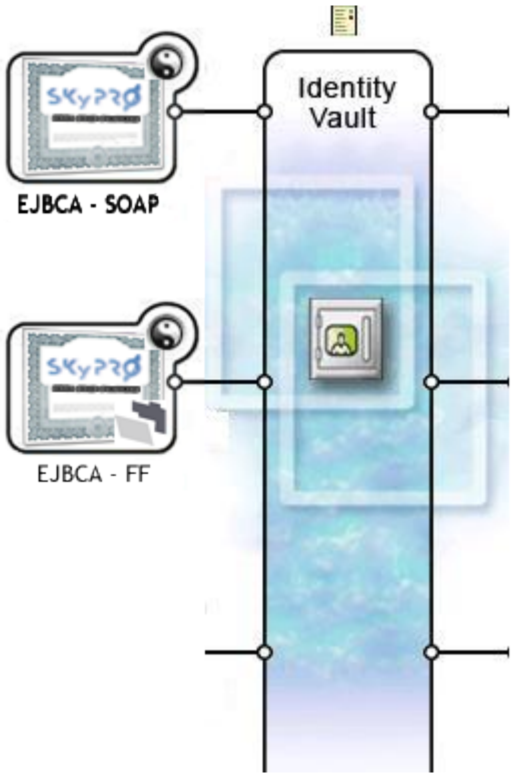
Technical Abstract

PrimeKey EJBCA PKI Driver

for MicroFocus Identity Manager



version:	2.0
author:	Andreas Fuhrmann
created:	23.11.2015
modified:	23.11.15 10:34
printed:	23.11.15 10:49
file:	EJBCA Driver technical description v2.0.docx

Inhaltsverzeichnis

Legal Notice

SKyPRO AG makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, SKyPRO AG reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. Further, SKyPRO AG makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, SKyPRO AG reserves the right to make changes to any and all parts of SKyPRO software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to the Swiss export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current Swiss export exclusion lists or to any embargoed or terrorist countries as specified in the Swiss export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. SKyPRO assumes no responsibility for your failure to obtain any necessary export approvals.

SKyPRO AG has intellectual property rights relating to technology embodied in the product that is described in this document.

SKyPRO AG
Gewerbestrasse 7
CH-6330 Cham

SWITZERLAND

www.skypro.com

About this document

This document summarizes the functions of the product “PrimeKey EJBCA drive” for MicroFocus Identity Manager.

Audience

This guide is intended for consultants and administrators designing and maintaining existing MicroFocus Identity Manager environment. You should have an understanding of drivers, workflows, eDirectory and the IDM Designer tool.

Feedback

We appreciate your feedback about this documentation. If you have any suggestions, comments, feature requests please contact us via

info@skypro.ch.

1 Abstract

Based on the open source Certificate Authority EJBCA (ejbca.sourceforce.net) the EJBCA driver creates certificates for user, workstation or any other object in your eDirectory. Based on J2EE technology EJBCA constitutes a robust, high performance and component based CA. EJBCA is an enterprise class PKI, meaning you can use EJBCA to build a complete PKI infrastructure for your organization.

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

- a SOAP driver, that communicates with the EJBCA infrastructure to create certificates
- a loopback driver, which exports certificates in PFX, CER and DER files and renews certificates

