**Slowly Changing Dimension Series**  
Part I : [SCD Type 1.](http://www.disoln.org/2013/01/slowly-changing-dimension-type-1-implementation-using-informatica-powercenter.html)  
Part II : [SCD Type 2.](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html)  
Part III : [SCD Type 3.](http://www.disoln.org/2013/01/slowly-changing-dimension-type-3-implementation-using-informatica-powercenter.html)  
Part IV : [SCD Type 4.](http://www.disoln.org/2013/04/SCD-Type-4-a-solution-for-Rapidly-Changing-Dimension.html)  
Part V : [SCD Type 6.](http://www.disoln.org/2013/04/Slowly-Changing-Dimension-Type-6-a-Combination-of-SCD-Type-1-2-3.html)

**SCD Type 1 Implementation using Informatica PowerCenter**

Unlike [SCD Type 2](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html), Slowly Changing Dimension Type 1 do not preserve any history versions of data. This methodology overwrites old data with new data, and therefore stores only the most current information. In this article lets discuss the step by step implementation of SCD Type 1 using Informatica PowerCenter.

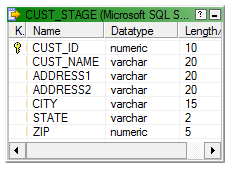
The number of records we store in SCD Type 1 do not increase exponentially as this methodology overwrites old data with new data  Hence we may not need the [performance improvement techniques](http://www.disoln.org/2012/07/design-approach-to-update-larger-tables.html) used in the [SCD Type 2](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html) Tutorial.

## Understand the Staging and Dimension Table.

For our demonstration purpose, lets consider the CUSTOMER Dimension. Below are the detailed structure of both staging and dimension table.

### Staging Table

In our staging table, we have all the columns required for the dimension table attributes. So no other tables other than Dimension table will be involved in the mapping. Below is the structure of our staging table.

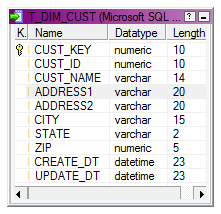
[](http://lh4.ggpht.com/-LAnDuT9SVMQ/UOpRtv7by3I/AAAAAAAAGo4/HMC6MonnA5k/s1600-h/image%25255B8%25255D.png)

#### Key Points

* 1. Staging table will have only one days data. [Change Data Capture](http://www.disoln.org/2012/10/change-data-capture-cdc-made-easy-using-mapping-variables.html)is not in scope.
  2. Data is uniquely identified using CUST\_ID.
  3. All attribute required by Dimension Table is available in the staging table

### Dimension Table

Here is the structure of our Dimension table.

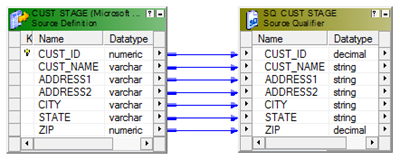
[](http://lh3.ggpht.com/-u4erQ3FM-js/UQn2G2m6wYI/AAAAAAAAHEs/vI7ObvmSFVk/s1600-h/image%25255B7%25255D.png)

#### Key Points

* 1. CUST\_KEY is the surrogate key.
  2. CUST\_ID is the Natural key, hence the unique record identifier.

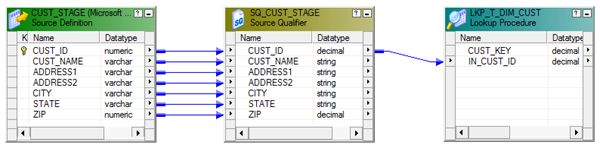
## Mapping Building and Configuration

**Step 1**  
Lets start the [mapping building process](http://www.disoln.org/2012/08/Understand-Informatica-PowerCenter-Mapping-Designer.html). For that pull the CUST\_STAGE source definition into the [mapping designer](http://www.disoln.org/2012/08/Understand-Informatica-PowerCenter-Mapping-Designer.html).

[](http://lh5.ggpht.com/-H0ueQlnoZyc/UOpRv8TLmWI/AAAAAAAAGpY/S_lGbs7Pk6E/s1600-h/image%25255B14%25255D.png)

**Step 2**  
Now using a [LookUp Transformation](http://www.disoln.org/2012/11/Working-with-Flat-File-Source-LookUp-Filter-Transformation.html" \t "_blank) fetch the existing Customer columns from the dimension table T\_DIM\_CUST. This lookup will give NULL values if the customer is not already existing in the Dimension tables.

* LookUp Condition : IN\_CUST\_ID = CUST\_ID
* Return Columns : CUST\_KEY

[](http://lh5.ggpht.com/-Ix8P1sldUG4/UQn2H5iIofI/AAAAAAAAHE8/ujwNjfmFzbs/s1600-h/image%25255B12%25255D.png)

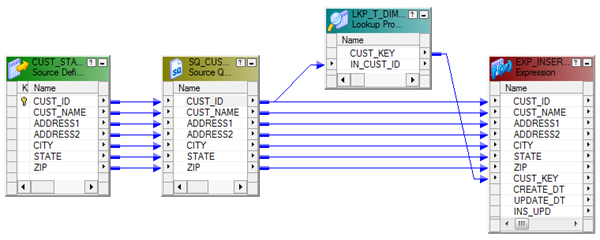
**Step 3**  
Use an Expression Transformation to identify the records for Insert and Update using below expression.

* + INS\_UPD :- IIF(ISNULL(CUST\_KEY),'INS', 'UPD')

Additionally create two output ports.

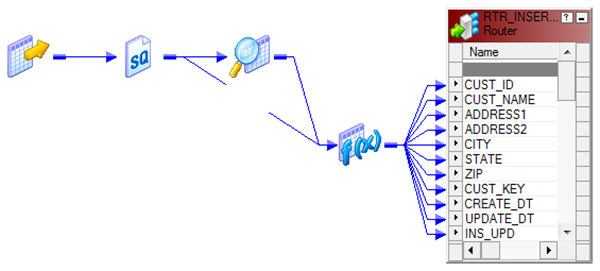
* + CREATE\_DT :- SYSDATE
  + UPDATE\_DT :- SYSDATE

See the structure of the mapping in below image.

[](http://lh5.ggpht.com/-peXvewOtPp4/UQn2Ihz1hBI/AAAAAAAAHFM/0Ad17GZxBzI/s1600-h/image%25255B16%25255D.png)

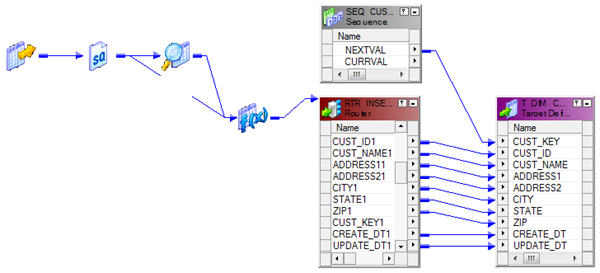
**Step 4**  
Map the columns from the Expression Transformation to a Router Transformation and create two groups (INSERT, UPDATE) in Router Transformation using the below expression. The mapping will look like shown in the image.

* + INSERT :- IIF(INS\_UPD='INS',TRUE,FALSE)
  + UPDATE :- IIF(INS\_UPD='UPD',TRUE,FALSE)

[](http://lh3.ggpht.com/-3fN3sWiUoD4/UQn2J9jqtMI/AAAAAAAAHFc/KMoTlKn8tnk/s1600-h/image%25255B20%25255D.png)

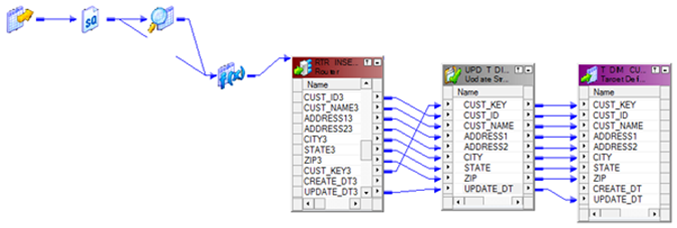
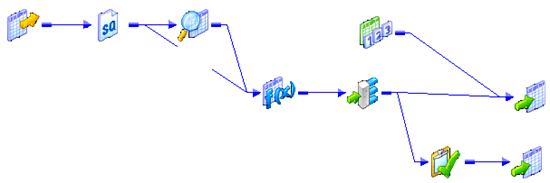
### INSERT Group

**Step 5**  
every records coming through the 'INSERT Group' will be inserted into the Dimension table T\_DIM\_CUST.   
  
Use a Sequence generator transformation to generate surrogate key CUST\_KEY as shown in below image. And map the columns from the Router Transformation to the target as shown below image.

[](http://lh6.ggpht.com/-bJoiia_dqE0/UQn2LXEr3BI/AAAAAAAAHFs/fh-qlcfmICk/s1600-h/image%25255B24%25255D.png)  
  
Note: [Update Strategy](http://www.youtube.com/watch?v=tAobsyxN_uw) is not required, if the records are set for Insert.

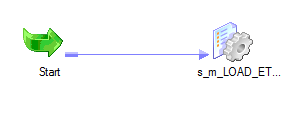
### UPDATE Group

**Step 6**  
Records coming from the 'UPDATE Group' will update the customer Dimension with the latest customer attributes. Add an [Update Strategy Transformation](http://www.youtube.com/watch?v=tAobsyxN_uw&feature=share&list=UUH3JfXqlnjvRWqhm-oz6b2A) before the target instance and set it as DD\_UPDATE. Below is the structure of the mapping.

[](http://lh5.ggpht.com/-xIAzHmV0L3Y/UQn2Mjdl7HI/AAAAAAAAHF8/Xc17chA8QwM/s1600-h/image%25255B28%25255D.png)  
We are done with the mapping building and below is the structure of the completed mapping.  
[](http://lh3.ggpht.com/-6K13mYGZVuE/UQn2S0ECAmI/AAAAAAAAHGM/f1MhB7gtqPE/s1600-h/image%25255B32%25255D.png)

## Workflow and Session Creation

There is not any specific properties required to be given during the [session configuration](http://www.disoln.org/2012/09/understand-informatica-powercenter-Workflow-Designer.html).

[](http://lh4.ggpht.com/-muemQ3GPEos/UHUDl9aT2uI/AAAAAAAAFn0/4tIjoT_QxYU/s1600-h/image%25255B34%25255D.png)

Below is a sample data set taken from the Dimension table T\_DIM\_CUST.

Initial Inserted Value for CUSTI\_ID 1003  
[Slowly Changing Dymention Type 1](http://lh4.ggpht.com/-H_xzflzXQG0/UQn_vgEQ3UI/AAAAAAAAHH0/IheCQ4LH1i4/s1600-h/image%25255B48%25255D.png)  
Updated Value for CUSTI\_ID 1003  
[Slowly Changing Dymention Type 1](http://lh5.ggpht.com/-tNuleLSFYO4/UQn_wz_Q9xI/AAAAAAAAHIE/kfLFnR7UwaA/s1600-h/image%25255B52%25255D.png)

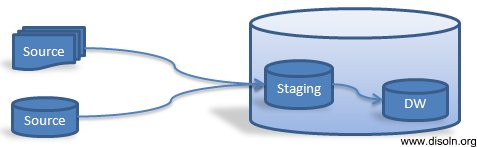
Hope you guys enjoyed this. Please leave us a comment in case you have any questions of difficulties implementing this.

# SCD Type 2 Implementation using Informatica PowerCenter

Slowly Changing Dimension Type 2 also known SCD Type 2 is one of the most commonly used type of Dimension table in a Data Warehouse.  SCD Type 2 dimension loads are considered to be complex mainly because of the data volume we process and because of the number of transformation we are using in the mapping. Here in this article, we will be building an Informatica PowerCenter mapping to load SCD Type 2 Dimension.

## Understand the Data Warehouse Architecture

Before we go to the mapping design, Lets understand the high level architecture of our Data Warehouse.

[](http://1.bp.blogspot.com/-zpKGZ3nQikI/UBsS82pNPGI/AAAAAAAAEok/gkpz5rR1gTM/s1600/Data-warehouse-architecture.png)

Here we have a staging schema, which is loaded from different data sources after the required data cleansing. Warehouse Tables are loaded from the staging schema directly. Both staging tables and the warehouse tables are in two different schemas with in a single database instance.

## Understand the Staging and Dimension Table.

### Staging Table

In our staging table, we have all the columns required for the dimension table attributes. So no other tables other than Dimension table will be involved in the mapping. Below is the structure of our staging table.

* CUST\_ID
* CUST\_NAME
* ADDRESS1
* ADDRESS2
* CITY
* STATE
* ZIP

### Key Points :

1. Staging table will have only one days data.
2. Data is uniquely identified using CUST\_ID.
3. All attribute required by Dimension Table is available in the staging table.

### Dimension Table

Here is the structure of our Dimension table.

* CUST\_KEY
* AS\_OF\_START\_DT
* AS\_OF\_END\_DT
* CUST\_ID
* CUST\_NAME
* ADDRESS1
* ADDRESS2
* CITY
* STATE
* ZIP
* CHK\_SUM\_NB
* CREATE\_DT
* UPDATE\_DT

### Key Points :

1. CUST\_KEY is the surrogate key.
2. CUST\_ID, AS\_OF\_END\_DT is the Natural key, hence the unique record identifier.
3. Record versions are kept based on Time Range using AS\_OF\_START\_DT, AS\_OF\_END\_DATE
4. Active record will have an AS\_OF\_END\_DATE value 12-31-4000
5. [Checksum value](http://www.disoln.org/2012/08/checksum-number-to-make-lookup-transformation-simple.html) of all dimension attribute columns are stored into the column CHK\_SUM\_NB

## Mapping Building and Configuration

Now we understand the ETL Architecture, Staging Table, Dimension Table and the design considerations, we can go to the mapping development. We are splitting the mapping development into six steps.

