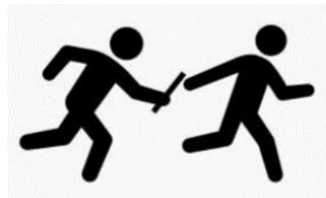




Prenumerationsrutinen

Information & Guidelines



Kontaktpersoner:

miriam.niskanen@vgregion.se

tuba.lundin@vgregion.se

michael.muller@vgregion.se

niklas.hendriksson@vgregion.se

Innehåll

Introduktion	3
Kortfattad information om vad som kan hanteras i PrenumerationsRutinen	3
Information om dokumentationen	3
Jobb som styr prenumerationsrutinen.....	4
1. Övervakning av prenumerationsrutinen	4
Statusmail.....	4
Datamodell övervakningsrapport för agentjobb och flöden	5
Övervaka agentjobb och flöden	5
Extra viktiga jobb att ha koll på	6
Nedan jobb bör köras minst var tredje dag. Så det är okej att jobbet misslyckas två dagar i rad men om den även misslyckas på den tredje dagen behöver det åtgärdas. Se vidare under rubriken Kontrolljobb för att se mer detaljerad info	6
Kontrolljobb.....	7
Vem ansvarar för källan eller prenumeranten.....	7
Kontrollera beroenden mellan jobb	7
Information om körningar	8
Kontrollera att klockslag stämmer överens för alla jobb i en AgentjobbGrupp	8
Kontrollera att cache- & remotetabeller uppdaterats korrekt	8
Kontrollera PBIRS Datarefresh	9
2. Administrera prenumerationsrutinen	9
Datamodell PrenumerationsRutin	10
Skapa ett jobb.....	10
Skapa ett agentjobb	10
Loggsteg.....	10
Skapa ett jobb som uppdaterar cache- eller remotetabeller i Denodo	10
Skapa ett jobs om uppdaterar statistik i Denodo.....	12
Lägga in ett agentjobb i prenumerationsrutinen	12
Jobb med flera beroenden	13
När prenumeranten inte ska starta direkt efter källan	13
När enbart prenumeranten ska startas via prenumerationsrutinen	13
Prenumeration.LaddInfo – Beskrivning av kolumner.....	14
Schemalägga PBIRS Refresh via prenumerationsrutinen	15
Jobb som alltid startas manuellt	15
3. Underhålla prenumerationsrutinen	16
4. Åtgärda prenumerationsrutinen	16

Låsningar på servern	16
Stoppa jobb manuellt	18
!!_Prenumeration_ETL_SjukhusEpisoder_Ssis_Daglig_Körning.....	19
!!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE.....	20
Starta om jobb som uppdaterar cache- eller remotetabeller	20

Introduktion

Prenumerationsrutinen används för att schemalägga agentjobb så att de startas i rätt ordning och vid rätt tidpunkt så att databaser/lösningar uppdateras korrekt. Ett flöde/jobb kan ha beroenden till flera andra jobb och invänta samtliga dessa jobb och omvänt kan ett flöde/jobb starta flera efterföljande flöden/jobb om detta är önskvärt. Det går även att schemalägga att PBI-rapporter så att dessa uppdateras så snart källan är uppdaterad.

Prenumerationsrutinen styrs av tabellen **Prenumeration.Laddinfo** som ligger i Orkestrering_DW på GRAL-PROD-IS01.

Vid frågor kontakta någon av oss som står med på första sidan så hjälper vi såklart till.

Kortfattad information om vad som kan hanteras i PrenumerationsRutinen

- Starta/uppdatera flöden av typen SQL-objekt, t.ex. anrop av Stored Procedures etc.
- Starta/uppdatera SSIS-flöden
- Starta/uppdatera SQL-Tabular
- Starta/uppdatera SQL-OLAP
- Starta/uppdatera PBIRS-Rapporter:
 - o Att låta PrenRutinen hantera start av schemalagring av PBIRS Refresh är bra dels p.g.a. att rapporten/rapporterna då blir uppdaterade vid rätt tillfälle. Samt att man undviker på detta sätt att användarna skapar upp flera PBIRS schemalagringar för samma rapport som körs flertalet gånger per dygn och belastar miljöerna i onödan p.g.a. osäkerheten kring när källan är uppdaterad.
- Starta/uppdatera Denodo-CACHE:
 - o Varianter: FULL Refresh med Truncate alt. FULL Refresh utan Truncate.
 - o Hantering av Inkrementell laddning ska implementeras för att endast hämta de senaste förändringarna. Vi vill undvika full-cache där det är möjligt.
- Starta/uppdatera Denodo-Remote
- Starta/uppdatera Denodo-Materialized table
- Starta/uppdatera Denodo-Update Statistics

Information om dokumentationen

Dokumentet består av 4 delar.

Den första delen beskriver hur och vilka verktyg som används för att övervaka prenumerationsrutinen.

Del 2 förklarar hur prenumerationsrutinen administreras och innehåller information om hur agent-jobb läggs till och schemaläggs i prenumerationsrutinen. Här finns även mallar och beskrivning av tabeller, kolumner och Stored Procedures som används för att administrera prenumerationsrutinen.

I del 3 finns instruktioner om hur ändringar i prenumerationsrutinen ska göras. T.ex. ändra en schemaläggning, ta bort en schemaläggning med mera.

Den 4de och sista delen innehåller vanliga fel och hur de åtgärdas. Vissa av jobben har återkommande problem och/eller problem som är specifika för det jobbet. Även information om det finns här.

Jobb som styr prenumerationsrutinen

Jobb som börjar med !!!_xxx är de jobb som styr och hanterar prenumerationsrutinen. Vissa jobb körs kontinuerligt hela dagarna och andra jobb körs på bestämda tider. Det centrala jobbet för prenumerationsrutinen heter **!!!_PrenumerationsRutin - Startar AgentJobb som ingår i prenumerationsRutinen SAMT AgentJobbStatus- KörsHelaDygnet.**

- ☑ !!!_PrenumerationsRutin - Förbereda_Inför_Ny_Månad_Nollställa_Periodicitet_För_Manuellt_Startade_jobb - Körs 1 gång per Månad
- ☑ !!!_PrenumerationsRutin - Sparar Prenumerations_OCH_Agentjobbs_Historik - Körs under helg
- ☑ !!!_PrenumerationsRutin - Sparar Prenumerations_OCH_Agentjobbs_info_som_Dagligt_Snapshot - Körs 1 gång per dag
- ☑ !!!_PrenumerationsRutin - Sparar Prenumerations_OCH_Agentjobbs_info_Underlag_Visa_Beroenden_Mellan_Jobb - Körs x-gång per dag
- ☑ !!!_PrenumerationsRutin - Sparar_Metadata_För_PrenumerationsRutinen
- ☑ !!!_PrenumerationsRutin - Startar AgentJobb som ingår i prenumerationsRutinen SAMT AgentJobbStatus- KörsHelaDygnet
- ☑ !!!_PrenumerationsRutin - Sänder StatusMail - Information om Dagens aktuella Prenumeranter - Visar bl.a Agentjobb
- ☑ !!!_PrenumerationsRutin - Sänder StatusMail kl: 00 SAMT Utför KILL - Visar eventuella låsningar på servern

Även mallar har prefixet !!!_xxx

!!!_PrenumerationsRutin – Sänder StatusMail kl: 00 SAMT Utför KILL – Visar eventuella låsningar på servern

Detta jobb körs en gång i timmen hela dygnet och hanterar låsningar på servern. Jobbet gör en kill på de frågor som körs som orsakar låsningen och skickar ut ett mail med information om låsningen. För mer information se proceduren

[Prenumeration].[sp_PrenumerationsRutin_Sända_StatusMail_SAMT_Utföra_KILL_Finns_det_låsningar_blocked_by]

1. Övervakning av prenumerationsrutinen

För att övervaka prenumerationsrutinen används en rapport för att kontrollera att alla flöden gått bra. Den innehåller alla agentjobb som startas av prenumerationsrutinen och deras status för dagens körning. Även de jobb som körs under eftermiddagen/kvällen dagen innan syns här. Eftersom vissa jobb startat efter alla gått hem för dagen.

När man övervakar prenumerationsrutinen är det bra att kolla rapporten ca 1 gång per timme. Det är viktigt att kolla historik-fliken varje måndag och efter helgdagar, för att kontrollera att inget slutat fungera under helgen/ledigheten.

