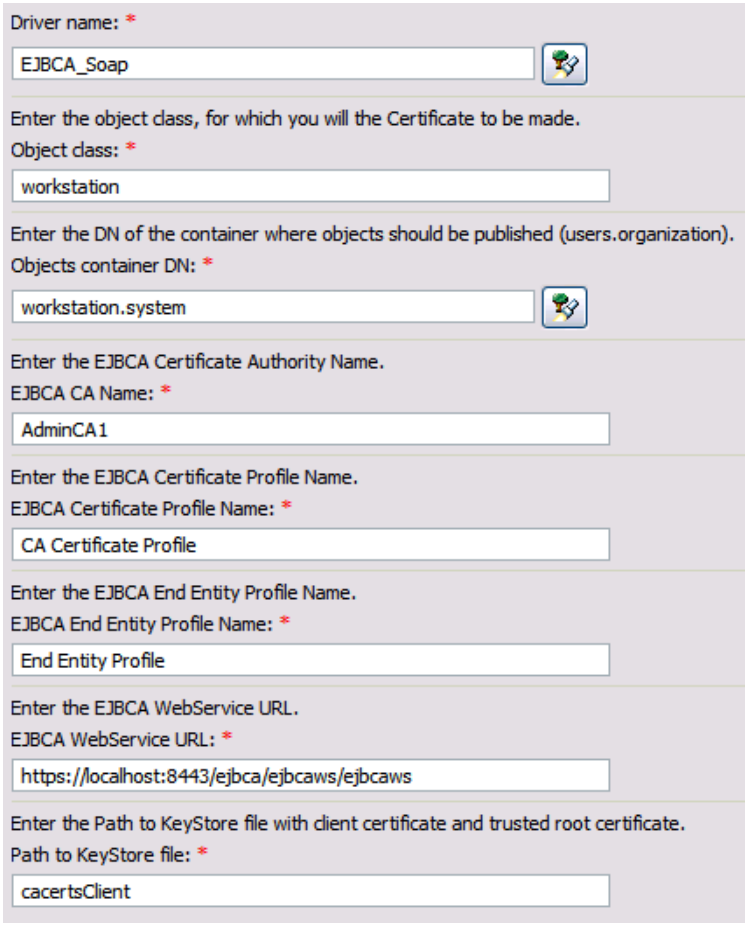
Now we're ready to import the driver into your Identity Manager environment. You can do this either with iManager or Designer.

2.7.1 Designer

Start Designer and open the project where you want to import the driver.

Import SOAP Driver

	Click right mouse bottom on your driver set and choose *Import from File...*
	Select *Browse* and go to the directory where you extracted the EJBCA driver. Choose the *EJBCA_Soap.xml* file you find in the subdirectory *drivers*. Press *Ok.*
	The driver import wizard appears. **driver name:** define any driver name. e.g. *EJBCA-SOAP* **object class:**



select the object class, for which the driver should create certificates.
e.g. *workstation*

Container DN:
define the container containing the objects for which the certificates should be created.
e.g. *workstation.system*

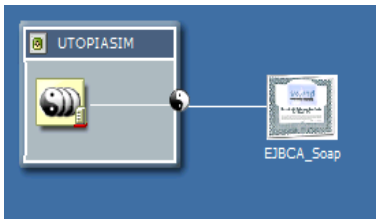
CA Name:
enter the name of your CA. You find the name of the CA under *basic functions* in you EJBCA Administration portal.