1. Join Staging Table and Dimension Table
2. Data Transformation
3. Generate Surrogate Key
4. [Generate Checksum Number](http://www.disoln.org/2012/08/checksum-number-to-make-lookup-transformation-simple.html)
5. Other Calculations
6. Identify Insert/Update
7. Insert the new Records
8. Update(Expire) the Old Version
9. Insert the new Version of Updated Record

### 1. Join Staging Table and Dimension Table

We are going to OUTER JOIN both the Staging (Source) Table and the Dimension (Target) Table using the SQL Override below. An OUTER Join gives you all the records from the Staging table and the corresponding records from Dimension table. if it is there is no corresponding record in the Dimension table, it returns NULL values for the Dimension table columns.

*SELECT*

*--Columns From Staging (Source) Tables CUST\_STAGE.CUST\_ID,*

*CUST\_STAGE.CUST\_NAME,*

*CUST\_STAGE.ADDRESS1,*

*CUST\_STAGE.ADDRESS2,*

*CUST\_STAGE.CITY,*

*CUST\_STAGE.STATE,*

*CUST\_STAGE.ZIP,*

*--Columns from Dimension (Target) Tables.*

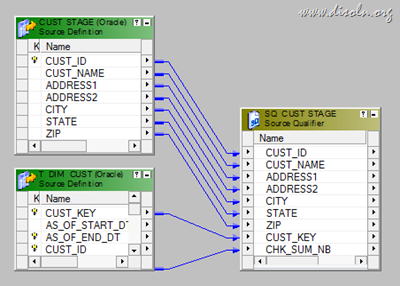
*T\_DIM\_CUST.CUST\_KEY,*

*T\_DIM\_CUST.CHK\_SUM\_NB*

*FROM CUST\_STAGE LEFT OUTER JOIN T\_DIM\_CUST*

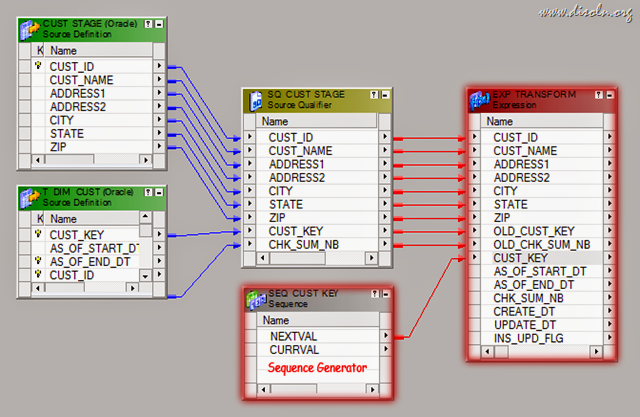
*ON CUST\_STAGE.CUST\_ID = T\_DIM\_CUST.CUST\_ID  -- Join On the Natural Key*

*AND T\_DIM\_CUST.AS\_OF\_END\_DT = TO\_DATE('12-31-4000','MM-DD-YYYY') – Get the active record.*

[](http://lh3.ggpht.com/-sBMWRoa5phI/UBzJdGSeW7I/AAAAAAAAEp8/wz5MOusEh74/s1600-h/Informatica-Source-Qualfier%25255B16%25255D.png)

### 2.  Data Transformation

Now map the columns from the Source Qualifier to an Expression Transformation. When you map the columns to the Expression Transformation, rename the ports from Dimension Table with OLD\_CUST\_KEY, CUST\_CHK\_SUM\_NB and add below expressions.

[](http://lh3.ggpht.com/-4VlBphzVa-o/UBt1wggBXDI/AAAAAAAAEqM/WI6BprYNvSY/s1600-h/Informatica-Source-Qualfier-Expression%25255B8%25255D.png)

* Generate Surrogate Key : A surrogate key will be generated for each and every record inserted in to the Dimension table
  + CUST\_KEY : Is the surrogate key, This will be generated using a Sequence Generator Transformation
* Generate Checksum Number: [Checksum number](http://www.disoln.org/2012/08/checksum-number-to-make-lookup-transformation-simple.html) of all dimension attributes. Difference in the Checksum value between the incoming and Checksum of the Dimension table record will indicate a changed column value. This is an easy way to identify changes in the columns than comparing each and every column.
  + CHK\_SUM\_NB : MD5(TO\_CHAR(CUST\_ID) || CUST\_NAME || ADDRESS1 || ADDRESS2 || CITY || STATE || TO\_CHAR(ZIP))
* Other Calculations :
  + Effective Start Date : Effective start date of the Record
    - AS\_OF\_START\_DT :  TRUNC(SYSDATE)
  + Effective end date  : Effective end date of the Record,
    - AS\_OF\_END\_DT : TO\_DATE('12-31-4000','MM-DD-YYYY')
  + Record creation date : Record creation timestamp, this will be used for the records inserted
    - CREATE\_DT :  TRUNC(SYSDATE)
  + Record updating date : Record updating timestamp, this will be used for records updated.
    - UPDATE\_DT :  TRUNC(SYSDATE)

### 3. Identify Insert/Update

In this step we will identify the records for INSERT and UPDATE.

* INSERT : A record will be set for INSERT if the record is not exist in the Dimension Table, We can identify the New records if  OLD\_CUST\_KEY is NULL, which is the column from the Dimension table
* UPDATE : A record will be set for UPDATE, if the record is already existing in the Dimension table and any of the incoming column from staging table has a new value.  If the column OLD\_CUST\_KEY is not null and the Checksum of the incoming record is different from the Checksum of the existing record (OLD\_CHK\_SUM\_NB <> CHK\_SUM\_NB), the record will be set for UPADTE
  + Following expression will be used in the Expression Transformation port INS\_UPD\_FLG shown in the previous step
  + INS\_UPD\_FLG : IIF(ISNULL(OLD\_CUST\_KEY), 'I', IIF(NOT ISNULL(OLD\_CUST\_KEY) AND OLD\_CHK\_SUM\_NB <> CHK\_SUM\_NB, 'U'))

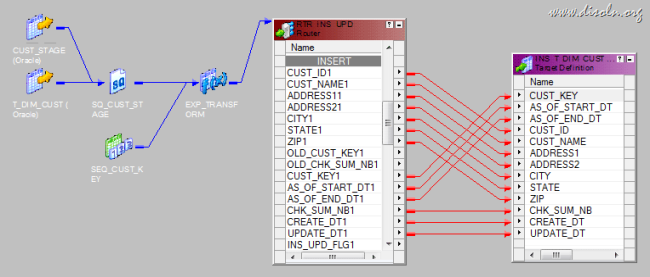
Now map all the columns from the Expression Transformation to a Router and add two groups as below

* + INSERT : IIF(INS\_UPD\_FLG = 'I', TRUE, FALSE)
  + UPDATE : IIF(INS\_UPD\_FLG = 'U', TRUE, FALSE)

[](http://lh6.ggpht.com/-pDRaByH2uns/UB1sYYN3LuI/AAAAAAAAErg/9CUiTqeEED4/s1600-h/Informatica-Slowly-changing-dimension-router.png)

### 4. Insert The new Records

Now map all the columns from the ‘INSERT’ group to the Dimension table instance T\_DIM\_CUST. While mapping the columns, we don’t need any column named OLD\_, which is pulled from the Dimension table.

[](http://lh3.ggpht.com/-muEqz_3_0so/UBzJh7Mv53I/AAAAAAAAErs/Mhge8TrRyIk/s1600-h/Informatica-Slowly-changing-dimension-insert.png)

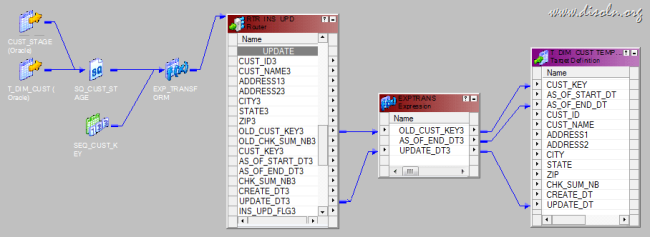
### 5. Update(Expire) the Old Version

The records which are identified for UPDATE will be inserted into a temporary table T\_DIM\_CUST\_TEMP. These records will then be updated into T\_DIM\_CUST as a post session SQL.  You can learn more about this [performance improvement technique](http://www.disoln.org/2012/07/design-approach-to-update-larger-tables.html) from one of our previous post.

We will be mapping below columns from ‘UPDATE’ group of the Router Transformation to the target table. To update(expire) the old record we just need the columns below list.

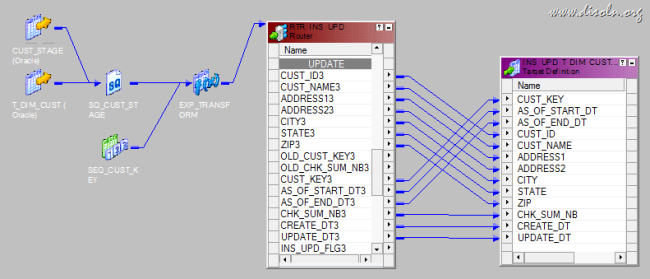
* + OLD\_CUST\_KEY : To uniquely identify  the Dimension Column.
  + UPDATE\_DATE : Audit column to know the record update date.
  + AS\_OF\_END\_DT : Record will be expired with previous days date.

While we map the columns, AS\_OF\_END\_DT will be calculated as ADD\_TO\_DATE(TRUNC(SYSDATE),'DD',-1) in an Expression Transformation. Below image gives the picture of the mapping.

[](http://lh3.ggpht.com/-ULR6I1y8GCo/UBzJjRgVSbI/AAAAAAAAEr0/jwDqNaA0VJQ/s1600-h/Informatica-Slowly-changing-dimension-update%25255B2%25255D.png)

### 6. Insert the new Version of Updated Record

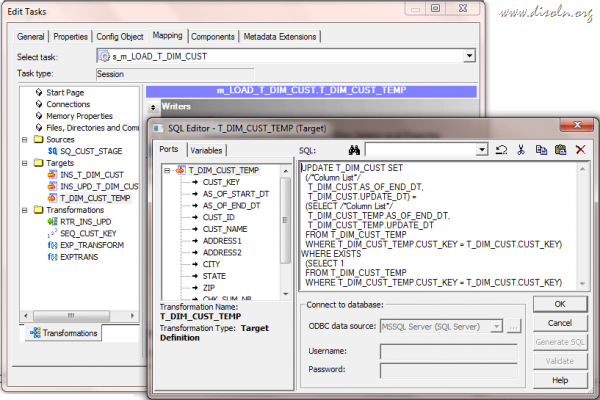
The records which are identified as UPDATE will have to have a new(active) version inserted.  Map all the ports from the ‘UPDATE’ group of the Router Transformation to target instance T\_DIM\_CUST. While mapping the columns, we don’t need any column named OLD\_, which is pulled from the Dimension table.

[](http://lh6.ggpht.com/-imrPk6_O6jo/UBzJkhUb9lI/AAAAAAAAEr8/L8rLycQm6RU/s1600-h/Informatica-Slowly-changing-dimension-insert-update.png)

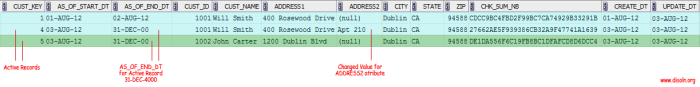
## Workflow and Session Creation

During the [session configuration process](http://www.disoln.org/2012/09/understand-informatica-powercenter-Workflow-Designer.html), add the below SQL as part of the Post session SQL statement as shown below. This correlated update SQL will update the records in T\_DIM\_CUST table with the values from T\_DIM\_CUST\_TEMP. Like we mentioned previously, this is a [performance improvement technique](http://www.disoln.org/2012/07/design-approach-to-update-larger-tables.html) used to update huge tables.

*UPDATE T\_DIM\_CUST SET  
(T\_DIM\_CUST.AS\_OF\_END\_DT,  
T\_DIM\_CUST.UPDATE\_DT) =  
(SELECT  
T\_DIM\_CUST\_TEMP.AS\_OF\_END\_DT,  
T\_DIM\_CUST\_TEMP.UPDATE\_DT  
FROM T\_DIM\_CUST\_TEMP  
WHERE T\_DIM\_CUST\_TEMP.CUST\_KEY = T\_DIM\_CUST.CUST\_KEY) WHERE EXISTS  
(SELECT 1  
FROM T\_DIM\_CUST\_TEMP  
WHERE T\_DIM\_CUST\_TEMP.CUST\_KEY = T\_DIM\_CUST.CUST\_KEY)*

[](http://lh3.ggpht.com/-3VKwc_E8EbQ/UB1seGvRP4I/AAAAAAAAEsE/IZE6qz2iNC4/s1600-h/Informatica-post-session-sql%25255B6%25255D.png)

Now lets look at the data see how it looks from the below image.

[](http://lh3.ggpht.com/-IU4ZJRwdMbg/UB1sfozv2rI/AAAAAAAAEsU/_TvUE9Ago64/s1600-h/slowly-changing-dimension-date-range%25255B6%25255D.png)

Hope you guys enjoyed this. Please leave us a comment in case you have any questions of difficulties implementing this.

# SCD Type 3 Implementation using Informatica PowerCenter

nlike [SCD Type 2](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html), Slowly Changing Dimension Type 3 preserves only few history versions of data, most of the time 'Current' and Previous' versions. The 'Previous' version value will be stored into the  additional columns with in the same dimension record.  In this article lets discuss the step by step implementation of SCD Type 3 using Informatica PowerCenter.