Länk till rapport finns under rubriken övervaka agentjobb och flöden.

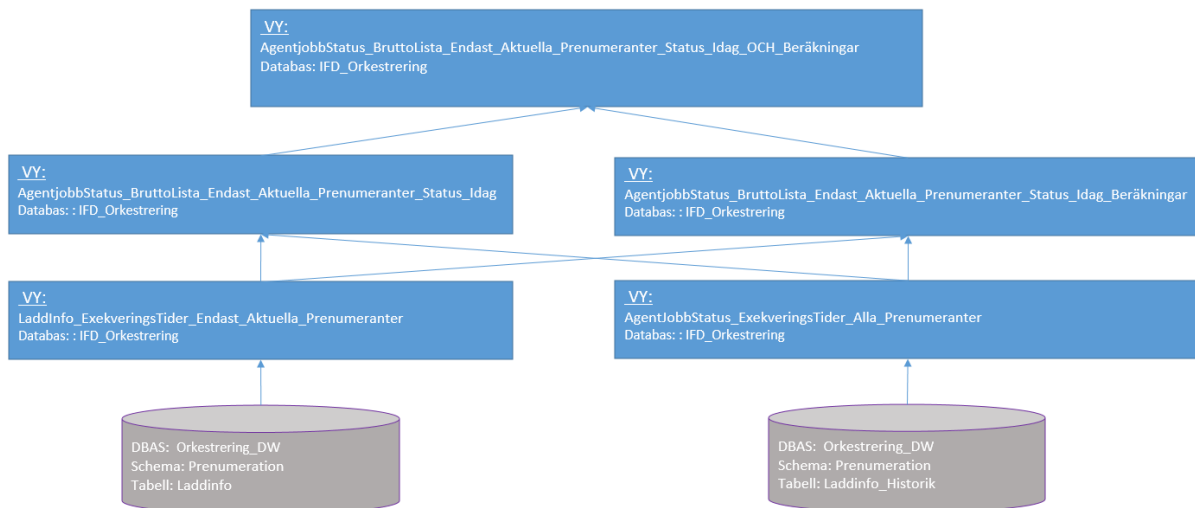
Statusmail

2 Gånger per dag ska statusmail skickas ut med information om hur dagens körningar har gått. Det första mailet skickas vid 09:00 och det andra vid 16:00. Nedan finns ett exempel på hur mailet ska se ut och vilken information som ska finnas med. Rubriken ska vara *Status Prenrutin 2024-XX-XX* och skickas till Michael Müller, Tuba Lundin, Nikals Hendriksson, Miriam Niskanen, Marcus Sandrén,

Status dagens körningar förmiddag

- Rullande: 2 Rullande idag och de har gått bra
- Daglig: Allt har gått bra hittills idag
- Vecka: 1 jobb ska starta runt 11:30
- Månad: 2 jobb idag och båda har gått bra.
- Gårdagens körningar: Allt gick bra igår.

Datamodell övervakningsrapport för agentjobb och flöden



Övervaka agentjobb och flöden

För att övervaka agentjobb och flöden som hanteras av prenumerationsrutinen används denna rapport: [joblog_masterdata_dw_pbris - Power BI Report Server \(vgregion.se\)](http://vgregion.se)

Rapporten uppdateras var 15de minut.

Monitorering Prenumerationsrutin

Är den viktigaste fliken och den som används under övervakningen.

Här visas alla aktiverade jobb och dagens status. Det går att filtrera ut jobbnamn, agentjobbgrupp (flera jobb som hör ihop), miljö och status. Kolumnen status visar om jobbet gått bra eller inte och kolumnen klar för dagen visar om jobbets sista körning är klar, är främst för kontroll av rullande jobb som kör flera gånger samma dag. Filtret periodicitet används för att filtrera jobb på hur ofta de körs. T.ex. daglig, månad eller rullande. På den här fliken syns även jobb som startade sent föregående dag. För ytterligare information om jobben högerklicka på ett jobbnamn. Välj granska på samma nivå och sedan vilken flik du vill gå till för mer information om jobbet.

Monitorering prenumerationsrutin – jobbsteg

Fliken används för att se status på ett specifikt jobbsteg. Används för att enkelt kunna se vilket steg i jobbet som gick fel.

Monitorering prenumerationsrutin – HISTORIK

Fliken visar alla jobb och status över tid. Den härfliken är extra viktig att kolla igenom på måndagar eller efter en helg. För att kontrollera att allt gått bra under ledigheten. Det är även viktigt att filtrera ut jobb med periodicitet MÅNAD eller VECKA för att se att inget sådant jobb misslyckats. Eftersom de bara körs en gång i månaden/veckan ser vi inte om något är fel för ens nästa tillfälle jobbet körs. Så det är viktigt att se över historiken.

Monitorering Prenumerationsrutin – inaktiva jobb

Fliken visar alla jobb som är inaktiverade men som finns kvar i prenumerationsrutinen. Det kan vara jobb som håller på att åtgärdas eller som behövt inaktiveras av någon anledning.

Monitorering prenumerationsrutin PBIRS / PBIRS HIST

Här visas status för dagens jobb som uppdaterar Power BI rapporter med ny information. HISTfliken visar även gamla körningar.

Monitorering – Ej PrenrutinJobb – Jobb utan prefix !!_

Fliken innehåller alla jobb som har !!_ men som vi ändå vill följa upp. Det kan vara t.ex. jobb som startas av prenumerationsrutinen men som rör dev- eller testmiljöer.

Prenumerationsrutin beroenden till källor

”Drill Through” flik. Används vid felsökning när ett jobb gått fel för att hitta vilka källor ett jobb är beroende av. Högerklicka på ett jobb på förstafliken, välj granska på samma nivå och sedanfliken prenumerationsrutin beroenden till källor.

Prenumerationsrutin beroenden till prenumeranter

”Drill Through#” flik. Används vid felsökning för att hitta vilka jobb som är prenumeranter som är beroende av ett jobb. Högerklicka på ett jobb på förstafliken, välj granska på samma nivå och sedanfliken prenumerationsrutin beroenden till prenumeranter.

Extra viktiga jobb att ha koll på

Följande jobb är extra viktiga att ha koll på och bör vara klara runt 08:00

- !!_Prenumeration_ETL_SjukhusEpisoder_SIS_Daglig_Körning
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE
- !!_Prenumeration_BI_CDW_Del_1__Enbart_SjukhusEpisoder_i_IFD_Sjukhusvård
- !!_Prenumeration_Kub_Belagging
- !!_Prenumeration_Kub_Akuten (*bör vara klart senast kl: 9*)

Nedan jobb bör köras minst var tredje dag. Så det är okej att jobbet misslyckas två dagar i rad men om den även misslyckas på den tredje dagen behöver det åtgärdas. Se vidare under rubriken Kontrolljobb för att se mer detaljerad info

!!_Prenumeration_BI_CDW_Del_2__Vårdkontakter_Samt_läkemedel_Utöver_SjukhusEpisoder_i_IFD_Sjukhusvård

Kontrolljobb

Vissa jobb är av typen 'kontrolljobb' vars uppgift är att kontrollera att källan är uppdaterad med den senaste informationen.

- Nedan ses ett exempel på ett sådant kontrolljobb.
!!_Prenumeration_KONTROLL_källa_klar_etl_SjukhusEpsisoder_Ssis_Daglig_Körning_____
Starta_Flödet_OM_Kontroll_ger_OK
- Nedan ses ett exempel på ytterligare en typ av kontrolljobb
Just nedan jobb fallerar ibland och det ligger så att säga i 'lösningsdesignen' att det blir så, denna källa är alltså otillgänglig vissa tider / ibland hela dagar. Men just detta kontrolljobb måste köras ok minst var tredje dag.
!!_Prenumeration_KONTROLL_källa_klar_Melior_RAMU_databaser_____
Starta_Flödet_OM_Kontroll_ger_OK

Vem ansvarar för källan eller prenumeranten

När ett jobb misslyckats är det bra att kontakta de som är ansvariga för jobbet och lösningen jobbet uppdaterar. För att hitta vem som är ansvarig för ett jobb kan den här proceduren i Orkestrering_DW användas. *Prenumeration.sp__Manuell_Kontroll_Vem_Är_Ansvrig_för_Viss_Prenumeration*.


