Nets DanID A/S
Lautrupbjerg 10
DK – 2750 Ballerup
T +45 87 42 45 00
F +45 70 20 66 29
info@danid.dk
www.nets-danid.dk

CVR no. 30808460

Specification document for LDAP API

Table of Contents

Version history

25 Nov 2009	Version 0, First revision of OCES I documentation	JRF
30 Nov 2009	Review	CR
8 February 2010	Version 1, Review and formatting	TechniWrite
10 June 2010	Version 1.1	MTV
20 January 2011	Version 1.2	MTV
1 February 2011	Version 1.3	MTV
23 February 2011	Version 1.4	CR/SEF
1 March 2012	Version 1.5	PKRIS

1. Purpose and target group

This document is a part of the Service Provider Package for NemID.

The purpose of the document is to provide a general introduction to LDAP and to describe how data are organised on NemID's LDAP servers. The document is relevant for the service provider if Nets DanID's Security Package does not contain the desired functionality in this area.

The document is aimed at the people at the service provider who are responsible for the implementation of NemID.

Summary of all documents in the Service Provider Package:

General documentation

- Introduction to NemID and the Service Provider Package
- Guidelines on the interaction design and user selection of applet
- Manuscript for migration to NemID
- Terms and concepts in NemID

Implementation documentation

- Implementation guidelines for NemID
- Configuration and setup

Test documentation

- Guidelines on the use of test tools
- Recommended Test Procedures

Reference documentation

- Specification document for the PID-CPR service
- Specification document for the RID-CPR service
- **Specification document for LDAP API**
- Specification document for OCSP
- Specification document for OCES II

2. LDAP databases

An LDAP database is an object-oriented way of storing data. In contrast to a traditional database, it does not contain rows and columns. Data are instead organised as objects with mapped attributes in a tree structure.

LDAP databases are traditionally used for storing, for example, user accounts in Unix systems or information from a certification authority. The information stored can be, for example, certificates or certificate revocation lists.

Each object in an LDAP database has a *distinguished name* (DN), which identifies the object unambiguously. The DN serves as a representation of a tree structure. The objects in LDAP can have child objects, and each object can thus be treated as a leaf or an inner node in a tree of objects.

An object has one or more object classes mapped, which indicate attributes that are mapped to the object. An attribute can be either mandatory or optional.

As a part of the NemID infrastructure, Nets DanID provides access to two publicly available LDAP databases:

- **Certificate LDAP** – With nodes that represent published end user certificates.

- **Certificate revocation list LDAP –** With nodes that represent CAs (Certificate Authorities) and certificate revocation lists.

3. Example of a person with NemID with a certificate

A citizen in Denmark who has a NemID certificate will also be included in NemID's certificate LDAP, if the person has chosen to publish his/her certificate in the address book.

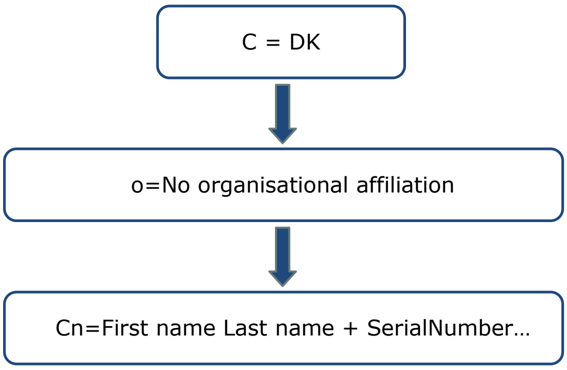
The person's DN may then look like this:

```
cn=Forename Surname+SerialNumber=PID:9208-2002-2-
123456789012, o=No organisational affiliation, c=DK
```

From this you can read:

- the person's name in the ***cn*** field (common name)
- the person's allocated PID in the ***SerialNumber*** field
- the person's organisational affiliation (none) in the ***o*** field (organization)
- the person's country in the ***c*** field (country).