Certificate Profile Name:
define the EJBCA certificate profile to be used by the driver. You can define certificate profiles in the EJBCA Administration portal under *Edit Certificate Profiles.*

End Entity Profile Name:
define the EJBCA end entity profile to be used by the driver. You can define end entity profiles in the EJBCA Administration portal under *Edit End Entity profiles.*

EJBCA WebService URL:
enter the URL of the EJBCA SOAP Service. You can test the URL in your web Browser. e.g. *https://<yourEJBCAhost>:8443/ejbca/ejbcaws/ejbcasw*

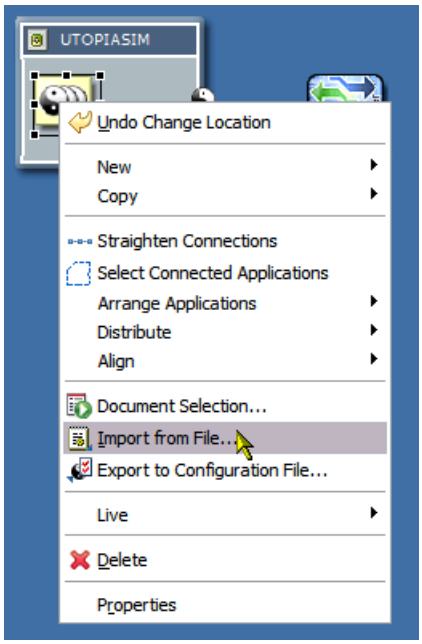
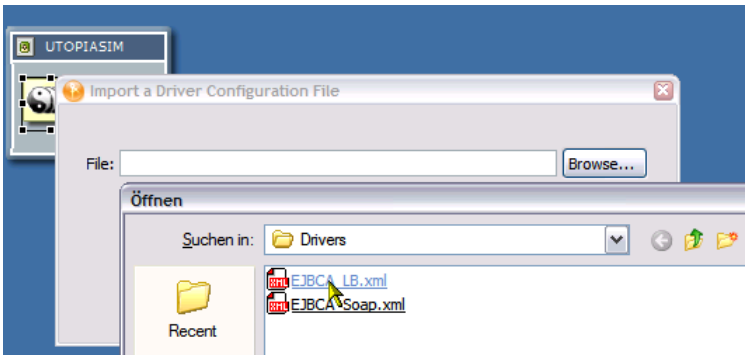
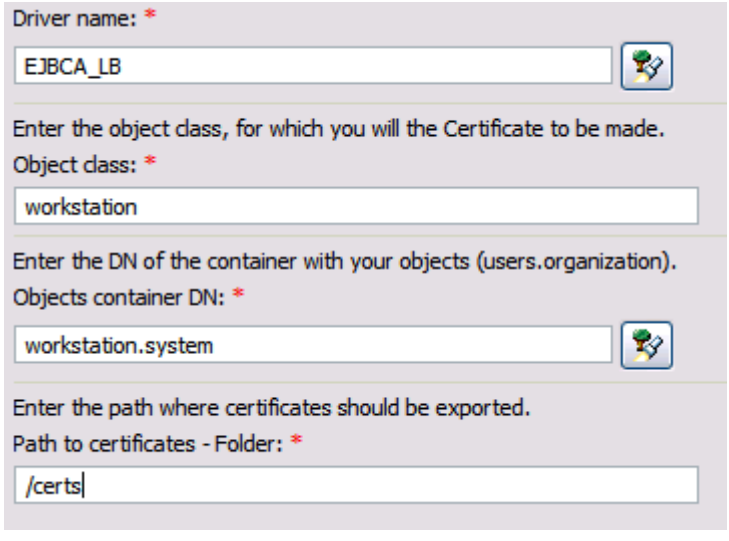
Path to keystore file
enter the path to your keystore file you created. e.g. /opt/novell/eDirectory/lib/dirxml/classes/cacertsClient

