SmartSignatur

Referencemanual for SmartSignatur® Server

LCES version 2.9.5 december 2014

Indholdsfortegnelse

Overblik

SmartSignatur Server applikationen er et Java web system, hvor brugeren kan udstede en NemID medarbejdersignatur hos Nets DanID og gemme den på lokal SmartSignatur server i sin virksomhed. Signaturen er efterfølgende til rådighed for brugeren i den lokale Windows session på enten en klassisk PC eller via eksempelvis VDI, RDP mv. Teknisk er SmartSignatur Server historisk navngivet LCES (Local Certificate Enrollment Service). Denne referencemanual bruger LCES og LCES2 fremefter.

LCES består af følgende dele:

1. En Java Appplet, som læser brugerens netværks brugerid. Denne hentes fra signaturserveren og afvikles i brugerens browser.
2. Java Appletten gemmer den resulterende medarbejdersignatur på brugerens Windows maskine.
3. En native Windows applikation og DLL fil, der betjener ovennævnte Java Applet. Den hedder *cv act pki/roamer*. For at, pki/roamer virker skal den installeres i henhold til vejledningen til cv act pki/roamer. Aktuel version af cv act pki/roamer er 2.4.6
4. En web applikation, der skal afvikles på virksomhedens server. Gennem HTML filer, Java Servlets, Java script og Java Applets betjener den brugeren. Den gemmer brugerens NemID medarbejdersignatur på signaturserveren i tilknytning til brugerens objekt i eDirectory.
5. Server databasen, som indeholder brugere og medarbejdersignaturer. Det er NetIQs *eDirectory*, derunder dens *SecretStore* del.
6. Endelig er der en lille administrator-frontend, som kan hjælpe systemadministratoren med at overvåge web applikationen.

Dette dokument beskriver, hvordan HTML-, DLL- og Java-koden, som er omtalt i punkterne 1-4 ovenfor, skal installeres og sættes op til at tilgå databasen, som er omtalt i punkt 5.

Java Applet

Den signerede Java jar fil, CIC294.jar, skal lægges i web-applikationens rodbibliotek. F.eks.:

```
tomcat/webapps/lces295/CIC294.jar
```

Hvis man ændrer denne fil, skal man sikre, at enhver klient-maskine, som har brugt en tidligere version, får sletten cachen, så den faktisk downloader den nye version ved næste brug. Bemærk, at cachen er en maskin-global Java cache; ikke browserens. Det sker i Windows ved at starte Kontrolpanelet, aktivere Java, faneblad *General*, knappen *Settings* ved *Temporary Internet Files.* Her kan man trykke på knappen *Delete files...* og så vælge *Cached Applications and Applets.*
Se også *http://www.java.com/en/download/help/plugin_cache.xml*.

cv act pki/roamer

Applikationen cv act PKI/roamer installeres via en standard Windows installer pakke.

eDirectory Schema udvidelse

For at kunne benytte alle funktioner omkring de ekstra OCES status felter (senere i dokumentet) skal eDirectory schema udvides med disse felter:

- OCES:CertificateType
- OCES:CertificateStatus
- OCES:CertificateSerialNumber
- OCES:ValidFrom
- OCES:ValidTo

Der findes sammen med LCES programmet en særlig lille eDirectory Schema udvidelsesfil med navnet "OCESv20.sch"

Bestilling af NemID medarbejdersignaturer

SmartSignatur Server kan via den indbyggede Identity Manager applikation konfigures til at automatisk at bestille og deaktivere NemID medarbejder direkte via Nets DanIDs særlige selfserviceWS SOAP interface. For at benytte dette SOAP interface skal der indgås særskilt aftale med Nets DanID.

Web applikationen

Sammenlignet med den tidligere LCES1 web-applikation har LCES2 følgende tilføjelser:

1. Java klasserne i pakke-gruppen `dk.liga.*`.
2. Jar-filen `jdom.jar`, version 1 og `xercesImpl-2.8.1.jar`. Det er XML parser klasser.
3. Jar-filen `log4j-1.2.15.jar` og `commons-logging-1.1.1.jar`. . LCES v.2.9.3 bruger logging-frameworket Log4j til logning af program-trace og -fejl samt til logning af forretningshændelser. Opsætningen er dokumenteret nedenfor i afsnittet *Opsætning af Log4j.*
4. Jar-filen `jce-jdk13-147.jar`, som er en del af Bouncycastle, en familie af krypteringsværktøjer, ref. http://www.bouncycastle.org/.
5. Jar-filen `jsso.jar`, som er NetIQs API til at tilgå *eDirectory*.
6. Jar-filen `mail-1.4.jar`, som er et e-mail API. Her bruges blot in Base64 encoder/decoder..
7. Jar-filen `ooapi-1.157.0-20130305.094222-9.jar`. Den bruges til håndtering af nøglekort (POCES) dialogen.
8. Jar-filen `plugin.jar`.
9. Model-XML filer til SOAP forespørgsler, som LCES2 foretager. Disse er RequestIssuanceInput.xml, IssueCertificateInput.xml, RequestRenewalInput.xml og RenewCertificateInput.xml. De skal ligge i Java classpath i `tomcat7/webapps/lces2x/WEB-INF/classes/resources`. Bemærk, at der også ligger nogle Java `*.class` filer i dette bibliotek.

10. Parameterfilen `LCESConfig.properties` kan fra LCES v.2.9.0 lægges udenfor Tomcat's `webapps/lces2xx`, eksempelvis i `/tomcat7/common/lces295`. Opsætningen af dette er dokumenteret nedenfor i afsnittet *Placering af parameterfilen uden for web applikationen*. Bemærk, at det er kun parameterfilen `LCESConfig.properties`, som kan lægges uden for classpath på denne måde.
11. Fra LCES v.2.9.2 var der en ny parameterfil, `nemid.properties`. Bemærk, at disse parametre er flyttet ind i den gamle parameterfil (`LCESConfig.properties`) fra v.2.9.3, så filen `nemid.properties` bruges ikke mere.