This corresponds to the following tree structure:



C = DK

o=No organisational affiliation

Cn=First name Last name + SerialNumber…

Please note that there is also a facility to provide extra information about the person in the **ou** field (organizationalUnit), which is inserted under the o-node. The user will then have the following DN:

```
cn=Forename Surname (Young person under 18)
+SerialNumber=PID:9208-2002-2-123456789012, ou=Young
person between 15 and 18,o=No organisational
affiliation, c=DK
```

3.1 Attributes for a person

There are the following extra attributes for a person in the Certificate LDAP:

Attribute name	Content
sn	User's surname
mail	The user's email address, which is included in the certificate. If the user has chosen not to include the email address in the certificate, this attribute will not be defined.
userCertificate	The user's certificate. Only the user's most recent certificate is accessible.

4. Example of a corporate signature

The LDAP properties for a corporate signature are the same as for a NemID employee signature. The only difference is that the serialNumber contains a UID instead of a RID, as would be the case with an employee signature.

5. Example of an employee with a NemID employee signature

An employee at a company in Denmark, who has a NemID employee signature, can be found in NemIDs certificate LDAP, if the employee has chosen to publish his or her certificate in the address book.

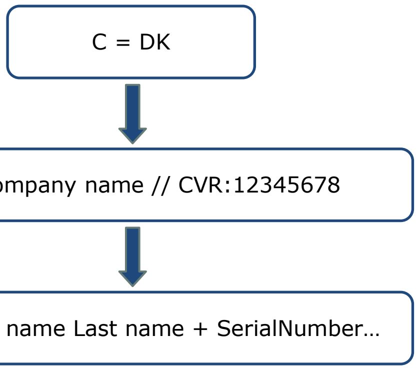
The employee's DN could look as follows:

```
cn = FirstName LastName + serialNumber = CVR :12345678-RID:
123456789012, o = Company Name / / TEL: 12345678, c = U.S.
```

Based on the DN one can conclude the following:

- The employee's name in the cn (common name)
- The employee's serial number (which contains the CVR number for his/her company and a serial number) in the serialNumber
- The employee's organizational affiliation in the field ***o*** (organization)
- The employee's country in the field ***c*** (country).

This corresponds to the following tree structure:


C = DK
o=Company name // CVR:12345678
Cn=First name Last name + SerialNumber…

Note that it is also possible to specify the organizational unit, to which the employee is assigned. This is done using the field ou (organizationalUnit), which is inserted under the *o*-node, so that the employee could have the following DN:

```
cn = FirstName LastName + serialNumber = CVR
:12345678-RID: 123456789012, OU = Test Department,
o = Company Name / / TEL: 12345678, c = U.S.
```

Possible attributes for an employee are the same as for a citizen.

6. Example: NemID issuing CA

NemID works with the aid of a root CA, which issues certificates exclusively to a number of issuing CAs. Each of these CAs issues a (large) number of certificates, after which it switches to an inactive state, in which it no longer issues certificates, but only administers (e.g. in terms of revocation) the certificates already issued.

A NemID certificate is thus always issued by one issuing CA – never by the root CA. Both the root CA and the issuing CAs are represented as nodes in the Certificate Revocation List LDAP.

The root CA has the following DN:

```
cn=TRUST2408 OCES Primary CA, o=TRUST2408, c=DK
```

The issuing CAs have DNs in the following format:

```
cn=TRUST2408 OCES CA n, o=TRUST2408, c=DK
```

6.1 Attributes for CA

The CA nodes' attributes include the following:

Attribute name	Content
caCertificate	The CA's certificate. For the root CA the certificate is self-signed; for an issuing CA it is signed by the root CA.
certificateRevocationList	The full certificate revocation list for the certificates that have been issued by this CA. For the root CA these certificates are for issuing CAs; for an issuing CA they are certificates for citizens.

7. Partial certificate revocation lists

Under each node that represents an issuing CA there are a number of nodes, each of which contains a part of the full certificate revocation list. These are called *partial certificate revocation lists*. A partial certificate revocation list contains information about up to 750 certificates, but it is faster and requires less capacity to access than the full certificate revocation list for a CA.

The nodes have DNs in the following format:

```
    cn=CRL1,cn=TRUST2408 OCES CA n, o=TRUST2408, c=DK
```

The actual certificate revocation list and the full certificate revocation list are stored in the *certificateRevocationList* attribute under the node.

To find the partial certificate revocation list that will contain a given citizen’s certificate, if the certificate is revoked you must first use the DN of the issuing CA that is in the certificate as issuerDN to locate the issuing CA in LDAP.