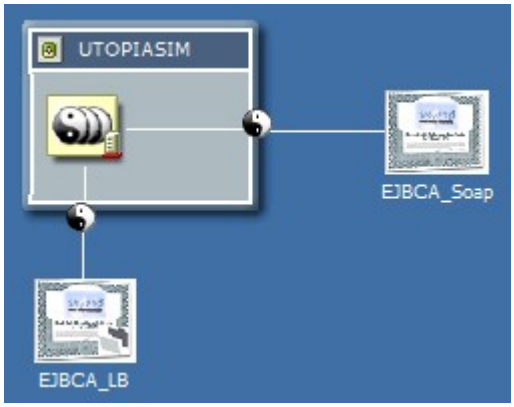
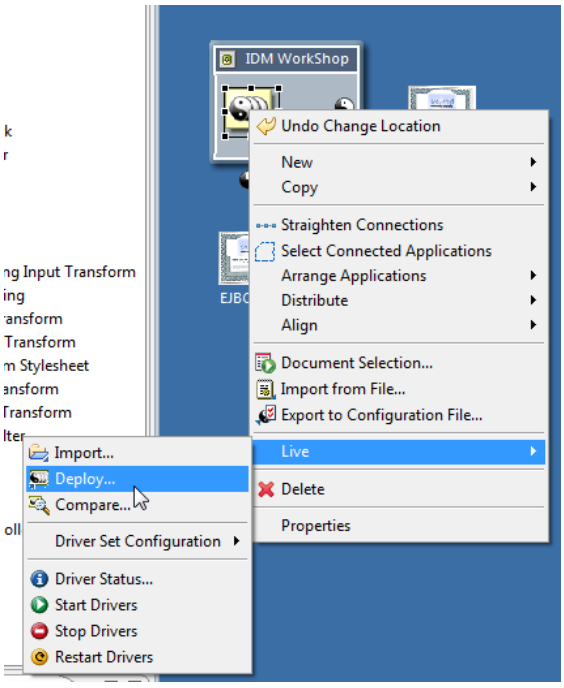


The driver has been created successfully.

We have provided the icons in the *icons* subdirectory. Double click the driver and select *iManager icon* to replace the image.

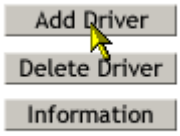
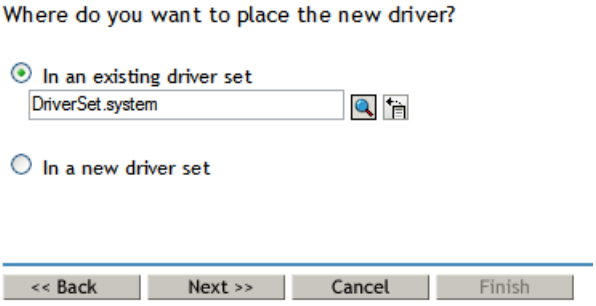
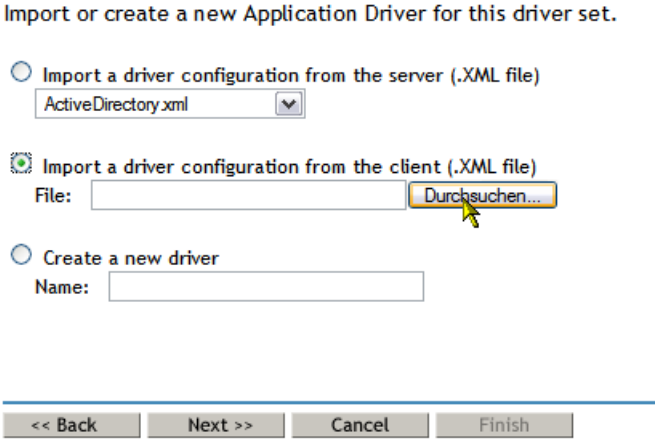
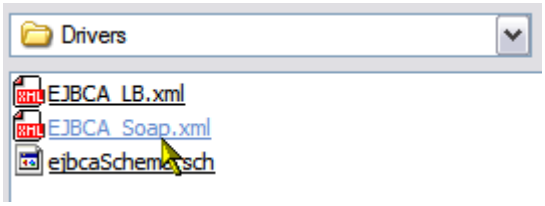
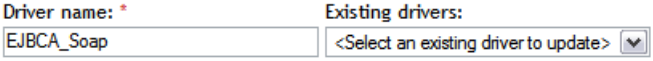
Import Loopback Driver

	Click right mouse bottom on your driver set and choose *Import from File...*
	Select *Browse* and go to the directory where you extracted the EJBCA driver. Choose the *EJBCA_LB.xml* file you find in the subdirectory *drivers*. Press *Ok.*
	The driver import wizard appears **Driver name** Enter any name for the driver **Object class** define the object class, for which the driver should export the certificates **Container DN** enter the name of the container containing your objects **Path to certificates** define the path, where the certificates should be exported to. Naming depends on operating system the driver is installed on.

