HAIBA

FGR-importer

Drifts og installationsvejledning

Indholdsfortegnelse

1 Formål 3

2 FGR-dataimporter 4

2.1 Komponenter 4

2.1.1 FGR-dataimporter 4

2.1.2 HAIBA database 4

2.1.3 Applikationsserver 4

3 Installation 5

3.1 Krav til driftsmiljø 5

3.1.1 Krav til applikationsservere 5

3.1.2 Krav til operativsystem 5

3.1.3 Krav til database 5

3.1.4 Krav til hardware 5

3.2 Installation af FGR-dataimporter 5

3.2.1 Installation af Tomcat Applikationsserver 5

3.2.2 Konfiguration af tomcat adgang til database 6

3.2.3 Konfiguration af FGR-dataimporter 7

3.2.4 Installation af MS SQLServer 7

3.2.5 Installation af HAIBA database 7

3.2.6 Installation af WAR fil 7

3.3 Opdatering til nye versioner 7

4 Daglig Drift 8

4.1 FGR-dataimporter 8

4.1.1 Fremgangsmåde for indlæsning af nye data 8

4.1.2 Konfiguration af FGR-dataimporter 8

4.2 Databaser 9

4.2.1 Skemafiler databasen 9

4.2.2 Databaseopsætning 9

5 Backup 10

6 Overvågning 11

6.1 Statusside 11

6.2 Logning 12

6.2.1 Fejlsøgning 12

7 Ændringslog 13

# Formål

Vejledning til drift, installation og konfiguration af FGR-dataimporter.

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

Drifts og installationsvejledningen 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. Ligeledes indeholder vejledningen en installationsvejledning, som bruges ved inintiel deployment, samt afsnit til efterfølgende delvis installation, eksempelvis når der sker opdateringer.

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.

# FGR-dataimporter

Dette dokument omfatter drift og installation af HAIBA FGR-dataimporteren

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

## Komponenter

### FGR-dataimporter

Importeren 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>/fgr-importer/status
* Filnavn: fgr-importer-<version ex. 1.0>.war

### HAIBA database

**Test:**

Databaseserver ligger på host: SRV-SQL-TEST01

Databasen hedder: HAIBA

Brugernavn: haiba\_app

Password: Indskrives ikke her

**Produktion:**

Databaseserver ligger på host: SQLSERVER02

Databasen hedder: HAIBA

Brugernavn: haiba\_app

Password: Indskrives ikke her

### Applikationsserver

**Test:**

Ligger på serveren: SRV-HAIBA-TEST

Port: 8080

**Produktion:**

Ligger på serveren: SRV-HAIBA-PROD

Port: 8080

# Installation

## Krav til driftsmiljø

### Krav til applikationsservere

FGR-dataimporteren er udviklet og testet på en TOMCAT 7 version 35 (64 bit).

Komponenterne kræver Oracle Java 6.0 eller højere.

Desuden skal MSSQL eller MySQL JDBC Connector Driver bruges.

### Krav til operativsystem

Der stilles ingen krav til operativsystemet, ud over det åbenlyse krav om at Java er understøttet på operativsystemet. Det forventes dog at operativsystemet er Microsoft Windows Server 2003 med servicepack 2 installeret

### Krav til database

Komponenten er testet mod MS SQLServer 2008 Og skal som minimum køre mod denne version

### Krav til hardware

Der er nogle minimumskrav for at kunne afvikle komponenten fornuftigt til test formål. Dog skal man forvente at bruge high-end hardware (både cpu, ram, netkort & diske).

Minimumskravene, for fornuftig performance på et test-setup er:

* Intel Core 2 eller lignende CPU
* 8 GB ram
* Nødvendig harddisk plads for at kunne håndtere HAIBA databasen (100+ GB)

## Installation af FGR-dataimporter

Importeren består af følgende komponenter:

1. Tomcat Applikationsserver
2. MS SQLServer
3. HAIBA database
4. EPI-MiBA data importer war fil

### Installation af Tomcat Applikationsserver

Tomcat kræver Java for at køre. så først skal den installeres.

Hent Java 6 version 38 (64 bit version, da det skal installeres på en Windows Server 2008), på følgende link: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

bemærk det skal være JDK og ikke JRE.

Log ind som administrator og installer Java

Da det er en 64 bit version af windows og der skal køres java på den, mangler der sikkert en version af Visual C runtime biblioteker. Disse kan hentes på følgende link: <http://www.microsoft.com/en-us/download/details.aspx?id=5555> og skal ligeledes installeres.

