Stamdata

Driftsvejledning

Indholdsfortegnelse

1 Formål 3

2 Omfattede komponenter 4

2.1 Stamdata importer komponenter 4

2.1.1 Stamdata Data Importere 4

3 Opdatering til nye versioner 5

4 Daglig Drift 6

4.1 Stamdata Importere 6

4.1.1 Fremgangsmåde for indlæsning af nye data 6

4.1.2 Eksempel for import af nye CPR-data 6

4.1.3 Konfiguration af Stamdata importere 7

4.2 Database 7

4.2.1 Skema til stamdata importere 7

4.2.2 Databaseopsætning 8

5 Backup 9

5.1 Backup af Data Manager input-filer 9

6 Overvågning 10

6.1 Statussider 10

6.1.1 Speciel overvågning af SOR og SOR-Relationer importerne 10

6.2 Logning 10

6.2.1 SLA-Log punkter 11

6.2.2 Fejlsøgning 11

7 Liste af Registre 12

8 Ændringslog 13

# Formål

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

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

Start/stop vejledning for komponenten beskrives, herunder hvilke andre applikationer der evt. skal genstartes.

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

Det bør angives 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 beskrives.

# Omfattede komponenter

Dette dokument omfatter driften af alle Stamdata importer komponenterne

Listen herunder beskriver hver komponent med type status URL og navnet på filen som skal deployes. Status URL’en kan løbende polles for at checke komponentens status. Status sider er beskrevet mere detaljeret senere i dokumentet.

## Stamdata importer komponenter

### Stamdata Data Importere

Hver stamdata importer ligger i sin egen WAR fil, hver importer har sin egen overvågnings URL, der enten fortæller om den enkelte importer er operationsdygtig (HTTP 200 OK), eller om der er fejl i importeren (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>/<komponent-navn>/
* Filnavn: <komponent>.war

# Opdatering til nye versioner

Når nye versioner af Stamdata komponenterne udkommer, vil der medfølge release notes som forklarer database-migrering, rollback-procedure, service vinduer mv. Til installation af første version af stamdata komponenterne henvises til installationsguiden.

# Daglig Drift

## Stamdata Importere

Stamdataimporterne er en gruppe filparsere og batch jobs som indlæser og vedligeholder data fra forskellige registre og gemmer dem i en MySQL database.

Hver importer har sin egen inbox-mappe som automatisk oprettes når servicen startes. Roden for disse inboxmapper er:

<JBOSS\_HOME>/domain/data/sdm4/

Komponenten kigger i sin inboxmappe for at se om der kommer nye filer til import. Det er driftens opgave at placere filer i inboxmapperne når tiden er inde for en opdatering. Hvilke filer der skal bruges og hvor ofte registrene skal opdateres er beskrevet i slutningen af dokumentet. Hver inbox er logisk navngivet efter den tilhørende importer.

Se afsnittet om overvågning for monitorering af servicen.

### Fremgangsmåde for indlæsning af nye data

Parserne forventer data at blive lagt i undermapper af deres rod-mappe, f.eks.:

<JBOSS\_HOME>/domain/data/sdm4/<importer>/20120822T201121S231/<file.txt>

<JBOSS\_HOME>/domain/data/sdm4/<importer>/20120822T201121S231/<file2.txt>

Undermappernes navne er underordnede. De importeres i leksikografisk orden,

og det vil derfor være oplagt at lægge dem så undermappernes navne er tids-

stempler som vist i eksemplet.

Skulle der ske en fejl under import, vil der blive lagt en fil ved navn

"LOCKED" i parserens inbox, f.eks.:

<JBOSS\_HOME>/domain/data/sdm4/<importer>/LOCKED

Parseren vil ikke forsætte før denne fil er slettet manuelt. Fejl-beskeden kan findes i loggen.

### Eksempel for import af nye CPR-data

Her en et eksempel på strukturen for en parsers filsystem. I dette tilfælde er det CPR-parseren:

<JBOSS\_HOME>/domain/data/sdm4/cprimporter/

Placeres filen ’D100312.L431101’ (som er en typisk CPR Person fil) i inputmappen begynder parseren at stabilisere data. Man kan gå ind på komponentens monitoreringsside for at se status for import:

http://<hostname>:<port>/cprimporter/status

Her vil man kunne se om CPRimporteren kører, hvornår CPRimporteren sidst har kørt, og hvilken status den har.