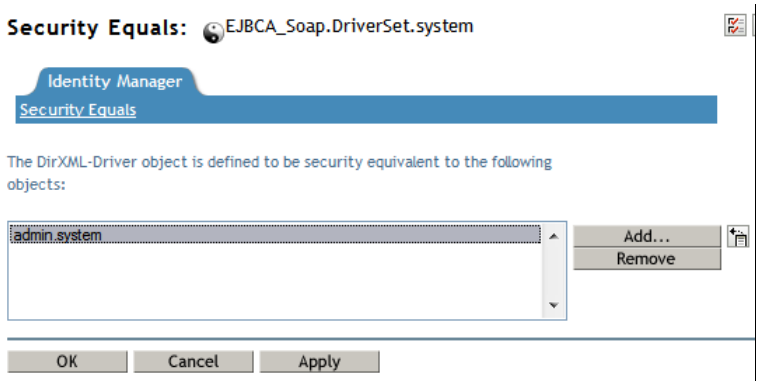
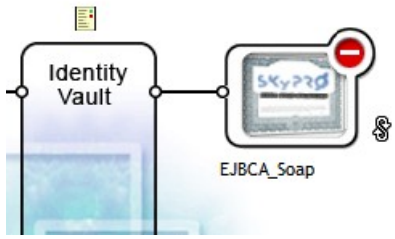
	e.g. /certs (for Linux)
	Confirm the successfully import of the driver.
	We have provided the icons in the *icons* subdirectory. Double click the driver and select *iManager icon* to replace the image.
	To deploy the drivers to your live system right click on your driver set and choose *Live/Deploy...* Select the two EJBCA drivers you have created to deploy.

