

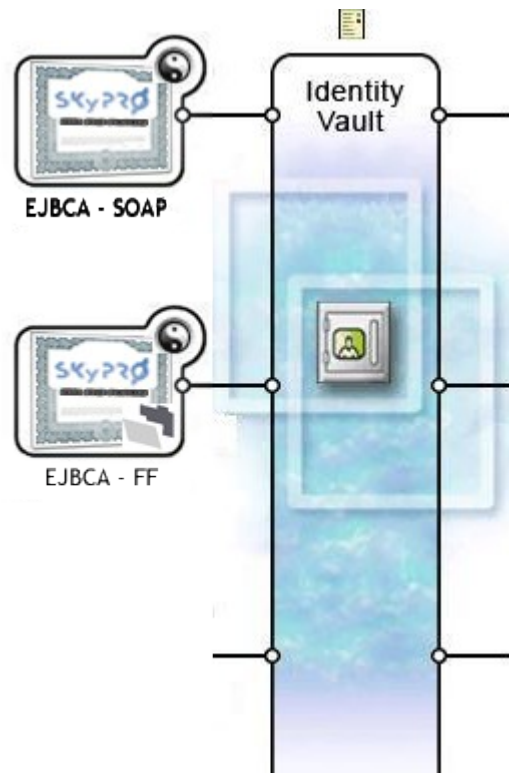


EJBCA Certificate Driver



for Novell Identity Manager

Technical Description



Version: 1.0

last updated: 24.10.2007
issue date: 24.10.2007

filename: EJBCA Driver technical description v1.0

Table of contents

1	ABSTRACT	3
1.1	SOAP DRIVER (ISSUING CERTIFICATES)	4
1.1.1	CA PROFILE.....	4
1.1.2	END ENTITY PROFILE.....	5
1.2	LOOPBACK DRIVER (EXPORTING CERTIFICATES)	7
2	EXAMPLE	8

1 Abstract

Based on the open source Certificate Authority EJBCA (ejbca.sourceforge.net) the EJBCA driver creates certificates for user, workstation or other any object in your central directory. 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.