LCES fra v.2.9.2 skal køre under Oracle's Java version 6 eller 7.

Direkte link til udstedelse og fornyelse

SmartSignatur serveren kan kaldes direkte via nedenstående URLs. Disse adresse kan eksempelvis indsættes i de standard mail skabeloner som Nets DanID sender direkte til den enkelte bruger.

1. Manuel indtastning: https://internserver/smartsignatur/index.html
2. Ny udstedelse: <URL>/LCES?REFNO=<refno>&pinotp=pin

   f.eks.: https://serverhost:8443/lces/LCES?REFNO=1234567890&pinotp=pin
3. POCES udstedelse: <URL>/LCES?REFNO=<refno>&pinotp=poces

   f.eks.: https://serverhost:8443/lces/LCES?REFNO=1234567890&pinotp=poces
4. Fornyelse og genudstedelse: <URL>/LCES?pinotp=renew

   f.eks.: https://serverhost:8443/lces/LCES?pinotp=renew

Integration til NetsDanID webservices

For at SmartSignatur serveren kan kommunikere med Nets DanIDs SOAP og HTTPS services skal en række forhold være etableret på både kunde og Nets DanID siden

1. Ved almindelige udstedelse og fornyelse kræves ikke andet integration end brug af almindelige SSL kommunikation med Nets DanID *IssueRenew* SOAP service

I alle andre sammenhænge end punkt. 1 skal der være indgået en aftale med Nets DanID

2. Ved anvendelse af Nets DanID *selfserviceWS* SOAP service skal der anvendes et VOCES certifikat til at etablere 2-vejs SSL forbindelse og Nets DanID skal eksplicit give dette certifikatrettigheder til at tilgå de enkelte SOAP services hos Nets.
3. Ved anvendelse af POCES applet (udstedelse af et MOCES certifikat med PIN kode dannet på baggrund af CPR tilknytning og brugerens private NemID) skal:
   A. VOCES certifikatet også af Nets DanID tildeles rettigheder til at kalde POCES appletten hos Nets DanID.
   B. ”logonto” parameter skal være sat hos Nets DanID.
   C. “serviceproviderid” paramenter skal være sat I henhold til det nummer man er tildelt fra Digitaliseringsstyrelsen når man opretter en PID til CPR aftale.
   D. Tiden på LCES serveren skal være sat korrekt.

Opsætning

Web applikationen er distribueret som en standard war-fil, som installeres og opdateres på standard vis. Den indeholder filen web.xml, som definerer disse applikationer:

1. *LCES*, som er den Servlet, som starter det andet skærmbillede i applikationen (den første er en rå HTML fil). Denne servlet skaffer ved hjælp af ovennævnte Java applet brugerens identifikation på dennes maskine.
2. *adm*, som er en servlet, som viser visse opsætnings- og statistik-data for LCES applikationen. Der findes ingen hyperlinks til denne applikation, så den kan kun aktiveres ved at referere *adresse*/lces295/adm. Applikationen er beskrevet i kapitlet *Applikationens administrator dialog* nedenfor.
3. Desuden definerer web.xml en servlet, som skal køre ved applikations-opstart: log4j.init, som aktiverer en Java klasse, Log4jSetup, jvf. afsnittet *Opsætning af Log4j* nedenfor.

Biblioteket css indeholder style sheet filer; biblioteket gfx indeholder billedfiler, begge dele til brug i applikationens web-sider.

Biblioteket WEB-INF/classes indeholder, foruden applikationens Java klassefiler:

1. Filen log4j.xml. Opsætning og modifikation af logningen er beskrevet i afsnittet *Opsætning af Log4j* nedenfor.
2. Directoriet logs, som ikke indeholder Java klassefiler, men som er default placeringen til logfiler fra systemet. Dette kan dog ændres, jvf. afsnittet *Opsætning af Log4j* nedenfor.
3. Parameterfilen LCESConfig.properties, som skal tilpasses lokalt, da den indeholder diverse sign-on parametre og valg af options. Indholdet i denne fil er beskrevet i afsnittet *Parametre i filen LCESConfig.properties* nedenfor og afsnittet *Placering af parameterfilen uden for web aplikationen* beskriver, hvordan man kan placere filen sådan, at den ikke bliver overskrevet ved næste opgradering af web-applikationen. Der er en speciel del af denne parameterfil, som indeholder parametre til nøglekort-dialogen. Biblioteket lces295 indeholder:

Biblioteket LCES295/WebContent indeholder:

1. Filen index.html, som er LCES systemets forside, en passiv HTML side med indtastningsfelter og knapper til at aktivere server systemet.
2. Fire HTML filer, som danner top og bund i de to Web-sider, som udgør applikationen for nyudstedelse med PIN kode, dvs. indtastning af PIN kode og dernæst overvågning af SOAP kaldet og opdatering i *SecretStore*. De hedder footer1/2.html og header1/2.html. De kan ændres lokalt og de kan erstattes af filer med andre navne, jvf. definitionen af parameteren html.lces.LCESWhoAmI.Header m.fl. i afsnittet *Parametre i filen LCESConfig.properties* nedenfor.
3. To HTML filer, som danner top og bund i nøglekort-dialogen, De hedder footer3.html og header3.html. De kan ændres lokalt. De refereres fra JSP filen otppanel.jsp, som også ligger i biblioteket lces295,
4. To HTML filer, som danner top og bund i *WhoAmI* dialogen ved nøglekortnøglekort-dialogen. De hedder footer4.html og header4.html og de kan ændres lokalt og deres navne er referet i parameteren html.lces.LCESWhoAmIOtp.Header m.fl. i afsnittet *Parametre i filen LCESConfig.properties* nedenfor.
5. Endnu to HTML filer, som danner top og bund i fornyelses-dialogen, hvor brugeren skal indtaste sit password til det gamle certifikat. De hedder footer5.html og header5.html og de kan ændres

lokalt. Deres navne er referet i parameteren `html.lces.LCESWhoAmIRen.Header` m.fl. i afsnittet *Parametre i filen LCESConfig.properties* nedenfor.