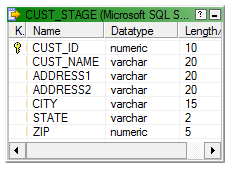
The number of records we store in SCD Type 3 do not increase exponentially as we do not insert a record for each and every historical record. Hence we may not need the [performance improvement techniques](http://www.disoln.org/2012/07/design-approach-to-update-larger-tables.html) used in the [SCD Type 2](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html) Tutorial.

## **Understand the Staging and Dimension Table.**

For our demonstration purpose, lets consider the CUSTOMER Dimension. Here we will keep previous version of CITY, STATE, ZIP into its corresponding PREV columns. Below are the detailed structure of both staging and dimension table.

### **Staging Table**

In our staging table, we have all the columns required for the dimension table attributes. So no other tables other than Dimension table will be involved in the mapping. Below is the structure of our staging table.

[](http://lh4.ggpht.com/-LAnDuT9SVMQ/UOpRtv7by3I/AAAAAAAAGo4/HMC6MonnA5k/s1600-h/image%25255B8%25255D.png)

#### Key Points

* 1. Staging table will have only one days data. [Change Data Capture](http://www.disoln.org/2012/10/change-data-capture-cdc-made-easy-using-mapping-variables.html)is not in scope.
  2. Data is uniquely identified using CUST\_ID.
  3. All attribute required by Dimension Table is available in the staging table

### Dimension Table

Here is the structure of our Dimension table.

[](http://lh5.ggpht.com/-xkwnyKC2-4A/UOpRu_djFvI/AAAAAAAAGpI/po9uwC1PQx0/s1600-h/image%25255B6%25255D.png)

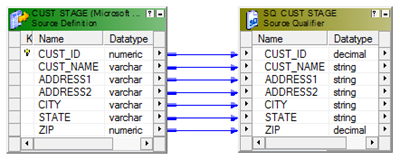
#### Key Points

* 1. CUST\_KEY is the surrogate key.
  2. CUST\_ID is the Natural key, hence the unique record identifier.
  3. Previous versions are kept in PREV\_CITY, PREV\_STATE, PREV\_ZIP columns.

## Mapping Building and Configuration

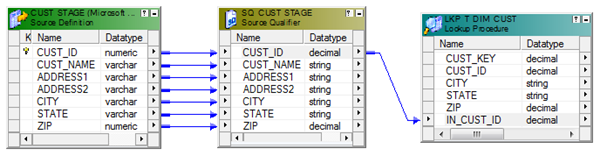
**Step 1**

Lets start the [mapping building process](http://www.disoln.org/2012/08/Understand-Informatica-PowerCenter-Mapping-Designer.html). For that pull the CUST\_STAGE source definition into the [mapping designer](http://www.disoln.org/2012/08/Understand-Informatica-PowerCenter-Mapping-Designer.html).

[](http://lh5.ggpht.com/-H0ueQlnoZyc/UOpRv8TLmWI/AAAAAAAAGpY/S_lGbs7Pk6E/s1600-h/image%25255B14%25255D.png)  
**Step 2**

Now using a [LookUp Transformation](http://www.disoln.org/2012/11/Working-with-Flat-File-Source-LookUp-Filter-Transformation.html" \t "_blank) fetch the existing Customer columns from the dimension table T\_DIM\_CUST. This lookup will give NULL values if the customer is not already existing in the Dimension tables. 

* LookUp Condition : IN\_CUST\_ID = CUST\_ID
* Return Columns : CUST\_KEY, CITY, STATE, ZIP

[](http://lh5.ggpht.com/-EYMbnKJcCyo/UOpRxaFxwBI/AAAAAAAAGpo/9qq-2e7AKFM/s1600-h/image%25255B19%25255D.png)  
**Step 3**

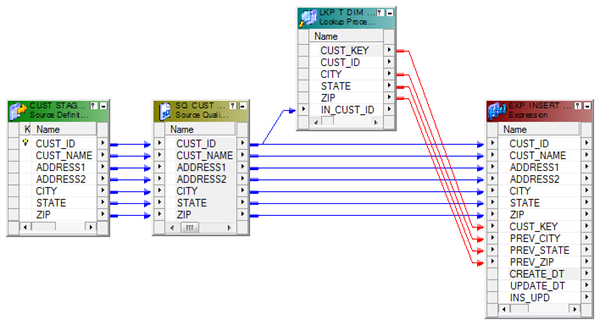
Using an Expression Transformation, identify the records for Insert and Update using below expression. Additionally, map the columns from the LookUp Transformation to the Expression as shown below.  With this we get both  the previous and current values of the CUST\_ID.

* + INS\_UPD :- IIF(ISNULL(CUST\_KEY),'INS', IIF(CITY <> PREV\_CITY OR STATE <> PREV\_STATE OR ZIP <> PREV\_ZIP, 'UPD'))

Additionally create two output ports.

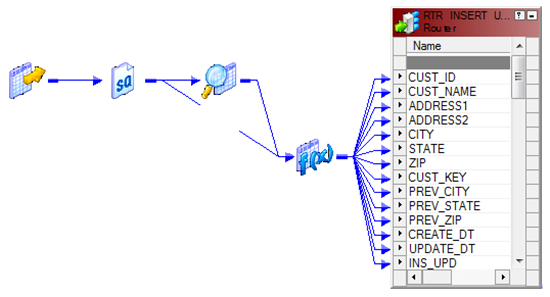
* + CREATE\_DT :- SYSDATE
  + UPDATE\_DT :- SYSDATE

Note : If in case there are too many columns to be compared  to build the INS\_UPD logic, make use of [CheckSum Number](https://www.google.com/url?q=http://www.disoln.org/2012/08/checksum-number-to-make-lookup-transformation-simple.html&sa=U&ei=xz7rUL2PFYjDiwLcqYH4Cw&ved=0CAcQFjAA&client=internal-uds-cse&usg=AFQjCNHhh5co3w7xRaKZqUSEwBOFpR5QGQ" \t "_blank) (MD5() Function) to make it simple.       

[](http://lh5.ggpht.com/-lF8VpC7_xW0/UOpRysZYJgI/AAAAAAAAGp4/H0z0PocfF-I/s1600-h/image%25255B25%25255D.png)  
**Step 4**

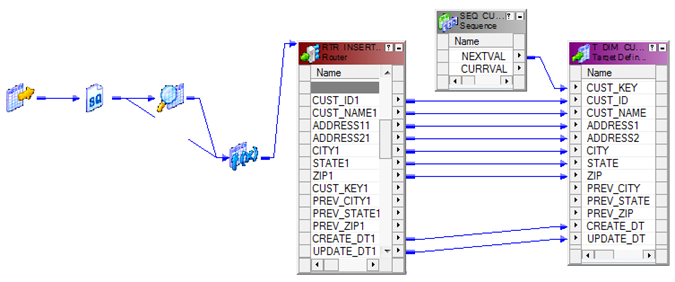
Map the columns from the Expression Transformation to a Router Transformation and create two groups (INSERT, UPDATE) in Router Transformation using the below expression. The mapping will look like shown in the image.

* + INSERT :- IIF(INS\_UPD='INS',TRUE,FALSE)
  + UPDATE :- IIF(INS\_UPD='UPD',TRUE,FALSE)

[](http://lh4.ggpht.com/-Y99pRRcTeWw/UOpRzrETcnI/AAAAAAAAGqI/Ju71sHBU0kc/s1600-h/image%25255B41%25255D.png)

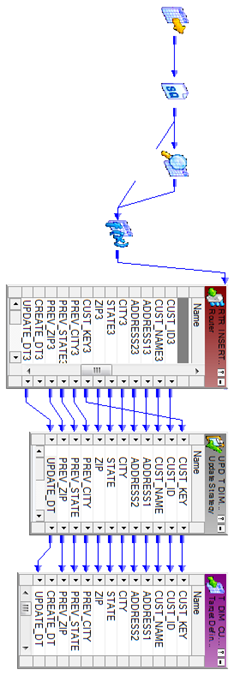
### INSERT Group

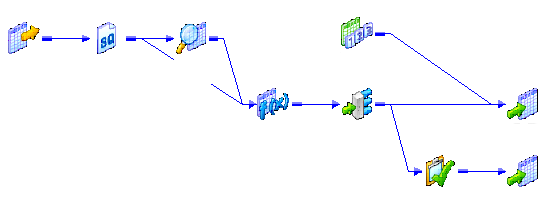
**Step 5**  
Every records coming through the 'INSERT Group' will be inserted into the Dimension table T\_DIM\_CUST.  
  
Use a Sequence generator transformation to generate surrogate key CUST\_KEY as shown in below image. And map the columns from the Router Transformation to the target. Leave all 'PREV' columns unmapped as shown below image.

[](http://lh5.ggpht.com/-cKr4fuCysZ8/UOpR0eGF3PI/AAAAAAAAGqY/v0xCK6W2zaI/s1600-h/image%25255B46%25255D.png)  
Note : [Update Strategy](http://www.youtube.com/watch?v=tAobsyxN_uw) is not required, if the records are set for Insert.

### UPDATE Group

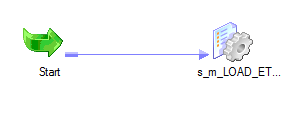
**Step 6**  
Records coming from the 'UPDATE Group' will update the customer Dimension with Current customer attributes and the 'PREV' attributes. Add an [Update Strategy Transformation](http://www.youtube.com/watch?v=tAobsyxN_uw&feature=share&list=UUH3JfXqlnjvRWqhm-oz6b2A) before the target instance and set it as DD\_UPDATE. Below is the structure of the mapping.

[](http://lh6.ggpht.com/-aQBhhpry0CE/UOpR1bpCzzI/AAAAAAAAGqo/NoV8c0IVKF8/s1600-h/image%25255B51%25255D.png)

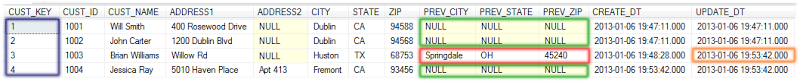
We are done with the mapping building and below is the structure of the completed mapping.  


## Workflow and Session Creation

There is not any specific properties required to be given during the session configuration.

[](http://lh4.ggpht.com/-muemQ3GPEos/UHUDl9aT2uI/AAAAAAAAFn0/4tIjoT_QxYU/s1600-h/image%25255B34%25255D.png)

Below is a sample data set taken from the Dimension table T\_DIM\_CUST. See the highlighted values.

[](http://lh3.ggpht.com/-elV_9OvbwtU/UOpR3FQwvHI/AAAAAAAAGrI/nv1OB_ueXo4/s1600-h/image%25255B35%25255D.png)

Hope you guys enjoyed this. Please leave us a comment in case you have any questions of difficulties implementing this.

# Informatica Performance Tuning Guide, Tuning and Bottleneck Overview - Part 1

[Performance](http://www.disoln.org/search/label/Performance%20Tips?&max-results=15) tuning process identifies the bottlenecks and eliminate it to get a better acceptable ETL load time. Tuning starts with the identification of bottlenecks in source, target, mapping and further to session tuning. It might need further tuning on the system resources on which the Informatica PowerCenter Services are running.

This performance tuning *article series* is split into multiple articles, which goes over specific areas of performance tuning. In this article we will discuss about the session anatomy and more about bottlenecks.

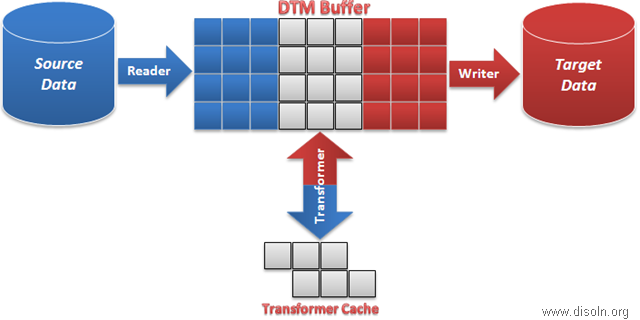
## Performance Tuning and Bottlenecks Overview

Determining the best way to improve performance can be complex. An iterative method of identifying one bottleneck at a time and eliminate it, then identify and eliminate the next bottleneck until an acceptable throughput is achieved is more effective.

The first step in performance tuning is to identify performance bottlenecks. Performance bottlenecks can occur in the source and target, the mapping, the session, and the system. Before we understand different bottlenecks, lets see the components of Informatica PowerCenter session and how a bottleneck arises.

## Informatica PowerCenter Session Anatomy

When a PowerCenter session is triggered, integration service start *Data Transformation Manager (DTM)*, which  is responsible to start reader thread, transformation thread and writer thread.   
  
*Reader thread* is responsible to read data from the sources. *Transformation threads* process data according to the transformation logic in the mapping and *writer thread* connects to the target and loads the data. Any data processing delay in these threads leads to a performance issue.

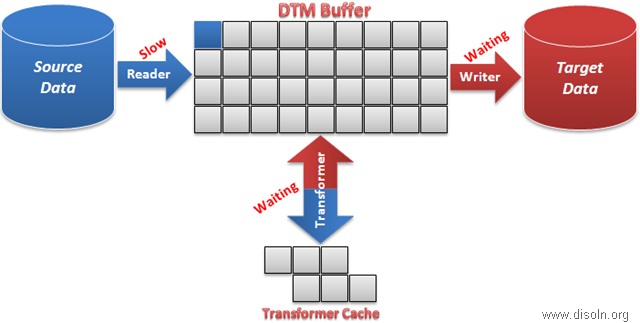
[](http://lh4.ggpht.com/-uzrz9XQNLmo/UhpvDP3pCUI/AAAAAAAAIgY/ehNWZXAiDU8/s1600-h/image%25255B13%25255D.png)

Above shown is the pictorial representation of a session. Reader thread reads data from the source and data transformation is done by transformation thread. Finally loaded into the target by the writer thread.