Hent Tomcat 7 version 35 (64 bit version) på følgende link:

<http://tomcat.apache.org/download-70.cgi>

Log ind som administrator og installer tomcat.

Når Tomcat spørger hvilken port den skal køre på, så se om 8080 er brugt v.h.a "netstat -a" kommandoen, alternativt vælg en anden port (eksempelvis 9090)

Vælg at installere tomcat som en windows service.

Test at tomcat er kørende ved at spørge på <http://localhost:8080> i en browser (eller brug alternativ port hvis 8080 var optaget).

For at Tomcat kan kommunikere med en Microsoft SQL server, skal der installeres en driver til dette. Den kan hentes på følgende link: <http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774>

Pak installationsprogrammet ud og tag filen sqljdbc4.jar og kopier den til <tomcat installation>/lib Tomcat skal efterfølgende genstartes. Bemærk dette behøves ikke hvis MySQL bruges som database

For at Tomcat kan kommunikere med MySQL, skal der installeres en driver til dette. Den kan hentes på følgende link: <http://dev.mysql.com/downloads/connector/j/>

Tag filen mysql-connector-java-5.1.22-bin.jar og kopier den til <tomcat installation>/lib Tomcat skal efterfølgende genstartes. Bemærk dette behøves ikke hvis Microsoft SQL Server bruges som database

### Konfiguration af tomcat adgang til database

Datasource til HAIBA databasen:

Indsæt reference til datakilden i Tomcats context <tomcat installation>/conf/context.xml Dette gøres ved at tilpasse følgende:

Microsoft SQL:

<Resource name="jdbc/HAIBADB" auth="Container" type="javax.sql.DataSource" maxActive="5" maxIdle="5" maxWait="10000" username="<bruger>" password="<password>" driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver" url=" jdbc:sqlserver://localhost:1433;databaseName=HAIBA"/>

MySQL:

<Resource name="jdbc/HAIBADB" auth="Container" type="javax.sql.DataSource" maxActive="5" maxIdle="5" maxWait="10000" username="<bruger>" password="<password>" driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/HAIBA"/>

Indsæt databasernes brugernavn ("username") og adgangskode ("password") i ovenstående ressourcer.

### Konfiguration af FGR-dataimporter

Importeren har indbygget nogle default konfigurationsindstillinger, disse kan dog overstyres så de passer ind i det miljø den er installeret på ved hjælp af filen config.properties.

* Opret en folder der hedder "shared\_config" under <tomcat installationsdir>
* Åbn filen <tomcat installationsdir>/catalina.properties
  + Find hvor der står "shared loader" og tilføj så der står følgende:

shared.loader=${catalina.base}/shared\_config

* Kopier filen "fgrconfig.properties" til shared\_config folderen
* Genstart tomcat.

### Installation af MS SQLServer

Microsoft SQL server er præ-installeret på SSI's miljø, så denne vejledning dækker ikke dette.

### Installation af HAIBA database

**Test og Produktion:**

Skema til HAIBA databasen ligger på følgende link:

<https://github.com/trifork/HAIBA-FGRimporter/tree/master/database> der ligger både en version til MySQL og til MSSQL

### Installation af WAR fil

Importer applikationen ligger i et JEE web arkiv (war fil), for at installere den i tomcat serveren skal den kopieres til <Tomcat installationsdir>/webapps -efterfølgende vil tomcat selv sørge for at klargøre (deploye) filen.

<TODO hvor skal war filen ligges hen? - CD, filsystem eller lign?>

## Opdatering til nye versioner

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

# Daglig Drift

## FGR-dataimporter

Importeren sørger for at importere filer fra ssi, for så at eksportere data til HAIBA databasen.

FGR-dataimporteren består af en Tomcat applikationsserver indeholdende en war fil som er selve applikationen. Desuden er der eksterne afhængigheder til en HAIBA database og SSI's hjemmeside hvor SKS klassifikationer ligger i filformat.

### Fremgangsmåde for indlæsning af nye data

Importeren kører periodisk (konfigurerbart pr. type).

Først hentes kildefilen. Hvis dette går godt, parses data. Herefter tømmes den tidligere importerede database og nye data lægges ind.