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

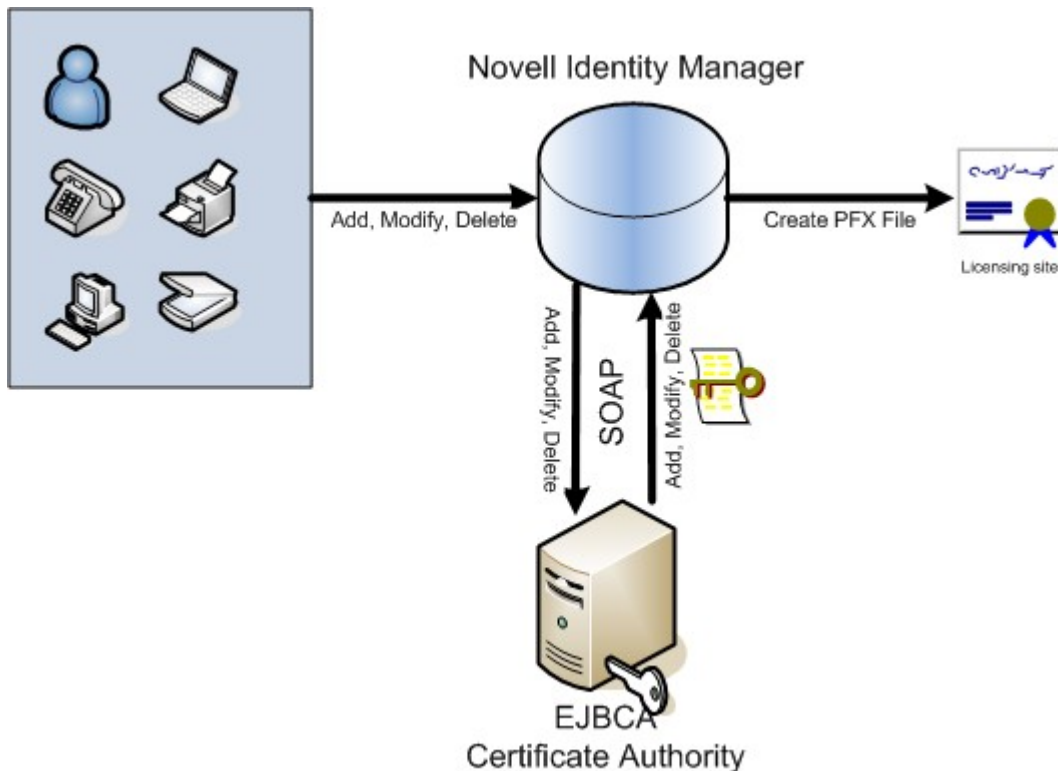


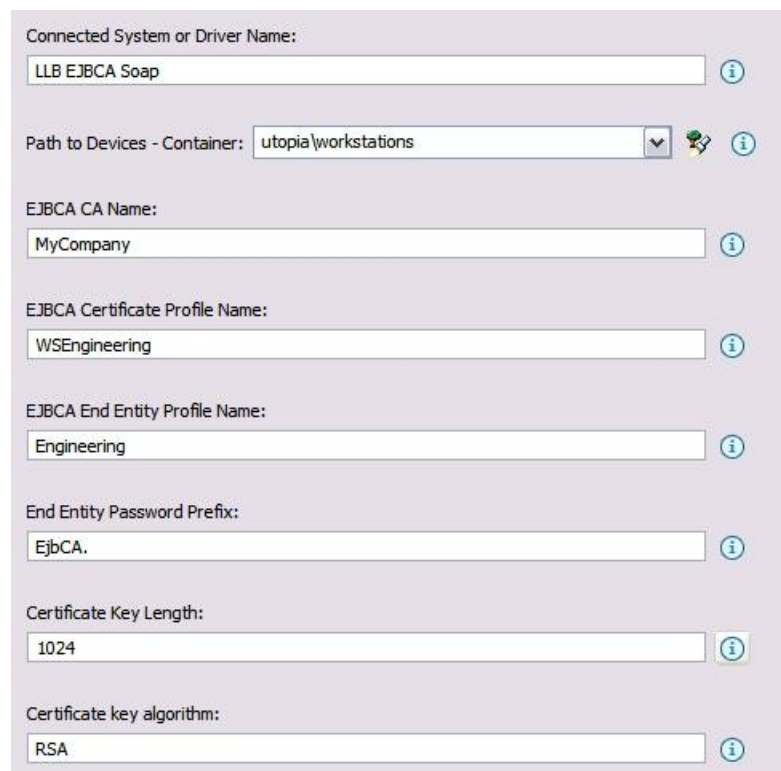
Image 1: EJBCA Overview

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 be synchronized with EJBCA, you can create certificate for any eDirectory object.

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 with via SOAP (simple object access protocol) protocol 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

Image 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 and functionality of the certificate, that 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

The image shows a configuration interface for EJBCA CA with two columns of settings. The left column includes options for ETSI QC Compliance, ETSI Secure Signature Creation Device, transaction value limits, custom QC-statement strings, and key usage settings. The right column includes validity settings, basic constraints, key usage, subject key ID, authority key ID, subject alternative names, and certificate policies. Several settings are checked, such as 'Use Basic Constraints', 'Use Key Usage', and 'Use Subject Key ID'.

Image 3: EJBCA CA: examples of available 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 central directory . 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 central directory, the driver deletes the entity in EJBCA and creates a new entity with a new certificate. After deletion of the object in the central directory, the entity is also removed in EJBCA.