When you have located this, you can use the X509 extension point *crlDistributionPoint* in the certificate to find DN for the sub-node under the issuing CA that contains the partial certificate revocation list – this enables you to perform a search that finds the correct node.

> If you only need to investigate whether a given certificate for a citizen has been revoked, it will as a rule be more appropriate to use the OCSP protocol. Read more about this in the document entitled **Specification document for OCSP**, which is part of the Service Provider Package.

8. Searches

When you want to perform a search, there are a number of necessary parameters:

1. Which LDAP server is to be searched:

    a. NemID's certificate LDAP is at *crtdir.certifikat.dk*.

    b. The certificate revocation list LDAP is at *crldir.certifikat.dk*.

    In both cases a TCP connection must be created to port 389.

2. In which search database the search is to start, i.e. which node in the tree is to serve as the root node for the search.

3. To what level you want to search: 1) in the search database node, 2) one level down or 3) in the whole sub-tree of the search database node.

4. Which search filter is to be matched against.

5. Which attributes you wish to have returned for matching nodes.

6. You must also specify whether you want to log on to perform the search. Only anonymous searches in the NemID LDAP databases are supported. It is also not possible to perform wildcard searches via the LDAP protocol.

The above parameters must be specified, regardless of which LDAP client is being used, but the mechanism for specifying them can, of course, vary.

8.1 Searches for a person with NemID with certificate

If you want to find certificates that belong to a person named Jens Hansen, you must specify the following parameters:

Search database	o=No organisational affiliation, c=DK
Search depth	Sub
Search filter	cn=Jens Hansen
Attribute	userCertificate

In this instance a person’s name proves to be a poor filter, as there are many people called Jens Hansen. You can instead use the person’s email address as a filter. If the Jens Hansen you are looking for has the email address **jens@hansen.dk**, this can be done using the following search parameters:

Search database	o=No organisational affiliation, c=DK
Search depth	Sub
Search filter	mail=jens@hansen.dk
Attribute	userCertificate

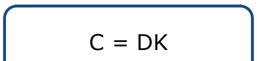
A limited number of search results Is returns. The limit is five from OCES I and five from OCES II. This means that a comprehensive search in both could provide up to ten entries.

8.2 Search for certificate revocation list for an issuing CA

To obtain the full revocation list for, for example, issuing CA number I, you can perform a search using the following parameters:

Search database	o=TRUST2408, c=DK
Search depth	One
Search filter	cn=TRUST2408 OCES CA I, o=TRUST2408, c=DK
Attribute	certificateRevocationList

The full certificate revocation list will then be returned.



9. LDAP clients

There are many different LDAP clients. Email clients will often have integrated LDAP in their address book functionality – this is true, for example, of email clients from Mozilla and Microsoft.

At *http://www.openldap.org* you will find the LDAP client ***ldapsearch***, which is a command line tool.

http://www.ldapadministrator.com has a GUI-based LDAP browser client.

9.1 Example of search with ldapsearch

The above three search examples could be performed with the following parameters for **ldapsearch**:

```
> ldapsearch –h crtdir.certifikat.dk –b “o=No organisational
affiliation, c=DK” ”cn=Jens Hansen” userCertificate

> ldapsearch –h crtdir.certifikat.dk –b “o=No organisational
affiliation, c=DK” ”mail=jens@hansen.dk” userCertificate

> ldapsearch –h crldir.certifikat.dk –b “o=TRUST2408, c=DK” –s
one “cn=TRUST2408 OCES CA I” certificateRevocationList
```

It is also possible to find a CA’s certificate with the following parameters for **ldapsearch**:

```
> ldapsearch -x -h crldir.certifikat.dk -s one -b
'o=TRUST2408,c=DK' '(cn=TRUST2408 OCES CA I)' cACertificate
```

With the addition of AIA in certificates, it is however possible to find the issuing CA directly by following the given HTTP reference in the user’s certificate.

9.2 Example of search with Internet Explorer

In addition to the address book’s GUI search, Internet Explorer also supports LDAP searches via URL encoding of the search string. This is described in more detail in RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt).
The results are displayed in the address book to maintain email security.