



PowerCenter Tips & Tricks

Lars Markhus

Informatica Produkteier / Affecto

BOB/NIB-konferansen 2011

Affecto



- **Hovedfokus: Business Intelligence**
- 900 ansatte i Norge, Sverige, Danmark, Finland, Estland, Latvia, Litauen og Polen
- **135** ansatte i Norge
- Etablert i Norge i **1997**
- Finansielt sterkt rustet med gode resultater
- Sterke **partnerskap** med SAP, Oracle, Microsoft, IBM og Informatica

INFORMATICA
NOW YOU KNOW™

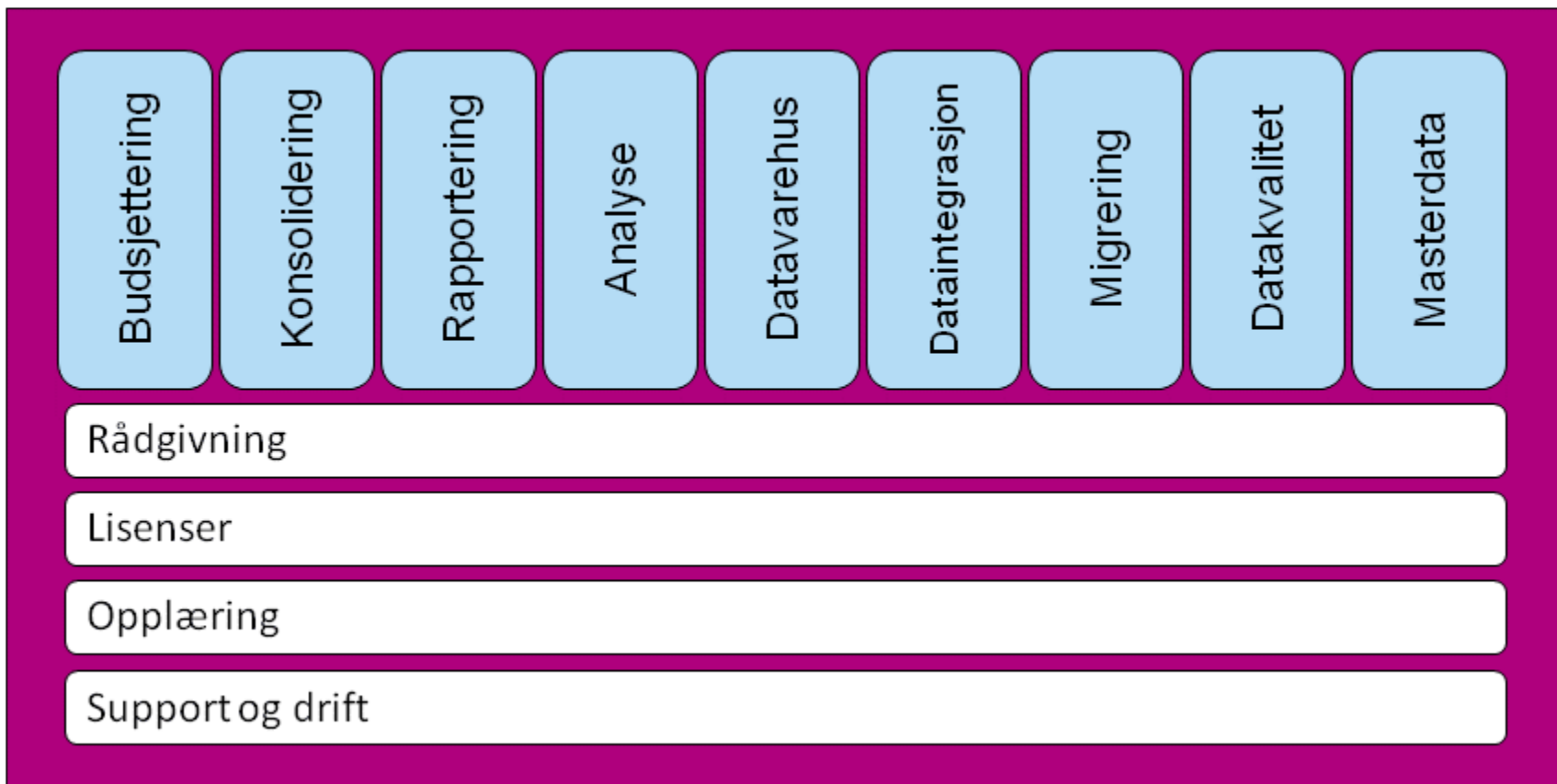
ORACLE®

Microsoft®

SAP®

Affecto
Be business intelligent

Totalleverandør av Business Intelligence



Agenda



- Feilhåndtering i PowerCenter
 - Vanlige feil / ”mangler” jeg ser der ute
 - Feilhåndtering i PowerCenter
 - Feillogging med PowerCenter
- Default Session Config
 - Overstyre default feilhåndtering på sesjoner
 - Styre andre default sesjonsparametre
- OS Profiler - Hva er det?
 - Kjøre under forskjellige OS brukere og standard pather
 - Prosjekt/område basert miljø konfigurasjon
 - Sikkerhet
- Bruk av Concurrent Workflows
 - Utfordring: Må prosessere mange like filer samtidig
Løsning: Parallellkjøre mange instanser av samme workflow
- Nyheter fra Informatica World 2010 - PowerCenter 9

Feilhåndtering i PowerCenter



- Vanlige feil i løsninger der ute
 - En kort oversikt over helt vanlige feil eller "mangler" jeg ofte ser der ute
 - Hvorfor nevne disse her? Fordi de er lett å overse og glemme. Forårsaker datafeil / tap av data.
 - Feilhåndtering i Workflows
 - Feilhåndtering i sesjoner (tasks)
 - Feilhåndtering i command tasks
 - Link conditions
- Mange muligheter for feilhåndtering
 - Ikke én riktig løsning, men mange varianter der ute
 - Hvordan støtter PowerCenter feilhåndtering?
 - Hvilke standard løsninger har den for logging av datafeil?

Vanlige feil

– Workflow - Link condition mangler



\$PrevTaskStatus=SUCCEEDED



Betingelse for at WF får gå vic

- **\$PrevTaskStatus=SUCCEEDED** (husk denne hvis ikke annet er tiltenkt)

Andre vanlige betingelser

- \$SrcFailedRows, \$SrcSuccessRows, \$TgtFailedRows, \$TgtSuccessRows, \$TotalTransErrors osv.
- Men noen ganger ønsker du faktisk at WF går videre uansett om foregående sesjon har feilet – bare dokumenter at dette er intensjonelt

Vanlige feil

- Sesjon / Task flagger ikke feil til Workflow



Fail parent if this task fails

- Vil flagge workflowen som feilet hvis denne sesjonen har feilet. Da synliggjøres dette på et høyere nivå.
- Hvis du schedulerer med PowerCenter Scheduler er ulempen at workflow faller ut av schedule og må re-scheduleres når du har rettet feilen.
- Noen ganger ønsker du ikke at en workflow skal feile. Da er det greit å utelate, men dokumenter da hvorfor du har gjort det.

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task:

Task type: Make reusable

Description:

Mapping name:

Resources:

Fail parent if this task fails: Fail parent if this task does not run: AND OR Disable this task

Fail parent if this task fails: Fail parent if this task does not run: AND OR Disable this task

Vanlige feil – Stop on error = 0

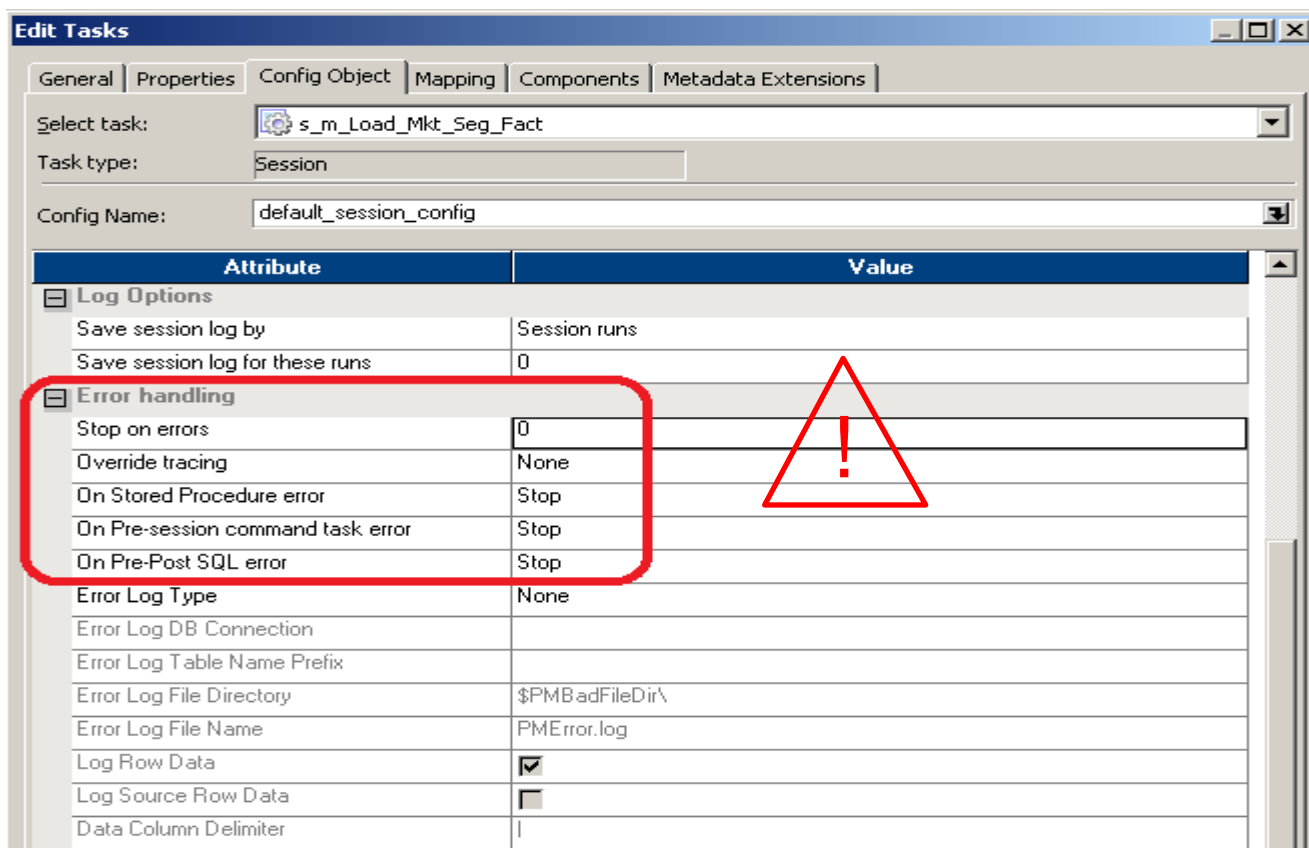
=> Sesjonen forkaster feilrader – data tapes!