## Source Bottlenecks

Performance bottlenecks can occur when the Integration Service reads from a source database. Slowness in reading data from the source leads to delay in filling enough data into DTM buffer. So the transformation and writer threads wait for data.  This delay causes the entire session to run slower.

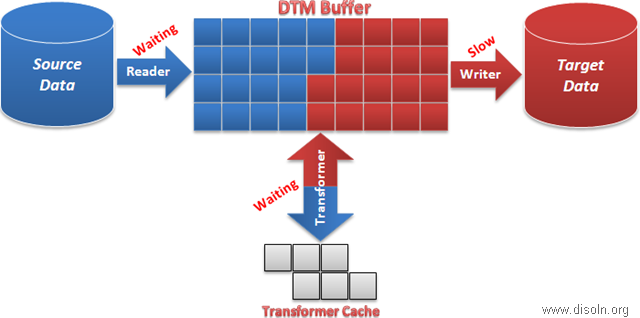
Inefficient query or small database network packet sizes can cause source bottlenecks.

[](http://lh6.ggpht.com/-nDR2lYUN2rM/UhqEBr8pKHI/AAAAAAAAIgw/u5TbUlrUvYg/s1600-h/image%25255B22%25255D.png)

## Target Bottlenecks

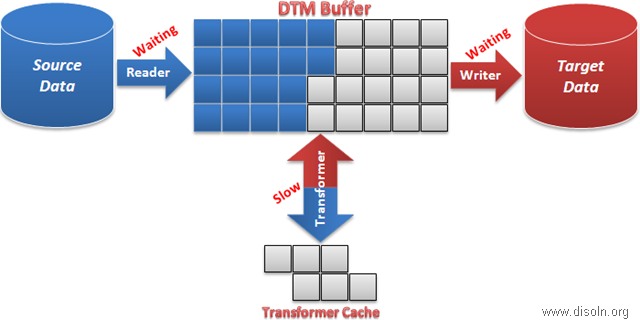
When target bottleneck occurs, writer thread will not be able to free up space for reader and transformer threads, until the data is written to the target. So the the reader and transformer threads to wait for free blocks. This causes the entire session to run slower.

Small database checkpoint intervals, small database network packet sizes, or problems during heavy loading operations can cause target bottlenecks.

[](http://lh4.ggpht.com/-AIq0hnT_RiY/UhqEDWGc_jI/AAAAAAAAIhA/ZqRFJCUHSDI/s1600-h/image%25255B32%25255D.png)

## Mapping Bottlenecks

A complex mapping logic or a not well written mapping logic can lead to mapping bottleneck. With mapping bottleneck, transformation thread runs slower causing the reader thread to wait for free blocks and writer thread to wait blocks filled up for writing to target.

[](http://lh3.ggpht.com/-djKMbgkobpc/UhqEE3YxNfI/AAAAAAAAIhQ/7z9FO7RBJr8/s1600-h/image%25255B44%25255D.png)

## Session Bottlenecks

If you do not have a source, target, or mapping bottleneck, you may have a session bottleneck. Session bottleneck occurs normally when you have the session memory configuration is not turned correctly. This in turn leads to a bottleneck on the reader, transformation or writer thread.  Small cache size, low buffer memory, and small commit intervals can cause session bottlenecks.

## System Bottlenecks

After you tune the source, target, mapping, and session, consider tuning the system to prevent system bottlenecks. The Integration Service uses system resources to process transformations, run sessions, and read and write data. The Integration Service also uses system memory to create cache files for transformations, such as Aggregator, Joiner, Lookup, Sorter, XML, and Rank.

# Informatica Performance Tuning Guide, Identify Performance Bottlenecks - Part 2

In our previous article in the [performance tuning](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html) series, we covered the basics of Informatica performance tuning process and the session anatomy. In this article we will cover the methods to identify different performance bottlenecks. Here we will use session thread statistics, session performance counter and workflow monitor properties to help us understand the bottlenecks.

## Source, Target & Mapping Bottlenecks Using Thread Statistics

Thread statics gives run time information from all the three threads; reader, transformation and writer thread. The session log provides enough run time thread statistics to help us understand and pinpoint the performance bottleneck.

### Gathering Thread Statistics

You can get thread statistics from the session long file. When you run a session, the session log file lists run time information and thread statistics with below details.

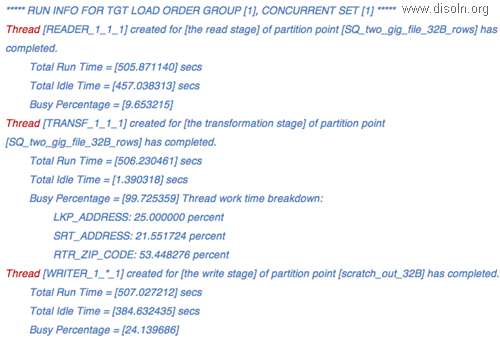
* **Run Time**: Amount of time the thread runs.
* **Idle Time** : Amount of time the thread is idle. Includes the time the thread waits for other thread processing.
* **Busy Time** : Percentage of the run time. It is (run time - idle time) / run time x 100.
* **Thread Work Time** : The percentage of time taken to process each transformation in a thread.

**Note** : Session Log file with normal tracing level is required to get the thread statistics.

### Understanding Thread Statistics

When you run a session, the session log lists run information and thread statistics similar to the following text.

If you read it closely, you will see reader, transformation and writer thread and how much time is spent on each thread and how busy each thread is. Additional to that, transformation thread shows how much  busy each transformation in the mapping is.

[](http://lh5.ggpht.com/-TEnAvKZV56U/UjAGfm4qG3I/AAAAAAAAInc/eUYUmOesc9U/s1600-h/Untitled1%25255B15%25255D.png)

The total run time for the transformation thread is 506 seconds and the busy percentage is 99.7%. This means the transformation thread was never idle for the 506 seconds. The reader and writer busy percentages were significantly smaller, about 9.6% and 24%. In this session, the *transformation thread is the bottleneck* in the mapping.

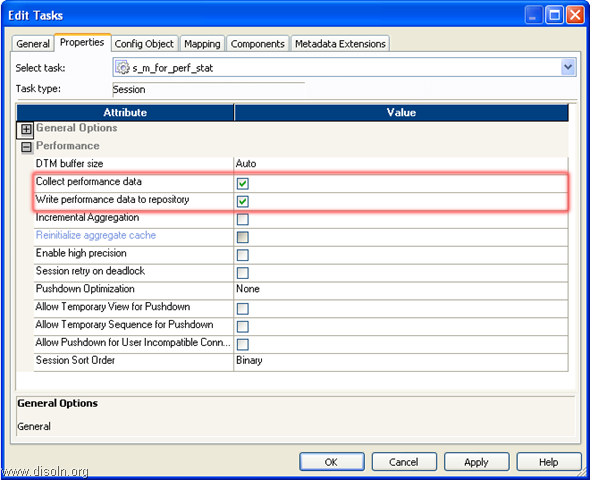
To determine which transformation in the transformation thread is the bottleneck, view the busy percentage of each transformation in the thread work time breakdown. The transformation RTR\_ZIP\_CODE had a busy percentage of 53%.  
  
**Hint** : Thread with the highest busy percentage is the bottleneck.

## Session Bottleneck Using Session Performance Counters

All transformations have counters to help measure and improve performance of the transformations. Analyzing these performance details can help you ***identify session bottlenecks***. The Integration Service tracks the number of input rows, output rows, and error rows for each transformation.

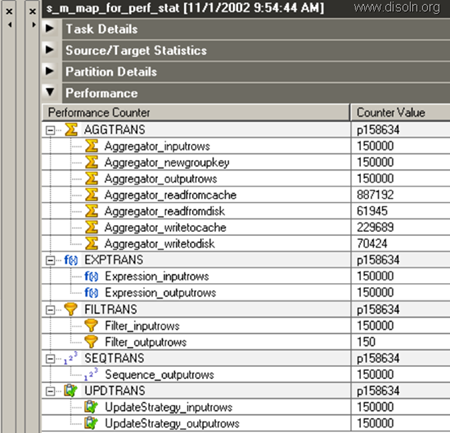
### Gathering Performance Counters

You can setup the session to gather performance counters in the workflow manager. Below image shows the configuration required for a session to collect transformation performance counters.

[](http://lh5.ggpht.com/-A8tlUr2revE/Ui01Dl4B-0I/AAAAAAAAIl0/2ZrvYtXrfYo/s1600-h/image%25255B18%25255D.png)

### Understanding Performance Counters

Below shown image is the performance counters for a session, which you can see from the workflow monitor session run properties.. You can see the transformations in the mapping and the corresponding performance counters.  
  
A *non-zero counts for readfromdisk and writetodisk indicate sub-optimal settings* for transformation index or data caches. This may indicate the need to tune session transformation caches manually.  
  
A *non-zero count for Errorrows*indicates you should eliminate the transformation errors to improve performance.

[](http://lh4.ggpht.com/-2YzD8NvLIOE/UjACiMB7nsI/AAAAAAAAInI/AHrjL_El4E0/s1600-h/image_thumb%25255B11%25255D.png)

* **Errorrows** : Transformation errors impact session performance. If a transformation has large numbers of error rows in any of the Transformation\_errorrows counters, you should eliminate the errors to improve performance.
* **Readfromdisk and Writetodisk**: If these counters display any number other than zero, you can increase the cache sizes to improve session performance.
* **Readfromcache and Writetocache** :  Use this counters to analyze how the Integration Service reads from or writes to cache.
* **Rowsinlookupcache**: Gives the number of rows in the lookup cache. To improve session performance, tune the lookup expressions for the larger lookup tables.

## Session Bottleneck Using Session Log File

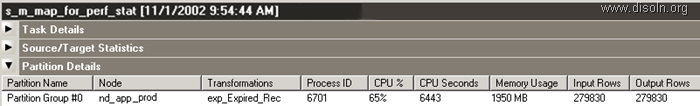
When the Integration Service initializes a session, it allocates blocks of memory to hold source and target data. Not having enough [buffer memory](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html#Anatomy) for [DTM process](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html#Anatomy), can slowdown reading, transforming or writing and cause large fluctuations in performance.

If the session is not able to allocate enough memory for the DTP Process, Integration service will write a warning message in to the session log file and gives you the recommended buffer size.  Below is a sample message seen in the session

*Message: WARNING: Insufficient number of data blocks for adequate performance. Increase DTM buffer size of the session. The recommended value is xxxx.*

## System Bottleneck Using the Workflow Monitor

You can view the Integration Service properties in the Workflow Monitor to see CPU, memory, and swap usage of the system when you are running task processes on the Integration Service. Use the following Integration Service properties to identify performance issues:

[](http://lh3.ggpht.com/-9QvcUUqtxRU/Ui1MUEg5w_I/AAAAAAAAImc/bky-nItnqf4/s1600-h/image%25255B46%25255D.png)

* **CPU%**: The percentage of CPU usage includes other external tasks running on the system. A high CPU usage indicates the need of additional processing power required by the server.
* **Memory Usage** : The percentage of memory usage includes other external tasks running on the system. If the memory usage is close to 95%, check if the tasks running on the system are using the amount indicated in the Workflow Monitor or if there is a memory leak. To troubleshoot, use system tools to check the memory usage before and after running the session and then compare the results to the memory usage while running the session.
* **Swap Usage**: Swap usage is a result of paging due to possible memory leaks or a high number of concurrent tasks.

# Informatica Performance Tuning Guide, Resolve Performance Bottlenecks - Part 3



In our previous article in the [performance tuning](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html) series, we covered different approaches to identify performance bottlenecks. In this article we will cover the methods to resolve different performance bottlenecks. We will talk about session memory, cache memory, source, target and mapping performance turning techniques in detail.

# I. Buffer Memory Optimization

When the Integration Service initializes a session, it allocates blocks of memory to hold source and target data. Sessions that use a large number of sources and targets might require additional memory blocks.

Not having enough [buffer memory](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html#Anatomy) for [DTM process](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html#Anatomy), can slowdown reading, transforming or writing and cause large fluctuations in performance. Adding extra memory blocks can keep the threads busy and improve session performance. You can do this by adjusting the buffer block size and [DTM Buffer](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html#Anatomy) size.  
  
**Note** : You can identify DTM buffer bottleneck from Session Log File, [Check here for details](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html#sesslog).

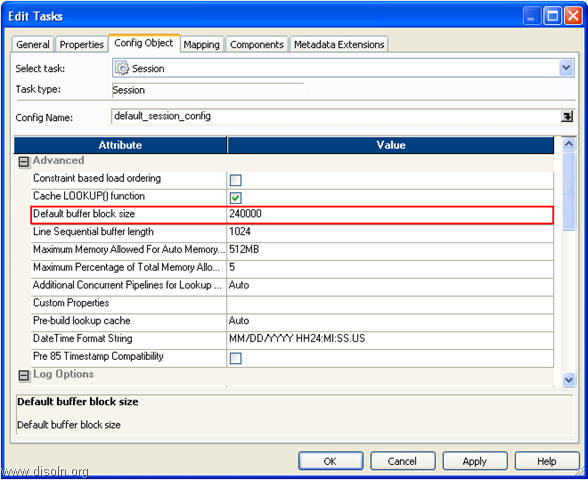
### 1. Optimizing the Buffer Block Size

Depending on the source, target data, you might need to increase or decrease the buffer block size.

To identify the optimal buffer block size, sum up the precision of individual source and targets columns. The largest precision among all the source and target should be the buffer block size for one row. Ideally, a buffer block should accommodates at least 100 rows at a time.