6. JSP filen otppanel.`jsp`, som starter nøglekort-dialogen, dvs. sætter parametrene op til DanID's POCES applet.
7. JSP filen `finiocapi.jsp`, som viser et statisk afslutningsbillede til det tilfælde, at parameteren `installCapi` ikke beder om, at afslutningsbilledet skal aktivere installation af brugerens nye certifikat på klientmaskinen. Siden opfordrer brugeren til at logge af og på klientmaskinen, hvilket vil installere den nye certifikat på klientmaskinen..
8. JSP-filen `statistics.jsp`, som viser *adm*-applikationen, nævnt ovenfor.
9. Filen errorPanel.html, som er en skabelon-fil for fejlmeddelelse-siden. Den har en linie midt i med "+++"er; denne linie erstattes så på kørselstidspunktet med en fejlmeddelelse.

Der skal også opsættes en Java *truststore* fil, jf. beskrivelsen af parametrene `TrustStore` og `TrustStorePassword` i afsnittet *Parametre i filen LCESConfig.properties* nedenfor.

Parametre i filen LCESConfig.properties

Parameter	**Beskrivelse**
IssueURL	URL'en til DanID's *IssueRenew* SOAP service. Bemærk, at Nets DanID har en adresse for test og en anden for produktion. Desuden leveres med denne applikation en DanID simulator, som kan bruges for test af applikationen, og den adresseres også med denne parameter.
IssueURLss	URL'en til Nets DanID's *SelfService* SOAP service, brugt af nøglekort dialog-delen. Bemærk, at DanID har en adresse for test og en anden for produktion. Desuden leveres med denne applikation en Nets DanID simulator, som kan bruges for test af applikationen, og den adresseres også med denne parameter.
keystoreFileName	Path og navn på den fil, som indeholder arbejdsgiverens VOCES certifikat. Bruges til HTTPS autorisation af *SelfService* SOAP kaldet ved brug af nøglekort. Path skal være fuld path. Filen kan fra LCES v.2.9.3 være at typen PKCS12 eller JKS. Dette skal være markeret ved filnavne suffix, som skal være ".`jks`" for JKS; alt andet anses for at være PKCS12. Bemærk, at denne fil kan være den samme, som den, der refereres fra parameteren `nemid.applet.parameter.signing.keystore`, forudsat at filen er en JKS fil og at dens filnavne suffix er ".`jks`".
keystoreFilePassword	Password til filen, som er angivet i parameteren `keystoreFileName`.
SecurityProtocol	Skal sættes til `ssl`.

Parameter	Beskrivelse
SecretStoreAdmin	DN (Distinguished User) værdien for den *eDirectory* bruger, som har rettigheder til at skrive i *SecretStore*. F.eks.: `cn=SecretStoreAdmin, ou=users, o=SmartSignatur`
SecretStoreAdminPassword	SecretStoreAdmin brugerens password
SecretStoreLDAPUrl	URL'en til den server, som kører eDirectory/SecretStore. F.eks.: `ldap://IP_or_DNS_name_to_eDirectory:636`
TrustStore	Path og navn på den Java truststore fil, som indeholder trustede rodcertifikater. Skal være sat op til at indeholde certifikater for de SLL forbindelser, som applikationen bruger. Bemærk, at der ikke følger nogen keystore fil med LCES installationen. Filen vil blive brugt af hele Tomcat serveren, så der bør laves én fil med alle certifikater i, og altså ikke en fil dedikeret til LCES. Path skal være fuld path. I øvrigt henvises til den almindelige Tomcat dokumentation.
TrustStorePassword	Password til filen, som er refereret i parameteren `TrustStore`.
KeyPair.Algorithm	Skal indeholde værdien `RSA`.
KeyPair.KeyLength	Skal indeholde værdien `2048`.
Request.SignatureAlgorithm	Skal indeholde værdien `SHA256WithRSA`.
Diverse html.cic.CICApplet.* og html.cic.CICWhoAmI.*	Parametre til Java appletterne i filen CIC294.jar. Bør ikke ændres lokalt undtagen i en situation, hvor nye versioner af appletterne er leveret med instruks om ændringer i disse parametre.
html.lces .LCESWhoAmI.Header .LCESWhoAmI.Footer .LCESWhoAmIOtp.Header .LCESWhoAmIOtp.Footer .LCESWhoAmIRen.Header .LCESWhoAmIRen.Footer .LCESInstallUserCert.Header .LCESInstallUserCert.Footer	Som nævnt i afsnittet *Opsætning* ovenfor referer disse parametre HTML filer, som udgør første og sidste dele af applikationens to aktive billeder: Indtastning af PIN kode og overvågning af SOAP kaldet og opdatering i *SecretStore*. Bemærk, at første billede, hvor brugeren indtaster *Referencenummer*, er en selvstændig passiv HTML side.