Stop on errors = 0 er default!

Sesjonen stopper ikke!
Rader med datafeil forkastes og tapes!
Feilrader skrives til sesjonslogg.

Sett den til 1 – stopp på første feil! Da oppdager du i det minste feilen.

Alternativt sett den til 0 og beskriv hvorfor du ikke ønsker å stanse sesjonen på radfeil!



The screenshot shows the 'Edit Tasks' configuration window for the task 's_m_Load_Mkt_Seg_Fact'. The 'Error handling' section is highlighted with a red box, and a red warning triangle is overlaid on the table.

Attribute	Value
Log Options	
Save session log by	Session runs
Save session log for these runs	0
Error handling	
Stop on errors	0
Override tracing	None
On Stored Procedure error	Stop
On Pre-session command task error	Stop
On Pre-Post SQL error	Stop
Error Log Type	None
Error Log DB Connection	
Error Log Table Name Prefix	
Error Log File Directory	\$PMBadFileDir\
Error Log File Name	PMError.log
Log Row Data	<input checked="" type="checkbox"/>
Log Source Row Data	<input type="checkbox"/>
Data Column Delimiter	

Sesjon - feilhåndtering



- **Stop on error =**
 - bestemmer hvordan IS (Integration Service) håndterer ikke-fatale feil
 - **0** (er default) – IS stanser ikke sesjonen ved ikke-fatale feil
 - (Normal) skrives feilrader til sesjonsloggen
 - **1** (burde vært default) – IS stanser sesjonen ved første ikke-fatale feil
 - Første feilrad skrives til sesjonsloggen
 - **n** – IS stanser sesjonen etter n ikke-fatale feil
 - Alle feilrader tom den n'te rad skrives til sesjonsloggen
 - Kan være mange gode grunner til å gjøre det!
- **Eks ikke-fatale feil:**
 - Oftes feil på data i en rad, f.eks:
 - konverteringsfeil (`to_date('31.13.2011','DD.MM.YYYY')`)
 - numerisk overflow
 - divisjon på 0 i en transformasjon
 - Forsøk på inserte/update en rad i target tabell som bryter constraints på tabellen.

Vanlige feil og hvorfor nevne de her



- Jeg har sett mange slike eksempler med Stop on error = 0 der ute. Noen intensjonelle, andre ikke.
- Standard konfigurasjon av en sesjon har stop on error = 0 og sesjonen går da i succeeded selv om rader forkastes.
- Uansett tapes data. Feil flagges ikke. Vanskelig å oppdage.
- Et sted var det trafikkdata som kom inn fra mange kilder og skulle bare brukes for statistikkformål. Noen få rader var helt korrupte, kunne uansett ikke brukes og ble bare forkastet. De ble ikke logget. Det var helt greit for formålet.
- Andre eksempler er økonomiske transaksjonsdata hvor noen få rader hadde spesialtegn som medførte radfeil. Data skulle brukes til avstemming av tall i økonomisystem. Regnskap likte ikke at det var avvik i tallene. Vanskelig å spore når feilet rad var forkastet. Et forklaringsproblem.
- Om litt skal vi se på hvordan PowerCenter kan logge radfeil for deg og hvordan de kan knyttes opp til aktuelle workflow, sesjon og kjøring som logget dem.

Vanlige feil - Command task



General | Properties | **Commands**

Select task: Command

Task type: Command

Use the controls on the right to change the execution order of the commands. Only commands tasks can add/delete/copy-paste commands.

	Name	Command
1	Run script A	/d1/projectX/script/script_a.sh
2	Run script B	/d1/projectX/script/script_b.sh
3	Run script C	/d1/projectX/script/script_c.sh

Her kjøres tre skript i sekvens. Hvis du ikke har satt Fail task if any command fails under Properties vil PowerCenter ignorere feilen og eksekvere neste skript. Command tasken går da uansett ut med Succeeded uansett om et eller flere skript har feilet.

General | **Properties** | Commands

Select task: Command

Task type: Command

Attribute

Fail task if any command fails

Recovery Strategy: Fail task and continue workflow

Sett også Fail parent hvis du ønsker å merke workflowen som feilet hvis dette oppstår.

General | Properties | Commands

Select task: Command

Task type: Command Make reusable

Description:

Fail parent if this task fails

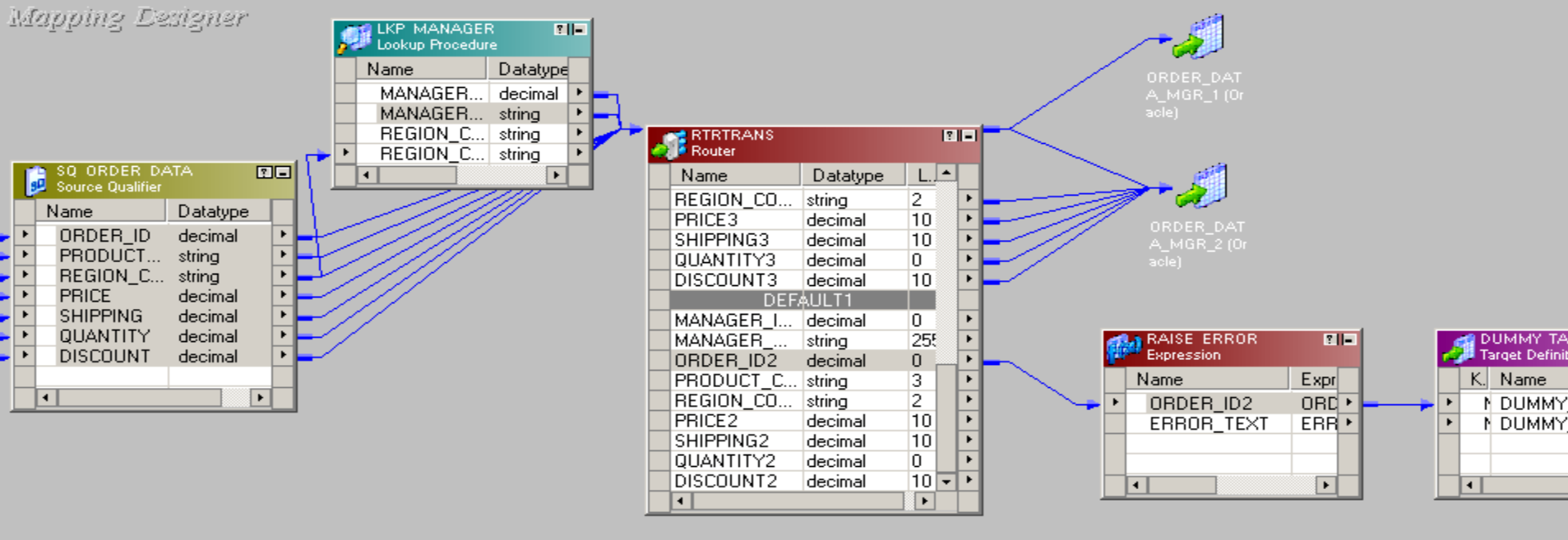
Fail parent if this task does not run

Disable this task

Create the input links as: AND OR

Vanligste feil

- Router mangler håndtering av default gruppen



- Hvis du har kompleks logikk i router gruppene og kriteriet ikke treffer disse gruppene vil raden havne i default gruppen. Hvis du ikke har håndtert (ikke har linke den videre) så vil raden(e) tapes helt. Hendelsen logges ikke og radene logges heller ikke.
- Som hovedregel: Håndter alltid default gruppen. Eks. Bruk en expression transformasjon (RAISE_ERROR) som flagger feil hvis en rad havner her:
- **ERROR('ERROR: ORDER_ID ' || TO_CHAR(ORDER_ID2) || ' havnet uventet i default router gruppen.')**