### Konfiguration af Stamdata importere

Hver importer har en default konfigurationsfil (default-config.properties) der er indlejret i war filen, de eneklte konfigurationsindstillinger kan overstyres ved af en miljø specifik konfigurationsfil (config.properties), der ligger i filsystemet på følgende lokation:

<JBOSS\_HOME>/modules/sdm4/config/<komponent>/main/config.properties

**Properties**

|  |  |
| --- | --- |
| **jdbc.JNDIName** | JNDI navn for datasourcen, der er konfigureret i JBOSS serveren, default: java:/SDMDS |
| **sdm.dataDir** | Folder navn hvor importeren kan finde datafilerne default: /pack/jboss/domain/data/sdm4 |
| **spooler.max.days.between.runs** | Parameter til overvågning af om en importer har modtaget data indenfor et forventet interval. Hvis denne frist overskrides vil overvågningen rapportere fejl. |

|  |  |
| --- | --- |
| **dk.nsi.dgws.sosi.dgwsLevels** | Kommasepararet liste af DGWS niveauer som kan bruges når man tilgår denne service. F.eks **dk.nsi.dgws.sosi.dgwsLevels=3,4**  Default: **3** |

## Database

De enkelte stamdata importere er testet på en MySQL 5.5 database.

### Skema til stamdata importere

Skema til stamdata importere opdateres automatisk af de enkelte importere, via det indbyggede migrerings framework.

Det skal dog understreges at så lang tid de enkelte NSP services er direkte afhængige af tabel layoutet for stamdata importerne, og der samtidig skal være bagud kompatibilitet vil tabel layoutet ikke ændre sige for stamdata importerne – men kun være splittet ud i importer war filerne.

### Databaseopsætning

Følgende indstillinger bør sættes specielt i databasen, disse indstillinger konfigureres i mysql’s konfigurationsfil.

|  |  |  |
| --- | --- | --- |
| **Indstilling** | **Værdi** | **Beskrivelse** |
| Max\_allowed\_packet | 16M | CPR opslagsservices kan returnere en pæn mængde data, derfor er max\_allowed\_packet default værdien på 1M for lidt - det anbefales at sætte den til 16M. |

# Backup

Alle tabeller i stamdatas database skal have daglig backup. Backup må ikke gemmes længere end 2 år pga. lovkrav.

Da der er tale om store mængder data er det vigtigt at holde for øje at det kan tage meget lang tid at genetablere et database image for stamdata.

## Backup af Data Manager input-filer

Alle filer fra dataset som lægges ind i Stamdata indbakker skal der laves backup af. Det vil sige at hver gang der f.eks. ankommer en ny fil fra CPR-registeret skal filen ligges i den pågældende parsers indbakke *og* samtidigt arkiveres. Hvordan filen arkiveres er op til driften, men filnavnet og dato for ankomst skal kunne spores.

# Overvågning

## Statussider

For hver komponent er der en status-side som periodisk kan kaldes for at tjekke om servicen kører. Hvis en service ikke kan overvåges via en simple status side vil det fremgå af dens driftsdokumentation.

Status sider fungerer over HTTP, og har følgende statuskoder:

|  |  |
| --- | --- |
| 200 | Alt er OK. |
| 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.

### Speciel overvågning af SOR og SOR-Relationer importerne

Da SOR behandles af to forskellige importere, er det et problem for datakonsistensen, hvis den ene af disse to parsere fejler på et datasæt som den anden parser ikke fejler på.

Skulle en af de to parsere afvise en import, er det en hastesag at få rettet den fejl der resulterer i afvisningen i den anden parser, så de to registre kan komme i sync så hurtigt som muligt.