* + - Buffer Block Size = Largest Row Precision \* 100

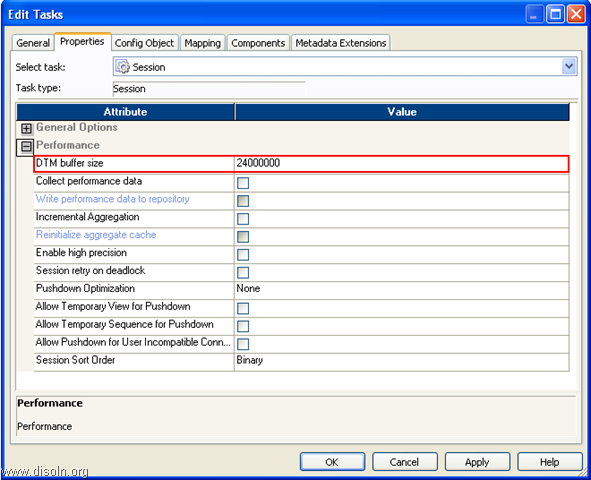
You can change the buffer block size in the session configuration as shown in below image.

[](http://lh6.ggpht.com/-yqFx6ZuIiA8/UleUxhhwC4I/AAAAAAAAIxg/n3_Kjz6SpOI/s1600-h/image%25255B46%25255D.png)

### 2. Increasing DTM Buffer Size

When you increase the DTM buffer memory, the Integration Service creates more buffer blocks, which improves performance. You can identify the required DTM Buffer Size based on below calculation.

* + - Session Buffer Blocks = (total number of sources + total number of targets) \* 2
    - DTM Buffer Size = Session Buffer Blocks \* Buffer Block Size / 0.9

You can change the DTM Buffer Size in the session configuration as shown in below image.[](http://lh4.ggpht.com/-6EzBFwjaOy0/UleUvxONjlI/AAAAAAAAIxQ/_hXmc7Ky6nA/s1600-h/image%25255B48%25255D.png)

# II. Caches Memory Optimization

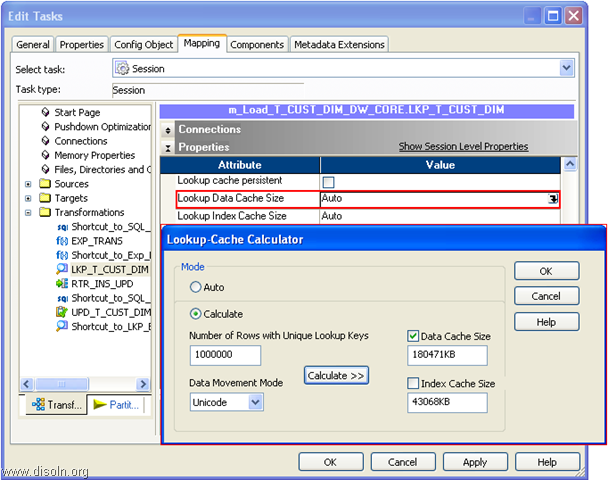
Transformations such as Aggregator, Rank, Lookup uses cache memory to store transformed data, which includes index and data cache. If the allocated cache memory is not large enough to store the data, the Integration Service stores the data in a temporary cache file. Session performance slows each time the Integration Service reads from the temporary cache file.

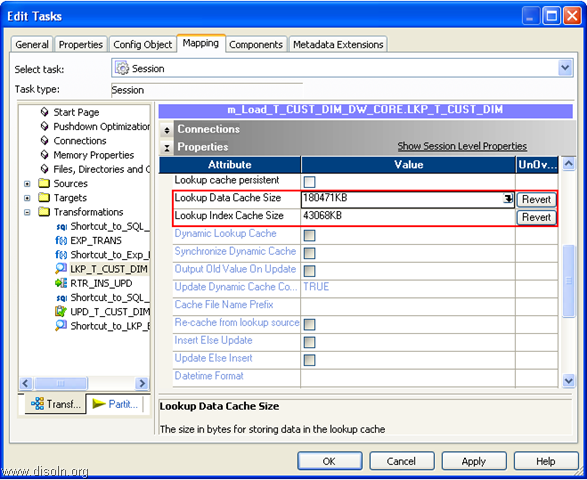
**Note** : You can examine the performance counters to determine what all transformations require cache memory turning, [Check here for details.](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html#session)

### 1. Increasing the Cache Sizes

You can increase the allocated cache sizes to process the transformation in cache memory itself such that the integration service do not have to read from the cache file.

You can calculate the memory requirements for a transformation using the Cache Calculator. Below shown is the Cache Calculator for Lookup transformation.

[](http://lh4.ggpht.com/-yZwOimoFyMw/UleUzcT3BOI/AAAAAAAAIxw/uRXT_-XK-AM/s1600-h/image%25255B41%25255D.png)

You can update the cache size in the session property of the transformation as shown below.[](http://lh4.ggpht.com/-Xxg332tVtHQ/Ulebdkp6c3I/AAAAAAAAIyY/h_lRcRcnfYQ/s1600-h/image%25255B58%25255D.png)

### 2. Limiting the Number of Connected Ports

For transformations that use data cache, limit the number of connected input/output and output only ports. Limiting the number of connected input/output or output ports reduces the amount of data the transformations store in the data cache.

# III. Optimizing the Target

The most common performance bottleneck occurs when the Integration Service writes to a target database. Small database checkpoint intervals, small database network packet sizes, or problems during heavy loading operations can cause target bottlenecks.

**Note** : Target bottleneck can be determined with the help of Session Log File, [check here for details](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html#mapping).

### 1. Using Bulk Loads

You can use bulk loading to improve the performance of a session that inserts a large amount of data into a DB2, Sybase ASE, Oracle, or Microsoft SQL Server database. When bulk loading, the Integration Service bypasses the database log, which speeds performance. Without writing to the database log, however, the target database cannot perform rollback. As a result, you may not be able to perform recovery.

### 2. Using External Loaders

To increase session performance, configure PowerCenter to use an external loader for the following types of target databases. External loader can be used for Oracle, DB2, Sybase and Teradata.

### 3. Dropping Indexes and Key Constraints

When you define key constraints or indexes in target tables, you slow the loading of data to those tables. To improve performance, drop indexes and key constraints before you run the session. You can rebuild those indexes and key constraints after the session completes.

### 4. Minimizing Deadlocks

Encountering deadlocks can slow session performance. You can increase the number of target connection groups in a session to avoid deadlocks. To use a different target connection group for each target in a session, use a different database connection name for each target instance.

### 5. Increasing Database Checkpoint Intervals

The Integration Service performance slows each time it waits for the database to perform a checkpoint. To decrease the number of checkpoints and increase performance, increase the checkpoint interval in the database.

### 6. Increasing Database Network Packet Size

If you write to Oracle, Sybase ASE, or Microsoft SQL Server targets, you can improve the performance by increasing the network packet size. Increase the network packet size to allow larger packets of data to cross the network at one time.

# IV. Optimizing the Source

Performance bottlenecks can occur when the Integration Service reads from a source database. Inefficient query or small database network packet sizes can cause source bottlenecks.

**Note** : Session Log File details can be used to identify Source bottleneck, [check here for details](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html#mapping).

### 1. Optimizing the Query

If a session joins multiple source tables in one Source Qualifier, you might be able to improve performance by optimizing the query with optimizing hints. Usually, the database optimizer determines the most efficient way to process the source data. However, you might know properties about the source tables that the database optimizer does not. The database administrator can create optimizer hints to tell the database how to execute the query for a particular set of source tables.

### 2. Increasing Database Network Packet Size

If you read from Oracle, Sybase ASE, or Microsoft SQL Server sources, you can improve the performance by increasing the network packet size. Increase the network packet size to allow larger packets of data to cross the network at one time.

# V. Optimizing the Mappings

Mapping-level optimization may take time to implement, but it can significantly boost session performance. Focus on mapping-level optimization after you optimize the targets and sources.

Generally, you reduce the number of transformations in the mapping and delete unnecessary links between transformations to optimize the mapping. Configure the mapping with the least number of transformations and expressions to do the most amount of work possible. Delete unnecessary links between transformations to minimize the amount of data moved.  
  
**Note** : You can identify Mapping bottleneck from Session Log File, [check here for details](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html#mapping).

### 1. Optimizing Datatype Conversions

You can increase performance by eliminating unnecessary datatype conversions. For example, if a mapping moves data from an Integer column to a Decimal column, then back to an Integer column, the unnecessary datatype conversion slows performance. Where possible, eliminate unnecessary datatype conversions from mappings.

### 2. Optimizing Expressions

You can also optimize the expressions used in the transformations. When possible, isolate slow expressions and simplify them.

* **Factoring Out Common Logic :** If the mapping performs the same task in multiple places, reduce the number of times the mapping performs the task by moving the task earlier in the mapping.
* **Minimizing Aggregate Function Calls :** When writing expressions, factor out as many aggregate function calls as possible. Each time you use an aggregate function call, the Integration Service must search and group the data. For example SUM(COL\_A + COL\_B) performs better than SUM(COL\_A) + SUM(COL\_B)
* **Replacing Common Expressions with Local Variables :**If you use the same expression multiple times in one transformation, you can make that expression a local variable.
* **Choosing Numeric Versus String Operations :** The Integration Service processes numeric operations faster than string operations. For example, if you look up large amounts of data on two columns, EMPLOYEE\_NAME and EMPLOYEE\_ID, configuring the lookup around EMPLOYEE\_ID improves performance.
* **Using Operators Instead of Functions :**The Integration Service reads expressions written with operators faster than expressions with functions. Where possible, use operators to write expressions.

### 3. Optimizing Transformations

Each transformation is different and the tuning required for different transformation is different. But generally, you reduce the number of transformations in the mapping and delete unnecessary links between transformations to optimize the transformation.  
  
**Note**: Tuning technique for different transformation will be covered as a separate article.

# Informatica Performance Tuning Guide, Performance Enhancements - Part 4

In our performance turning article series, so far we covered about the [performance turning basics](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html), [identification of bottlenecks](http://www.disoln.org/2013/09/Informatica-Performance-Tuning-Guide-Identify-Performance-Bottlenecks.html) and [resolving different bottlenecks](http://www.disoln.org/2013/10/Informatica-Performance-Tuning-Guide-Resolve-Performance-Bottlenecks-Part-3.html). In this article we will cover different performance enhancement features available in Informatica PowerCener. In addition to the features provided by PowerCenter, we will go over the designs tips and tricks for ETL load performance improvement.

## Performance Enhancements Features

The main PowerCenter features for [Performance](http://www.disoln.org/search/label/Performance%20Tips?max-results=15) Enhancements are.

**Pushdown Optimization.**

**Pipeline Partitions.**

**Dynamic Partitions.**

**Concurrent Workflows.**

**Grid Deployments.**

**Workflow Load Balancing.**

**Other Performance Tips and Tricks.**

## **1. Pushdown Optimization**

Pushdown Optimization Option enables data transformation processing, to be pushed down into any relational database to make the best use of database processing power. It converts the transformation logic into SQL statements, which can directly execute on database. This minimizes the need of moving data between servers and utilizes the power of database engine.

* + - **Read More** about [*Pushdown Optimization*](http://www.disoln.org/2013/07/Informatica-PowerCenter-Pushdown-Optimization-an-ELT-Approach.html)*.*

## **2. Session Partitioning**

The Informatica PowerCenter Partitioning Option increases the performance of PowerCenter through parallel data processing. Partitioning option will let you split the large data set into smaller subsets which can be processed in parallel to get a better session performance.

* + - **Read More** about [*Session Partitioning*](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-for-Parallel-Processing.html)*.*

## **3. Dynamic Session Partitioning**

Informatica PowerCenter [session partition](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-When-Where-and-How.html) can be used to [process data in parallel](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-for-Parallel-Processing.html) and achieve faster data delivery. Using Dynamic Session Partitioning capability, PowerCenter can dynamically decide the degree of parallelism. The Integration Service scales the number of session partitions at run time based on factors such as source database partitions or the number of CPUs on the node resulting significant [performance improvement](http://www.disoln.org/search/label/Performance%20Tips?&max-results=15).

* + - **Read More** about[*Dynamic Session Partition*](http://www.disoln.org/2013/08/Dynamic-Partitioning-to-Increase-Parallelism-Based-on-Resources-Availability.html)*.*

## **4. Concurrent Workflows**

A concurrent workflow is a workflow that can run as multiple instances concurrently. A workflow instance is a representation of a workflow. We can configure two types of concurrent workflows. It can be concurrent workflows with the same instance name or unique workflow instances to run concurrently.

* + - **Read More** about [*Concurrent Workflows*](http://www.disoln.org/2012/11/Informatica-Concurrent-Workflows-to-Reduce-Warehouse-ETL-Load-Time.html)*.*

## **5. Grid Deployments**

When a PowerCenter domain contains multiple nodes, you can configure workflows and sessions to run on a grid. When you run a workflow on a grid, the Integration Service runs a service process on each available node of the grid to increase performance and scalability. When you run a session on a grid, the Integration Service distributes session threads to multiple DTM processes on nodes in the grid to increase performance and scalability.

* + - **Read More** about [*Grid Deployments*](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html)*.*

## **6. Workflow Load Balancing**

Informatica Load Balancing is a mechanism which distributes the workloads across the nodes in the [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html). When you run a workflow, the Load Balancer dispatches different tasks in the workflow such as Session, Command, and predefined Event-Wait tasks to different [nodes](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html) running the Integration Service. Load Balancer matches task requirements with resource availability to identify the best node to run a task. It may dispatch tasks to a single node or across nodes on the [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html).

* + - **Read More** about [*Workflow Load Balancing*](http://www.disoln.org/2013/11/Informatica-PowerCenter-Load-Balancing-for-Workload-Distribution.html)*.*

## **7. Other Performance Tips and Tricks**

Throughout this blog we have been discussing different tips and tricks to improve your ETL load performance.  We would like to reference those tips and tricks in this article for your reference.