2.7.2 iManager

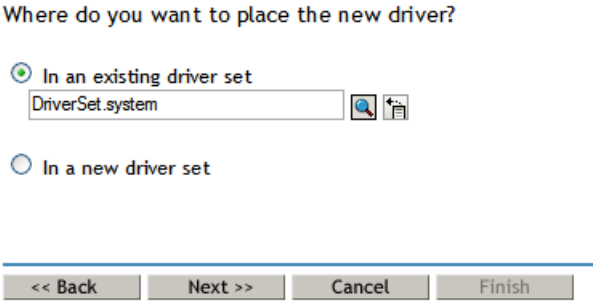
Import SOAP Driver

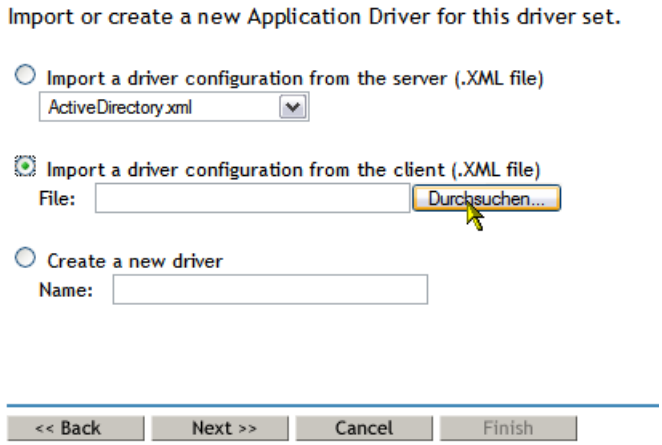
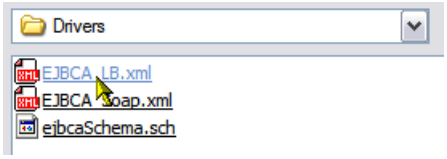
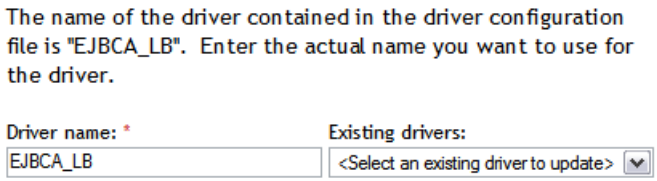
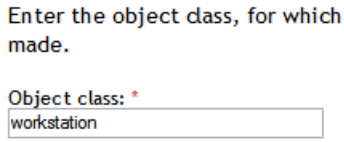
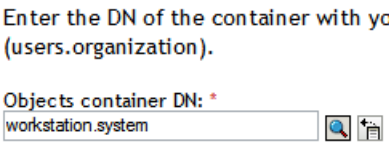
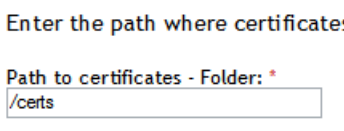
	Start iManager and login with admin rights. Go to the Identity Manager Overview. In the overview press the „*Add Driver*" buttom.
	Decide whether you want to place the new driver in an existing driver set or in a new driver set. Press *Next*
	Select to import a driver from the client as XML file and press the *Browse* bottom.
	Browse to your download directory where you extracted the EJBCA driver. Select the *EJBCA-Soap.XML* file in the subdirectory *Drivers*. Press *Next*
	The driver import wizard appears
	**driver name:** define any driver name. e.g. *EJBCA-SOAP*

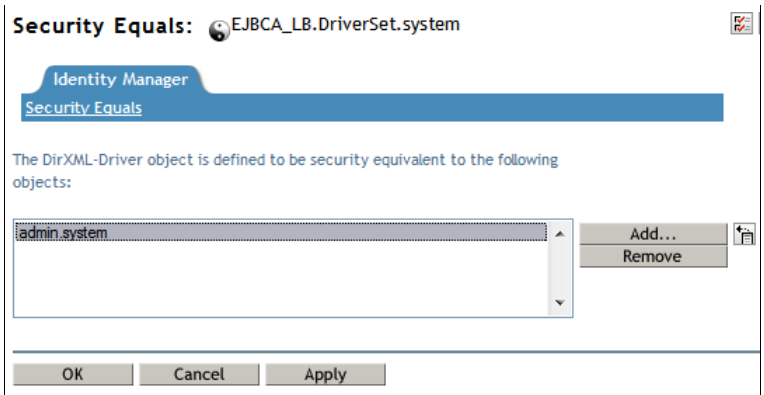
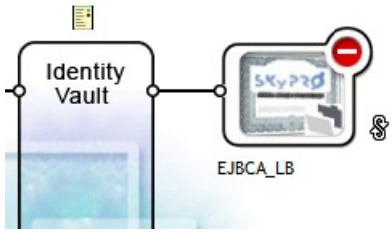
Enter the object class, for which you will the Certificate to be made. Object class: * workstation	**object class:** select the object class, for which the driver should create certificates. e.g. *workstation*
Enter the DN of the container where objects should be published (users.organization). Objects container DN: * workstations.system	**Container DN:** define the container containing the objects for which the certificates should be created. e.g. *workstation.system*
Enter the EJBCA Certificate Authority Name. EJBCA CA Name: * AdminCA1	**CA Name:** enter the name of your CA. You find the name of the CA under *basic functions* in you EJBCA Administration portal.
Enter the EJBCA Certificate Profile Name. EJBCA Certificate Profile Name: * CA Certificate Profile	**Certificate Profile Name:** define the EJBCA certificate profile to be used by the driver. You can define certificate profiles in the EJBCA Administration portal under *Edit Certificate Profiles.*
Enter the EJBCA End Entity Profile Name. EJBCA End Entity Profile Name: * End Entity Profile	**End Entity Profile Name:** define the EJBCA end entity profile to be used by the driver. You can define end entity profiles in the EJBCA Administration portal under *Edit End Entity profiles.*
Enter the EJBCA WebService URL. EJBCA WebService URL: * https://localhost:8443/ejbca/ejbcaws/ej	**EJBCA WebService URL:** enter the URL of the EJBCA SOAP Service. You can test the URL in your web Browser. e.g.: *https://<yourEJBCAhost>:8443/ejbca/ejbcaws/ejbcasw*
Enter the Path to KeyStore file with client certificate and trusted root certificate. Path to KeyStore file: * novell/eDirectory/lib/dirxml/cacertsClient	**Path to keystore file** enter the path to your keystore file you created. e.g. */opt/novell/eDirectory/lib/dirxml/classes/cacertsClient* Press *Next*