The screenshot displays the configuration parameters for an end entity profile in EJBCA. The interface is organized into several sections:

- User Information:** Username (Required, Modifiable), Password (Autogenerated, Required).
- Batch generation (clear text pwd storage):** Use (checkbox), Default (checkbox), Required (checkbox).
- Subject DN Fields:** EMail, EmailAddress in DN (dropdown), Add button.
- CN, Common Name:** (text input), Required, Modifiable.
- Subject Alternative Name Fields:** Other Name (dropdown), Add button.
- Reverse Subject DN and Subject Alt Name Checks:** (checkbox).
- Email Domain:** (text input), Use (checkbox), Required (checkbox), Modifiable (checkbox).
- Subject Directory Attribute Fields:** Date of birth (yyyyymmdd) (dropdown), Add button.
- Default Certificate Profile:** ENDUSER (dropdown).
- Available Certificate Profiles:** List including Andy, ENDUSER, LLB CA Certificate Profile, OCSPSIGNER.
- Default CA:** AdminCA1 (dropdown).
- Available CAs:** List including AdminCA1, lb.
- Default Token:** P12 file (dropdown).
- Available Tokens:** List including User Generated, P12 file, JKS file, PEM file.
- Types:**
 - Administrator: Use (checkbox), Default (checkbox), Required (checkbox).
 - Send Notification: Use (checkbox), Default (checkbox), Required (checkbox).
- Notification Sender (Email Address):** (text input).
- Notification Subject:** (text input).
- Notification Message:** (large text area).
- Printing of user data:** Use (checkbox), Default (checkbox), Required (checkbox).

Image 4: EJBCA: 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 certificate in a file based directory. On storing the certificate information in the eDirectory, the driver exports the certificate into a PFX, CER, DER or PEM 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.

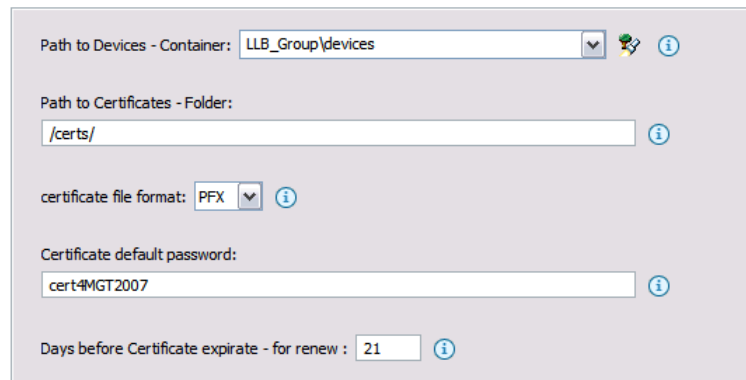
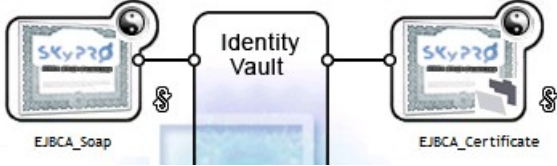
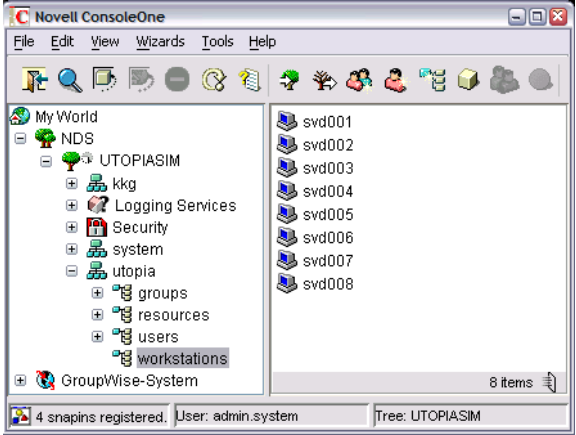
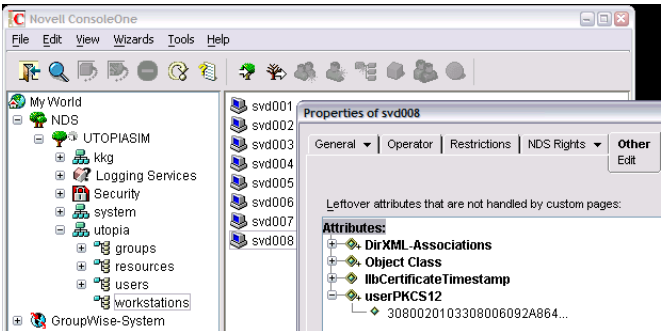