Task (sesjon) (3)

– PowerCenter alternativer for feillogging



Error Log Type:

- **None** – skriver til sesjonsloggen
- **Flat File** – Skriver feilrader til fil
- **Relational Database** – Skriver feilrader til loggtabeller på DB

The screenshot shows the 'Edit Tasks' dialog box with the following configuration:

- General | Properties | **Config Object** | Mapping | Components | Metadata Extensions
- Select task: s_m_Load_Mkt_Seg_Fact
- Task type: Session
- Config Name: default_session_config

Attribute	Value
Log Options	
Save session log by	Session runs
Save session log for these runs	0
Error handling	
Stop on errors	0
Override tracing	None
On Stored Procedure error	Stop
On Pre-session command task error	Stop
On Pre-Post SQL error	Stop
Error Log Type	None
Error Log DB Connection	
Error Log Table Name Prefix	
Error Log File Directory	\$PMBadFileDir\
Error Log File Name	PMErrror.log
Log Row Data	<input checked="" type="checkbox"/>
Log Source Row Data	<input type="checkbox"/>
Data Column Delimiter	

Session

– Error Log Type = Flat File



Skriver loggfil til
filsystemet
under
katalogen
\$PMBadFileDir\

Attribute	Value	UnOverri...
Log Options		
Save session log by	Session runs	
Save session log for these runs	0	
Error handling		
Stop on errors	1000	Revert
Override tracing	None	
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	
On Pre-Post SQL error	Stop	
Error Log Type	Flat File	Revert
Error Log DB Connection		
Error Log Table Name Prefix		
Error Log File Directory	\$PMBadFileDir\	
Error Log File Name	PMErrror.log	
Log Row Data	<input checked="" type="checkbox"/>	
Log Source Row Data	<input checked="" type="checkbox"/>	Revert
Data Column Delimiter		

Error Log File (1/2)

- Workflow / Sesjons info



- *****

- Repository GUID: a192d9fe-031c-45f7-8251-325858d30739
- Repository: REP_SVC_EDW_DEV
- Folder: Developer_20
- Workflow: wfl_TEST
- Session: s_m_Load_Tgt
- Mapping: m_Load_Tgt
- Workflow Run ID: 128
- Worklet Run ID: 0
- Session Instance ID: 36
- Session Start Time: 02/08/2011 00:35:46
- Session Start Time (UTC): 1297121746

Eksempel på hvordan PowerCenter logger til feilfil. Her med oversikt over hvor jobben kjørte, hvilken workflow og kjøring, hvilken sesjon og kjøre id. Du kan knytte datafeil logget her mot respektive kjøring i PowerCenter logger.

Se neste side for Data forekomster...

Error Log File (2/2)

- Dataforekomster



- Transformation||Transformation Mapplet Name||Transformation Group||Partition Index||Transformation Row ID||Error Sequence||Error Timestamp||Error UTC Time||Error Code||Error Message||Error Type||Transformation Data||Source Mapplet Name||Source Name||Source Row ID||Source Row Type||Source Data
- ORDER_DATA_MGR_2||N/A||Input||1||3||1||02/08/2011 00:35:48||1297121748||8361||Error loading row to target table [ORDER_DATA_MGR_2]. Error message [\nORA-01400: kan ikke sette inn NULL i ("EDW"."ORDER_DATA_MGR_2"."MANAGER_ID")\n\nDatabase driver error...\nFunction Name : Execute\nSQL Stmt :
- INSERT INTO
ORDER_DATA_MGR_2(MANAGER_ID,MANAGER_NAME,ORDER_ID,PRODUCT_CODE,REGION_CODE,PRICE,SHIPPING,QUANTIT Y,DISCOUNT) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?) \nDatabase driver error...\nFunction Name : Execute Multiple\nSQL Stmt : INSERT INTO
ORDER_DATA_MGR_2(MANAGER_ID,MANAGER_NAME,ORDER_ID,PRODUCT_CODE,REGION_CODE,PRICE,SHIPPING,QUANTIT Y,DISCOUNT) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
||2||N:|N:|D:125.00000000000000|D:ANO|D:QX|D:14.75000000000000|D:13.96000000000000|D:0.00000000000000|D:-
4.7600000000000000||N/A||SQ_ORDER_DATA||3||0||D:125.00000000000000|D:ANO|D:QX|D:14.75000000000000|D :13.96000000000000|D:0.00000000000000|D:-4.760000000000000
- Her logges feilmeldingene og raden som feilet
- Feilen er knyttet opp mot workflow/sesjon run_id og rad nr i kjøringen.
- Nyttig for sporing, men filen er mer komplisert å bruke maskinelt.
- PowerCenter kan også logge til DB. Se neste slide.

Task (sesjon)

– Error Log Type = Relational Database



Skriver logfil til DB:

- Oppgi DB Connection

• Evt prefiks for loggtabellene – eks kode for forretningsområde

Logger til følgende tabeller:

- PMERR_DATA
- PMERR_MSG
- PMERR_SESS
- PMERR_TRANS

Attribute	Value	UnOverri...
Log Options		
Save session log by	Session runs	
Save session log for these runs	0	
Error handling		
Stop on errors	1	Revert
Override tracing	None	
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	
On Pre-Post SQL error	Stop	
Error Log Type	Relational Database	Revert
Error Log DB Connection	Relational:EDW	Revert
Error Log Table Name Prefix		
Error Log File Directory	\$(MBAAdminDir)	
Error Log File Name	PMError.log	
Log Row Data	<input checked="" type="checkbox"/>	
Log Source Row Data	<input checked="" type="checkbox"/>	Revert
Data Column Delimiter		

Error Log Type = Relational Database

- Eksempler



- **PMERR_SESS** – Metadata om sesjonen som meldte feilen.
- **PMERR_MSG** – Lagrer alle feilmeldingene knyttet til workflow, mapping, transformasjon etc
- **PMERR_TRANS** – Ved transformasjonsfeil logger den metadata om transformasjonen som f.eks. datatypen på transformasjonsportene.
- **PMERR_DATA** – Når Integrasjonstjenesten treffer en radfeil skriver den raden til PMERR_DATA. Hver kolonne flagges med status for om feltet var gyldig, overflow, trunkert, ugyldig osv.

.

Eks: PMERR_DATA



COLUMN	VALUE
REPOSITORY_GID	a192d9fe-031c-45f7-8251-325858d30739
WORKFLOW_RUN_ID	133
WORKLET_RUN_ID	0
SESS_INST_ID	36
TRANS_MAPPLET_INST	N/A
TRANS_NAME	EXPTRANS
TRANS_GROUP	Input
TRANS_PART_INDEX	1
TRANS_ROW_ID	2
TRANS_ROWDATA	D:2.0000000000000000 T:LARS MA D:124.00000000000000 D:GRE D:WA D:12.45000000000000 D:1.34000000000000 00 D:0.0000000000000000 D:-0.5600000000000000
SOURCE_ROW_ID	2
SOURCE_ROW_TYPE	0
SOURCE_ROW_DATA	D:124.00000000000000 D:GRE D:WA D:12.45000000000000 D:1.3400000000000000 D :0.0000000000000000 D:-0.5600000000000000
LINE_NO	1

- Data Column Indicator: D – Valid, N – Null, T – Truncated, B – Binary, U – data Unavailable
- Her logges feilraden med all kobling tilbake til aktuelle sesjon og kjøring.

Eks: PMERR_MSG



- "Transformation [EXPTRANS] had an error evaluating output column [PRICE_BY_UNIT]. Error message is [<<Expression Error>> [/]: divisor is zero... f:(f:12.45 / f:0)]."
- "There is an error in the port [PRICE_BY_UNIT]: The default value for the port is set to: ERROR(<<Expression Error>> [ERROR]: transformation error... nl:ERROR(u:'transformation error'))."
- An error occurred moving data from the transformation EXPTRANS: to the transformation ORDER_DATA_MGR_2.

Oppsummering: Feilhåndtering



- PowerCenter har mange muligheter for å håndtere feil
- Ulike metoder anvendes i ulike prosjekt – dokumenter deres standard
- Se opp for vanlige feil
 - dokumenter en standard for feilhåndtering hos deg
 - dokumenter i jobbene når du avviker fra standard og hvorfor
- PowerCenter tilbyr
 - Flere måter for feillogging
 - Workflow / Sesjonslogger
 - Standardiserte feillogger på fil eller DB
 - Gir sporbarhet (Audit krav)
- Hvordan kan man gjøre om standard feilhåndtering i sesjoner m.m.? – Neste slide: Default Session Config

Default Session Configuration



- Et objekt på folder nivå (også versjonert hvis dere har versjonert repository)
- Kan ha flere session config
- Konfigurerer egenskaper ved sesjoner:
 - Minne størrelser
 - Log opsjoner
 - Feilhåndtering
 - Partisjoneringsopsjoner

Edit Session Configuration

General Properties

Name: default_session_config

Type: Configuration

Attribute	Value
Advanced	
Constraint based load ordering	<input type="checkbox"/>
Cache LOOKUP() function	<input checked="" type="checkbox"/>
Default buffer block size	Auto
Line Sequential buffer length	1024
Maximum Memory Allowed For Auto ...	512MB
Maximum Percentage of Total Memo...	5
Additional Concurrent Pipelines for L...	2
Custom Properties	
Pre-build lookup cache	Auto
DateTime Format String	MM/DD/YYYY HH24:MI:SS.US
Pre 85 Timestamp Compatibility	<input type="checkbox"/>
Log Options	
Save session log by	Session runs
Save session log for these runs	5
Error handling	

Default Session Config



Objektet er litt skjult!
Du finner det i
Workflow Manger
øverst til høyre:



Hvis du har versjonert repository må du sjekke ut objektet! Husk å sjekke det inn for at det skal ta effekt!