* + - **Read More** about [*Other Performance Tips and Tricks*](http://www.disoln.org/search/label/Performance%20Tips?max-results=15)*.*

# Informatica PowerCenter Pushdown Optimization a Hybrid ELT Approach

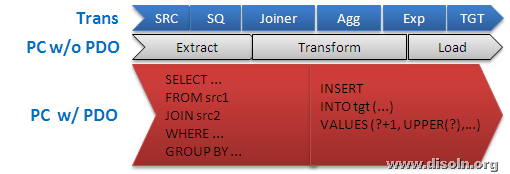
Informatica Pushdown Optimization Option [increases performance](http://www.disoln.org/search/label/Performance%20Tips?&max-results=5) by providing the flexibility to push transformation processing to the most appropriate processing resource. Using Pushdown Optimization, data transformation logic can be pushed to source database or target database or through the PowerCenter server.  This gives the option for the ETL architect to choose the best of the available resources for data processing.

## What is Pushdown Optimization

Pushdown Optimization Option enables data transformation processing, to be pushed down into any relational database to make the best use of database processing power. It converts the transformation logic into SQL statements, which can directly execute on database. This minimizes the need of moving data between servers and utilizes the power of database engine.

## How Pushdown Optimization Works

When you run a session configured for pushdown optimization, the Integration Service analyzes the mapping and transformations to determine the transformation logic it can push to the database. If the mapping contains a mapplet, the Integration Service expands the mapplet and treats the transformations in the mapplet as part of the parent mapping. The Integration Service converts the transformation logic into SQL statements and sends to the source or the target database to perform the data transformation. The amount of transformation logic one can push to the database depends on the database, transformation logic, and mapping and session configuration.

[](http://lh5.ggpht.com/-wXjRkiDSpJo/UfnqCoAZ71I/AAAAAAAAIYY/lPTBTTwnIp0/s1600-h/image%25255B89%25255D.png)

## Different Type Pushdown Optimization

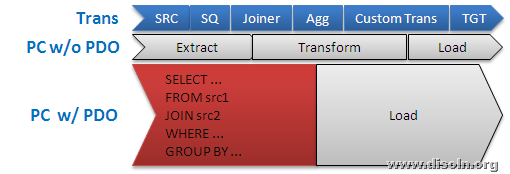
You can configure pushdown optimization in the following ways.

* + 1. **Source-side pushdown optimization**
    2. **Target-side pushdown optimization**
    3. **Full pushdown optimization**

### Source-side pushdown optimization

When you run a session configured for source-side pushdown optimization, the Integration Service analyzes the mapping from the source to the target or until it reaches a downstream transformation it cannot push to the database.

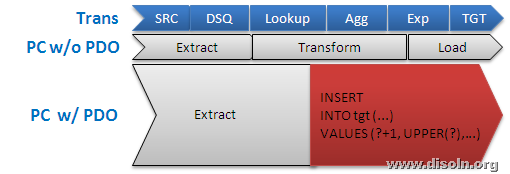
The Integration Service generates a SELECT statement based on the transformation logic for each transformation it can push to the database. When you run the session, the Integration Service pushes all transformation logic that is valid to push to the database by executing the generated SQL statement. Then, it reads the results of this SQL statement and continues to run the session.

[](http://lh4.ggpht.com/-nHoMsFu05lk/UfnqEOquDSI/AAAAAAAAIYo/H4E9Vacf4Ug/s1600-h/image%25255B79%25255D.png)

If you run a session that contains an SQL override or lookup override, the Integration Service generates a view based on the override. It then generates a SELECT statement and runs the SELECT statement against this view. When the session completes, the Integration Service drops the view from the database.

### Target-side pushdown optimization

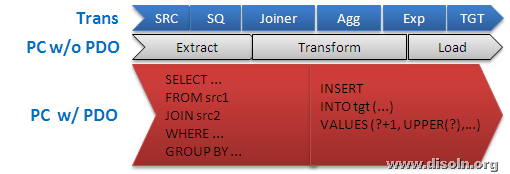
When you run a session configured for target-side pushdown optimization, the Integration Service analyzes the mapping from the target to the source or until it reaches an upstream transformation it cannot push to the database.

[](http://lh4.ggpht.com/-t23k86bVtOM/UfnqFS9J-NI/AAAAAAAAIY4/PuqhLQst7FY/s1600-h/image%25255B100%25255D.png)

The Integration Service generates an INSERT, DELETE, or UPDATE statement based on the transformation logic for each transformation it can push to the database, starting with the first transformation in the pipeline it can push to the database. The Integration Service processes the transformation logic up to the point that it can push the transformation logic to the target database. Then, it executes the generated SQL.

### Full pushdown optimization

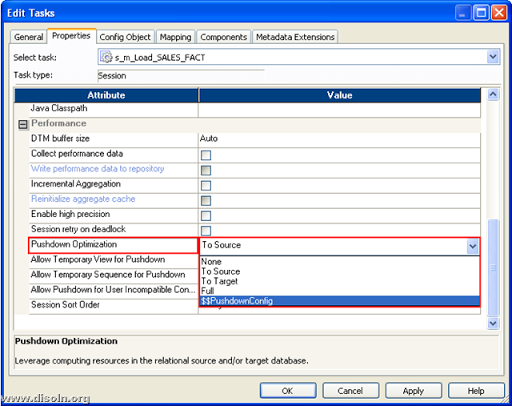
The Integration Service pushes as much transformation logic as possible to both source and target databases. If you configure a session for full pushdown optimization, and the Integration Service cannot push all the transformation logic to the database, it performs partial pushdown optimization instead.

[](http://lh5.ggpht.com/-wXjRkiDSpJo/UfnqCoAZ71I/AAAAAAAAIYY/lPTBTTwnIp0/s1600-h/image%25255B89%25255D.png)

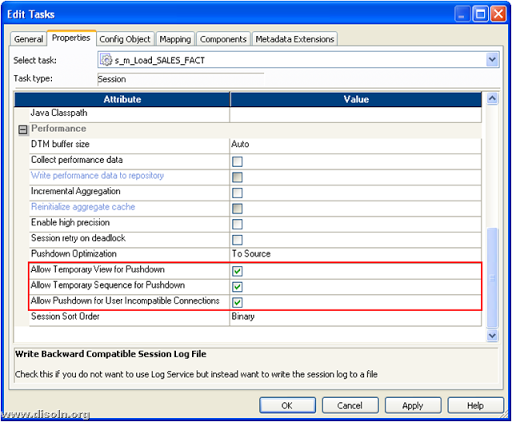
To use full pushdown optimization, the source and target must be on the same database. When you run a session configured for full pushdown optimization, the Integration Service analyzes the mapping starting with the source and analyzes each transformation in the pipeline until it analyzes the target. It generates SQL statements that are executed against the source and target database based on the transformation logic it can push to the database. If the session contains an SQL override or lookup override, the Integration Service generates a view and runs a SELECT statement against this view.

## Configuring Session for Pushdown Optimization

A session can be configured to use pushdown optimization from informatica powercenter workflow manager. You can open the session and choose the Source, Target or Full pushdown optimization as shown in below image.

[](http://lh3.ggpht.com/-0yjcfJTkmgY/Ufn3ISW8iKI/AAAAAAAAIZQ/0KrhUenliqE/s1600-h/image%25255B146%25255D.png)  
You can additionally choose few options to control how integration service push data transformation into SQL statements. Below screen shot shows the available options.

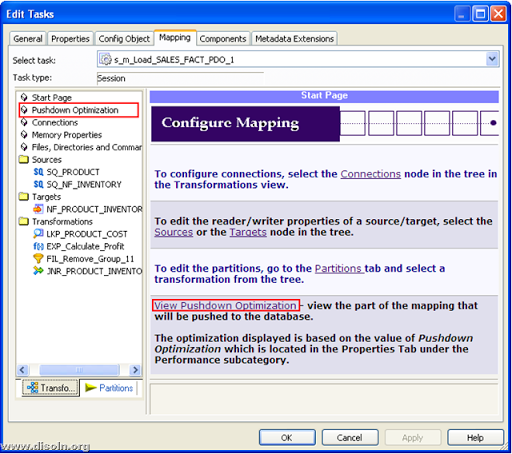
* **Allow Temporary View for Pushdown.** Allows the Integration Service to create temporary view objects in the database when it pushes the session to the database.
* **Allow Temporary Sequence for Pushdown.** Allows the Integration Service to create temporary sequence objects in the database.
* **Allow Pushdown for User Incompatible Connections.** Indicates that the database user of the active database has read permission on the idle databases.

[](http://lh6.ggpht.com/--NScj6z8pcM/Ufn3L63h3GI/AAAAAAAAIZg/Ihw4kzjG8kw/s1600-h/image%25255B141%25255D.png)

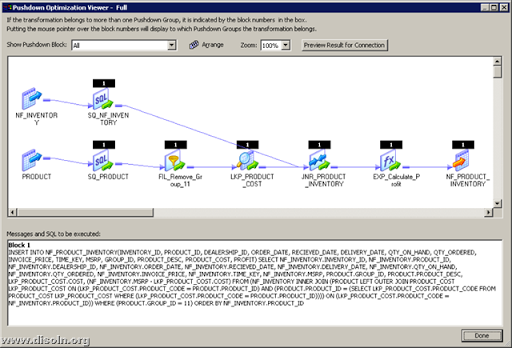
## Using Pushdown Optimization Viewer

Use the Pushdown Optimization Viewer to examine the transformations that can be pushed to the database. Select a pushdown option or pushdown group in the Pushdown Optimization Viewer to view the corresponding SQL statement that is generated for the specified selections.

You can invoke the viewer from highlighted '*Pushdown Optimization'* as shown in below image.

[](http://lh5.ggpht.com/-hxoZutignSA/Ufn3NquxZ8I/AAAAAAAAIZw/qeCqqgmp7R4/s1600-h/image%25255B137%25255D.png)

Pushdown optimizer viewer pops up in a new window and it shows how integration service converts the data transformation logic into SQL statement for a particular mapping. When you select a pushdown option or pushdown group in the viewer, you do not change the pushdown configuration. To change the configuration, we must update the pushdown option in the session properties.

[](http://lh5.ggpht.com/-rqIs6h7vV2E/Ufn3PVeOOQI/AAAAAAAAIaA/zgfdM-qMX1U/s1600-h/image%25255B130%25255D.png)

## Things to Consider before Using Pushdown Optimization

When you run a session for full pushdown optimization, the database must run a long transaction, if the session contains a large quantity of data. Consider the following database performance issues when you generate a long transaction.

* A long transaction uses more database resources.
* A long transaction locks the database for longer periods of time, and thereby reduces the database concurrency and increases the likelihood of deadlock.
* A long transaction can increase the likelihood that an unexpected event may occur.

# Informatica PowerCenter Partitioning for Parallel Processing and Faster Delivery

In addition to a better [ETL design](http://www.disoln.org/search/label/Performance%20Tips), it is obvious to have a session optimized with no bottlenecks to get the best [session performance](http://www.disoln.org/search/label/Performance%20Tips). After optimizing the session performance, we can further improve the performance by exploiting the under utilized hardware power. This refers to parallel processing and we can achieve this in Informatica PowerCenter using Partitioning Sessions.

## What is Session Partitioning

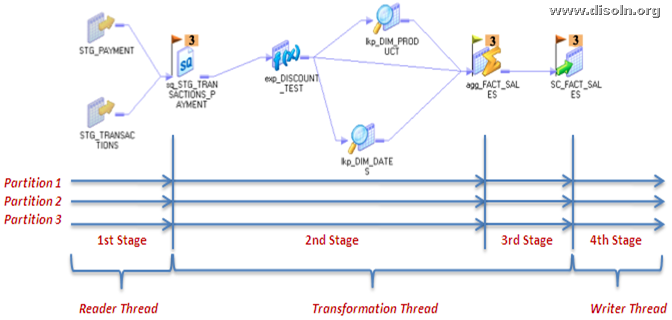
The Informatica PowerCenter Partitioning Option increases the performance of PowerCenter through parallel data processing. Partitioning option will let you split the large data set into smaller subsets which can be processed in parallel to get a better session performance.

## Partitioning Terminology

Lets understand some partitioning terminology before we get into mode details.

* **Partition** **:** A partition is a subset of the data that executes in a single thread.
* **Number of partitions :**We can divide the data set into smaller subset by increasing the number of partitions. When we add partitions, we increase the number of processing threads, which can improve session performance.
* **Stage :**Stage is the portion of a pipeline, which is implemented at run time as a thread.
* **Partition Point :**This is the boundary between two stages and divide the pipeline into stages. Partition point is always associated with a transformation.
* **Partition Type :**It is an algorithm for distributing data among partitions, which is always associated with a partition point. The partition type controls how the Integration Service distributes data among partitions at partition points.

Below image shows the points we discussed above. We have three partitions and three partition points in below session demo.

[](http://lh3.ggpht.com/-uQYFQwF7v80/UeTH7UC1DVI/AAAAAAAAIQc/PjCWYq2iGPE/s1600-h/image%25255B20%25255D.png)

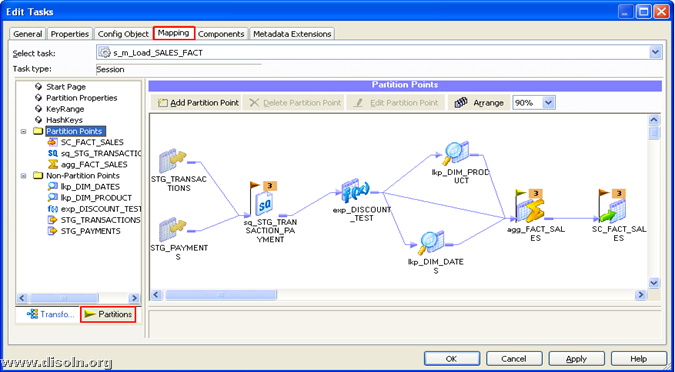
## Type of Session Partitions

Different type of partition algorithms are available.