## Logning

Stamdata importerne på DoDi’en bruger ikke JBOSS logging, men her hver deres egen log konfiguration som placeres følgende:

<JBOSS\_HOME>/modules/sdm4/config/<Importer navn>/main/

Her ligges en log4j.properties, der beskriver hvorledes importeren skal logge, samt konfiguration for SLA logging for netop denne komponent

### SLA-Log punkter

For at sikre en struktureret logning på tværs af de forskellige services, benyttes nsp-util også til SLA-Logning fra sdm4 importerne.

Der logges fra 2 logpunkter pr. importer ét punkt fra sdm4-core og ét fra selve importeren, sdm4-core er delt for alle importere og der dannes én linje for hver gang en importer startes.

En log fra sdm4-core ser ud som nedenfor:

03-06-2013 13:34:46,752 [pool-10-thread-1] INFO dk.sdsd.nsp.slalogdata - LogPoint="<importernavn>.ParserExecutor" LogPointUniqId="SDM4CORE\_ENTRY" StartTime="2013-06-03 13:34:45.650" EndTime="2013-06-03 13:34:46.752" Duration="1101893 microseconds" MessageId="vitaminimporter-1370259285650" RequestSize=0 ReplySize=0 Result=OK ClientIP="<empty>" SOAPOperation="<empty>" SOAPEndpoint="<empty>" SOAPAction="<empty>" TargetSOAPOperation="<empty>" TargetSOAPEndpoint="<empty>" GenericCallParms(0)= { }

Fra de enkelte importere logges følgene pr datasæt:

03-06-2013 13:34:46,750 [pool-10-thread-1] INFO dk.sdsd.nsp.slalogdata - LogPoint="<importernavn>.process" LogPointUniqId="SDM4. <importernavn>.process" StartTime="2013-06-03 13:34:45.782" EndTime="2013-06-03 13:34:46.750" Duration="967593 microseconds" MessageId="vitaminimporter-1370259285650" RequestSize=0 ReplySize=0 Result=OK ClientIP="<empty>" SOAPOperation="<empty>" SOAPEndpoint="<empty>" SOAPAction="<empty>" TargetSOAPOperation="<empty>" TargetSOAPEndpoint="<empty>" GenericCallParms(2)= { ("input","/pack/jboss/domain/data/sdm4/vitaminimporter/vitaminer") , ("processed\_records","634") }

GenericCallParms indeholder altid ”input” som peger på input folderen samt ”processed\_records” som indeholder antal indsatte eller ændrede records i databasen.

MessageID indeholder en unik id for kørslen således man kan kæde sdm4-core loggen sdm4-importeren sammen pr. kørsel.

### Fejlsøgning

Opstår der en fejlsituation i en komponent, skal driften undersøge den pågældende komponents logfil 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 der hændelsen bør undersøges. Komponenterne vil i så vid udstrækning som muligt forsøge at forsætte på trods af fejl.

Anvendes Splunk til indeksere logfiler bør alle de konfigurerede filer indekseres. Der kan opsættes alarmer i Splunk som aktiveres hvis en hændelse med ERROR-niveau logges. Dette niveau anvendes udelukkende ved alvorlige fejl. Der udover er også hændelser på WARN-niveau interessante da de f.eks. fortæller om folk forsøger at tilgå servicen uden tilladelse ol.

# Liste af Registre

Hvert register har sin egen registerspecifikations fil som ligger i register mappen sammen med dokumentationen.

# Ændringslog

Kilden til dette dokument kan findes på:

<https://github.com/trifork/sdm4-core/blob/master/doc/Driftsvejledning.docx>

| Version | Dato | Ændring | Ansvarlig |
| --- | --- | --- | --- |
| 1.0 | 2011-04-28 | Initielt Dokument | Trifork |
| 1.1 | 2011-09-12 | Opdateret med CPR services | Trifork |
| 1.2 | 2012-06-18 | Tilføjet informationer om schema opdateringer, samt mysql settings og v2 parsere | Trifork |
| 1.3 | 2012-08-22 | Opdateret med opsplitning af stamdata importere i moduler, samt JBOSS 7 konfiguration på DoDi | Trifork |
| 1.4 | 2012-08-24 | Fjernet al dokumentation der ikke er Importer specifik | Trifork |
| 1.5 | 2013-06-03 | Tilføjet afsnit om sla-log | Trifork |