NB! Du kan ikke se om session config objektet er utsjekket eller ikke slik du normalt ser folder nivå objekter i Navigator vinduet i Workflow Manager som for eksempel mappinger. Bruk Find Checkouts funksjonen på folderen! Vanlig at man glemmer å sjekke det inn!

Session Config er imidlertid et objekt listet opp i navigatorvinduet i Repository Manager. Der synes det.

Konfigurer minne, lookup cache, sesjonslogg og feilhåndtering.

Edit Session Configuration

Attribute	Value
Advanced	
Constraint based load ordering	<input type="checkbox"/>
Cache LOOKUP() function	<input checked="" type="checkbox"/>
Default buffer block size	Auto
Line Sequential buffer length	1024
Maximum Memory Allowed For Auto ...	512MB
Maximum Percentage of Total Memo...	5
Additional Concurrent Pipelines for L...	2
Custom Properties	
Pre-build lookup cache	Auto
DateTime Format String	MM/DD/YYYY HH24:MI:SS.US
Pre 85 Timestamp Compatibility	<input type="checkbox"/>
Log Options	
Save session log by	Session runs
Save session log for these runs	5
Error handling	
Stop on errors	1
Override tracing	None
On Stored Procedure error	Stop
On Pre-session command task error	Stop
On Pre-Post SQL error	Stop
Error Log Type	None
Error Log DB Connection	
Error Log Table Name Prefix	

Sess Config i ulike miljø - Hvordan?



- Ulike miljø har ulike ressurser tilgjengelig (minne, cpu, etc) og ulike krav til feilhåndtering og logging
- Bruk session config for å gi sesjonen ulike egenskaper i miljø med ulike tilgjengelige ressurser:
 - Eks: Begrensede ressurser i utv/test – Mer ressurser i produksjon
- Eks: Ulike miljø kan ha ulike krav til logging:
 - Eks: UTV – enhetstesting – ønsker ikke bevare / har ikke plass på DB
 - TEST – Ønsker sporbarhet sessjonslogger siste n-kjøringer og alle feiltrader til DB
 - PROD – Audit krav at alle sesslogger bevares – Logger alle feiltrader til DB.

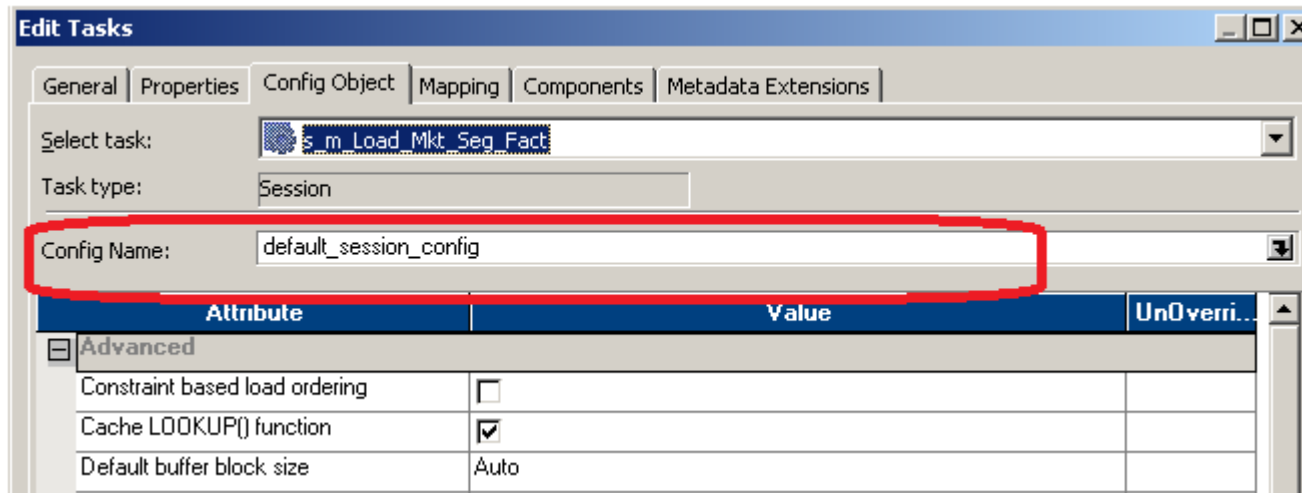
Default_Sess_Config	UTV	TEST	PROD
Max Memory Allowed	512 MB	512MB	1024MB
Save log by	Session runs	Session runs	Timestamp (*)
Num logs	5	20	(*) Audit krav: Logger resirkuleres ikke – alle bevares
Error log type	File (ikke plass i DB)	Relational DB	Relational DB

Eks: Flere typer egendefinerte Session Config i samme folder



Eks:	Sess_Config_Small	Sess_Config_Large	Sess_Config_XLarge
Max Memory Allowed	512 MB	1024MB	4096MB

Du velger sesjonens konfigurasjon under Config Object tab:

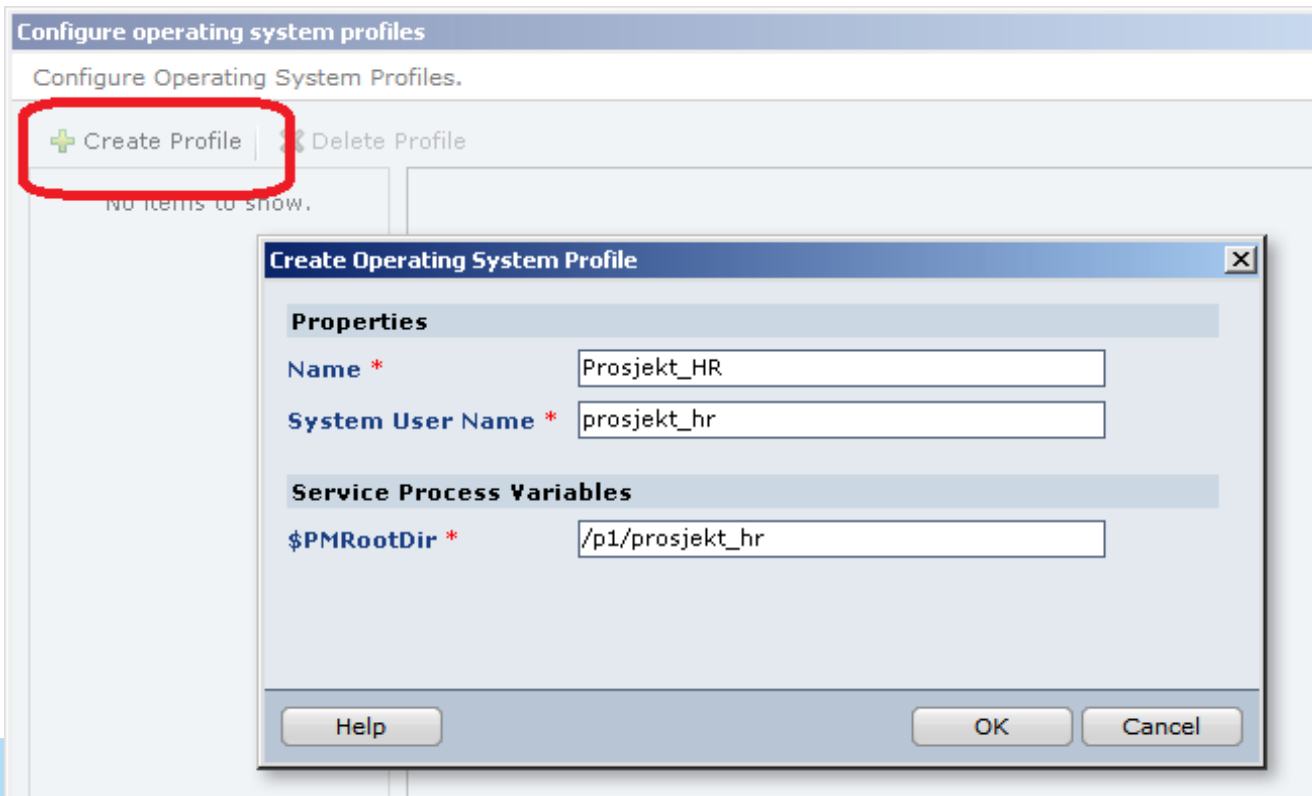
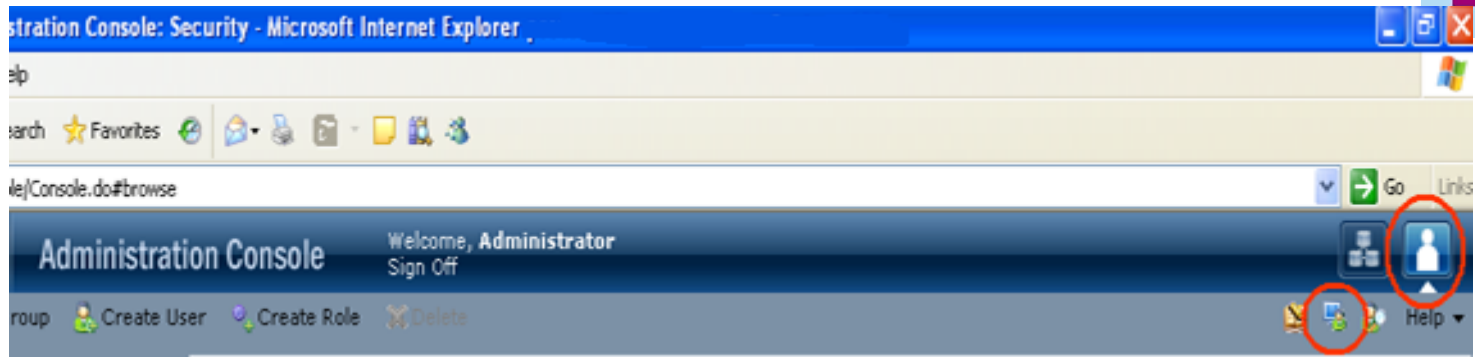


OS Profiler – Hva er det?