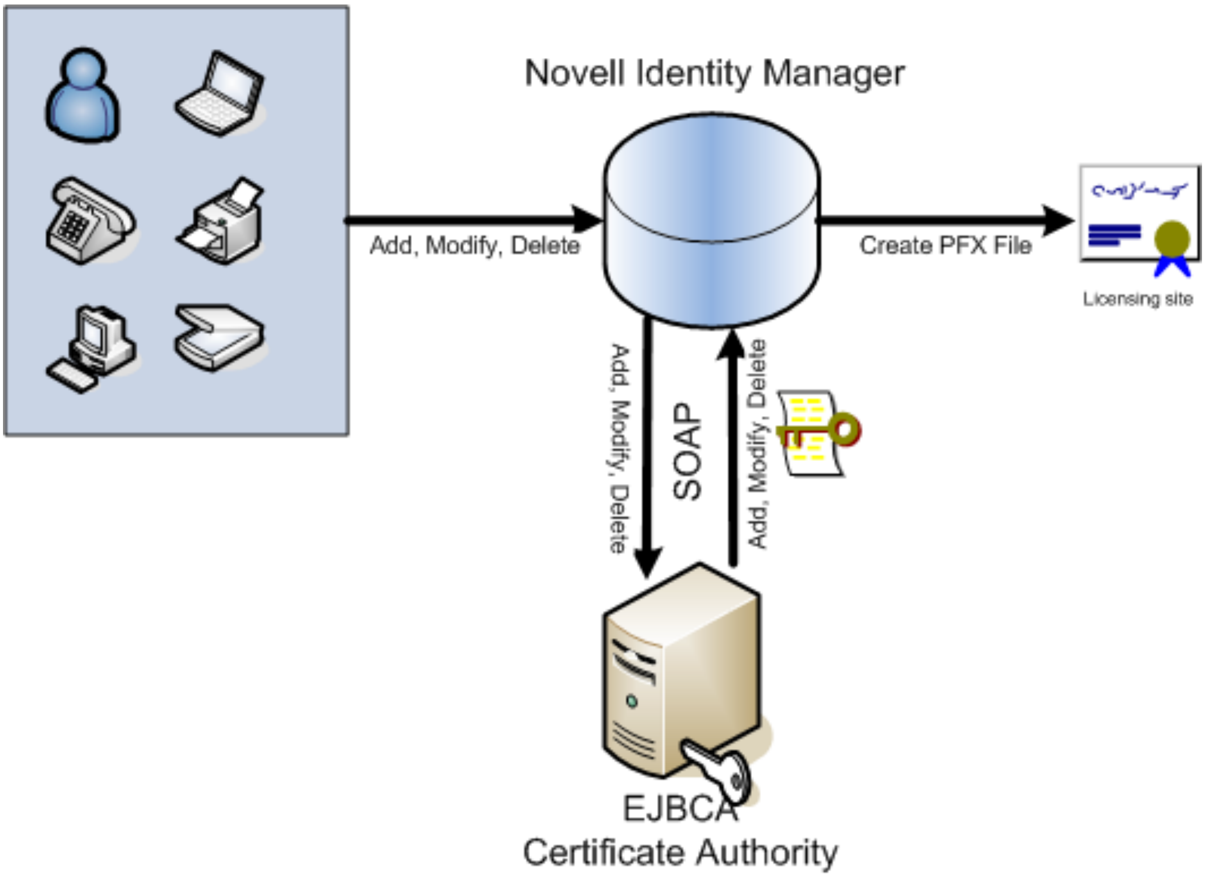


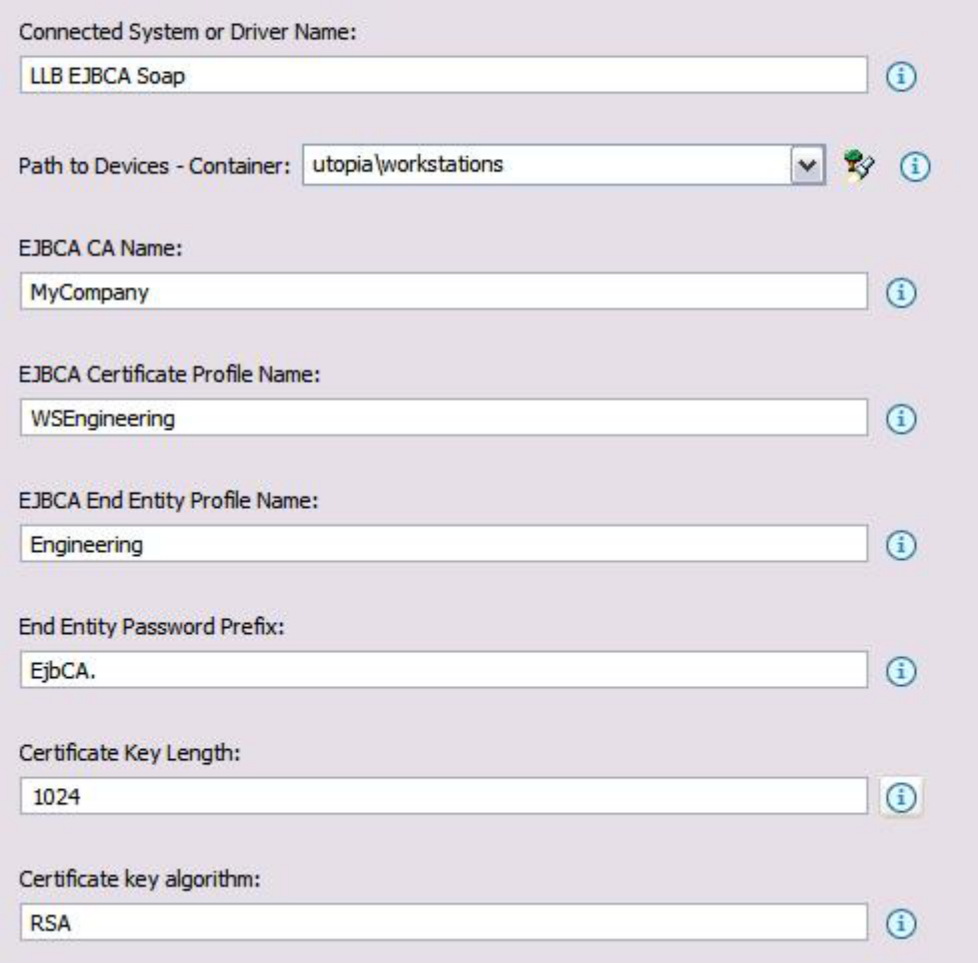
figure 1: Driver Overview

The SOAP driver synchronizes objects from eDirectory with 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 into eDirectory by the SOAP driver. Since all eDirectory object classes can by synchronized with EJBCA, you can create certificate for any eDirectory object (e.g. user, server, workstation, server, laptop, printer, phone etc.).

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

1.1 SOAP Driver (issuing certificates)

