HAIBA

LPR-kontakter, data behandler

Driftsvejledning

version 1.0

Indholdsfortegnelse

1 Formål 3

2 HAIBA LPR-databehandler 4

2.1 LPR-databehandler komponenter 4

2.1.1 LPR-databehandler 4

2.1.2 LPR database 4

2.1.3 Indlæggelses database 4

3 Opdatering til nye versioner 5

4 Daglig Drift 6

4.1 LPR Databehandler 6

4.1.1 Fremgangsmåde for indlæsning af nye data 6

4.1.2 Eksempel for import af data 6

4.1.3 Konfiguration af LPR Databehandler 6

4.2 Databaser 7

4.2.1 Skemafiler databasen 7

4.2.2 Databaseopsætning 7

5 Backup 8

6 Overvågning 9

6.1 Statusside 9

6.2 Logning 9

6.2.1 Fejlsøgning 9

7 Ændringslog 10

# Formål

Dokumentet er målrettet systemadministratorer og driftspersoner, som skal kunne håndtere driftsmæssige aspekter af komponenten.

Driftsvejledningen indeholder information om komponentens version, standard placering af logfiler og konfigurationsfiler, eksterne afhængigheder, og evt. krav til genstart af applikationen hvis komponenten bliver ikke-responsiv.

Kendte fejlkoder som skrives i logfiler er dokumenteret, så disse evt. kan overvåges, og tillige danne baggrund for fejlsøgning. En generel læsevejledning til logfiler vedlægges.

Det er angivet hvorledes komponenten bedst lader sig overvåge, dvs. en generisk beskrivelse af overvågningen, der ikke er værktøjsafhængig.

Evt. specielle krav til backup beskrives, ligesom procedure ved reetablering af komponenten ud fra backup er beskrevet.

# HAIBA LPR-databehandler

Dette dokument omfatter driften af HAIBA LPR-databehandleren

Listen herunder beskriver hver del komponent med type, evt. status URL og filnavne. Status URL’en kan løbende polles for at checke komponentens status.

## LPR-databehandler komponenter

### LPR-databehandler

LPR Databehandleren ligger i sin egen WAR fil, den har sin egen overvågnings URL, der enten fortæller om den er operationsdygtig (HTTP 200 OK), eller om der er fejl i den (HTTP 500 ERROR), Overvågningssiden vil give et bud på hvad fejlen er, dog bør man kigge i log-filen for at få alle detaljer med.

* Type: Batch
* Status Url: http://<hostname>:<port>/lpr-importer/status
* Filnavn: lpr-importer-<version ex. 1.0>.war

### LPR database

**Test:**

Databaseserver ligger på host: SRV-SQL-TEST01

Databasen hedder: HAIBA

Brugernavn: haiba\_app

Password: Indskrives ikke her

**Produktion:**

<TODO>

### Indlæggelses database

**Test:**

Databaseserver ligger på host: SRV-SQL-TEST01

Databasen hedder: HAIBA

Brugernavn: haiba\_app

Password: Indskrives ikke her

**Produktion:**

<TODO>

# Opdatering til nye versioner

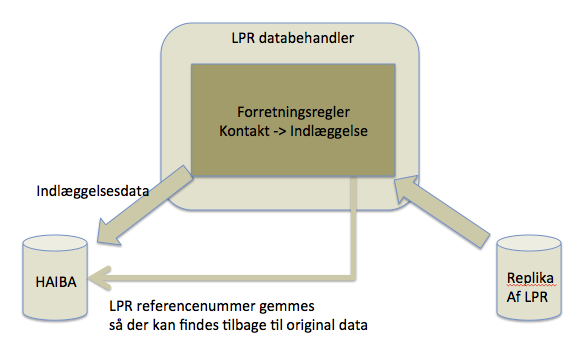
Når nye versioner af LPR Databehandleren udkommer, vil der medfølge release notes som forklarer database-migrering, rollback-procedure, service vinduer mv.

Til installation af første version, og eventuel geninstallation, af LPR Databehandleren henvises til installationsvejledningen.

# Daglig Drift

## LPR Databehandler

LPR databehandleren sørger for at importere data fra Landspatientregisteret (LPR), for så via nogle forretningsregler igen at eksportere data til HAIBA indlæggelsesdata.



LPR databehandleren består af en Tomcat applikationsserver indeholdende en war fil som er selve applikationen. Desuden er der eksterne afhængigheder til en LPR database og en HAIBA database

### Fremgangsmåde for indlæsning af nye data

LPR databehandleren står med jævne mellemrum og kigger i LPR databasen for at se om der er kommet nye data, dette interval kan sættes op via konfigurationsfilen. Opdager den nye data vil den indlæse data og køre det igennem forretningsreglerne. Skulle der ske en databehandlingsfejl, vil denne fejl blive skrevet ned i en fejl-kø (fil), som efterfølgende skal behandles af en dataansvarlig.