Eller kör den här koden


```
USE [Masterdata_DW]
GO
DECLARE      @return_value int
EXEC        @return_value =
[Prenumeration].[sp__Manuell_Kontroll_Vem_Är_Ansvrig_för_Viss_Prenumeration]
@Ange_Agentjobbnamn = N'agentjobbnamn'
SELECT      'Return Value' = @return_value
GO
```

Kontrollera beroenden mellan jobb

När ett jobb misslyckats är det viktigt att ta reda på om jobbet har beroenden till andra jobb. Stannar ett jobb som flera andra jobb har beroenden till kommer inte de andra jobben starta på förväntad tid.

För att hitta beroenden mellan jobb exekvera någon av de här procedurerna. Den nedra inkluderar hur senaste körningen har gått. Den övre visar endast beroenden mellan jobben.

 Prenumeration.sp__Manuell_Laddinfo_Kontrollera_Beroenden_mellan_Källor_OCH_Prenumeranter_för_ett_visst_Agentjobb

 Prenumeration.sp__Manuell_Laddinfo_Kontrollera_Beroenden_mellan_Källor_OCH_Prenumeranter_för_ett_visst_Agentjobb_SAMT_Senaste_KörningsStatus

Det går även bra att köra denna kod:

```
USE [Orkestrering_DW]
GO
DECLARE      @return_value int
EXEC        @return_value =
[Prenumeration].[sp__Manuell_Laddinfo_Kontrollera_Beroenden_mellan_Källor_OCH_Prenumeranter_för_ett_visst_Agentjobb]
@Ange_Agentjobbnamn = N'!!_Prenumeration_Datamart_SVF_Och_Utdata_SVF',
@Ange_Om_hela_Kedjan_Av_Beroenden_Skall_Visas_J_N = N'N'
SELECT      'Return Value' = @return_value
GO
```

Byt ut agentjobbnamn mot det jobbet du vill kontrollera beroenden för.

Information om körningar

Koden nedan kan användas för att få fram information om körningar och vilken status de har. För att t.ex. ta reda på varför ett jobb inte startat. *Ytterligare beskrivning?*

```
SELECT TOP (1000)
[EventLogID],[Database],[Name],[Scope],[StartTime],[EndTime],[Total_KörTid_HH_MM_S
S],[ErrorDetected],[WarningDetected],[Message],[Rad_Skapad_DatumTid],[AppExecLogID
],[ParentEventLogID]

FROM [Audit].[Orkestrering_Prenumeration].[Audit_AdminEventlog__SP_Körtider] where
message like '%Datamart_Beläggning%'
```

Kontrollera att klockslag stämmer överens för alla jobb i en AgentjobbGrupp

Beskrivning av kod nedan?

```
SELECT l1.Källa_AgentjobbNamn, l2.Källa_AgentjobbNamn, l1.Prenumerant_AgentJobbNamn,
l2.Prenumerant_AgentJobbNamn, l1.Tidigaste_Klockslag_På_Dygnet_Agentjobb_Får_Starta,
l2.Tidigaste_Klockslag_På_Dygnet_Agentjobb_Får_Starta,
l1.Avser_Agentjobb_Periodicitet_OCH_Klockslag_Enbart_Prenumerant_J_N,
l2.Avser_Agentjobb_Periodicitet_OCH_Klockslag_Enbart_Prenumerant_J_N
FROM [Orkestrering_DW].[Prenumeration].[Laddinfo] as l1

FULL JOIN [Orkestrering_DW].[Prenumeration].[Laddinfo] as l2
--ON l2.Källa_AgentjobbNamn = l1.Prenumerant_AgentJobbNamn
ON l2.Prenumerant_AgentJobbNamn = l1.Prenumerant_AgentJobbNamn

--ON l2.Källa_AgentjobbNamn = l1.Källa_AgentjobbNamn
--AND l2.Prenumerant_AgentJobbNamn = l1.Källa_AgentjobbNamn

WHERE 1=1
AND l1.Prenumerant_AgentJobbNamn <> 'PRENUMERANT_SAKNAS'
AND l1.Tidigaste_Klockslag_På_Dygnet_Agentjobb_Får_Starta <>
l2.Tidigaste_Klockslag_På_Dygnet_Agentjobb_Får_Starta
AND ISNULL(l1.Avser_Agentjobb_Periodicitet_OCH_Klockslag_Enbart_Prenumerant_J_N,
'N') = 'N'
```

Kontrollera att cache- & remotetabeller uppdaterats korrekt

För att kontrollera att cache- och remotetabeller uppdaterats kan man kolla i tabellen

DenodoRefreshLog_Information i audit databasen på gral-prod-is01. I den tabellen finns två kolumner, rows_before och rows_after. De visar hur många rader som fanns före och efter tabellen laddats. Är kolumnen rows_after 0 har inte vyer uppdaterats.

Det finns även vyer i databasen GRAL_DRIFT_DW i Denodo där det går att se status för cachevyerna.

- Vdp_lista_vyer_med_cache_fysiskt_lagrad: läser cache tabell och uppdateras var 15de minut.
- Vdp_lista_vyer_med_cache: endast de med adminbehörighet kan använda denna.
- Prenumeration_status_denodorefreshlog_db_denodo_dev: innehåller det som uppdateras via prenumerationsrutinen.

Vyerna uppdateras via jobben nedan.vDe körs var 15de minut men det går att köra jobben manuellt vid behov / eller om man behöver information direkt.

```
☑ I_DENODO_Datarefresh_CACHE_Uppdatera_VDP_Lista_Vyer_Med_Cache_Avser_Denodo_DEV_Miljö
☑ I_DENODO_Datarefresh_CACHE_Uppdatera_VDP_Lista_Vyer_Med_Cache_Avser_Denodo_PROD_Miljö
☑ I_DENODO_Datarefresh_CACHE_Uppdatera_VDP_Lista_Vyer_Med_Cache_Avser_Denodo_TEST_Miljö
```

Kontrollera PBIRS Datarefresh

För att kontrollera att en PBI rapport uppdateras som den ska via prenumerationsrutinen kör den här koden men byt ut mot PBIRS schemalaggnings du vill kontrollera.

```
SELECT *
FROM [VGDB0856\01,52255].[ReportServer_PBIRS_Log].[Report_Log].[Subscriptions]
WHERE description like 'Pbirs_report_datarefresh_IFD_Sjukhusvård%'
```

Kolumnen LastStatus visar status om rapporten uppdaterats med ny data och kolumnen LastRunTime visar senast gången rapporten uppdaterades.

För att endast se de rapporter som uppdateras via prenumerationsrutinen kör koden nedan

```
SELECT TOP (1000) [Rapportnamn],
[Folder],
[Scheduleringsnamn],
[ScheduleringsSchema_Starttid_MAX],
[ScheduleringsSchema_Sluttid_MAX],
[Message],
[ItemAction],
[RapportStatus],
[UserName],
[name]
FROM
[VGDB0856\01,52255].[ReportServer_PBIRS_Log].[Report_Log].[v_Rapportstatus_Senaste_7_D
agarna_Där_Datarefresh_Sker_Via_PrenRutin]
where Scheduleringsnamn like 'Pbirs_report_datarefresh_%'
```