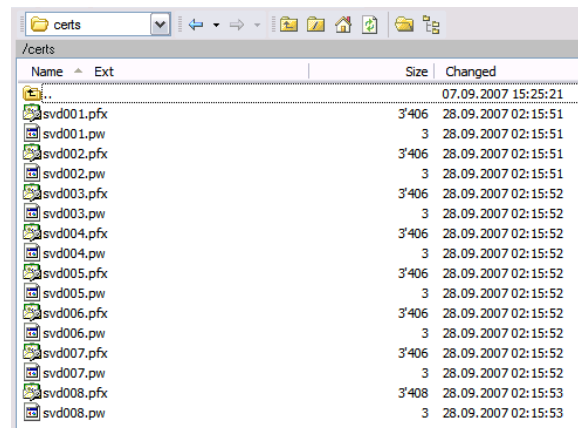
Image 5: Loopback driver global configuration values

The loopback driver also polls the central directory for certificates which are running out. The driver allows defining an automatic "in time" renewal process for such certificates. You can instruct the driver how many days before reaching the expiration date a new certificate will be created and exported.

2 Example

<p>Both drivers are up and running.</p>																																																																
<p>e.g. we create workstation objects in eDirectory in a specific container</p>																																																																
<p>All workstation objects are created as <i>end entity</i> in the EJBCA PKI infrastructure. The appropriate certificates are generated.</p>	<table border="1"> <tr><td><input type="checkbox"/></td><td>svd001</td><td>lib</td><td>svd001</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd002</td><td>lib</td><td>svd002</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd003</td><td>lib</td><td>svd003</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd004</td><td>lib</td><td>svd004</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd005</td><td>lib</td><td>svd005</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd006</td><td>lib</td><td>svd006</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd007</td><td>lib</td><td>svd007</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>svd008</td><td>lib</td><td>svd008</td><td></td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> <tr><td><input type="checkbox"/></td><td>tomcat</td><td>AdminCA1</td><td>www.ejbcatest.local</td><td>EJBCA Sample</td><td>Generated</td><td>View End Entity Edit End Entity View Certificates View History</td></tr> </table>	<input type="checkbox"/>	svd001	lib	svd001		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd002	lib	svd002		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd003	lib	svd003		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd004	lib	svd004		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd005	lib	svd005		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd006	lib	svd006		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd007	lib	svd007		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	svd008	lib	svd008		Generated	View End Entity Edit End Entity View Certificates View History	<input type="checkbox"/>	tomcat	AdminCA1	www.ejbcatest.local	EJBCA Sample	Generated	View End Entity Edit End Entity View Certificates View History
<input type="checkbox"/>	svd001	lib	svd001		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd002	lib	svd002		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd003	lib	svd003		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd004	lib	svd004		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd005	lib	svd005		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd006	lib	svd006		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd007	lib	svd007		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	svd008	lib	svd008		Generated	View End Entity Edit End Entity View Certificates View History																																																										
<input type="checkbox"/>	tomcat	AdminCA1	www.ejbcatest.local	EJBCA Sample	Generated	View End Entity Edit End Entity View Certificates View History																																																										
<p>In ConsoleOne you see the attribute <i>userPKCS1</i>. This attribute holds the certificate including private and public key material.</p>																																																																

All certificates are exported as PFX file including the password file by the loopback driver.



The screenshot shows a Windows Explorer window with the address bar set to '/certs'. The main pane displays a list of files in a table format with columns for Name, Ext, Size, and Changed. The files are organized into pairs of PFX and PW files, numbered from 001 to 008.

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