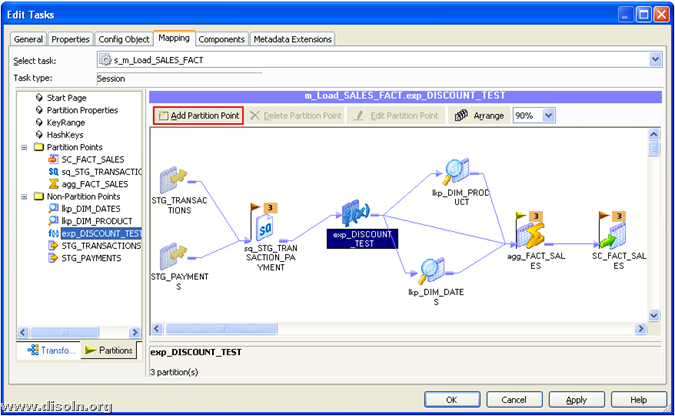
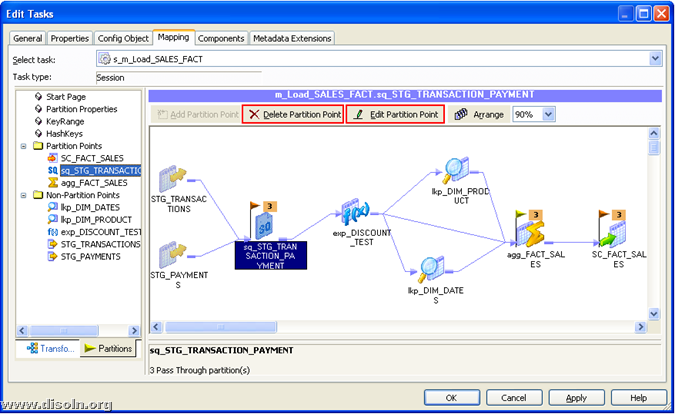
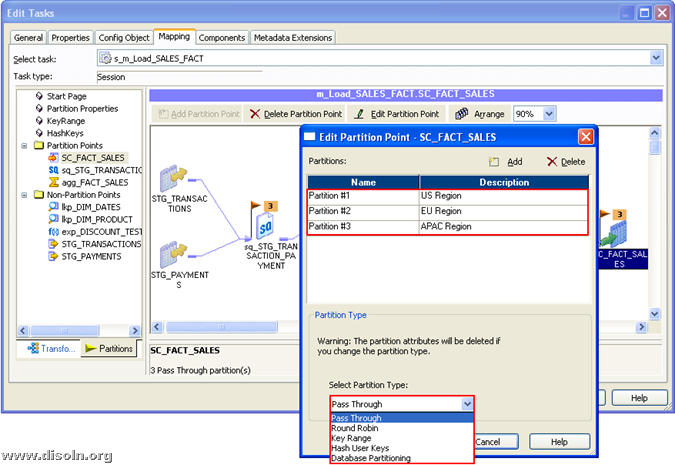
* **Database partitioning**: The Integration Service queries the database system for table partition information. It reads partitioned data from the corresponding nodes in the database.
* **Round-Robin Partitioning** : Using this partitioning algorithm, the Integration service distributes data evenly among all partitions. Use round-robin partitioning when you need to distribute rows evenly and do not need to group data among partitions.
* **Hash Auto-Keys Partitioning** : The PowerCenter Server uses a hash function to group rows of data among partitions. When hash auto-key partition is used, the Integration Service uses all grouped or sorted ports as a compound partition key. You can use hash auto-keys partitioning at or before Rank, Sorter, and unsorted Aggregator transformations to ensure that rows are grouped properly before they enter these transformations.
* **Hash User-Keys Partitioning** : Hash user keys. The Integration Service uses a hash function to group rows of data among partitions based on a user-defined partition key. You choose the ports that define the partition key.
* **Key Range Partitioning**: With this type of partitioning, you specify one or more ports to form a compound partition key for a source or target. The Integration Service then passes data to each partition depending on the ranges you specify for each port.
* **Pass-through Partitioning** : In this type of partitioning, the Integration Service passes all rows at one partition point to the next partition point without redistributing them.

## Setting Up Session Partitions

Lets see what is required to setup a session with partition enabled.  
  
We can invoke the user interface for session partition as shown in below image from your session using the menu Mapping -> Partitions.

[](http://lh5.ggpht.com/-CRSpTnXb9cU/UeX4ETxeAeI/AAAAAAAAIQ0/2XXRAQlNuVQ/s1600-h/image37.png)

The interface will let you Add/Modify Partitions, Partition Points and Choose the type of partition Algorithm. Choose any transformation from the mapping and the "Add Partition Point" button will let you add additional partition points.

[](http://lh3.ggpht.com/-Uqby1xqtOzU/UeX4FuxbvqI/AAAAAAAAIRA/gCNNsogm_Y0/s1600-h/image72.png)  
Choose any transformation from the mapping and the "Delete Partition Point" or "Edit Partition Point" button will let you modify partition points.  
[](http://lh5.ggpht.com/-KGtZuSYSEAA/UeX4GvIhz3I/AAAAAAAAIRU/LZyCVJXTSnU/s1600-h/image82.png)  
The "Add/Delete/Edit Partition Point" opens up an additional window which let you modify the partition and choose the type of the partition algorithm as shown in below image.  
[](http://lh4.ggpht.com/-CGdTPFFArr8/UeX4HlVMEjI/AAAAAAAAIRk/MiPCA3utaHw/s1600-h/image78.png)

# Dynamic Partitioning to Increase Parallelism Based on Resources Availability

Informatica PowerCenter [session partition](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-When-Where-and-How.html) can be used to [process data in parallel](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-for-Parallel-Processing.html) and achieve faster data delivery. Using Dynamic Session Partitioning capability, PowerCenter can dynamically decide the degree of parallelism. The [Integration Service](http://www.disoln.org/2012/09/understand-informatica-powercenter-Workflow-Designer.html) scales the number of session partitions at run time based on factors such as source database partitions or the number of CPUs on the node resulting significant [performance improvement](http://www.disoln.org/search/label/Performance%20Tips?&max-results=15).

## Dynamic Partitioning Methods

The Integration Service can decide the number of session partitions at run time based different factors.

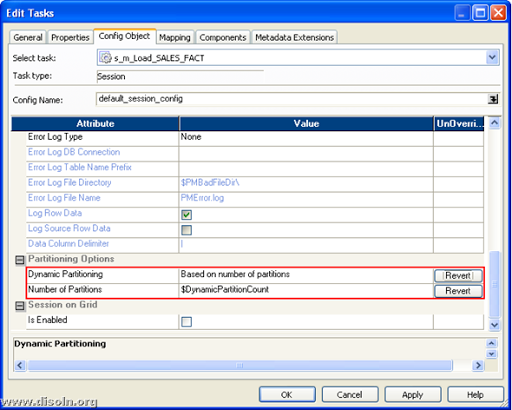
* **Based on source partitioning :** Determines the number of partitions using database partition information. The number of partitions is the maximum of the number of partitions at the source.
* **Based on number of CPUs :** Sets the number of partitions equal to the number of CPUs on the node that prepares the session. If the session is configured to run on a grid, dynamic partitioning sets the number of partitions equal to the number of CPUs on the node that prepares the session multiplied by the number of nodes in the grid.
* **Based on number of nodes in grid :** Sets the partitions to the number of nodes in the grid running the session. If you configure this option for sessions that do not run on a grid, the session runs in one partition and logs a message in the session log.
* **Based on number of partitions :** Sets the partitions to a number that you define in the Number of Partitions attribute. Use the *$DynamicPartitionCount*session parameter, or enter a number greater than 1.

**Note** : Do not configure dynamic partitioning for a session that contains manual partitions. If you set dynamic partitioning and you manually partition the session, the session will be invalid.

## Session Configuration With Dynamic Partitioning

[Session](http://www.disoln.org/2012/09/understand-informatica-powercenter-Workflow-Designer.html) can be setup for dynamic partition from *Config Object* Tab in the session properties. You can choose the available partitioning options from the drop down list as shown in below image.  
[](http://lh4.ggpht.com/-OEGpsnCT_Ao/UgNJgT1r8vI/AAAAAAAAIbg/VMaMbND5usE/s1600-h/image%25255B17%25255D.png)

If the dynamic partition is setup based on *"Based on number of partitions", "Number of Partitions"*property can be set up using the session parameter $DynamicPartitionCount, or enter a number greater than 1 as shown in below image.

[](http://lh6.ggpht.com/-sgXc3FeHprU/UgNOix7LV8I/AAAAAAAAIb0/p_Sgu3oGGJk/s1600-h/image%25255B32%25255D.png)

## Dynamic Partitioning with Different Partition Types

When using different partition types with dynamic partition, following rules must be considered.

**Pass-through partitioning :** If you change the number of partitions at a partition point, the number of partitions in each pipeline stage changes. If you use pass-through partitioning with a relational source, the session runs in one partition in the stage.

**Key range partitioning :** You must define a closed range of numbers or date keys to use dynamic partitioning. Dynamic partitioning does not scale partitions with key range partitioning on relational targets.

**Database partitioning :** When you use database partitioning, the Integration Service creates session partitions based on the source database partitions. This can be used only with Oracle and IBM DB2 sources.

**Hash auto-keys, Hash user keys, Round robin :** Use hash user keys, hash auto-keys, and round-robin partition types to distribute rows with dynamic partitioning. Use hash user keys and hash auto-keys partitioning when you want the Integration Service to distribute rows to the partitions by group. Use round-robin partitioning when you want the Integration Service to distribute rows evenly to partitions.

## Rules and Guidelines for Dynamic Partitioning

Use the following rules and guidelines with dynamic partitioning.

* Dynamic partitioning uses the same connection for each partition.
* You cannot use dynamic partitioning with XML sources and targets.
* Sessions that use SFTP fail if you enable dynamic partitioning.
* When you set dynamic partitioning to a value other than disabled, and you manually partition the session on the Mapping tab, you invalidate the session.

## Using Dynamic Partitioning with Partition Types

The following rules apply to using dynamic partitioning with different partition types.

**Pass-through partitioning :-** If you change the number of partitions at a partition point, the number of partitions in each pipeline stage changes. If you use pass-through partitioning with a relational source, the session runs in one partition in the stage.

**Key range partitioning :-** You must define a closed range of numbers or date keys to use dynamic partitioning. The keys must be numeric or date datatypes. Dynamic partitioning does not scale partitions with key range partitioning on relational targets.

**Database partitioning :-**When you use database partitioning, the Integration Service creates session partitions based on the source database partitions. Use database partitioning with Oracle and IBM DB2 sources.

**Hash auto-keys, hash user keys, or round-robin :-**Use hash user keys, hash auto-keys, and round-robin partition types to distribute rows with dynamic partitioning. Use hash user keys and hash auto-keys partitioning when you want the Integration Service to distribute rows to the partitions by group. Use round-robin partitioning when you want the Integration Service to distribute rows evenly to partitions.

Hope you enjoyed this article and it is informative. Please leave us your comments and feedback.

**Concurrent Workflows to Reduce Warehouse ETL Load Time**

In large data integration projects, it is quite common to source data from [multiple systems](http://www.disoln.org/2012/10/Working-With-Multiple-Data-Sources-and-Aggregator-Transformation.html), sources, regions etc... As the number of data sources increases, the [ETL load time](http://www.disoln.org/search/label/Performance%20Tips) also increases, because of the increasing data volume. One way to reduce the load time is by running different ETL process in parallel. Informatica PowerCenter's capability to run workflow concurrently can be used in such scenarios to reduce the ETL load time. 

## What is Concurrent Workflows

A concurrent workflow is a workflow that can run as multiple instances concurrently. A workflow instance is a representation of a workflow. We can configure two types of concurrent workflows. 

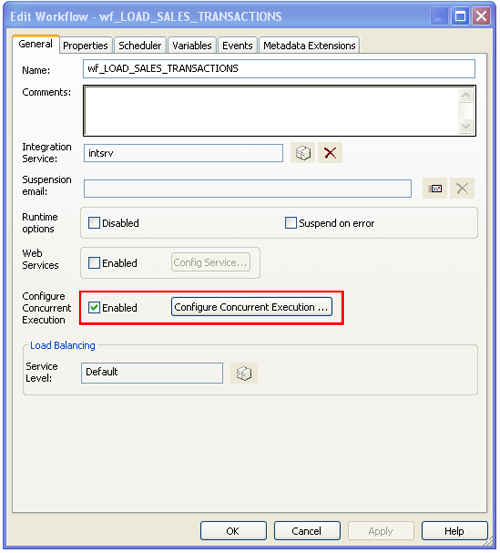
**1. Allow concurrent workflows with the same instance name.** Configure one workflow instance to run multiple times concurrently. Each instance has the same source, target, and variables parameters. The Integration Service identifies each instance by the run ID.

**2. Configure unique workflow instances to run concurrently.** Define each workflow instance name and configure a workflow parameter file for the instance. You can define different sources, targets, and variables in the parameter file.