	Define an admin security equivalent for the driver and list every object, residing in the configured container, you want explicitly to exclude from the driver. Press *Next*
	Confirm the driver summary and press *Finish.*
	Your new driver has successfully been added.

Import Loopback Driver

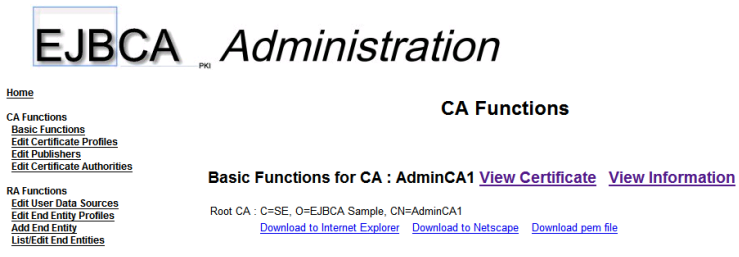
	Start iManager and login with admin rights. Go to the Identity Manager Overview. In the overview press the „*Add Driver*" bottom.
	Decide whether you want to place the new driver in an existing driver set or in a new driver set. Press *Next*

	Select to import a driver from the client as XML file and press the *Browse* bottom.
	Browse to your download directory where you extracted the EJBCA driver. Select the *EJBCA-LB.XML* file in the subdirectory *Drivers*. Press *Next*
	**driver name:** define any driver name. e.g. *EJBCA-LB*
	**Object class** define the object class, for which the driver should export the certificates
	**Container DN** enter the name of the container containing your objects
	**Path to certificates** define the path, where the certificates should be exported to. Naming depends on operating system the driver is installed on. e.g. /certs (for Linux) Press *Next*

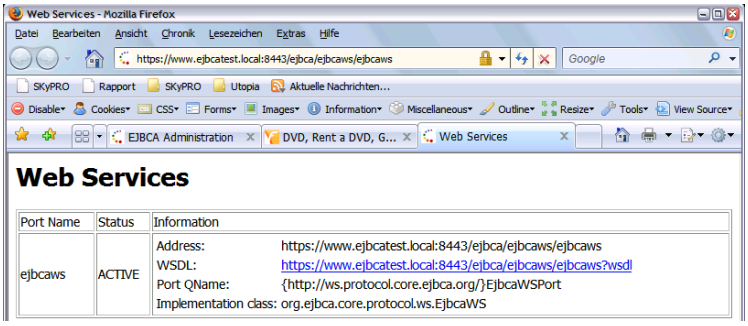
	Define an admin security equivalent for the driver and list every object, residing in the configured container, you want explicitly to exclude from the driver. Press *Next*
	Confirm the driver summary and press *Finish.*
	Your new driver has successfully been added.

2.8 EJBCA Administration Overview

For your convinience we provide you at this place with some information about EJBCA Administration. During the configuration of the driver you have to configure some parameters with information from EJBCA. Here you see where you'll find these informations.