This driver communicates via EJBCA’s SOAP (simple object access protocol) interface with the EJBCA server. Object classes and context of objects, which have to be synchronized, are freely definable in the driver’s configuration. Each object is created as an end entity in the EJBCA infrastructure.



Connected System or Driver Name: LLB EJBCA Soap

Path to Devices - Container: utopia\workstations

EJBCA CA Name: MyCompany

EJBCA Certificate Profile Name: WSEngineering

EJBCA End Entity Profile Name: Engineering

End Entity Password Prefix: EjbCA.

Certificate Key Length: 1024

Certificate key algorithm: RSA

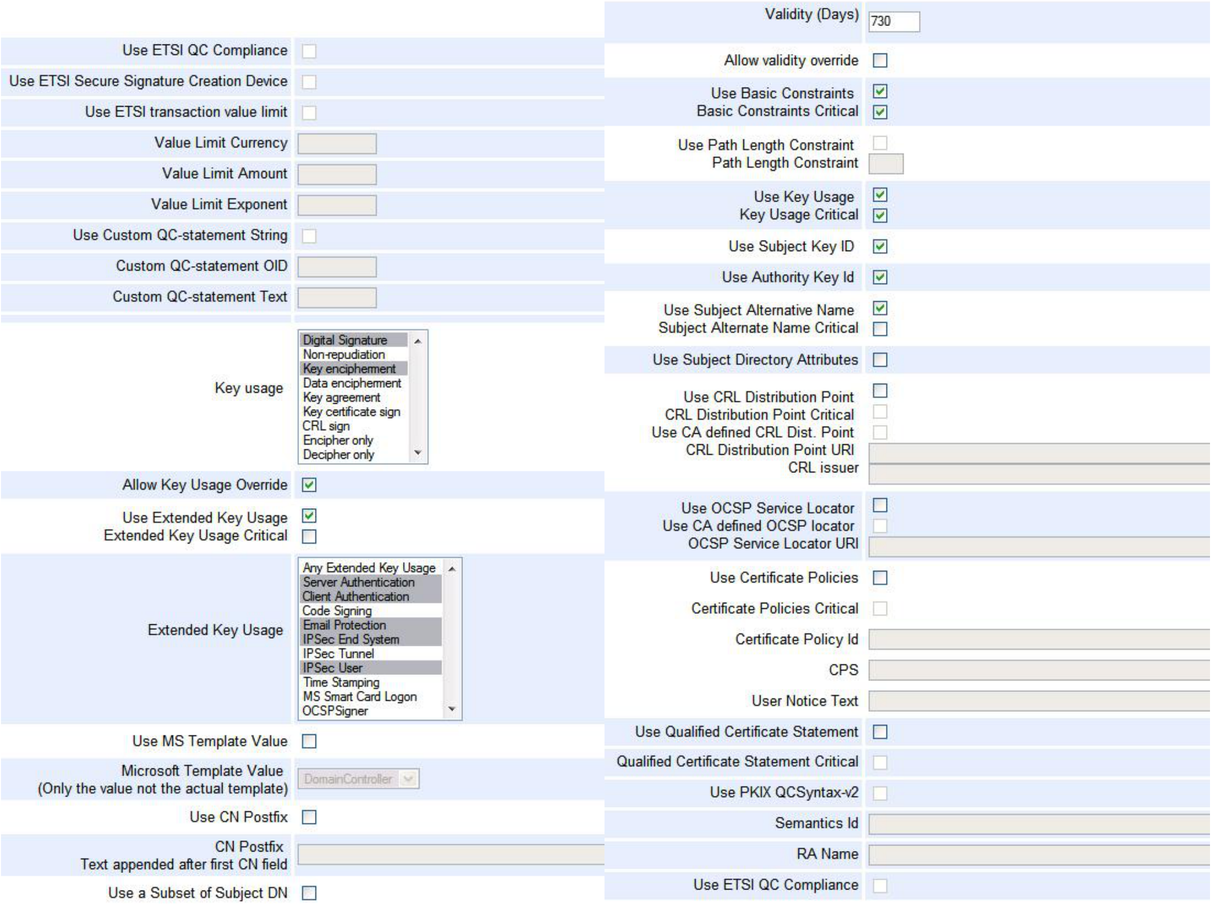
figure 2: SOAP driver global configuration values