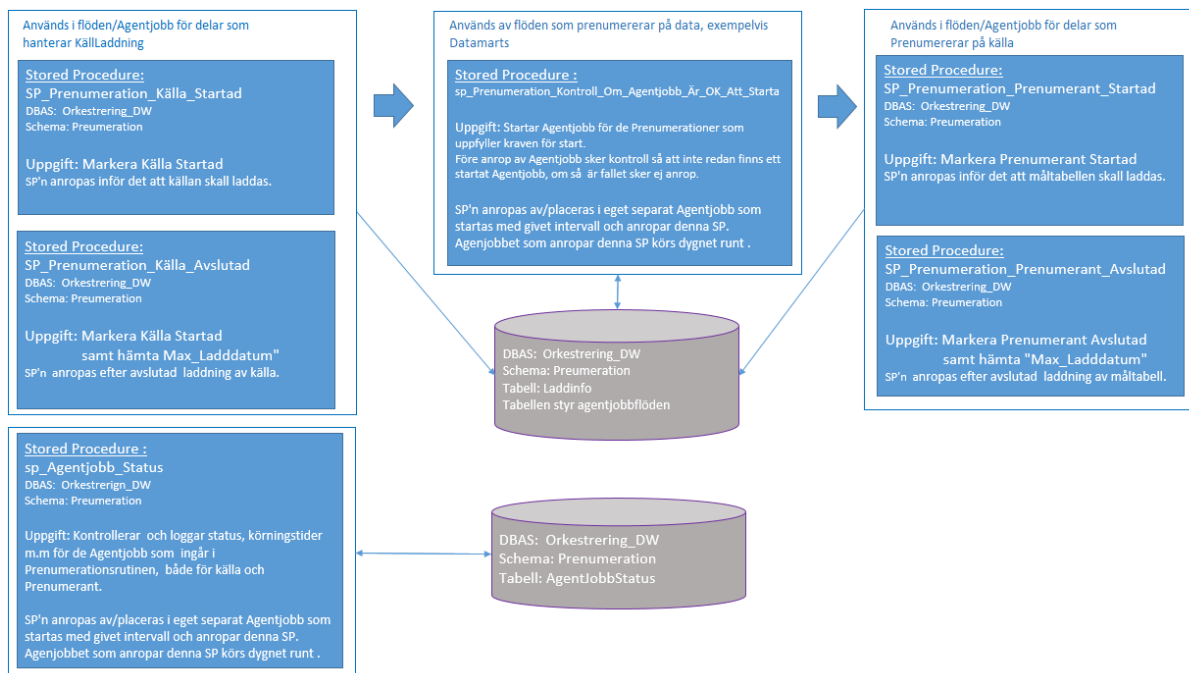
2. Administrera prenumerationsrutinen

Prenumerationsrutinen administreras genom att lägga till, ta bort eller justera rader i tabellen Prenumeration.LaddInfo i databasen Orkestrering_DW på GRAL-PROD-IS01.

Prenumerationsrutinen är främst avsedd för jobb som rör produktionsmiljöer. I undantagsfall kan jobb för dev eller test läggas in men fråga gärna någon av kontaktpersonerna för prenumerationsrutinen innan.

Det finns mallar att utgå ifrån i Orkestrering_DW som kan användas för att lägga till ett jobb i prenumerationsrutinen. Det finns även mallar för hur agentjobb ska se ut med mera. Mallarna börjar med prefixet !!!_

Datamodell PrenumerationsRutin



Beroende på krav vad gäller tillgänglighet på data så finns alternativ på när anrop skall ske. Ställs tydliga krav på exakthet vad gäller starttiden väljer man att anropa proceduren i direkt anslutning av källan. Om kraven är lägre så kan alternativet vara att anropa proceduren som första steg i Flödet/Agentjobbet

Skapa ett jobb

Skapa ett agentjobb

1. Jobbets namn måste börja med prefixet `!!_Prenumeration`.
 - a. Jobb som består av flera delar namnges med `del_1`, `del_2` osv.
 - b. Jobb som rör **dev**- eller **test**miljöer ska avslutas med suffixet avser **dev**_miljö eller avser **test**_miljö
2. För att jobbet ska startas av prenumerationsrutinen måste jobbet ha **loggsteg**.
 - a. Jobb utan beroenden behöver 2 loggsteg: `Källa_Startad` & `Källa_Avslutad`.
 - b. Jobb med beroenden behöver 4 loggsteg: `Källa_Startad` & `Källa_Avslutad` **samt** `Prenumerant_Startad` & `Prenumerant_Avslutad`.
3. Loggstegen för start ska vara först i jobbet. Loggstegen för avslutad ska vara sist i jobbet.
4. Loggstegen ska även startas om när de misslyckas. Sätt `retry attempts` till 20 och `retry interval` till 2.

Loggsteg

Loggsteg för prenumerant används för att kontrollera att nästa jobb i kedjan startat och avslutats. I de jobb som har alla 4 loggstegen ska loggstegen för prenumeranten vara före loggsteg för källan.

Mall för loggsteg finns bland agentjobben på GRAL-PROD-IS01.

!!!_MALL_Prenumeration_LOGGSTEG

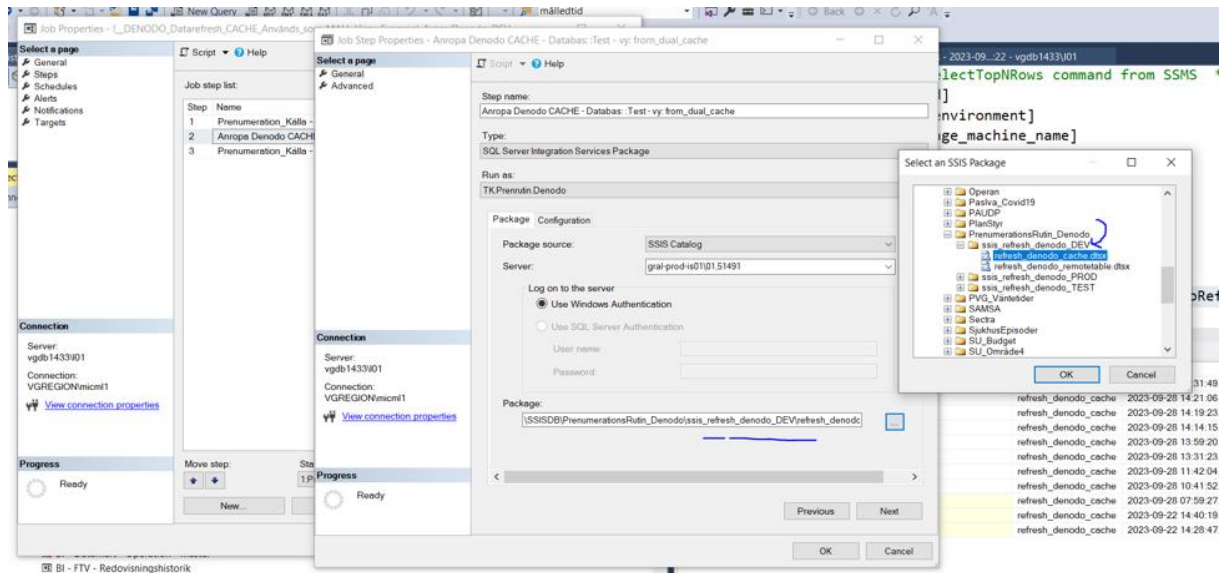
Skapa ett jobb som uppdaterar cache- eller remotetabeller i Denodo

Det finns tre mallar som går att ha som exempel. Det finns ett exempel per miljö.

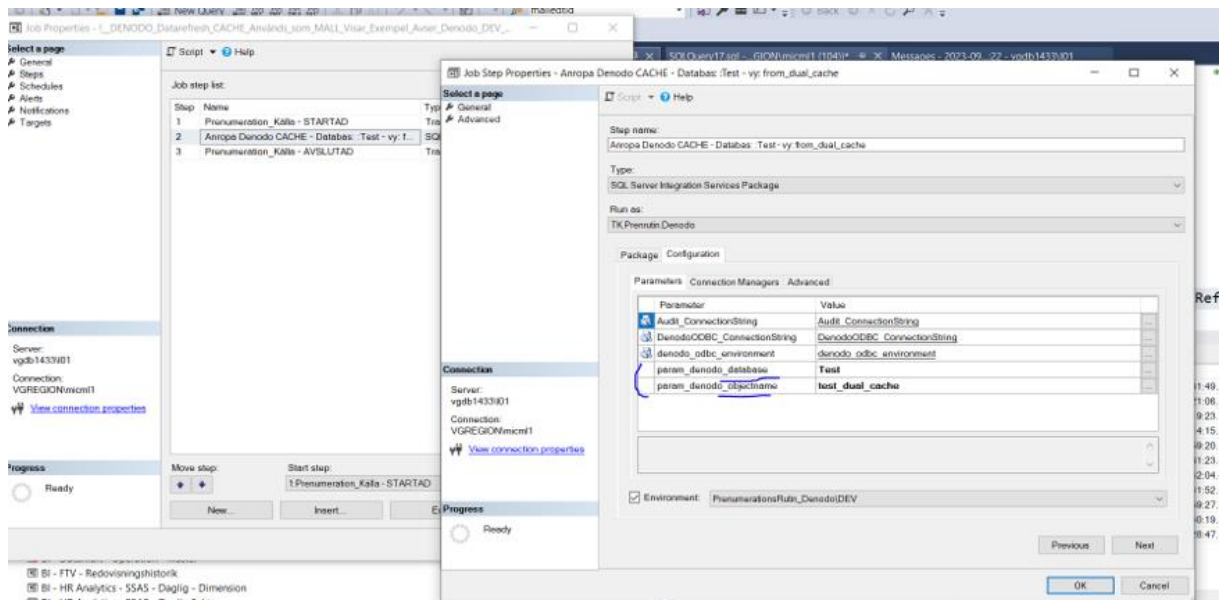
- !_DENODO_Datarefresh_CACHE_Anvands_som_MALL_Visar_Exempel_Avser_Denodo_DEV_Miljö
- !_DENODO_Datarefresh_CACHE_Anvands_som_MALL_Visar_Exempel_Avser_Denodo_TEST_Miljö
- !_DENODO_Datarefresh_CACHE_Anvands_som_MALL_Visar_Exempel_Denodo_PROD_Miljö

Byt ut 'Används som mall visar exempel' till din databas eller lösning som ska cachas.