Parameter	Beskrivelse
html.lces .LCESTemplateWriter.Form	Referer e HTML *form* navn , som bruges ved link mellem HTML siderne. Skal have værdien LCES.
storeOcesFields	Bestemmer, om visse data fra certifikatet (fra-/til-datoer, SerialNumber og en tilstandskode) skal gemmes i dedikerede "OCES"-felter i den LDAP, som er vært for *SecretStore* eller ej. *yes:* Gem værdierne i LDAP; *no:* Gem dem ikke. Optional, default værdien er *no.* Hvis *yes,* bemærk de næste par parametre, også. Mere om disse OCES felter i afsnittet nedenfor.
storeOcesStateValues	Komma-separeret liste af tilstands-værdier, som sættes i feltet *OCESCertificateStatus* i LDAP. Der skal gives netop syv værdier med betydningerne: 1. Renewal processen starter 2. Renewal processen fejlede 3.Begynd forespørgsler til DanID. 4. Forespørgsel til DanID fejlede. 5. Begynd opdatering af SecretStore. 6. SecretStore opdatering fejlede. 7. SecretStore opdatering sluttede OK. Værdierne er af strengformat, men LDAP feltet kan være sat til kun at acceptere tal. Værdierne skal koordineres med LDAP administratoren. Hvis en position har værdien -1, bliver den tilsvarende hændelse ikke markeret i LDAP feltet. Skal gives, hvis storeOcesFields er *yes,* negligeres ved *no.*
validStartOcesStateValues	Een eller flere komma-separerede værdier, som LDAP feltet *OCESCertificateStatus* skal indeholde, for at applikationen vil starte. Bruges kun ved nyudstedelse af certifikat, ikke ved fornyelse. Negligeres ved storeOcesFields = *no.* Ellers optional; ikke givet betyder intet opstarts-check.

Parameter	Beskrivelse
installCapi	Afgør, om billedet efter fremskaffelse af certifikatet fra DanID (og upload til *SecretStore*) skal installere certifikatet på Windows klientmaskinen eller ej. Værdier: • `yes` eller `always`: Gem altid på klientmaskinen • `inssuanceonly`: Gem *Issuance* certifikater på klientmaskinen, men gem ikke *Renewal* certifikater. • `no` eller `never`: Gem aldrig på klientmaskinen Kan udelades, default værdi er `installCapi=yes`.
saveSecretStoreEntries	Bestemmer, om det gamle nøgle/certifikat par bliver gemt under et andet navn i SecretStore. Ny i LCES v.2.9.3. `yes`: de bliver gemt. `no`, default: de bliver ikke gemt Gemte entries har navnet \\smartsignatur.moces\privatekey_yyyymmdd-hhmmss, henholdsvis -\certificate_ …
OCESCertificateStatus OCESCertificateSeria NumberOCESValidFrom OCESValidTo	Mulighed for at angive nøgleværdier for hver af de fire LDAP felter, som LCES opdaterer, hvis parameteren storeOcesFields er sat til "`yes`". Alle fellter har default-værdier, som er de samme som nøgleværdierne. F.eks.: Hvis man ikke angiver parameteren OCESValidTo, vil LCES søge feltet i LDAP under navnet `OCESValidTo`. Men hvis man har angivet denne parameter: `OCESValidTo=expirationDate` bruges LDAP nøglen `expirationDate`.
deleteSecretStoreBeforeWrite	Bestemmer, om brugerens gamle entries i SecretStore skal slettes før det nye key/cert par tilfjøes, eller omvendt. Ny i LCES v.2.9.4.1/v.2.9.5 *yes*: slet gamle entries før tilføj nye. *no*, default: tilføj nye entries før sletning af gamle.

Parameter	Beskrivelse
removeSecretNoErrorChecking	Bestemmer, om en ”not found” fejl ved sletning af enkelt-entries i SecretStore skal medføre logning og stop eller om den skal negligeres. Introduceret pga. en eriodisk SecretStore fejl, hvor sletning af eet entry kan slette alle. *yes*: negliger fejl under sletning af enkelt-entry i SecretStore. *no*, default: Foretag normal fejlhåndtering.
POCES udstedelse	De følgende parametre er til opsætning af nøglekort (POCES) dialogen og parameterisering af Nets DanID appletten. Denne dialog tillader en medarbejder at få tildelt en medarbejdersignatur ved at bruge en blanding af en udstedelse hos Nets DanID og en log-on, autoriseret ved ḿedarbejderes private NemID nøglekort. Den funktionalitet kræver en særskilt aftale med Nets DanID om anvendelse af deres SelfServiceWS SOAP service.
nemid.applet.server.url.prefix	Første del af den URL, som bruges til at lokalisere nøglekort-Appletten fra DanID. For test miljøet: https://appletk.danid.dk, for produktion: https://applet.danid.dk
nemid.applet.parameter.signing .keystore	Path og navn på den JKS fil, som skal bruges til signering af DanID applettens forespørgsel. Det er det samme certifikat, som skal være givet i LCESConfig.properties parameteres `keystoreFileName`, hvor certifikatet dog kan være pakket som en PKCS12 fil. Bemærk, at path er absolut, i modsætning til reglerne for LCES v.2.9.2, hvor det var relativ til Java classpath. Denne parameter kan referere den samme fil, som parameteren keystoreFileName gør, forudsat at filen er en JKS fil og at dens filnavne suffix er ”. `jks`”.
nemid.applet.parameter.signing .password	Password til keystore filen, som givet ovenfor
nemid.applet.parameter.signing .alias	Alias, også kaldet *brugervenligt navn,* på det certifikat i keystore filen, refereret med ovenstående parameter.
nemid.applet.parameter.signing .keypassword	Password til den *Alias*, som er givet i overstående parameter.
nemid.pidservice .serviceprovideri d	Bruges ikke af LCES, men bør have en numerisk værdi
nemid.pidservice.environment	Værdien: OCESII_DANID_ENV_PREPROD eller OCESII_DANID_ENV_PROD

Parameter	Beskrivelse
nemid.environment	Værdien: OCESII_DANID_ENV_PREPROD eller OCESII_DANID_ENV_PROD
openoces.applet.server.url	For pp miljøet: https://opensign.pp.certifikat.dk/, for produktion: https://opensign.certifikat.dk/
openoces.applet.name	Skal sættes til OpenSign-bootstrapped.jar.
nemid.serviceprovider.logonto	Et navn, som arbejdsgiveren er tildelt af DanID til denne service.