You can define key length and key algorithm as well as CA profile and end entity profile. The CA profile defines the desired type of certificate, whereas the end entity profile works as a template for the end entity.

1.1.1 CA profile

The CA profile defines the usage und functionality of the certificate, which is created for the entity in EJBCA. For example the CA profile defines:

- validity of the certificate (in days)
- key usage (digital signature, key or data encipherment, key agreement, CRL sign etc.)
- extended key usage (server or client authentication, email protection, IPSec etc.)
- available key lengths (up to 4096 bits)
- signing CA
- and much more



Use ETSI QC Compliance	□
Use ETSI Secure Signature Creation Device	□
Use ETSI transaction value limit	□
Value Limit Currency	
Value Limit Amount	
Value Limit Exponent	
Use Custom QC-statement String	□
Custom QC-statement OID	
Custom QC-statement Text	
Key usage	Digital Signature Non-repudiation Key encipherment Data encipherment Key agreement Key certificate sign CRL sign Encipher only Decipher only
Allow Key Usage Override	☑
Use Extended Key Usage	☑
Extended Key Usage Critical	□
Extended Key Usage	Any Extended Key Usage Server Authentication Client Authentication Code Signing Email Protection IPSec End System IPSec Tunnel IPSec User Time Stamping MS Smart Card Logon OCSPSigner
Use MS Template Value	□
Microsoft Template Value (Only the value not the actual template)	DomainController
Use CN Postfix	□
CN Postfix Text appended after first CN field	
Use a Subset of Subject DN	□

Validity (Days)	730
Allow validity override	□
Use Basic Constraints	☑
Basic Constraints Critical	☑
Use Path Length Constraint	□
Path Length Constraint	
Use Key Usage	☑
Key Usage Critical	☑
Use Subject Key ID	☑
Use Authority Key Id	☑
Use Subject Alternative Name	☑
Subject Alternate Name Critical	□
Use Subject Directory Attributes	□
Use CRL Distribution Point	□
CRL Distribution Point Critical	□
Use CA defined CRL Dist. Point	□
CRL Distribution Point URI	
CRL issuer	
Use OCSP Service Locator	□
Use CA defined OCSP locator	□
OCSP Service Locator URI	
Use Certificate Policies	□
Certificate Policies Critical	□
Certificate Policy Id	
CPS	
User Notice Text	
Use Qualified Certificate Statement	□
Qualified Certificate Statement Critical	□
Use PKIX QCSyntax-v2	□
Semantics Id	
RA Name	
Use ETSI QC Compliance	□