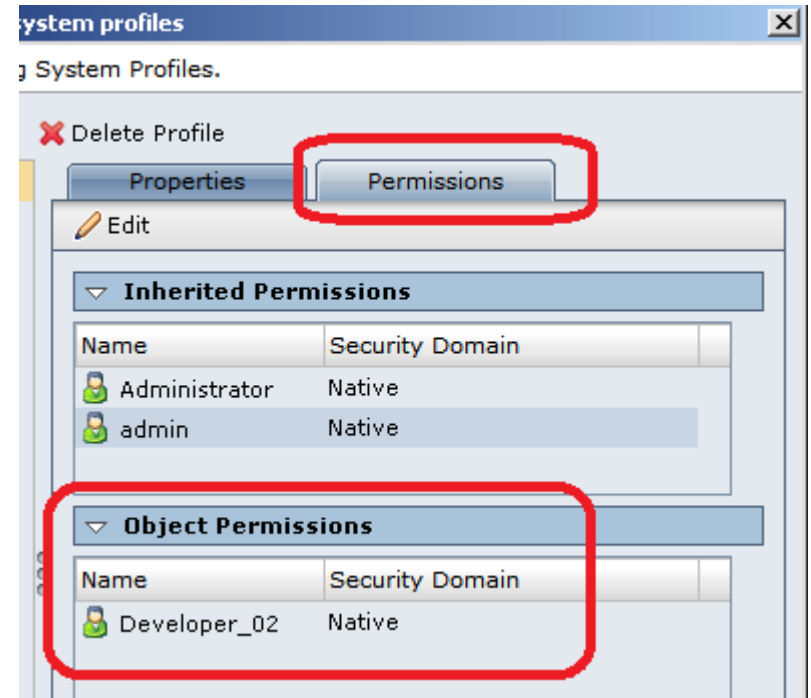
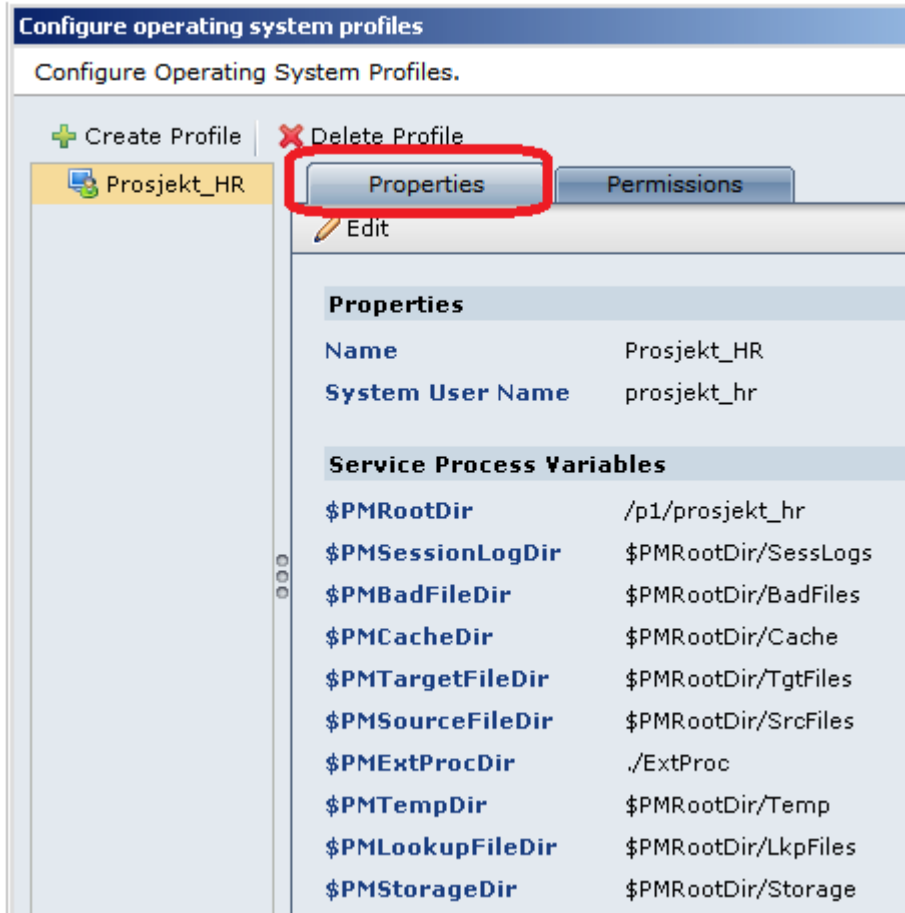


- Mulig å kjøre PowerCenter Workflows under ulike operativsystembrukere (OS brukere) på samme server (node)
- Kan kjøre Workflows under annen OS bruker enn Informatica installasjonen selv kjører under. Eks:
 - Utviklere får ikke tilgang til Informatica Server install – hindrer utvikler feil på server installasjonen
- Gjør det mulig å definere egne OS brukere / filsystem for ulike prosjekter som kjører på samme server (node).
 - Skille ut sensitive data under egne brukere med særskilte tilganger. Eks: HR (personopplysning) / Børssensitiv informasjon
 - Robustifisering: Bruk dedikerte disk-mounts for ulike prosjekter. Hindrer at et prosjekt fyller diskene og tar ned hele systemet for andre prosjekter
- OS Profiler settes / gjelder på foldernivå i PowerCenter

OS Profiler – Hvordan opprettes de?



OS Profiler – Hvordan defineres de?



- Dedikert OS bruker – sikkerhet
- Dedikert filstruktur - robusthet

• Informatica bruker / gruppestyrt tilgang

OS Profiler – settes på folder nivå



- I Repository Manager -> Folder -> Edit
- OS Profilen settes på foldernivå
 - Gjelder da alle workflows i denne folder

The screenshot shows the 'Folder Properties' dialog box with the 'Permissions' tab selected. The 'Name' field contains 'HR_integrasjon' and the 'Descriptions' field contains 'Prosjekt HR folder'. The 'OS Profile' dropdown menu is highlighted with a red box and is set to 'Prosjekt_HR'. The 'Options' section has the 'Allow Shortcut' checkbox checked. The 'Status' dropdown is set to 'Active'. The dialog box has 'OK', 'Cancel', and 'Help' buttons at the bottom.

OS Profil – Hvordan skru det på



- Enables på Integrasjonstjenesten:
 - Gå inn Informatica Administrator Consollet
 - Under Integrasjonstjenesten -> Advanced Properties -> Sett “Use Operating System Profiles = Yes” in Admin Console under “Your Integration Service”, Advanced Properties
 - Må restarte Integrasjonstjenesten
- Unix/Linux: Må endre egenskaper på en PowerCenter Service exec filen **pmimpprocess** for at PowerCenter skal klare å kjøre under ulike OS brukere:
 - [PowerCenter install path]/server/bin/pmimpprocess må eies av user id=0 (root). Bare root kan gjøre dette: **chown root pmimpprocess**
 - The sticky bit must be set
 - for the user and group: **chmod og+s pmimpprocess**; and
 - execute permission enabled for other: **chmod o+rx pmimpprocess**
 - OS Profilens OS bruker må være medlem av samme gruppe som pmimpprocess kjører under.
Example:
`$ls -l pmimpprocess`
`-r-sr-sr-x 1 root support 32768 Dec 11 09:12 pmimpprocess`
 - **NB! Du må sjekke denne innstillingen også etter hver oppgradering av Informatica! Når du lager ny INFA_HOME for neste versjon blir ikke disse rettighetene satt på den nye installasjonen. Du må utføre det selv.**

OS Profiler – Oppsummert



- Muligheter for å definere prosjektbaserte
 - Filstrukturer og \$PM-variable
 - Eks: kalle script med variabelbasert path fra CMD tasker: \$PMRootDir/script/run_a.sh – nyttig hvis du har forskjellig root path i ulike prosjekt / miljø pga krav fra drifts organisasjonen.
 - Eller du kjører ulike instanser av koden i forskjellige PowerCenter foldre på samme node for ulike formål som utvikling, test osv.
 - Kan da kjøres under ulike OS brukere som også kan hindre lekkasje mellom prosjekt når det kjøres mange varianter på samme maskin.
- Muligheter for å konfigurere sikkerhet / tilgang
 - OS nivå / filsystemnivå
 - Applikasjonsnivå - PowerCenter folder nivå
 - Tilgangsstyrt med personlige brukere / grupper
 - Gir mulighet for inndeling i prosjektbaserte sikkerhetsnivåer
- Legger du til LDAP / AD security med personlig pålogg på Unix og su / sudo opp til applikasjonbruker for prosjektet oppnås:
 - Tilgangskontroll på applikasjonbruker
- Legger du til audit logging på personlig brukers su / sudo til applikasjonsbruker får du sporbarhet
- NB! I 8.6.1 er OS Profiler fortsatt bare tilgjengelig for Unix servere, men dessverre ikke Windows servere.