Opdatering af LDAP felter uden for SecretStore

Programmet understøtter stempling af hændelser i visse felter i den LDAP, som er vært for *SecretStore*. Dette sker, hvis parameteren storeOcesFields er sat til `yes`. Felternes LDAP navne starter pr. default med OCES, f.eks. OCESCertificateStatus.

Man kan dog ændre LDAP navnene ved parametre i filen `LCESConfig.properties` efter disse regler, jvf. gennemgangen af parametre ovenfor: Hvis man ønsker et andet navn end default, giver man en parameter af default navnet. Dennes værdi angives så som det ønskede LDAP navn. F.eks.: Hvis parameteren OCESValidTo ikke er givet, bliver *ValidTo* værdien gemt under navnet `OCESValidTo`. Hvis parameteren er givet som her:

```
OCESValidTo=expirationDate
```

bliver *ValidTo* værdien gemt under navnet `expirationDate`.

I det følgende antages, at man bruger default-navnene.

1. OCESCertificateStatus: En tilstands-kode, som fortæller, hvor i flowet, den aktuelle bruger er. Værdierne gives i parameteren `storeOcesStateValues`.
2. OCESCertificateSerialNumber, som udfyldes med det serienummer, som DanID har givet som et ekstra felt i brugerens id (cn= værdien).
3. OCESValidFrom, som udfyldes med certifikatets *gælder fra* dato i formatet YYYY-MM-DD hh:mm:ss.
4. OCESValidTo, som udfyldes med certifikatets *gælder til* dato i formatet YYYY-MM-DD hh:mm:ss.

Dette er styret af de tre parametre `storeOcesFields`, `storeOcesStateValues` og `validStartOcesStateValues`. De er beskrevet kort i parameter-skemaet ovenfor. Hvis det første felt er sat til *yes,* explicit eller pr. default, sættes disse felter, og derfor skal det andet felt gives og det tredje felt kan gives.

Det andet felt angiver de tilstands-værdier, programmet vil sætte i LDAP feltet OCESCertificateStatus undervejs i flowet. De er komma-separerede, og der skal som nævnt i skemaet ovenfor have netop syv værdier. Det kan se sådan ud:

```
storeOcesStateValues=56, 57, 60, 61, 70, 71, 80
```

Bemærk, at værdien ”-1” i en position tolkes, som at den til positionen svarende tilstand ikke skal stemples i feltet OCESCertificateStatus.

Det tredie felt angiver den eller de statuskode-værdier, som feltet OCESCertificateStatus skal indeholde, for at programmet vil starte nyudstedelse af en signatur feltet bruges ikke ved fornyelse. Det kan se sådan ud:

```
validStartOcesStateValues=60,70
```

Hvis denne parameter ikke er givet, foretages intet opstarts-check.

System-validerings siden *adm* validerer disse parametre.

Placering af parameterfilen uden for web applikationen

Ændring her, nu er denne indirektion obligatorisk.

Konfigurationsfilen `LCESConfig.properties` ligger i applikationens distributionsfil og vil ved installation blive lagt i *`tomcat`*`/webapps/lces295/WEB-INF/classes`. Derved kan en tidligere lokalt tilrettet version blive overskrevet, hvilket ikke er ønskeligt, fordi der altid vil være en del lokale tilpasninger netop i denne fil. For at undgå dette, kan man, og fra LCES v.2.9.4 skal man, lægge properties filen et andet sted og sætte en Tomcat-variabel, som peger på det bibliotek, som filen er lagt i.

Placeingen skal angives i Tomcat filen `tomcat7/conf/context.xml`. Disse linier tilføjes i denne fil:

```
<Environment name="lces295/ParmPlace"
 value="/var/lib/tomcat7/common/lces295/LCESConfig.properties"
 type="java.lang.String"
 override="true" />
```

`name` værdien skal gives som vist, og `value` værdien er path til properties filen samt filens navn. Bemærk, at syntaksen af `name` værdien er lidt ændret i forhold til LCES v.2.9.0 og at value værdien fik tilføjet filnavnet ved LCES v.2.9.3. Bemærk: Filen kan have det navn, man ønsker, men delen efter punktummet skal være `properties`.

Hvis denne variabel ikke er defineret, eller hvis properties filen ikke kan findes på det give sted, forsøger programmet at finde den i Java classpath, dvs. i `webapps/lces295/WEB-INF/classes`.

Bemærk, at det er kun parameterfilen `LCESConfig.properties`, som kan lægges uden for Classpath på denne måde.

Applikationens status-side (URL: `-/lces295/adm`) viser fuld path på den anvendte parameterfil.

Opsætning af Log4j

LCES v.2.9.1 og frem bruger lognings-frameworket *log4j* til logning af programmers trace- og fejl-informationer samt til logning af forretningshændelser, så som: *nu beder vi om data fra DanID serveren* og: *nu er kaldet færdigt*; med indikation af, om kaldet gik godt eller skidt. Så vidt muligt indeholder sådanne log-linier en identifikation, så som bruger-id eller nøgleværdien for et *SecretStore* entry.