figure 3: examples of profile parameters

1.1.2 End Entity Profile

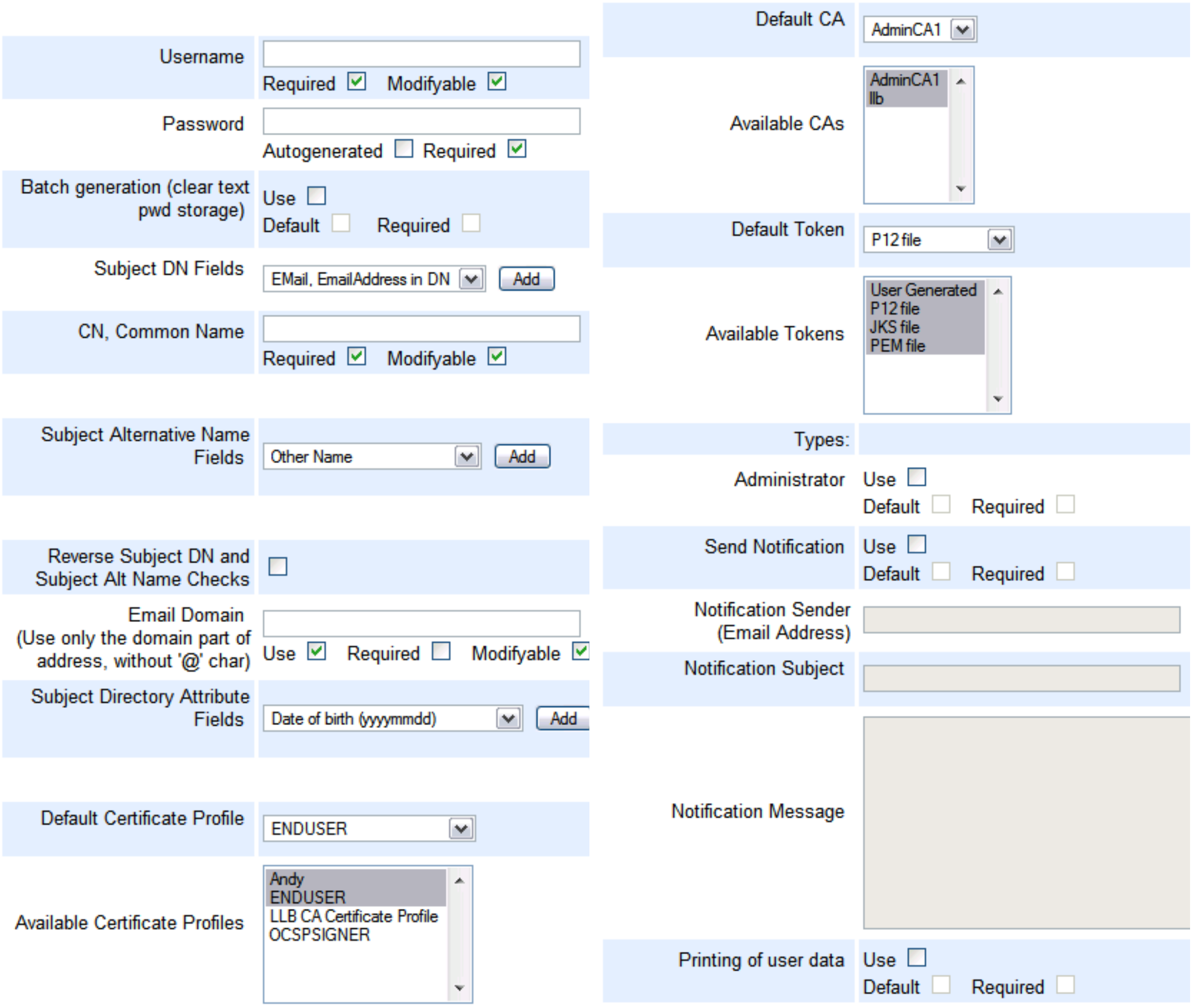
The end entity profile defines many parameters and attributes for the end entity. These are for example:

- attribute for object naming
- alternative naming fields
- required fields
- by which CA profile the entity can be created
- supported tokens (P12, JKS, PEM)
- and much more

Since EJBCA allows defining different CA profiles and end entity profiles, the driver is extremely flexible. You can use different driver instances for different object classes or contexts, which use different CA profiles or end entity profiles.

The certificate information including private and public keys are stored in your eDirectory . A separate attribute holds the public key of the certificate for LDAP validation purposes. Additionally the driver also stores the creation and the expiration date of certificate.

If the naming attribute of the object changes in your eDirectory, the driver deletes the entity in EJBCA and creates a new entity with a new certificate. If you delete the object in eDirectory, the entity is also removed in EJBCA.



Username	Required ☑ Modifyable ☑
Password	Autogenerated ☐ Required ☑
Batch generation (clear text pwd storage)	Use ☐ Default ☐ Required ☐
Subject DN Fields	EMail, EmailAddress in DN Add
CN, Common Name	Required ☑ Modifyable ☑
Subject Alternative Name Fields	Other Name Add
Reverse Subject DN and Subject Alt Name Checks	☐
Email Domain (Use only the domain part of address, without '@' char)	Use ☑ Required ☐ Modifyable ☑
Subject Directory Attribute Fields	Date of birth (yyyymmdd) Add
Default Certificate Profile	ENDUSER
Available Certificate Profiles	Andy ENDUSER LLB CA Certificate Profile OCSPSIGNER