Concurrent Workflows

- Hva er det?



- Muliggjør kjøring av parallelle instanser av samme workflow
 - Instansene kan startes og stoppes uavhengig av hverandre
 - Kan kjøre i parallell
- Med samme instansnavn
 - Kjører samme workflow i parallell med samme source, target og parameter variable
 - Typisk ved lesing fra real-time kilder som meldingskjører og web-services
- Med forskjellige navngitte instanser
 - Kan kjøre instanser i parallell med forskjellige source, target og variable
 - For eksempel bruke samme jobb til å laste opp forskjellige schema / tabeller med ulike datasett i parallelle kjøring basert på samme kode (workflow)
 - Navngitte instanser kan ha persistente workflow-/mappingvaribale mellom hver kjøring

Concurrent workflows

- Case

- Behov for å parallell prosessere mange store filer samtidig
- Filene kan ankomme på ulike tidspunkt og må behandles så fort hver enkelt fil er mottatt
- Må kunne skille datasettene når vi parallell prosesserer

GRUPPE_A (20 kunder)

GRUPPE_B (130 kunder)

GRUPPE_B126 (15 kunder)

GRUPPE_T45 (3 kunder)

GRUPPE_X11 (60 kunder)

Løsning: PowerCenter Concurrent Workflows



- 5 likt formaterte filer skal behandles samtidig
- Mapping / session / workflow definert bare én gang
- 5 navngitte concurrent workflow instanser – samme kode anvendt i samtidige instanser
- 5 distinkte datasett – en for hver instans
- Parameterfiler opprettes for hver instans og benyttes for å sette workflow-/mappingparametre. Disse brukes for å styre hvilket datasettet som behandles i hver mapping.

The screenshot shows the 'Edit Workflow' dialog box with the 'Configure Concurrent Execution' checkbox checked and circled in red. The 'Concurrent Execution Configuration' dialog is open, showing the following options:

- Allow concurrent run with same instance name
- Allow concurrent run only with unique instance name

The 'Concurrent Execution Configuration' dialog also displays a table with the following data:

Instance Name	Parameter File
WFL_TESTINSTANCE1	/project/param_wfl_testinstance1.par
WFL_TESTINSTANCE2	/project/param_wfl_testinstance2.par
WFL_TESTINSTANCE3	/project/param_wfl_testinstance3.par
WFL_TESTINSTANCE4	/project/param_wfl_testinstance4.par
WFL_TESTINSTANCE5	/project/param_wfl_testinstance5.par

Hvordan identifisere et korrekt datasett med korrekt concurrent WF instans?



PowerCenter Mapping variable identifiserer WF og INSTANS:

- `$PMWorkflowName`
- `$PMWorkflowRunInstanceName`

Kobler dette mot styretabeller som inneholder hvilket datasett denne instansen skal behandle

DATASET_WF_INSTANCE:

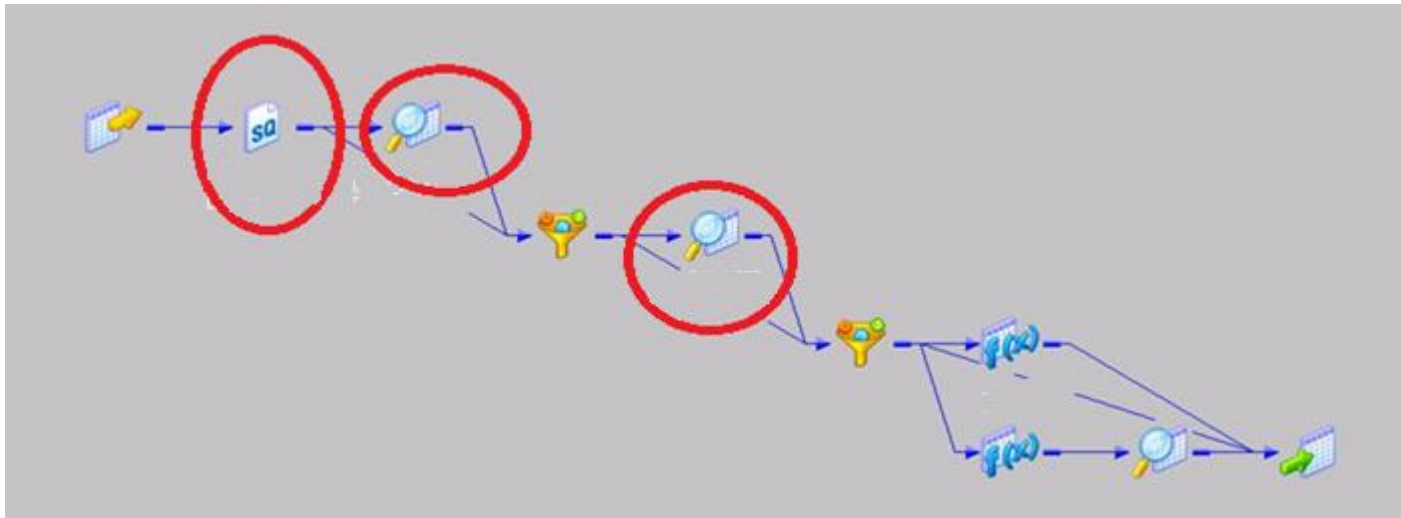
DATA_SET	WF_NAME_INSTANCE
GRUPPE_A	Wfi_TESTINSTANCE1
GRUPPE_B	Wfi_TESTINSTANCE2
GRUPPE_B126	Wfi_TESTINSTANCE3
GRUPPE_T45	Wfi_TESTINSTANCE4
GRUPPE_X11	Wfi_TESTINSTANCE5