Se till att rätt paket anropas i jobbet. Det finns ett paket per miljö. Extra noga om du vill skapa ett nytt jobb utan mallen och se till att ha med alla loggsteg i jobbet.

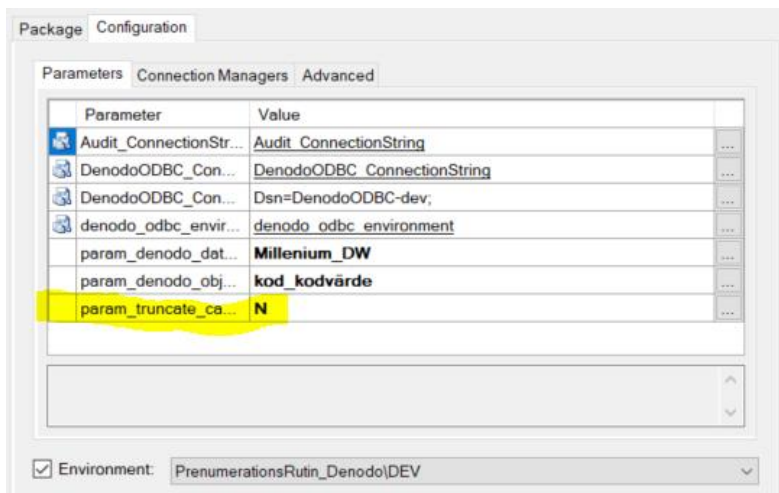


Under 'konfiguration' fliken går det att se vilken miljö, databas och tabell som ska cachas.



Cacha vyer för Millenium

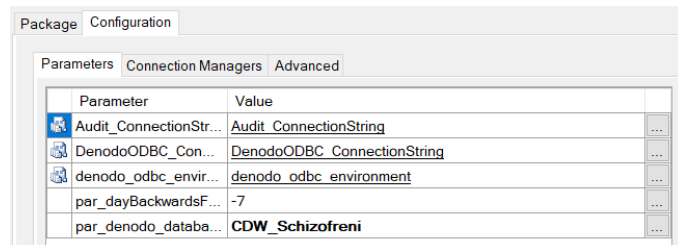
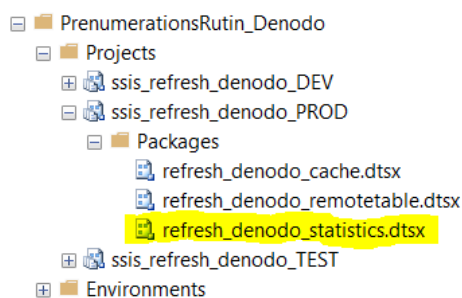
När vyer för millenium ska cachas är det viktigt att inte göra en truncate innan cachen körs om. Sätt truncate variabeln till N enligt bilden nedan. Då det finns mycket beroenden till millenium är det viktigt att vyer aldrig blir tomma. Därför vill vi inte trunkera innan vi cachar om. Istället görs rader som redan finns till 'invalid'. De ogiltiga raderna raderas sedan en gång i veckan. Se exempel på bilden nedan.



Skapa ett jobbs om uppdaterar statistik i Denodo

Det går uppdatera statistik på vyer i denodo via prenumerationsrutinen.

Det gör du genom att skapa ett jobb som vanligt med loggsteg och samma namnstandard som för cachejobb. Men byt ut CACHE i namnet mot STATISTIK. I jobbet skapa sedan ett eller flera jobbsteg som kör SSIS paketet i bilden nedan. Det finns ett paket per miljö så se till att välja rätt. Input till SSIS paketet är databasen där vyerna finns och hur ofta statistiken ska uppdateras. -7 i exemplet nedan innebär att statistiken uppdateras var sjunde dag.

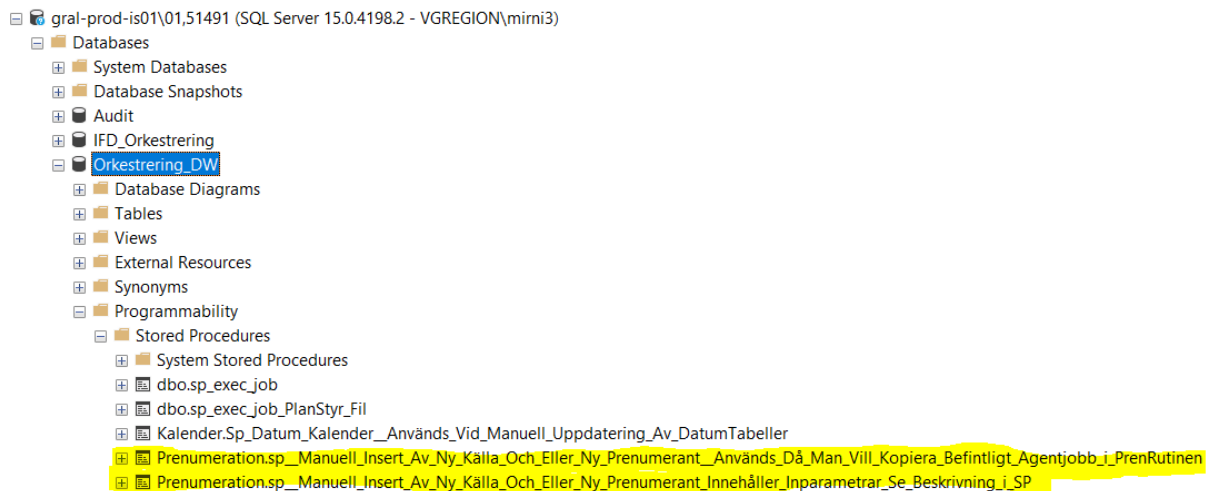


Lägga in ett agentjobb i prenumerationsrutinen

Ett jobb läggs till i prenumerationsrutinen genom att lägga in en eller flera rader för jobbet/jobben i tabellen **Prenumeration.Laddinfo i Orkestrering_DW** på GRAL-PROD-IS01. Det finns två stored procedures som kan användas som en mall. I mallarna finns beskrivningar av alla kolumner och vilken information som behöver fyllas i.

Sista jobbet i kedjan behöver endast läggas till som en prenumerant. Den behöver ingen egen rad där jobbet även är en källa.

De två markerade procedurerna nedan innehåller en beskrivning av alla kolumner och hur/vilken information som behöver fyllas i för att ett jobb ska starta via prenumerationsrutinen.



Jobb med flera beroenden

Om ett jobb är beroende av flera jobb läggs det till en rad för varje beroende. T.ex. Jobb A är beroende av jobb B och jobb C. Då skapas en rad där jobb B är källa och jobb A prenumerant **samt** en till rad där jobb C är källa och jobb A är prenumerant. Då kommer inte jobb A startas före både jobb B och jobb C gått klart.

När prenumeranten inte ska starta direkt efter källan

I de fall när prenumeranten **inte** ska starta direkt efter källan anges J i kolumnen

Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N. Om ett jobb är källa till flera prenumeranter är det viktigt att välja raden med rätt prenumerant för att inte råka försena andra körningar.