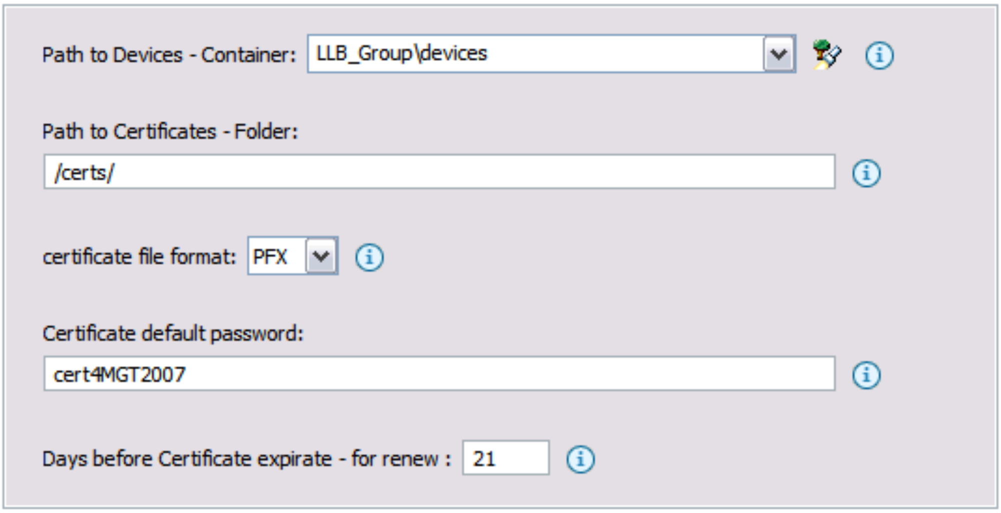
Default CA	AdminCA1
Available CAs	AdminCA1 llb
Default Token	P12 file
Available Tokens	User Generated P12 file JKS file PEM file
Types:	
Administrator	Use ☐ Default ☐ Required ☐
Send Notification	Use ☐ Default ☐ Required ☐
Notification Sender (Email Address)	
Notification Subject	
Notification Message	
Printing of user data	Use ☐ Default ☐ Required ☐

figure 4: end entity profile parameters

The SOAP communication is secured by a client certificate, that needs to be issued by the EJBCA CA. No unauthorized client can access the SOAP services. All transferred data is SSL encrypted.

1.2 Loopback driver (exporting certificates)

The loopback driver exports the certificates in a file based directory. At the moment the certificate information is written into eDirectory by the SOAP driver, the loopback driver exports the certificate into a PFX, CER or DER file. The file format as well as the destination directory is configurable. Using the pfx file format allows you to protect the file with a password.



Path to Devices - Container: LLB_Group\devices

Path to Certificates - Folder: /certs/

certificate file format: PFX

Certificate default password: cert4MGT2007

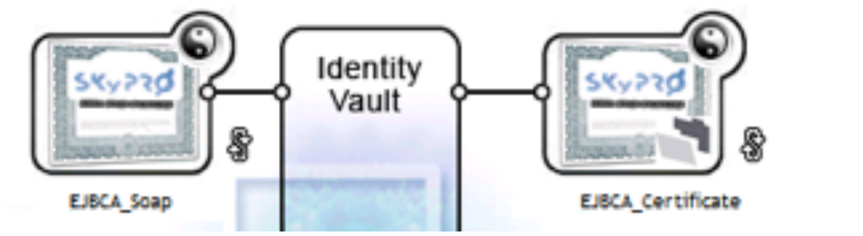
Days before Certificate expirate - for renew : 21

figure 5: loopback driver global configuration values

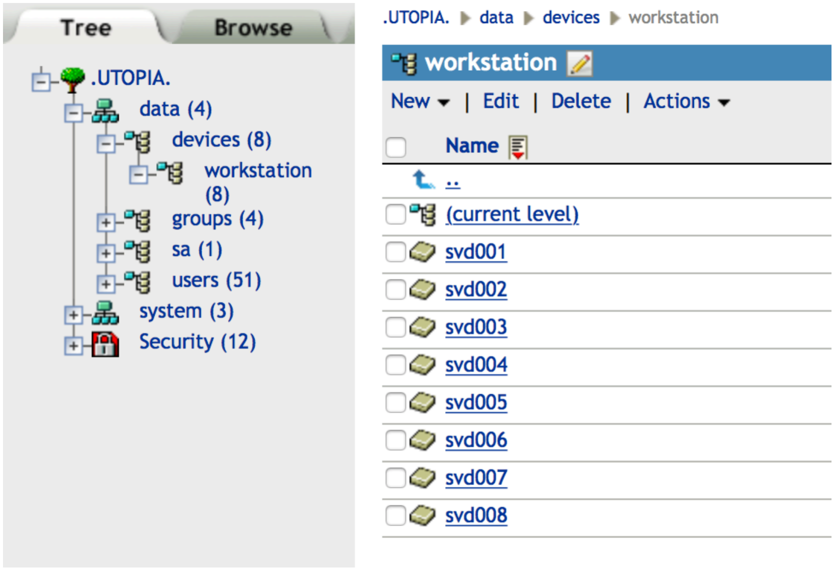
The loopback driver also polls the central directory for all certificates, reaching their expiration date. The driver allows defining an automatic “in time” renewal process for these certificates. In the driver parameters you’re able to define how many days before reaching the expiration date a new certificate will be created and exported automatically.