### Konfiguration af FGR-dataimporter

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

**Properties**

|  |  |
| --- | --- |
| **cron.shak.import.job, cron.sks.import.job, cron.sor.import.job** | Spring scheduler cron konfiguration. Default ”0 0 0 \* \* \*” hvilket svarer til en kørsel kl 0000 hver morgen. Se eks. http://javahunter.wordpress.com/2011/05/05/cronscheduler-in-spring/ |
| **jdbc.dialect** | Hvilken type database HAIBA-importeren bruger, værdier er "MySQL" eller "MSSQL" |
| **jdbc.haibaJNDIName** | JNDI navn for datasourcen til HAIBA, der er konfigureret i Tomcat serveren, default: java:comp/env/jdbc/HAIBADB |
| **jdbc.haibatableprefix** | Prefix til Organisation- og GenericSKSLine-tabellerne til at angive mssql-skemaer |
| **max.days.between.runs** | Antal dage mellem kørsler. Hvis perioden siden sidste kørsel overskrider denne værdi, angives det på statussiden. Default 1 |
| **shak.remoteurl** | URL til shakfil, default http://filer.sst.dk/sor/data/shak/shakcomplete/shakcomplete.txt |
| **sks.remoteurl** | URL til sksfil, default http://filer.sst.dk/sor/data/sor/sorxml/v\_2\_0\_0/Sor.zip |
| **sor.filenameinziptoparse** | Den fil i sor.zip, der skal pakkes ud og parses, default Sor.xml |
| **sor.remoteurl** | URL til sorfil, default ftp://filer.sst.dk/filer/sks/data/skscomplete/SKScomplete.txt |

## Databaser

Importeren 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 FGR-importeren:

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

Her ligger både skemaer for Microsoft SQLserver og for MySQL

### Databaseopsætning

der er ingen specielle indstillinger til databasen.

# Backup

FGR-dataimporteren 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 nulstille transaktions-id'erne i HAIBA databasen <TODO, beskriv dette nåt det bliver udviklet>, men da der er meget data der skal behandles kan dette tage lang tid derfor kan det være en god ide at tage backup af HAIBA databasen.

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

# Overvågning

## Statusside

FGR-dataimporteren 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.

Eksempel på statusvisning (efter manual kørsel) kan ses her:

------------------

shak

------------------

Last import started at: 2014-02-07T13:39:25.727+01:00 and ended at: 2014-02-07T13:43:10.180+01:00. Processing took 224 seconds. Outcome was SUCCESS

------------------

Manual start importer

Scheduled start importer

status: MANUAL

------------------

sks

------------------

Last import started at: 2014-02-07T13:39:19.957+01:00 and ended at: 2014-02-07T13:56:41.077+01:00. Processing took 1041 seconds. Outcome was SUCCESS

------------------

Manual start importer

Scheduled start importer

status: MANUAL

------------------

sor

------------------

Last import started at: 2014-02-07T13:39:08.787+01:00 and ended at: 2014-02-07T13:40:35.677+01:00. Processing took 86 seconds. Outcome was SUCCESS

------------------

Manual start importer

Scheduled start importer

status: MANUAL

## Logning

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

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 "fgrdebug.log". hver gang den maksimale filstørrelse nås bliver der således lavet en ny fil " haibaepimiba.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.dk.nsi.haiba.fgrimporter.importer.ImportExecutor=DEBUG, DEBUGFILE

log4j.logger.dk.nsi.haiba.fgrimporter=DEBUG, DEBUGFILE

log4j.additivity.dk.nsi.haiba.fgrimporter.importer.ImportExecutor=false

log4j.additivity.dk.nsi.haiba.fgrimporter.dao.impl=false

log4j.appender.DEBUGFILE=org.apache.log4j.RollingFileAppender

log4j.appender.DEBUGFILE.File=fgrdebug.log

log4j.appender.DEBUGFILE.Append=true

log4j.appender.DEBUGFILE.MaxFileSize=50MB

log4j.appender.DEBUGFILE.MaxBackupIndex=10

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

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

### 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.

# Ændringslog

Kilden til dette dokument kan findes på:

[https://github.com/trifork/HAIBA-FGRimporter/blob/master/doc/Drifts og installationsvejledning.docx](https://github.com/trifork/HAIBA-FGRimporter/blob/master/doc/Drifts%20og%20installationsvejledning.docx)

| Version | Dato | Ændring | Ansvarlig |
| --- | --- | --- | --- |
| 1.0 | 2013-02-13 | Initielt Dokument | Trifork  Aksel Schmidt |
|  |  |  |  |