Exempel där prenumeranten inte startar direkt efter källan:

```
SELECT * FROM [Orkestrering_DW].[Prenumeration].[Laddinfo]
WHERE Källa_AgentjobbNamn = '!!_Prenumeration_BI_CDW_Covid_19_LND_De12'
OR Prenumerant_AgentJobbNamn = '!!_Prenumeration_BI_CDW_Covid_19_LND_De12'
```

Eller

```
SELECT * FROM [Orkestrering_DW].[Prenumeration].[Laddinfo]
WHERE Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N = 'J'
```

När enbart prenumeranten ska startas via prenumerationsrutinen

I de fall då endast prenumeranten ska startas av prenumerationsrutinen men källan har en egen schemulering och **inte** startas av prenumerationsrutinen.


Se till att jobbet som är källan börjar med !!_Prenumeration så att jobbet kommer med i Laddinfo tabellen även fast det inte startas via prenumerationsrutinen. Lägg även till loggstegen i jobbet. Saknas loggstegen kommer inte prenumerationsrutinen veta när det är okej för prenumeranten att starta, eftersom källan måste ha en avslutstid innan prenumeranten får starta.


Gör sedan som vanligt för att lägga till en ny rad i Laddinfo tabellen. Men sätt kolumnen **Skall_Inledande_AgentJobb_Startas_Automatiskt_Via_PrenumerationsRutin_J_N** till N. Eftersom källan inte ska startas via prenumerationsrutinen. Sätt kolumnen

Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N till J eftersom endast prenumeranten startas av prenumerationsrutinen.

Prenumeration.LaddInfo – Beskrivning av kolumner

Alla fält finns beskrivna med exempel i de här procedurerna i databasen Orkestrering_DW på gral-prod-is01.

 Prenumeration.sp_Manuell_Insert_Av_Ny_Källa_Och_Eller_Ny_Prenumerant_Används_Då_Man_Vill_Kopiera_Befintligt_Agentjobb_i_PrenRutinen

 Prenumeration.sp_Manuell_Insert_Av_Ny_Källa_Och_Eller_Ny_Prenumerant_Innehåller_Inparametrar_Se_Beskrivning_i_SP

Nedan finns endast de kolumner som inte har default värden.

Kolumn	Beskrivning
AgentjobbGrupp	Data som hänger ihop och blir en gemensam lösning bildar en grupp.
Ansvarig_för_Specifik_Prenumerationspost	Mailadresser till de som är ansvariga för jobben och bör informeras om jobbet misslyckats. Ange vilka som ansvarar för källan och prenumeranten.
AgentjobbServer	Server där jobbet finns. Ange för både källa och prenumerant.
Avser_Agentjobb_Att_Leverans_Är_Helt_Klar	Visar om jobbet är det sista i kedjan. Sätts till J eller N. Sista delen i kedjan av jobb är ofta en KUB eller PBIRS schemulering. Används för status-gral rapporten som visar om en lösning är uppdaterad som den ska.
Prenumeration_Aktiverad_J_N	Avgör om jobbet startas av prenumerationsrutinen eller inte. Sätts till J för att jobbet ska startas automatiskt av prenumerationsrutinen.
Är_AgentJobb_Inledande_Jobb_för_AgentjobbGrupp_J_N	Visar om ett jobb är första jobbet i kedjan. Sätts till J för första jobbet. Råkar det stå ett J för ett jobb som EJ är först i kedjan är detta ingen fara.
Skall_Inledande_AgentJobb_Startas_Automatiskt_Via_PrenumerationsRutin_J_N	Ska vara J om man vill att PrenRutinen ska starta jobbet. <i>Undantag: Vill man istället ha en schemulering i själva Agentjobbet så sätts nedan till N. Då startar källan via agentsjobbet schemulering men prenumeranten startas av prenumerationsrutinen.</i>
Agentjobb_Periodicitet	Hur ofta jobbet ska köras. Giltiga värden är DAGLIG, VECKA, MÅNAD, RULLANDE. Hänger ihop med kolumnen <i>Agentjobb_Periodicitet_Specifikation</i> .
Agentjobb_Periodicitet_Specifikation	Hur ofta jobbet ska köras. Är Agentjobb_Periodicitet DAGLIG så ska denna kolumn ha värdet VARJE DAG. För mer information se kod i stored procedures i Orkestrering_DW på gral-prod-is01.

Tidigaste_Klockslog_På_Dygnet_Agentjobb_Får_Starta	Tidigaste klockslog som jobb kan starta. Om jobbet är beroende av andra jobb i kedjan så kan starten bli senare än vad som angetts i kolumnen.
Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N	Tilldelas J i de undantagsfall då man vill att klocksloget endast ska gälla Prenumeranten. Med andra ord i de fall man INTE vill att Prenumeranten ska starta direkt efter källan.
Prenumerant_AgentJobbNamn	Agentjobb som är prenumeranten. Alltså det jobb som ska starta EFTER källan. Om ett jobb inte har en prenumerant anges "PRENUMERANT_SAKNAS". För sista jobbet i kedjan räcker det att jobbet finns med som en prenumerant.
Källa_AgentjobbNamn	Agentjobbet som är källan.

Schemalägga PBIRS Refresh via prenumerationsrutinen

Det går att schemalägga uppdateringar av PBI rapporter via prenumerationsrutinen. Då uppdateras rapporten direkt efter att datakällan är uppdaterad. Det är bra att göra detta för att undvika att rapporter uppdateras i onödan och belastar Report Servers servrar.

1. Skapa en schemaläggning av rapporten på report server. Sätt att schemaläggningen bara ska gälla en dag och följ namnstandard enligt exempel nedan.

Edit Pbirds_report_datarefresh_CDW_Schizofreni_0010 At 03:00 on 2024-02-20 and ending 2024-02-21

Byt ut CDW_Schizofreni mot din datakälla / lösning. 0010 anger att det är första körningen. Består lösningen av flera rapporter eller uppdateringar så ökas det till 0020 osv för senare körningar. Det blir då en schemaläggning per körning som behövs.

2. Skapa ett agentjobb enligt den här mallen

!!!_MALL_Prenumeration_RAPPORTPAKET_Pbirds_report_datarefresh_Description_XXXXXXXXXX

Byt ut Description_XXXX mot din lösning eller databas

3. Kontakta **Michael Müller, Miriam Niskanen, Tuba Lundin eller Niklas Hendriksson** så kan vi skapa upp Power Shell scriptet som ska köras i jobbet. Power Shell scriptet kommer heta samma sak som schemaläggningen på report server. Exempelvis:
powershell.exe -file "D:\BI\PowerShell\Pbirds_report_datarefresh_CDW_Schizofreni_0010.ps1" -NonInteractive
4. Lägg till jobbet som prenumerant i prenumerationsrutinen till källan som ska uppdateras innan rapporten.

Jobb som alltid startas manuellt

Det finns vissa jobb som inte ska ha någon schemaläggning som ska köras helt ad-hoc/vid behov. T.ex. !!_Prenumeration_BI – PA_UDP Månadsladdning är ett sådant jobb. Det jobbet startas manuellt av Tony Niva. Se det jobbet som exempel.

Jobb som ska startas manuellt ska ha periodicitet 'MANUELL-START-MÅNAD'. Jobbet måste även innehålla ett steg som anropar den här procedur där agentjobbnamnet är input:

sp_Prenumerant_Tilldela_Periodicitet_För_Dag_i_Månad_Då_Körningsdatum_Ej_Är_Fastställt.

Inför varje ny månad behöver 'dag i månaden' nollställas. Det görs via ett jobb som körs automatiskt en gång i månaden.

- Jobb Namn: !!!_PrenumerationsRutin -
Förbereda_Inför_Ny_Månad_Nollställa_Periodicitet_För_Manuellt_Startade_jobb - Körs 1 gång per Månad
- Anropar SP:
[Prenumeration].[sp_Prenumerant_Tilldela_Periodicitet_Värde_För_Att_Nollställa_Inför_Ny_Månad]

3. Underhålla prenumerationsrutinen

Exempel på underhåll kan vara att ändra en starttid för ett jobb, ta bort ett jobb, inaktivera ett jobb eller lägga till en prenumerant. **OBS! - Om du lägger till en prenumerant till en källa se till att även lägga till loggsteg för prenumeranten i agentjobbet som är källan.** Annars kommer inte prenumeranten startas som den ska.

Här beskrivs hur ändringar bör göras i prenumerationsrutinen. T.ex. ändra eller ta bort en schemaläggning.