Der er disse led i at få logningen til at køre:

For det første: Der findes en Servlet, som sætter Log4j op for LCES applikationen ved Tomcat's opstart. Den ligger i dette normale udviklingsmiljø (og i den normale war-fil) og den hedder `dk.liga.cert.util.servlet.Log4jSetup`. Den indeholder bl.a. disse liner:

```
String prefix = getServletContext().getRealPath("/");
String file = getInitParameter("log4j-init-file");
```

Første linie skaffer applikationens path, f.eks: `/var/lib/tomcat7/webapps/lces295/`.

Anden linie referer navnet på en parameter, som skal være sat i web.xml filen, se nedenfor. Denne parameter skal give resten af path til Log4j's parameterfil, sådan at fuld path er prefix + file.

For det andet: applikationens web.xml fil skal indeholde denne del:

```
<servlet>
  <servlet-name>log4j-init</servlet-name>
  <servlet-class>dk.liga.cert.util.util.Log4jSetup</servlet-class>
  <init-param>
    <param-name>log4j-init-file</param-name>
    <param-value>WEB-INF/classes/log4j.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

De første to linier navngiver den servlet, som skal sætte logningen op for denne applikation. Det er forklaret ovenfor. <init-param> delen definerer værdien af den symbolske parameter log4j-init-file, som bruges af opstart-sevletten, som vist ovenfor. Bemærk <load-on-startup>1</..., som betyder, at Tomcat starter denne servlet under sin opstart.

På denne måde refereres Log4j's parameterfil som <prefix>/WEB-INF/classes/log4j.xml.

For det tredje: Log4jSetup servletten indeholder også disse to statements:

```
String logRoot = prefix + "WEB-INF/classes/logs";
System.setProperty("logRoot", logRoot);
```

De sætter systemvariablen logRoot til at pege på applikationens directory classes/logs, hvilket er default directoriet for systemets logning. De refereres i log4j.xml filen to steder således:

```
<appender name="fileTrace" class="org.apache.log4j.RollingFileAppender">
  <param name="append" value="true" />
  <param name="file" value="${logRoot}/LcesTrace.log"/>
```

Log4j.xml filen kan ændres lokalt, hvis et andet log-directory ønskes.

For det fjerde: Log4j parameterfilen indeholder en *category* for en trace-logger og én for en forretnings-logger. De referer hver sin *appender,* hvoraf én er vist her:

```
<appender name="fileTrace" class="org.apache.log4j.RollingFileAppender">
  <param name="append" value="true"/>
  <param name="file" value="${logRoot}/LcesTrace.log"/>
  <param name="MaxFileSize" value="1000KB" />
  <param name="MaxBackupIndex" value="5" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern"
        value="%d{yyyy-MM-dd HH:mm:ss} %5p %F: %m%n"/>
  </layout>
  <!-- levels: all < debug < info < warn < error < fatal < off -->
  <filter type="ThresholdFilter" level="info"/>
```

```
</appender>
```

Bemærk, at der her er tale om standard brug af Log4j, hvilket vil sige, at man kan trimme på `RollingFileAppender`'s parametre, på datoformatet, på de faste felter til venstre i hver log-linie og endda bruge en anden appender, f.eks. er der standard appendere til een-fil-pr-dag, til e-mails, socket, SQL, JMS, XML, HTML og en del mere. Appenders kan desuden filtrere på indhold og der kan være flere appenders til det, der fra applikationen opfattes som én logger (*category*). Alt dette kan sættes i filen `log4j.xml`. I øvrigt henvises til dokumentationen for Log4j.

Endelig, hvordan bruger applikationerne dette: Ved en statisk metode i Log4j's `Logger` klasse kan man få fat i en singleton logger, som svarer til en given *category* i log4j parameterfilen. Den kan en hvilken som helst klasse i applikationen så få fat i og skrive til. For at løsne koblingen mellem alle programmer og *category* tag'ets *name* værdi, bruger applikationen normalt en hjælper, en mini-factory-klasse, `dk.liga.cert.util.util.LoggerFactory`, som har en getter metode for hver af de tilsigtede loggere, én for programtrace og én for logning af forretningshændelser. Disse to metoder er statiske og hedder `getTraceLogger`, henholdsvis `getBusinessLogger`. De tager ingen parametre og returnerer en `org.apache.log4j.Logger` instans med de parametre, som er sat op i `log4j.xml` parameterfilen.

Et program kunne have dette statement i toppen:

```
private Logger loggerTrace = LoggerFactory.getTraceLogger();
```

og så logge rundt omkring med statements som detteher:

```
this.loggerTrace.trace("blah blah, secret name: " + secretName);
```

Husk den gamle regel om at gøre tidskrævende streng-konkateneringer betinget af, om loggeren overhovedet har tænkt sig at logge det niveau, man skriver til, så ovenstående statement burde nok se sådan ud:

```
if (this.loggerTrace.isTraceEnabled()) {
  this.loggerTrace.trace("blah blah, secret name: " + secretName);
}
```

Bemærkninger

Web-applikationens klient kan kun køre på en Windows maskine. Det er på grund af `CIC294.jar` funktionaliteten, som arbejder mod Windows' bruger-database og produktet `pki-roamer`, som er et Windows program.

LCES fra version 2.9.1 kan køre under Java version 7.

Fejlmeddelelser er i en blanding af engelsk og dansk. Der er flere grunde til det, bl.a.:

1. At visse Java Exceptions skrives i brugerens browser temmelig rå, så som *access denied*.
2. At fejlmeddelelser fra Nets DanId's server gives rå, og de er på engelsk.
3. At i hvert fald den nye DanID v.2 rettede del skriver sammenhængs-information i en fejlmeddelelse sammen med en rå fejlmeddelelse. Derved kan dansk og engelsk faktisk blive blandet i en fejlmeddelelse, i stil med *Første SOAP kald gav fejl: Access denied.*

Egentlige fejlmeddelelser, som er hard-kodede i web-applikationen, er (for det meste) på dansk.

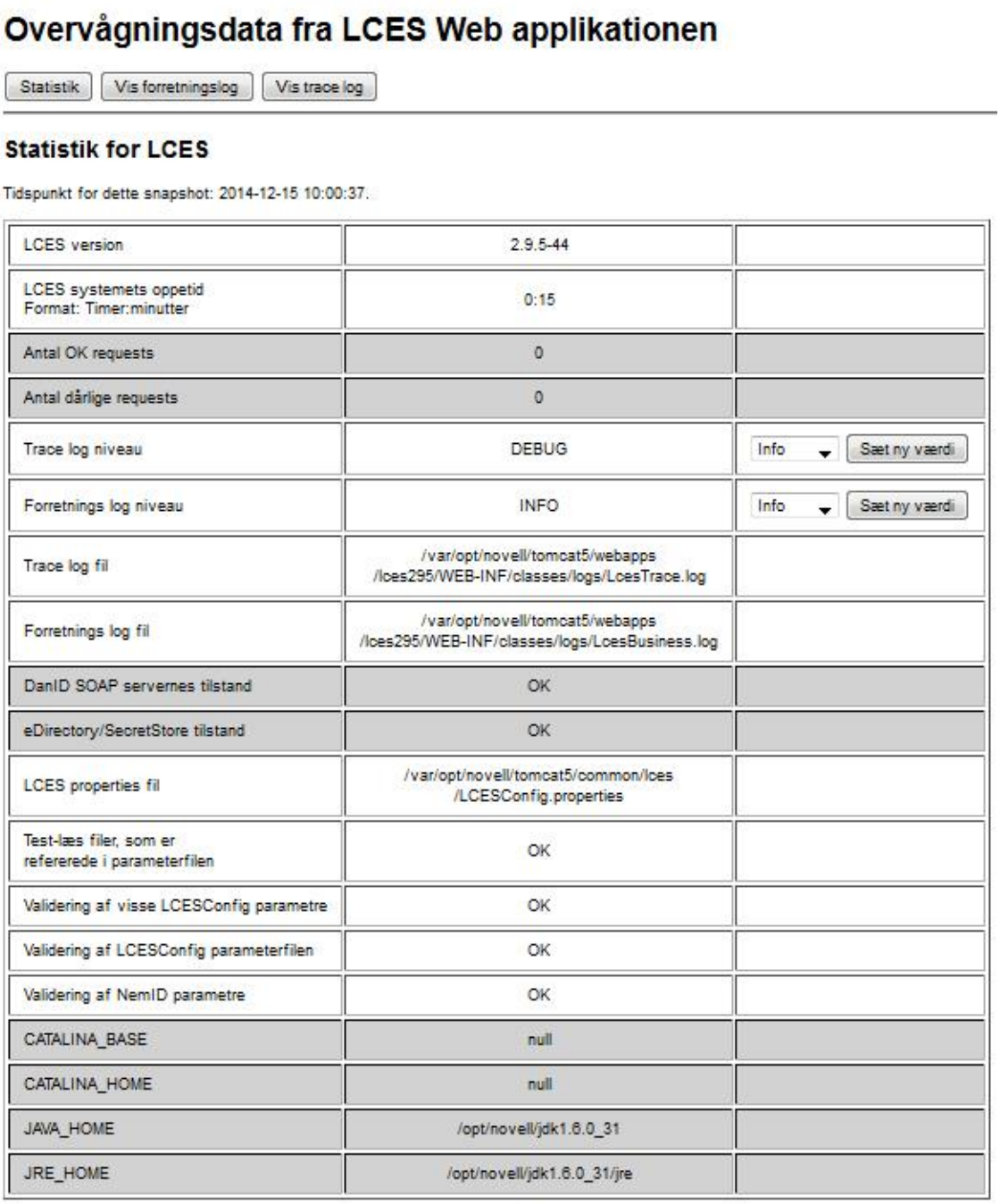
Applikationens administrator dialog

Foruden den brugerrettede web-applikation `lces295/` findes en lille administrator web-applikation, som nås ved adressen `lces295/adm/`. Denne side har tre dele, som hver kan nås ved at trykke på den relevante knap øverst på enhver af siderne: *Statistik, Vis forretningslog* og *Vis trace log*.

Bemærk, at alle sider er statiske og viser dermed situationen, som den var på visningstidspunktet. Brugeren må gen-aktivere en side ved at trykke på én af de tre knapper i toppen for at genaktivere en side. *Statistik* siden viser tidspunktet for sidste visning.

Statstik siden viser, foruden visse statistik-tal, også resultatet af visse checks, som alle skal vise OK for at nogen bruger kan bruge applikationen. Det er derfor afgørende, at man får denne side til at vise OK for disse checks, før brugerne slippes løs på selve web-applikationen.

Siden ser sådan ud:



Overvågningsdata fra LCES Web applikationen

Statistik | Vis forretningslog | Vis trace log

Statistik for LCES

Tidspunkt for dette snapshot: 2014-12-15 10:00:37.

LCES version	2.9.5-44	
LCES systemets oppetid Format: Timer:minutter	0:15	
Antal OK requests	0	
Antal dårlige requests	0	
Trace log niveau	DEBUG	Info Sæt ny værdi
Forretnings log niveau	INFO	Info Sæt ny værdi
Trace log fil	/var/opt/novell/tomcat5/webapps /lces295/WEB-INF/classes/logs/LcesTrace.log	
Forretnings log fil	/var/opt/novell/tomcat5/webapps /lces295/WEB-INF/classes/logs/LcesBusiness.log	
DanID SOAP servernes tilstand	OK	
eDirectory/SecretStore tilstand	OK	
LCES properties fil	/var/opt/novell/tomcat5/common/lces /LCESConfig.properties	
Test-læs filer, som er refererede i parameterfilen	OK	
Validering af visse LCESConfig parametre	OK	
Validering af LCESConfig parameterfilen	OK	
Validering af NemID parametre	OK	
CATALINA_BASE	null	
CATALINA_HOME	null	
JAVA_HOME	/opt/novell/jdk1.6.0_31	
JRE_HOME	/opt/novell/jdk1.6.0_31/jre	

Her er en beskrivelse af hver del:

Statistik siden

Denne side viser først nogle tal: LCES versionen (version-build nummer), oppetiden, antal OK gennemførte og antal fejlede DanID requests.

Dernæst vises log niveauerne for de to logfiler, som web-applikationen har, jf. afsnittet *Opsætning af Log4j.* I kolonne til højre for de viste log-niveauer kan man ændre log-niveauet for hver af loggerne. Bemærk, at en sådan ændring kun virker indtil genstart af web-applikationen, hvor Log4j's parameterfil bliver læst igen og taget til efterretning.

De næste fire rækker i tabellen er ikke statistik, men viser resultatet af visse checks for, om selve web-applikationen kan fungere. Der checkes for disse ting:

1. DanID SOAP server tilstand: Der checkes for, om denne applikation kan tilgå DanID's SOAP server ved at bruge de parametre, som selve web-applikationen bruger.
2. eDirectory/SecretStore tilstand: Der checkes for, om denne applikation kan tilgå eDirectory og SecretStore, hvor certifikaterne fra DanID skal gemmes. Også denne funktion baserer sig på de parametre, som selve web-applikationen bruger.
3. Test-læs filer, som er refererede i parameterfilen: Der checkes for, om filerne, som er refererede fra parameterfilen LCESConfig.properties, faktisk findes og kan læses. Der er tale om disse parametre: *TrustStore, html.lces.LCESWhoAmI.Header, html.lces.LCESWhoAmI.Footer, html.lces.LCESWhoAmIOtp.Header, html.lces.LCESWhoAmIOtp.Footer, html.lces.LCESWhoAmIRen.Header, html.lces.LCESWhoAmIRen.Footer, html.lces.LCESInstallUserCert.Header* og *html.lces.LCESInstallUserCert.Footer.*
4. Validering af visse LCESConfig parametre: Der checkes for, at alle obligatoriske parametre er givet. Visse obligatoriske parametre checkes dog ikke her, fordi de checkes i testen *Test-læs filer, som er refererede i parameterfilen.*
   Desuden checkes for, om parameteren `storeOcesFields` har værdien *yes*, dvs. at den er givet med værdien *yes* eller at den ikke er givet, hvorved den defaulter til y*es.* Hvis den på den ene eller anden måde har værdien *yes*, checkes for, om den da obligatoriske parameter `storeOcesStateValues` er givet og at den har netop syv komma-separerede værdier.
5. Validering af LCES parameterfilen. Denne validering checker for dobbelte parametre - samme nøgle er givet mere end én gang, hvilket det øvrige system ikke detekterer, det bruger blot den sidst givne værdi. Den checker desuden for ukendte parametre - en linie i parameterfilen med nøgle, som ikke er kendt i LCES, hvilket sandsynligvis er en stavefejl.
6. Validering af visse LCESConfig parametre, dedikeret til POCES funktionaliteten, dvs. parametre, som bruges til at sætte parametrene til *NemID* dialogen op. Der checkes for, at man kan tilgå én af parametrene, hvis navn starter med ”nemid.”. Derved indlæses alle parametre, og en eventuel fejl på dette tidspunkt (manglende mandatory parameter) vil blive meldt.
7. Desuden vises visse parametre, som ikke er sat i `LCESConfig.properties` filen: Path og navn på netop denne parameterfil samt værdierne af nogle systemvariable, som er vigtige for Tomcat miljøet: CATALINA_BASE, CATALINE_HOME, CATALINA_TMPDIR, JAVA_HOME og JRE_HOME. Det gælder for disse systemvariable, at de, som ikke har en værdi, bliver ikke vist på siden.

Vis forretningslog siden

Denne side viser de sidste 100 linier fra Log4j forretningsloggen. Bemærk, at nederste linie er den nyeste.

Vis trace log siden

Denne side viser de sidste 100 linier fra Log4j program-trace loggen. Bemærk, at nederste linie er den nyeste.

Simulator SOAP service for test af programmet

Der er lavet en Tomcat applikation, som fungerer som en simulator af nogle af DanID's SOAP services. Netop SOAP kaldene til udstedelse af ny signatur (i service termen: *Issuance*) samt til *SelfService's* kald `GetOrderDetailsByPocesSignatureAndReferenceNumberV300` kan sendes til denne simulator og den sender et hard kodet svar tilbage. Den kan sende OK og fejlsvar, afhængig af en bestemt værdi i et af request-felterne.

Denne simulator er en separat Tomcat applikation ved navn `netssimulator`. Hvor denne simulator en installeret, kan dens mini-dokumentation ses ved at lade en web-browser læse *`tomcat`*`/webapps/netssimulator/`.

Kontaktinformationer

Samara ApS

International House

Center Boulevard 5

2300 København S

Telefon 35 36 95 05

Mail: post@smartsignatur.dk

Web: www.smartsignatur.dk