Logikk i mapping

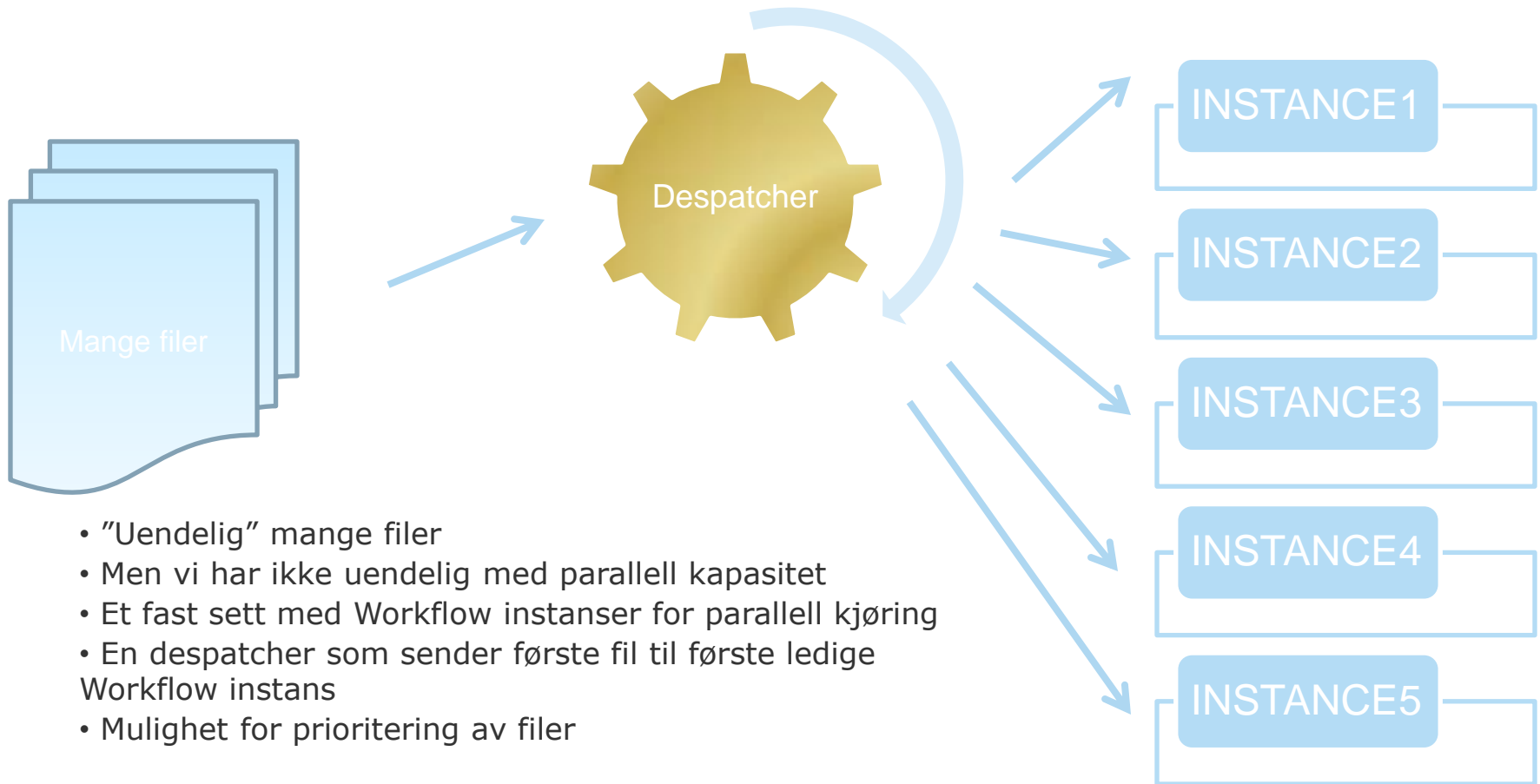


- SQL Override i alle **source kvalifiere og lookups** for å begrense datasettet til kun det som tilhører denne navngitte instansen av workflowen:

```
SELECT *  
FROM KUNDE, DATASETT_WF_INSTANCE  
WHERE KUNDE.DATA_SET = DATASETT_WF_INSTANCE.DATA_SET  
AND DATASETT_WF_INSTANCE.WF_NAME_INSTANCE =  
'$PMWorkflowName$PMWorkflowRunInstanceName'
```



Despatcher håndterer start av workflows



Concurrent workflows og parallellitet på Oracle DB tabeller



- Alle datasett håndteres i en og samme tabell i ulike partisjoner
 - en kode for tabellen – mindre vedlikehold
 - flere partisjoner - en per datasett / workflow instans
- Hvert datasett håndteres i egen partisjon på en Oracle DB
- Hver tabell partisjonert pr DATA_SET
- PK på tabell utvidet med DATA_SET

Partisjonering:

- Hver partisjon har lokal indeks PK fra kilde + DATA_SET
- Delete på partisjonerte temp tabeller tar alt for lang tid
- Trunkering av partisjoner svært raskt og mulig samtidig med at vi laster andre partisjoner for andre datasett

Løsning:

- Prosedyre kalles fra mapping og trukerer partisjon for relevant DATA_SET
- Bruker også sesjonsvariable
 - **\$PMWorkflowName** for å identifisere target tabellen
 - **\$PMWorkflowRunInstanceName** for å identifisere hvilket data_set / partisjoner som skal trukeres

Oppsummering: Concurrent Workflows



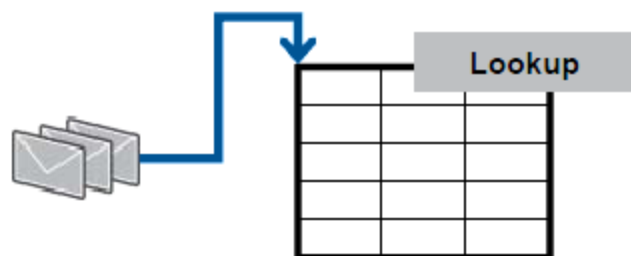
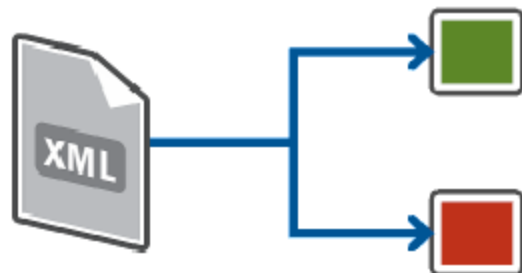
- En versjon av koden
 - en workflow pr kilde
 - mindre vedlikehold av kode
 - Instansstyrt prosessering av datasett
- Mulig å kjøre samme workflow i mange parallelle instanser for flere uttrekk fra samme kilde
- Styring av datasett i hver instans
 - `$PMWorkflowName`
 - `$PMWorkflowRunInstanceName`

Nyheter i PowerCenter 9



- Fra Informatica World 2010
- Ny støtte real-time mappinger og håndtering av ugyldige XML
- LOOKUP:
 - Lookup policy on multiple match = Usa ALL values – gjør transformasjonen aktiv
 - Dynamisk lookup cache – re-cache hvis inkommende data er nyere enn cache tidspunktet
 - Husk den eksisterende: Report Error on multiple match!
- Mulighet for rotering av sesjonslogger
- Pipeline lookup – muliggjør partisjonering av lookup source for raskere caching av lookup
- Mapping Architect for Visio – nye muligheter - generer dine mappinger fra Visio templates

24x7 Operations



WHERE incoming data row timestamp
newer than cached timestamp

Handle invalid XML

If XML contains errors, continue processing valid data and reroute the errors for remediation

NEW

Dynamic lookup cache

Incoming real-time transactions not in chronological order can result in older data overwriting newer data. Use expression logic to conditionally update lookup cache

NEW

Session log rotation

Avoid generating very large session logs. Instead users can control the rotation of session logs

NEW

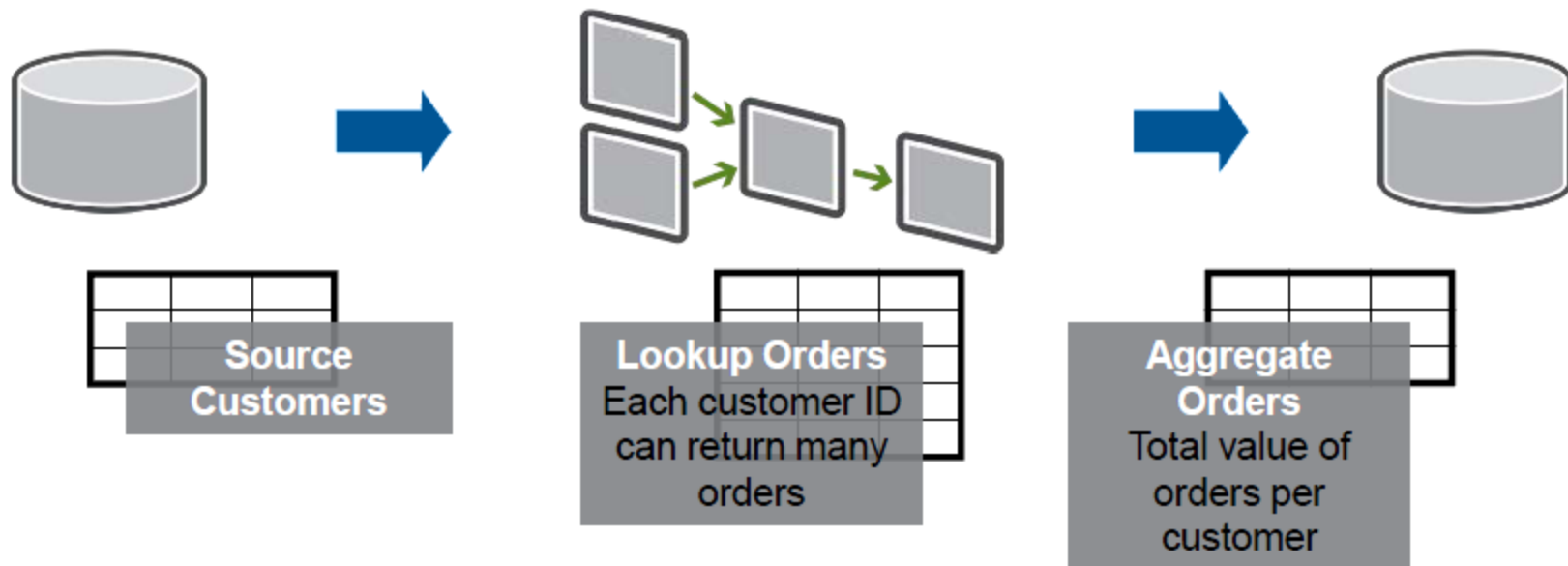


PowerCenter



Session Logs

Lookup Transformations



Lookup Policy on Multiple Match:

- Report Error
- Use First Value
- Use Last Value
- Use Any Value
- **Use All Values** **NEW**

Data may be too large to build a lookup cache of everything



Instead, build cache using a subset of the data

```
WHERE order_date > to_date('01-01-2005')
```

NEW

Transformation Tips

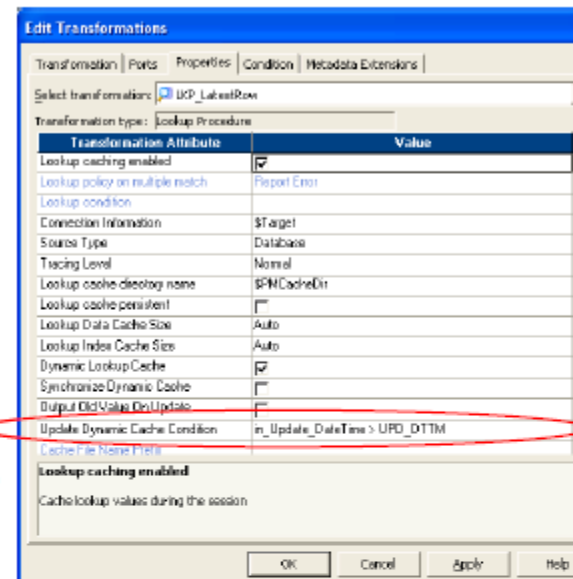
Lookups **INFORMATICA**[®]