Som första steg är det viktigt att ta en kopia på **LaddInfo** tabllen ifall något blir fel under ändringen. Det kan göras med koden nedan

```
DECLARE @TSQL1 varchar(MAX)
DECLARE @Currtime datetime
SET @Currtime = GETDATE()

SELECT @TSQL1 = (
'SELECT * INTO
Prenumeration.[Laddinfo_SäkerhetsKopia_Före_förändring_Datum_Klockslog_' +
CONVERT(CHAR(23), @Currtime , 25) + ']' from [Prenumeration].[Laddinfo]')
----PRINT @TSQL1

EXEC (@TSQL1)
```

Sedan är det okej att göra ändringen.

Ska du göra ändringen via en UPDATE, gör först en SELECT för att hitta raden som ska ändras. Används sedan **radID** i where-villkoret så att inte fel rad uppdateras. Det går även att göra ändringen direkt via designverktyget.

Ska något tas bort gör samma som vid UPDATE fast DELETE istället med **radID** i villkoret.

4. Åtgärda prenumerationsrutinen

Här finns förklaringar och lösningar på fel som kan uppstå. Vissa fel är generella men här finns även lösningar på specifika problem.

Fler fel / åtgärder som borde finnas med?

Låsningar på servern

Ibland kan jobb låsa sig på servern.

För att hitta vilka queries som fastnat kör denna kod:

```
SELECT TOP (1000) *
FROM [Audit].[dbo].[Currently_Running_Queries]
```

För att stoppa låsningar behöver man göra en **kill** på körningen med denna SQL:

```
INSERT INTO [Prenumeration].[Laddinfo_Historik]
([AgentjobbGrupp], [Beskrivning_Av_Specifik_Prenumerationspost],
[Ansvarig_för_Specifik_Prenumerationspost]
,[AgentjobbServer], [Avser_Agentjobb_Att_Leverans_Är_Helt_Klar],
[Prenumeration_Aktiverad_J_N], [Är_Källa_OCH_ELLER_Prenumerant_Agentjobb_Disablade]
,[Är_AgentJobb_Inledande_Jobb_för_AgentjobbGrupp_J_N],
Skall_Inledande_AgentJobb_Startas_Automatiskt_Via_PrenumerationsRutin_J_N
,[Inledande_Agentjobb_Som_Startas_Automatiskt_Har_Stannat_J_N, Agentjobb_Periodicitet,
Agentjobb_Periodicitet_Specifikation
,[Agentjobb_Periodicitet_Är_Uppfylld_idag_J_N,
[Tidigaste_Klockslog_På_Dygnet_Agentjobb_Får_Starta]
,[Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N],
[Källa_AgentjobbNamn], [Källa_Starttidpunkt]
,[Källa_Avslutadtidpunkt], [Källa_Status],
[Inledande_Källa_AgentJobb_Har_Anropats_Tidpunkt], [Prenumerant_AgentJobbNamn]
,[Prenumerant_Starttidpunkt], [Prenumerant_Avslutadtidpunkt], [Prenumerant_Status],
[Prenumerant_AgentJobb_Har_Anropats_Tidpunkt])
SELECT
AgentjobbGrupp], [Beskrivning_Av_Specifik_Prenumerationspost],
[Ansvarig_för_Specifik_Prenumerationspost],[AgentjobbServer],
[Avser_Agentjobb_Att_Leverans_Är_Helt_Klar], [Prenumeration_Aktiverad_J_N],
[Är_Källa_OCH_ELLER_Prenumerant_Agentjobb_Disablade]
,[Är_AgentJobb_Inledande_Jobb_för_AgentjobbGrupp_J_N],
Skall_Inledande_AgentJobb_Startas_Automatiskt_Via_PrenumerationsRutin_J_N
,[Inledande_Agentjobb_Som_Startas_Automatiskt_Har_Stannat_J_N, Agentjobb_Periodicitet,
Agentjobb_Periodicitet_Specifikation
,[Agentjobb_Periodicitet_Är_Uppfylld_idag_J_N,
[Tidigaste_Klockslog_På_Dygnet_Agentjobb_Får_Starta],
[Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N]
,[Källa_AgentjobbNamn] ,[Källa_Starttidpunkt] ,[Källa_Avslutadtidpunkt]
,[Källa_Status] ,[Inledande_Källa_AgentJobb_Har_Anropats_Tidpunkt]
,[Prenumerant_AgentJobbNamn] ,[Prenumerant_Starttidpunkt]
,[Prenumerant_Avslutadtidpunkt] ,[Prenumerant_Status]
,[Prenumerant_AgentJobb_Har_Anropats_Tidpunkt]
FROM Prenumeration.[Laddinfo]
----WITH (NOLOCK)
-- Utöver det som är oförändrat
EXCEPT SELECT [AgentjobbGrupp], [Beskrivning_Av_Specifik_Prenumerationspost],
[Ansvarig_för_Specifik_Prenumerationspost], [AgentjobbServer],
[Avser_Agentjobb_Att_Leverans_Är_Helt_Klar], [Prenumeration_Aktiverad_J_N],
[Är_Källa_OCH_ELLER_Prenumerant_Agentjobb_Disablade],
[Är_AgentJobb_Inledande_Jobb_för_AgentjobbGrupp_J_N],
Skall_Inledande_AgentJobb_Startas_Automatiskt_Via_PrenumerationsRutin_J_N,
Inledande_Agentjobb_Som_Startas_Automatiskt_Har_Stannat_J_N, Agentjobb_Periodicitet,
Agentjobb_Periodicitet_Specifikation, Agentjobb_Periodicitet_Är_Uppfylld_idag_J_N,
[Tidigaste_Klockslog_På_Dygnet_Agentjobb_Får_Starta],
[Avser_Agentjobb_Periodicitet_OCH_Klockslog_Enbart_Prenumerant_J_N],
[Källa_AgentjobbNamn], [Källa_Starttidpunkt], [Källa_Avslutadtidpunkt],
[Källa_Status], [Inledande_Källa_AgentJobb_Har_Anropats_Tidpunkt],
[Prenumerant_AgentJobbNamn], [Prenumerant_Starttidpunkt],
[Prenumerant_Avslutadtidpunkt], [Prenumerant_Status],
[Prenumerant_AgentJobb_Har_Anropats_Tidpunkt] FROM Prenumeration.[Laddinfo_Historik]
--WITH (NOLOCK)
```

Beskriv vad koden gör?

Stoppa jobb manuellt

Ibland kan det ligga kvar en gammal körning som inte avslutats korrekt. Det kan göra att jobb inte startar som de ska. Har ett jobb inte fått ett avslutsdatum kommer det inte starta igen nästa gång det ska köras. Ett jobb som tenderar till att haka upp sig är till exempel

```
!!_Prenumeration_ETL_Prokap_Datamart_Prokap_Städad.
```

Om kolumnen stop_execution_date är *null* betyder det att körningen **inte** avslutats korrekt och ligger kvar och skräpar.