2 Example

Both drivers are up and running.



e.g. we create workstation objects in eDirectory in a specific container



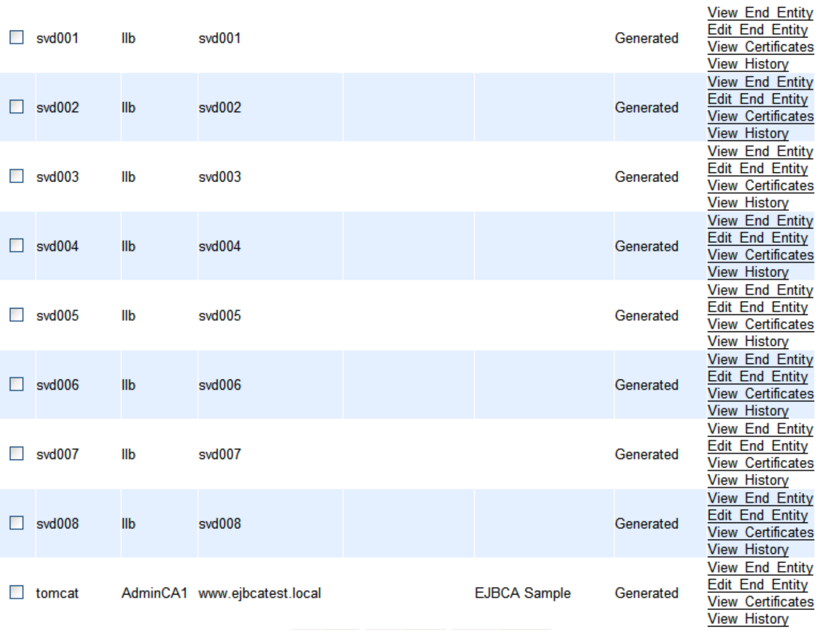
.UTOPIA.
- data (4)
  - devices (8)
    - workstation (8)
  - groups (4)
  - sa (1)
  - users (51)
- system (3)
- Security (12)

.UTOPIA. ▶ data ▶ devices ▶ workstation

workstation

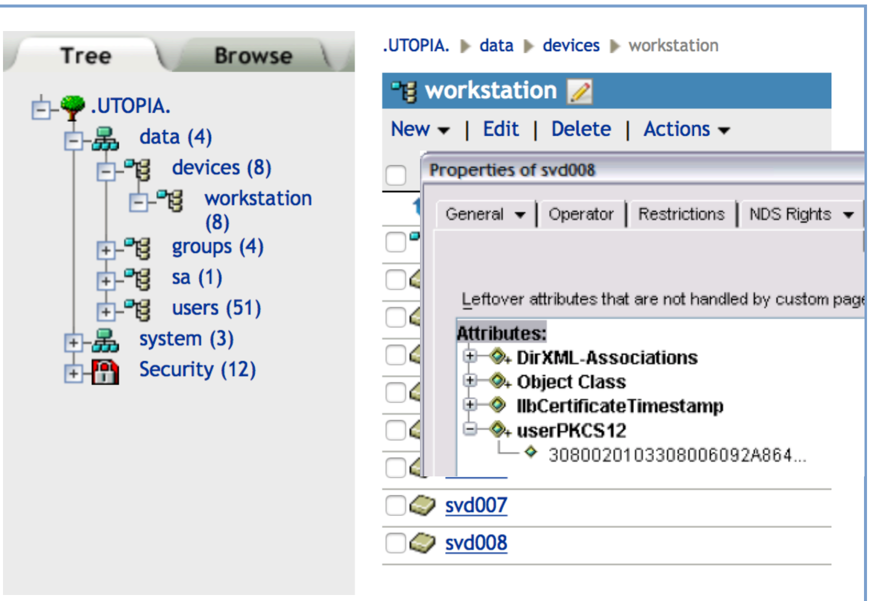
..
□ (current level)
□ svd001
□ svd002
□ svd003
□ svd004
□ svd005
□ svd006
□ svd007
□ svd008

All workstation objects are create as *end entity* in the EJBCA PKI infrastructure. The appropriate certificates are generated.