Bliver data succesfuldt behandlet, vil LPR databasen blive opdateret med et tidsstempel.

### Eksempel for import af data

<TODO>

### Konfiguration af LPR Databehandler

LPR databehandleren har en default konfigurationsfil (default-config.properties) der er indlejret i war filen, de enkelte konfigurationsindstillinger kan overstyres ved af en miljø specifik konfigurationsfil (config.properties), der ligger i filsystemet på følgende lokation: <Tomcat installationsdir>/shared\_config

**Properties**

|  |  |
| --- | --- |
| **jdbc.lprJNDIName** | JNDI navn for datasourcen til LPR, der er konfigureret i Tomcat serveren, default: java:comp/env/jdbc/LPRDB |
| **jdbc.haibaJNDIName** | JNDI navn for datasourcen til HAIBA, der er konfigureret i Tomcat serveren, default: java:comp/env/jdbc/HAIBADB |
| **jdbc.dialect** | Hvilken type database LPR-importeren bruger, værdier er "MySQL" eller "MSSQL" |
| **disable.database.errorlog** | parameter der fortæller om fejl fra forretningsregler også skal logges i databasen. default værdi er *false,* hvilket betyder at der også logges til databasen, sæt denne til *true* for at slå logningen til databasen fra (der logges dog stadig til fil uanset hvad.) |
| **hours.between.contacts.same.hospital** | Max. antallet af timer mellem LPR kontakter for samme person og fra samme hospital før det skal betragtes som en indlæggelse. Default værdien er 4 - **Bemærk:** ændres der på denne værdi, bør man genbehandle alt data. |
| **hours.between.contacts.different.hospital** | Max. antallet af timer mellem LPR kontakter for samme person og fra forskellige hospitaler før det skal betragtes som en indlæggelse. Default værdien er 10 - **Bemærk:** ændres der på denne værdi, bør man genbehandle alt data. |

## Databaser

LPR Databehandleren er testet på en MSSQL 2008 v2 og en MySQL 5.5 database, sidstnævnte er brugt til udvikling.

### Skemafiler databasen

SQL skemaer ligger på Github sammen med kildekoden, man skal altid referere til den gældende version af LPR-importeren:

<https://github.com/trifork/HAIBA-LPRimporter/tree/master/database>

Her ligger både skemaer for Microsoft SQLserver og for MySQL

### Databaseopsætning

<TODO, beskriv disse hvis der er nogen>

Følgende indstillinger bør sættes specielt i databasen:

|  |  |  |
| --- | --- | --- |
| **Indstilling** | **Værdi** | **Beskrivelse** |
|  |  |  |

# Backup

LPR databehandlere kan reetableres ud fra installations og driftsvejledning i tilfælde af nedbrud.

Man skal dog i tilfælde af ændringer til konfigurationsindstiller sørge for at de bliver noteret ned i dokumenterne således det er nemt at re-etablere systemet.

Data kan re-etableres ved at genbehandle LPR data, men da der er meget data der skal behandles kan dette tage lang tid <TODO, når der er lavet en måling give et estimat hvor lang tid det vil tage> derfor kan det være en god ide at tage backup af HAIBA databasen, sammen med det bør der også tages backup af LPR T\_ADM tabellen, da databehandlere indsætter et tidsstempel for hvornår data er behandlet.

<TODO, henvis til drifstafdelingens procedurer for backup af data.>

# Overvågning

## Statusside

LPR databehandleren har en status-side som periodisk kan kaldes for at tjekke om servicen kører.

Statussiden vil overvåge om komponenten kører, samt om der er forbindelse til databaserne, og så vidt muligt skrive hvad en eventuel fejl er.

Status siden fungerer over HTTP, og har følgende statuskoder:

|  |  |
| --- | --- |
| 200 | Alt er OK. |
| 404 | Siden ikke fundet, er serveren startet? |
| 500 | Der er opstået en fejl, og driften bør undersøge komponentens log for fejlmeddelelser. Kan fejlen ikke opklares simpelt, bør driften kontakte support. |

URL’s for status sider kan findes tidligere i dette dokument.