För att kontrollera om det ligger kvar någon gammal körning kör koden nedan. Kör select först för att hitta körningar som saknar ett stop_execution_date. Kör sedan update för att sätta ett datum för stop_execution_date för att stoppa körningen.

```
-----  
DECLARE @max_instance_id BIGINT  
DECLARE @Källa_AgentJobb_Status VARCHAR(200)  
  
drop table #temp_lång_körtid_Föregående  
  
select 'Agentjobbstatus - Föregående körningar (Alltså EJ senaste körningen) : ' AS  
AgentjobbStatusFöregåendeKörningar, sj.name as Jobname, start_execution_date,  
stop_execution_date  
-----ISNULL(COUNT(*), 0)  
  
into #temp_lång_körtid_Föregående  
FROM msdb.dbo.sysjobactivity AS sja1 WITH (NOLOCK)  
  
INNER JOIN msdb.dbo.sysjobs AS sj WITH (NOLOCK)  
ON sja1.job_id = sj.job_id  
WHERE 1=1  
and sj.name like '!!_Prenumeration%'  
and sj.name = '!!_Prenumeration_VALIDERING_CDW'  
AND sja1.start_execution_date IS NOT NULL  
--AND sja.stop_execution_date IS NULL  
  
and sja1.start_execution_date = (select MAX(sja2.start_execution_date)  
from msdb.dbo.sysjobactivity sja2  
where sja2.job_id = sja1.job_id  
-- Nedan villkor medför att kontrollen sker på om det finns någon gammal  
körning som 'skräpar' där stop_execution_date is null  
-- (i kombination med att det finns en nyare körning som kört klart med  
stop_execution_date ifyllt)  
and sja2.start_execution_date < (select max(start_execution_date) from  
msdb.dbo.sysjobactivity sja3 where sja3.job_id = sja2.job_id)  
and sja2.stop_execution_date is null)  
  
select 'Agentjobbstatus - Föregående körningar (Alltså EJ senaste körningen) : ' AS  
AgentjobbStatusFöregåendeKörningar, isnull(COUNT(*), 0) as Antal  
from #temp_lång_körtid_Föregående  
  
-----  
-- Uppdatering för jobb som har en gammal körning där värdet för stop_execution_date  
IS NULL  
-----  
--update sja  
--set stop_execution_date = last_executed_step_date --- sja.start_execution_date ----  
-last_executed_step_date ----start_execution_date  
  
select t.Jobname, sja.*  
FROM msdb.dbo.sysjobactivity AS sja WITH (NOLOCK)
```

```

INNER JOIN msdb.dbo.sysjobs AS sj WITH (NOLOCK)
ON sj.job_id = sj.job_id

join #temp_lång_körtid_Föregående t
on t.Jobname = sj.name
and t.start_execution_date = sj.start_execution_date
--and t.stop_execution_date = sj.stop_execution_date
WHERE 1=1
-- and sj.name = @Källa_AgentjobbNamn

AND sj.start_execution_date IS NOT NULL
AND sj.stop_execution_date IS NULL
--and stop_execution_date is null
--and Jobname <>
'!!_Prenumeration_KONTROLL_källa_klar_Melior_RAMU_databaser_____Starta_Flödet_OM_Kon
troll_ger_OK'
--and Jobname = '!!_Prenumeration_ETL_Prokap_DataMart_Prokap_Städad'
-----
and t.Jobname = '!!_Prenumeration_VALIDERING_CDW'

```

Om ett jobb stoppats manuellt måste det startas om manuellt efteråt med. Eftersom jobbet avslutades mitt under körningen körs inte de sista loggstegen som säger till prenumerationsrutinen att jobbet är avslutat. Har inte ett jobb ett avslutsdatum kommer det inte starta nästa gång.

Om du stoppat !!_Prenumeration_ETL_Prokap_Datamart_Prokap_Städad. Vänta tills 14:00 innan du startar om det igen.

!!_Prenumeration_ETL_SjukhusEpisoder_Ssis_Daglig_Körning

Jobbet läser in XML-filer som ibland kan bli korrupta.

För att hitta vilka filer som är korrupta kör koden nedan på gral-prod-db03

```

SELECT TOP (100) *
FROM [SjukhusEpisoder_LND].[import].[XMLData__Korrupta_Filer]
order by etl_RowCreated desc

```

Korrupta filer får null i kolumnen XML_Data och kolumnen VarBinaryData innehåller data. T.ex. såhär:

etl_RowCreated	etl_ImportDataID	etl_AppExecLogID	etl_LogicalBatchDTS	etl_SourceSystem	SourceFileName	Term	XMLData	VarBinaryData	Filuttagsdatum	AntalPoster	SchemaNamn
2023-05-31 02:35:50.493	487242	158697	2023-05-31 02:34:19.000	NULL	\\region.selappl\B\Fileshare\SjukhusEpisodPro...	NULL	NULL	0x3C3F786D6C2076657273696F6E3D22312E302220656E63...	NULL	NULL	NULL
2023-05-31 02:29:11.850	487164	158695	2023-05-31 02:29:07.000	NULL	\\region.selappl\B\Fileshare\SjukhusEpisodPro...	NULL	NULL	0x3C3F786D6C2076657273696F6E3D22312E302220656E63...	NULL	NULL	NULL

OBS! Det ligger några gamla filer i tabellen som exempel. De behöver inte åtgärdas.

Motsvarande korrupta filer hamnar även på Applikation (P:) i mappen

!_Korrupta_Filer_GemensamYta_För_alla_Korrupta_filer

Kontakta sutf@insieme.se och skriv vilka filer de behöver skicka om

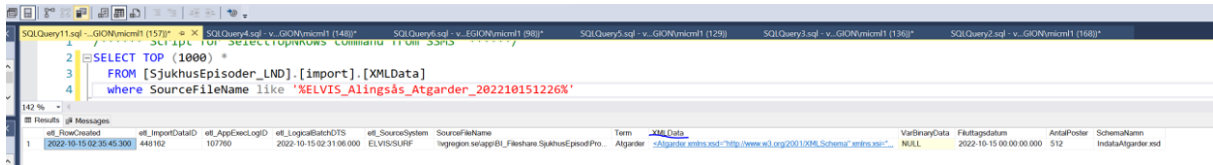
Kontrollera att insieme skickat om filerna genom att kolla i tabellen imprt.XML_data.

```

SELECT TOP (100) *
FROM [SjukhusEpisoder_LND].[import].XMLData
WHERE SourceFileName = '' --namn på fil du vill kontrollera

```

Resultat av SQL: n ovan ska se ur så här. Kolumnen XML_Data ska vara ifylld som i exemplet. Då är filen korrekt.



Kontrollera hur körningen går genom att köra koden nedan

```

SELECT TOP (1000) *
FROM [Audit].[dbo].[Audit_AdminEventlog__SP_Körtider]

```

När filerna har skickats om och de blivit inlästa kan vi ta bort dem från tabellen XMLData__Korrupta_Filer och från mappen på (P:).

!!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE Även om jobbet har lyckats kan en eller flera cache- och remotetabeller innehålla 0 rader. För att se att allt är okej ska kolumnen **status_övergripande_vpk_vecka** ha värdet 'OK'.

För att kontrollera att alla vyer fyllts på med nya data kör koden nedan i databasen GRAL_DRIFT_DW i Denodo. Rader där **rows_after** är 0 har inte fått ny data.

```

SELECT *
FROM "GRAL_DRIFT_DW".prenumeration_status_denodorefreshlog_db_denodo_prod
WHERE denodo_database = 'Datamart_ProKap' order by start_datetime desc;

```

För att fylla på med data i vyn som inte uppdaterats: Kör om jobbsteg som uppdaterar cache- eller remotetabellen som saknar rader. Se även till att köra loggstegen för start och avslut så att körningen loggas korrekt i prenumerationsrutinen. Får körningen inget avslut kommer jobbet inte startas nästa gång det ska köras.

ProKap lösningen består av flera jobb. Så kolla i jobben nedan för att hitta rätt steg att köra om (*ej jobb som rör dev- eller testmiljö*)

- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE_Avser_denodo_DEV_Miljö
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE_Specifikt_PIB_Vyer
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_CACHE_Specifikt_Pub_Vyer
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_operation_Remote
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_operation_Remote_Avser_Denodo_DEV_Miljö
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_sjukhusvård_Remote
- !!_Prenumeration_DENODO_Datarefresh_Datamart_ProKap_Veckoplanering_sjukhusvård_Remote_Avser_Denodo_DEV_Miljö
- !!_Prenumeration_ETL_ProKap_PlanerHistorik
- !!_Prenumeration_ETL_ProKap_DataMart_ProKap_Städad

Starta om jobb som uppdaterar cache- eller remotetabeller

Har ett jobb som uppdaterar cache- eller remotetabeller misslyckats behöver det startas om. Det är viktigt att vyerna i produktionsmiljön får nya data som de ska. Gäller jobbet dev- eller testmiljön är det inte lika bråttom att lösa problemet.

Vanliga orsaker till att ett cache- eller remotetabellers jobb misslyckas är att Denodo är nere eller att en vy tagits bort eller bytt namn. För att se felmeddelandet gå till mappen integration services catalogs på gral-prod-is01. Gå sedan till mappen SSISDB och slutligen PrenumerationsRutin_Denodo. Välj reports och sedan all executions. Hitta körningen som gick fel och där finns ett felmeddelande.

Starta om jobb när felet är åtgärdat

1. Kolla vilket steg som misslyckades. Öppna steget och gå till fliken 'configuration' för att se vilken databas och vy det gäller.
2. Starta om steget som misslyckades och se till att även alla loggsteg körs igen. Körs inte loggstegen korrekt riskerar jobbet att inte starta vid nästa tillfälle jobbet ska köras.