□	svd001	llb	svd001			Generated	View End Entity Edit End Entity View Certificates View History
□	svd002	llb	svd002			Generated	View End Entity Edit End Entity View Certificates View History
□	svd003	llb	svd003			Generated	View End Entity Edit End Entity View Certificates View History
□	svd004	llb	svd004			Generated	View End Entity Edit End Entity View Certificates View History
□	svd005	llb	svd005			Generated	View End Entity Edit End Entity View Certificates View History
□	svd006	llb	svd006			Generated	View End Entity Edit End Entity View Certificates View History
□	svd007	llb	svd007			Generated	View End Entity Edit End Entity View Certificates View History
□	svd008	llb	svd008			Generated	View End Entity Edit End Entity View Certificates View History
□	tomcat	AdminCA1	www.ejbcatest.local		EJBCA Sample	Generated	View End Entity Edit End Entity View Certificates View History

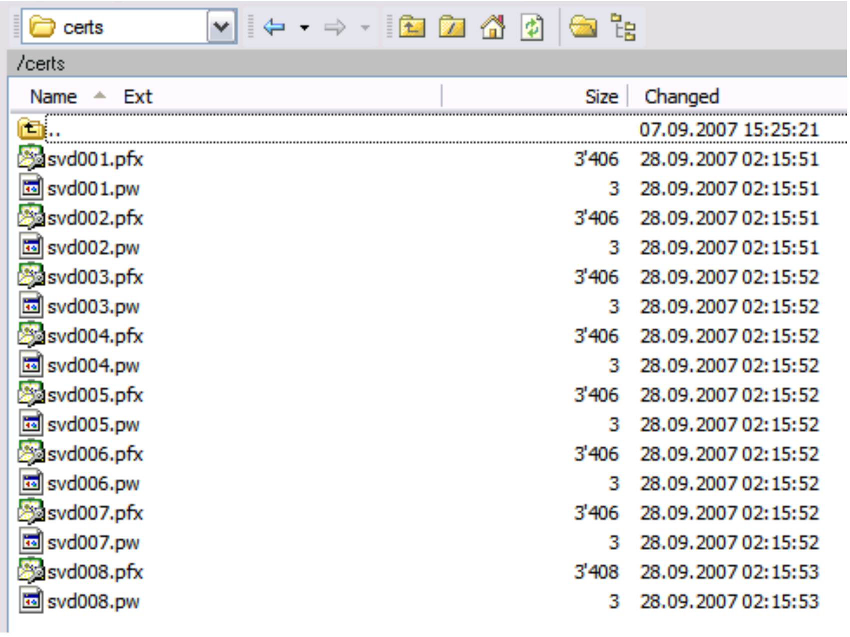
In iManager you see the attribute *userPKCS12* for all objects, which have received a certificate from EJBCA. This attribute holds the certificate including private and public key material.



Leftover attributes that are not handled by custom page

Attributes:
- DirXML-Associations
- Object Class
- IlbCertificateTimestamp
- userPKCS12
  - 3080020103308006092A864...

All certificates are exported as PFX, CER or DER file including the password (PFX only) defined in the loopback driver.



Name	Ext	Size	Changed
..			07.09.2007 15:25:21
svd001.pfx		3'406	28.09.2007 02:15:51
svd001.pw		3	28.09.2007 02:15:51
svd002.pfx		3'406	28.09.2007 02:15:51
svd002.pw		3	28.09.2007 02:15:51
svd003.pfx		3'406	28.09.2007 02:15:52
svd003.pw		3	28.09.2007 02:15:52
svd004.pfx		3'406	28.09.2007 02:15:52
svd004.pw		3	28.09.2007 02:15:52
svd005.pfx		3'406	28.09.2007 02:15:52
svd005.pw		3	28.09.2007 02:15:52
svd006.pfx		3'406	28.09.2007 02:15:52
svd006.pw		3	28.09.2007 02:15:52
svd007.pfx		3'406	28.09.2007 02:15:52
svd007.pw		3	28.09.2007 02:15:52
svd008.pfx		3'408	28.09.2007 02:15:53
svd008.pw		3	28.09.2007 02:15:53