## Logning

<TODO, beskriv log opsætning, og hvordan den konfigureres>

### Fejlsøgning

Opstår der en fejlsituation i komponenten, skal driften undersøge logfilen for loghændelser på ERROR-niveau. F.eks. i tilfælde af at komponenten ikke kan forbinde til databasen. Visse andre fejl er ikke-kritiske. Det vil sige at komponenten kan forsætte med at fungere. De bliver også logget på ERROR-niveau da disse hændelser bør undersøges. Komponenten vil i så vid udstrækning som muligt forsøge at forsætte på trods af fejl.

<TODO beskriv typiske fejl som kan ses i logfilen>

# Fejlhåndtering fra forretningsregler

LPR databehandleren sørger for at data fra LPR bliver sendt igennem en række forretningsregler (disse er beskrevet i løsningsbeskrivelsen)m som sørger for at tilrette data så det passer ind i HAIBA indlæggelsesmodellen.

Men der kan være data som på den ene eller anden måde falder uden for disse regler, derfor er der lavet et fejlhåndteringsmodul det kan håndtere dette.

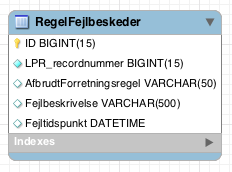
Fejlhåndteringen sørger for at logge disse fejl til både en tabel og til log-filer.

Overvågningsmodulet kigger ikke efter fejl i regelbehandlingen, det er en manuel proces at holde øje med dette.

det er ligeledes en manuel proces at finde ud af hvad der skal ske med det data som fejler i regelbehandlingen, f.eks. om det skyldes en fejl i kode eller en fejl i data.

Hvis der opstår en fejl som følge af databehandlingen, vil alle kontakter tilknyttet det CPR-nummer som fejlen opstod på ikke blive behandlet.

## Fejlbeskrivelse

Tabel layout er:

Log fil layout er:

<Datotidsstempel> [INFO] BusinessRulesErrors - LPR Recordnummer [?], Regel [?], fejlbesked [?]

Hvor de enkelte felter betyder følgende:

|  |  |
| --- | --- |
| **Felt** | **Beskrivelse** |
| LPR recordnummer | Referencen for kontakten i LPR |
| Regel | Forretningsreglen som fejlede på grund af data. |
| Fejl beskrivelse | Tekstuel beskrivelse af fejlen |
| Tidspunkt | Logningstidspunkt for fejlen |

Et eksempel på en fejl kunne være følgende:

2013-01-25 15:29:43,767 [INFO] BusinessRulesErrors - LPR Recordnummer [1234], Regel [LPR dato og tid regel], fejlbesked [Proceduredato findes ikke]

Eksemplet viser at LPR referencenummer 1234 ikke kan behandles af "Dato og Tid" reglen, da der er en tilknyttet procedure hvor proceduredatoen ikke findes.

### Logfiler og logtabel

Logfilen er konfigureret via log4j properties filen, der ligger i <Tomcat installationsdir>/appconfig

Log appenderen hedder "BusinessRulesErrors", det er denne appender LPR Databehandleren skriver fejlene til. Log appenderen er sat op som en rullende fil appender, så der er styr på hvor meget diskplads der maksimalt bruges, og den skriver til en fil der hedder "forretningsregel-fejl.log". hver gang den maksimale filstørrelse nås bliver der således lavet en ny fil " forretningsregel-fejl.log.1" o.s.v indtil det maksimale antal filer bliver nået, derefter slettes den ældste fil hver gang en ny skrives.

Default opsætning ses her:

log4j.logger.BusinessRulesErrors=INFO, ERRORFILE

log4j.appender.ERRORFILE=org.apache.log4j.RollingFileAppender

log4j.appender.ERRORFILE.File=forretningsregel-fejl.log

log4j.appender.ERRORFILE.Append=true

log4j.appender.ERRORFILE.MaxFileSize=100MB

log4j.appender.ERRORFILE.MaxBackupIndex=10

log4j.appender.ERRORFILE.layout=org.apache.log4j.PatternLayout

log4j.appender.ERRORFILE.layout.ConversionPattern=%d [%-2p] %c - %m%n

Logtabellen ligger i HAIBA databasen og hedder RegelFejlbeskeder, der bliver logget det samme i denne tabel som i logfilerne, men der er ikke noget der sikrer at der maksimalt logges eksempelvis 1 GB.

Derimod kan database logning slås til/fra ved at bruge "disable.database.errorlog" parameteren som beskrevet tidligere.

# Ændringslog

Kilden til dette dokument kan findes på:

<https://github.com/trifork/HAIBA-LPRimporter/blob/master/doc/Driftsvejledning.docx>

| Version | Dato | Ændring | Ansvarlig |
| --- | --- | --- | --- |
| 1.0 | 2013-01-03 | Initielt Dokument | Trifork  Kjeld Froberg |
|  |  |  |  |