	**CA Name:** On the home page of your EJBCA administration portal you find the name of your CA. In this example *AdminCA1*
	**Add Certificate Profiles** Select *Edit Certificate Profiles* to create and edit EJBCA certificate profiles. Please review EJBCA documentation for more information on how to create certificate profiles.
	Add End Entity Profile

Select *Edit End Entity Profiles* to create and edit EJBCA end entity profiles. Please see EJBCA documentation for further informations about end entity profiles.

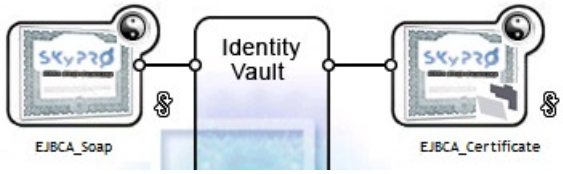
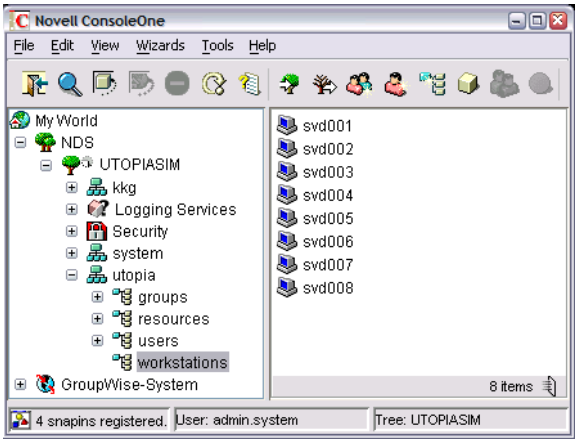
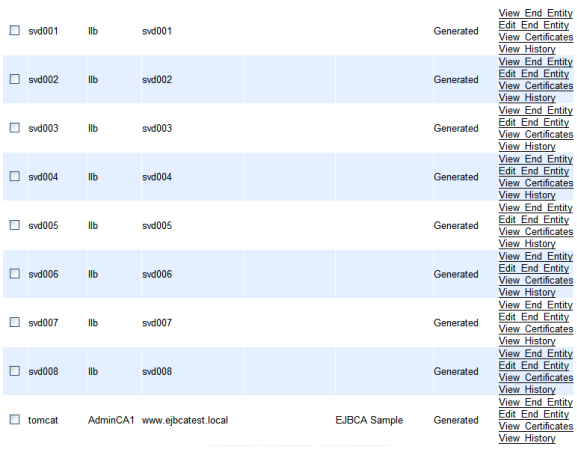


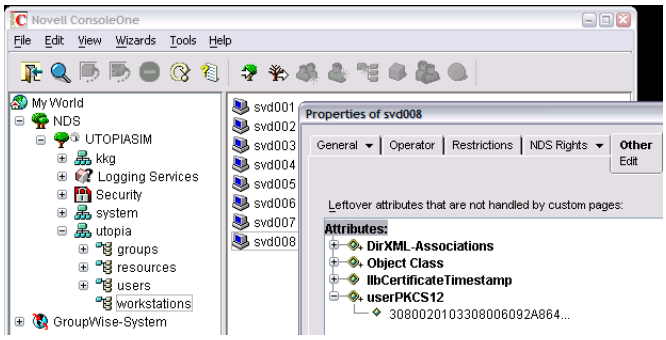
Web Service URL

If the web service is installed correctly you can test it from your browser. e.g.

https://ejbca.local.test:8443/ejbca/ejbcaws/ejbcasw

3 Run and test the drivers

In iManager check if both drivers are up and running.	
We create some workstation objects in eDirectory in the specified container.	
All workstation objects are create as *end entity* in the EJBCA PKI infrastructure. The appropriate certificates are generated.	

In ConsoleOne you see the attribute *userPKCS12*. This attribute holds the certificate including private and public key material.	
All certificates are exported by the loopback driver as PFX file including the password file.