- **Cache Updates**

- Update the dynamic lookup cache with results of an expression

Use Case: Data is not in chronological order

Add WHERE incoming row timestamp > cached timestamp



- **SQL Overrides for Uncached Lookups**

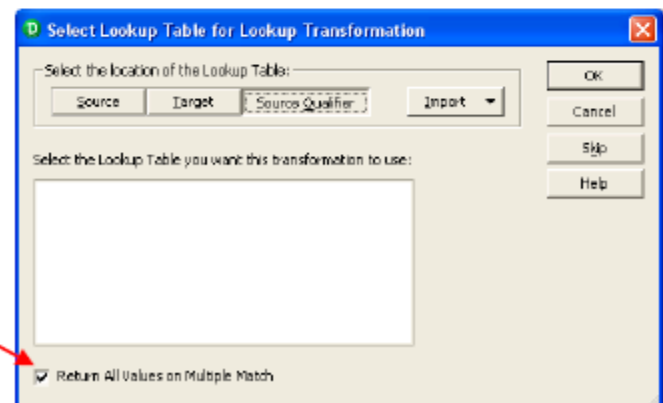
- You must choose the Use Any Value on Lookup Policy on Multiple Match condition to override the lookup query for an uncached lookup

- **Multiple Rows Return**

Use Case: Aggregate customer orders and store the total value

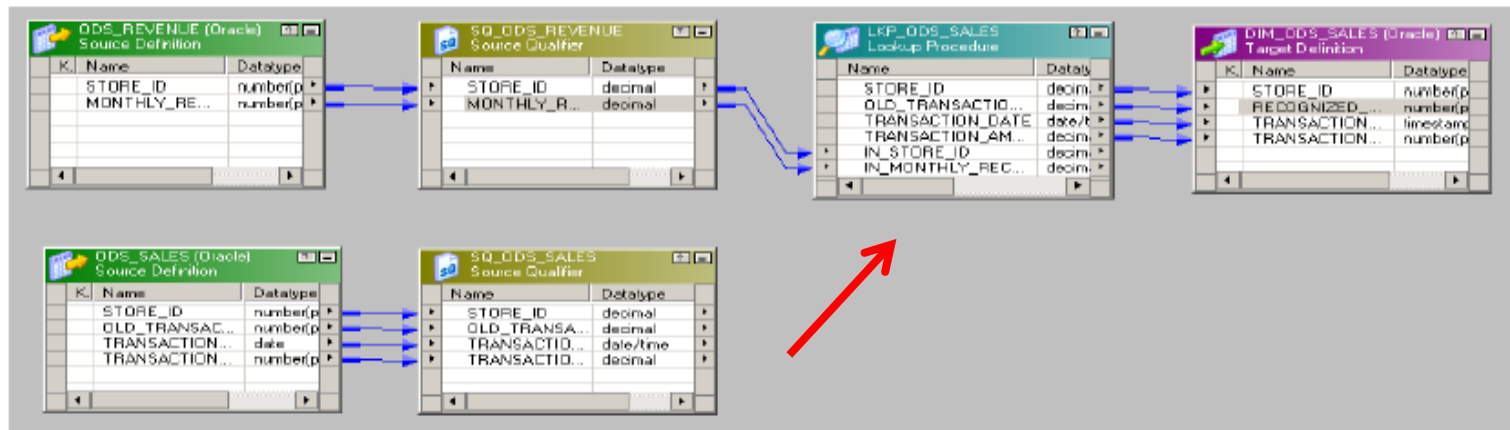
- **Database Deadlock Resilience**

- NumOfDeadlockRetries
- DeadlockSleep



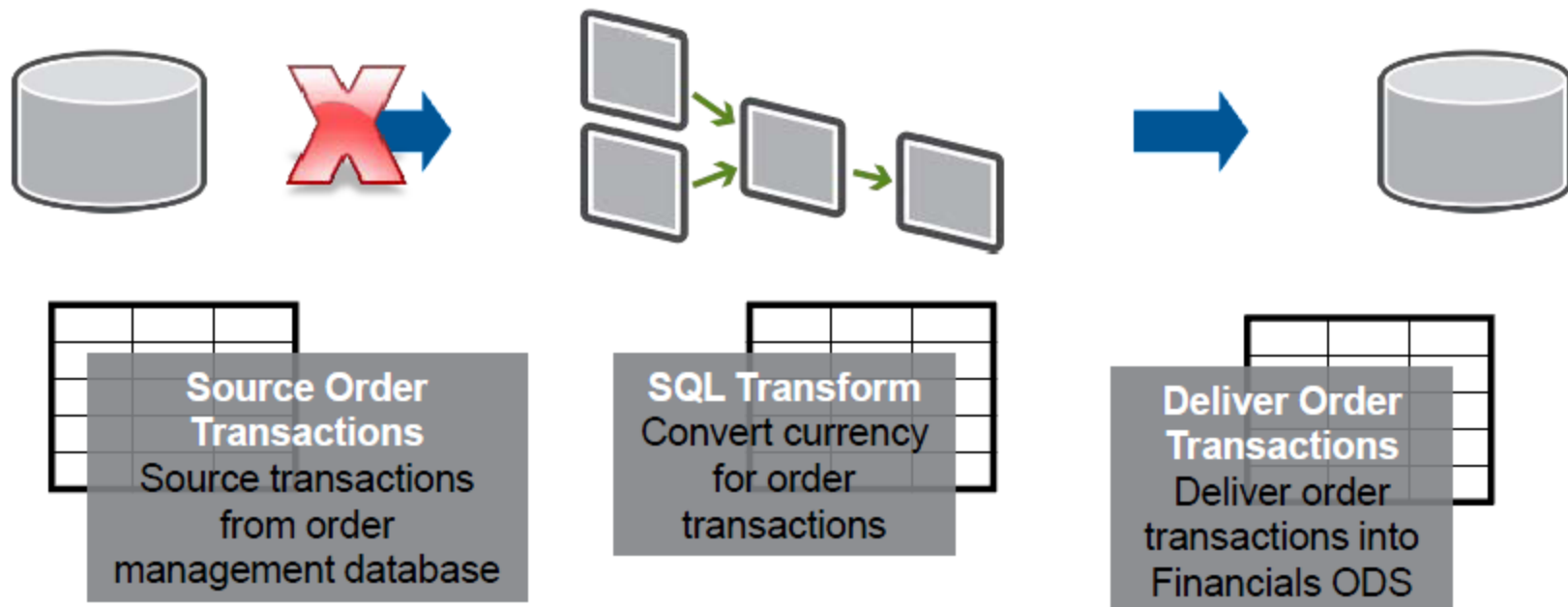
Pipeline Lookup

- Perform a lookup on an application source that is not a relational table or file
 - Partial pipeline contains Source & Source Qualifier but no target
 - Integration Service reads source data and passes to Lookup Transformation to create cache
 - Create partitions to improve performance



SQL Transform

Guaranteed exactly-once semantics



A passive transformation does not change the number of rows

An active transformation may change the number of rows

Set active or passive at design time

NEW

Guarantee exactly-once semantics for passive transformations

NEW

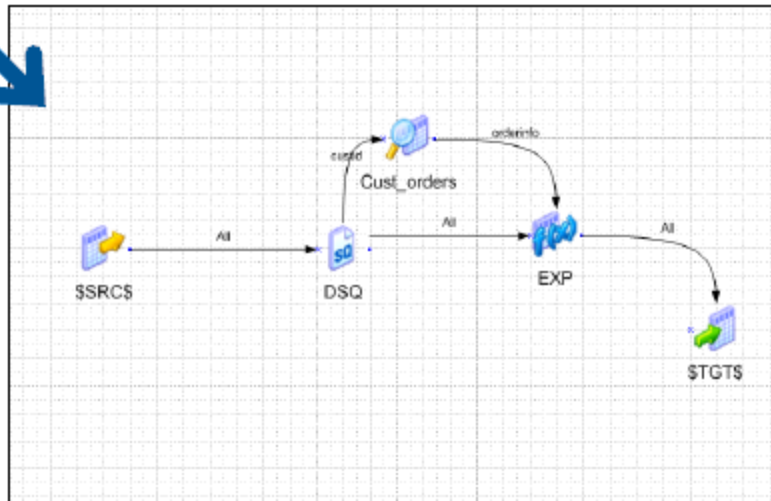
Mapping Architect for Visio



DI Architect

Develops Visio mapping template, logical flow

Generate Mapping



DI Developer

**Augments, Tunes
Generated Mappings**

**Supports PowerExchange
data sources**

**Additional transformation
support**

Midstream Normalizer
Transformation

**Supports transformation
shortcuts**

**Configure and edit custom
transformations after
mapping generation**

Affecto arrangementer fremover



Affecto events:

- ILM seminar 10.mars
- DVH/BI 2.0 seminar 23.mars
- SAP tech-brief i mars
- INFA tech-brief i april
- ABC-seminar 5. mai

Affecto kurs:

- BI Fundamentals
- Kravspesifikasjon for BI
- Dimensjonsmodellering
- SAP BO-kurs
- Informatica-kurs
- MS-kurs