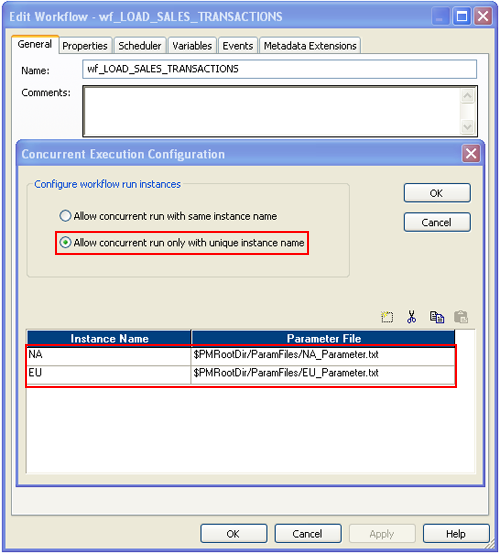
## Concurrent Workflows Configuration

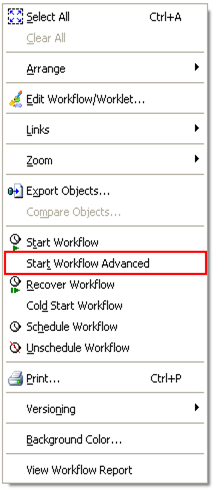
For the demonstration, lets consider a scenario where we need to load daily transaction data from North America, Europe region.  These two files are expected to be available around the same time.

Here we will create one workflow to load the sales transaction data and the same work will be used to load both the file, which can execute concurrently.  
Once the workflow is created, enable concurrent execution as shown in below image.

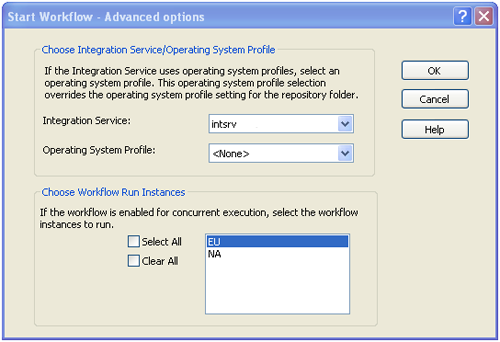


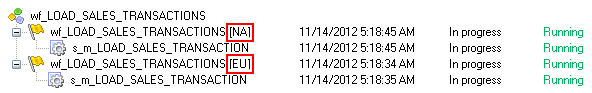
Now Click on “Configure Concurrent Execution” and given the properties as in below image. Provide two different parameter files, which contains the source file information of corresponding region.

  
With that concurrent workflow configuration is done. Now to trigger the workflow, you can start the workflow using "Start Workflow Advanced" option as shown below.



Choose the workflow instance name from the pop up window and click OK to run the selected workflow instance.

  
Form the workflow monitor you can see the running instance of the workflow. As shown in below image you can see the workflow run instance, which is running concurrently.



# Informatica PowerCenter on Grid for Greater Performance and Scalability

Informatica has developed a solution that leverages the power of [grid computing](http://www.tdan.com/view-articles/9378) for greater data integration scalability and [performance](http://www.disoln.org/search/label/Performance%20Tips?&max-results=15). The grid option delivers the load balancing, [dynamic](http://www.disoln.org/2013/08/Dynamic-Partitioning-to-Increase-Parallelism-Based-on-Resources-Availability.html) [partitioning](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-When-Where-and-How.html), [parallel processing](http://www.disoln.org/2013/07/Informatica-PowerCenter-Partitioning-for-Parallel-Processing.html) and [high availability](http://www.disoln.org/2013/07/Workflow-Recovery-Configuration-for-Informatica-PowerCenter-Workflows.html) to ensure optimal scalability, performance and reliability. In this article lets discuss how to setup Infrmatica Workflow to run on grid.

## What is PowerCenter On Grid

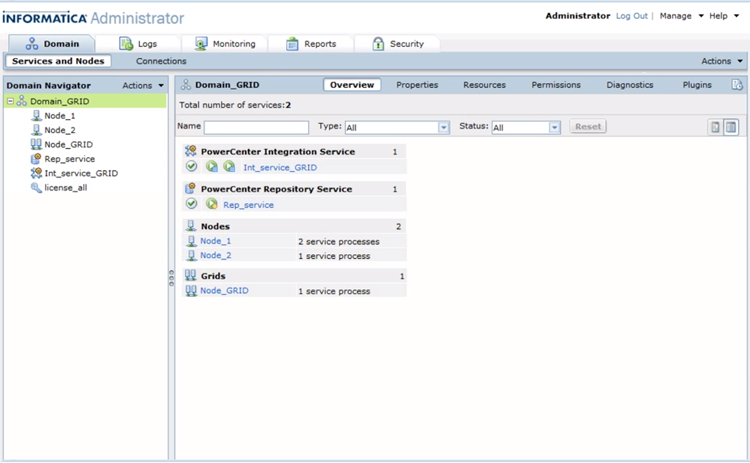
When a PowerCenter domain contains multiple nodes, you can configure workflows and sessions to run on a grid. When you run a workflow on a grid, the Integration Service runs a service process on each available node of the grid to increase performance and scalability. When you run a session on a grid, the Integration Service distributes session threads to multiple DTM processes on nodes in the grid to increase performance and scalability.

**Domain**: A PowerCenter domain consists of one or more nodes in the grid environment. PowerCenter services run on the nodes. A domain is the foundation for PowerCenter service administration.

**Node**: A node is a logical representation of a physical machine that runs a PowerCenter service.

## Admin Console with Grid Configuration

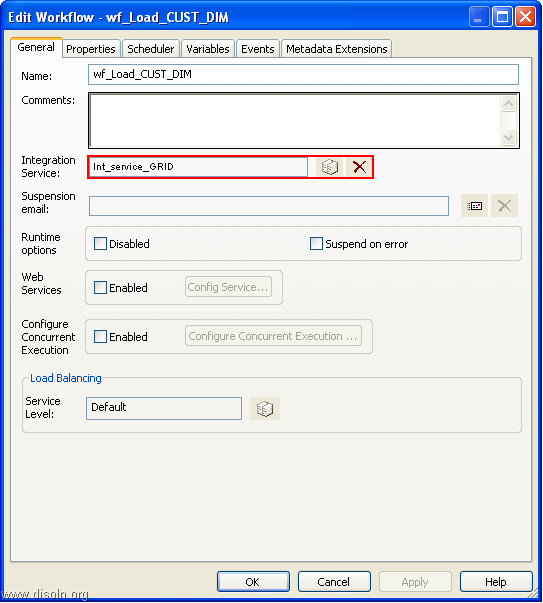
Below shown is an Informatica Admin Console, with two node Grid configuration. We can see two nodes Node\_1, Node\_2 and the Node\_GRID grid created using two nodes. The integration service Int\_service\_GRID is running on the grid.

[](http://lh5.ggpht.com/-zuQDripbCoM/UncAJGR9cHI/AAAAAAAAI2g/QZvjvcW4Kmw/s1600-h/image%25255B22%25255D.png)

## Setting up Workflow on Grid

When you setup a workflow to run grid, the Integration Service distributes workflows across the nodes in a grid. It also distributes the Session, Command, and predefined Event-Wait tasks within workflows across the nodes in a grid.

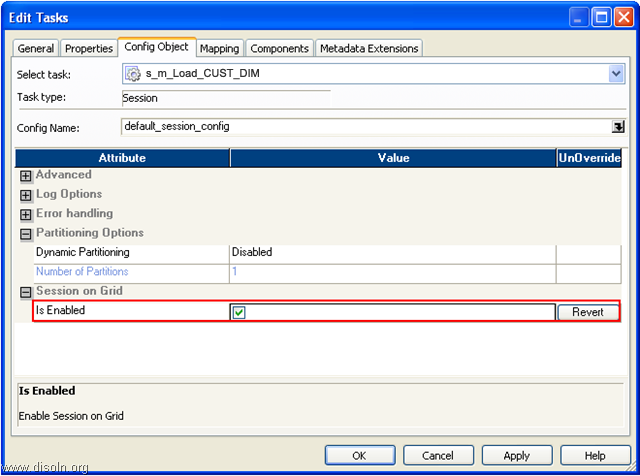
You can setup the workflow to run on grid as shown in below image.You can assign the integration service, which is configured on grid to run the workflow on grid.

[](http://lh5.ggpht.com/-AQF_zRIajXo/UncD939SInI/AAAAAAAAI20/wRQ5cGdpPqY/s1600-h/image%25255B34%25255D.png)

## Setting up Session on Grid

When you run a session on a grid, the Integration Service distributes session threads across nodes in a grid. The Load Balancer distributes session threads to [DTM processes](http://www.disoln.org/2013/08/Informatica-PowerCenter-Performance-Turning-A-to-Z-Guide.html#Anatomy) running on different nodes.  You might want to configure a session to run on a grid when the workflow contains a session that takes a long time to run.

You can setup the session to run on grid as shown in below image.

[](http://lh3.ggpht.com/-Ghgx_cGeOPI/UnciFbxOMcI/AAAAAAAAI3w/cianuzxjCtM/s1600-h/image%25255B87%25255D.png)

## Workflow Running on Grid

Below workflow monitor screen shots sows a workflow running on grid. You see two of the session in the workflow wf\_Load\_CUST\_DIM run on Node\_1 and other one on Node\_1 from 'Task Progress Details' Window.

## [image](http://lh5.ggpht.com/-0E6ndXTtIRs/UncguGINhgI/AAAAAAAAI3M/LH5f9oO2wXc/s1600-h/image%25255B74%25255D.png)

## Key Features and Advantages of Grid

* **Load Balancing** : While facing spikes in data processing, load balance guarantees smooth operations by switching the data processing between nodes on the grid. The node is chosen dynamically based on process size, CPU utilization, memory requirements etc...
* **High Availability** : Grid complements the [High Availability](http://www.disoln.org/2013/07/Workflow-Recovery-Configuration-for-Informatica-PowerCenter-Workflows.html) feature or PowerCenter by switching the master node in case of a node failure. This ensures the monitoring and the shorten time needed for recovery processes.
* **Dynamic Partitioning** : [Dynamic Partitioning](http://www.disoln.org/2013/08/Dynamic-Partitioning-to-Increase-Parallelism-Based-on-Resources-Availability.html) helps making the best use of currently available nodes on the grid. By adapting to available resources, it also helps increasing the performance of the whole ETL process.

# Informatica PowerCenter Load Balancing for Workload Distribution on Grid

Informatica PowerCenter Workflows runs on [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html), distributes workflow tasks across nodes in the [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html). It also distributes Session, Command, and predefined Event-Wait tasks within workflows across the nodes in a [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html). PowerCenter uses load balancer to distribute workflows and session tasks to different nodes. This article describes, how to use load balancer to setup high workflow priorities and how to allocate resources.

## What is Informatica Load Balancing

Informatica load Balancing is a mechanism which distributes the workloads across the nodes in the [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html). When you run a workflow, the Load Balancer dispatches different tasks in the workflow such as Session, Command, and predefined Event-Wait tasks to different [nodes](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html) running the Integration Service. Load Balancer matches task requirements with resource availability to identify the best node to run a task. It may dispatch tasks to a single node or across nodes on the [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html).

## Identifying the Nodes to Run a Task

Load Balancer matches the resources required by the task with the resources available on each node. It dispatches tasks in the order it receives them. You can adjust the workflow priorities and the assign resources needs for tasks, such that load balancer can distribute the tasks to the right nodes and right priority.

|  |  |
| --- | --- |
|  | **Assign service levels** : You assign service levels to workflows. Service levels establish priority among workflow tasks that are waiting to be dispatched. |
|  | **Assign resources** : You assign resources to tasks. Session, Command, and predefined Event-Wait tasks require PowerCenter resources to succeed. If the Integration Service is configured to check resources, the Load Balancer dispatches these tasks to nodes where the resources are available.   Assigning Service Levels to Workflows Service levels determine the order in which the Load Balancer dispatches tasks from the dispatch queue. When multiple tasks are waiting to be dispatched, the Load Balancer dispatches high priority tasks before low priority tasks. You create service levels and configure the dispatch priorities in the Administrator tool.  Integration service will be limited to run You give Higher Service Level for the workflows, which needs to be dispatched first, when multiple workflows are running in parallel. Service Levels are set up in the Admin console.  You assign service levels to workflows on the General tab of the workflow properties as shown below.  [Informatica PowerCenter Load Balancing for Workload Distribution on Grid](http://lh4.ggpht.com/-xKrleZ94Khg/UoBT2cmaEsI/AAAAAAAAI4c/tbylOzlkJAs/s1600-h/image%25255B58%25255D.png) Assigning Resources to Tasks If the Integration Service runs on a [grid](http://www.disoln.org/2013/10/Informatica-PowerCenter-Workflows-on-Grid-for-Performance-and-Scalability.html) and is configured to check for available resources, the Load Balancer uses resources to dispatch tasks. The Integration Service matches the resources required by tasks in a workflow with the resources available on each node in the grid to determine which nodes can run the tasks.   You can configure the resource requirements by the tasks as shown in below image.  Below configuration shows that, the source qualifier needs source file from *File Directory* *NDMSource*, which is accessible only from one node. Available resource on different nodes are configured from Admin console.  [Informatica PowerCenter Load Balancing for Workload Distribution on Grid](http://lh6.ggpht.com/-crzegtRbC5I/UoBWHntH_4I/AAAAAAAAI5A/UjvBSzRZnCs/s1600-h/image%25255B98%25255D.png) |

# Informatica Incremental Aggregation Implementation and Business Use Cases

Incremental Aggregation is the perfect [performance improvement](http://www.disoln.org/search/label/Performance%20Tips) technique to implement; when you have to do aggregate calculations on your incrementally changing source data. Rather than forcing the session to process the entire source data and recalculate the same data each time you run the session, incremental aggregation persist the aggregated value and adds the [incremental changes](http://www.disoln.org/2012/10/An-ETL-Framework-for-Change-Data-Capture-CDC.html) to it. Lets see more details in this article.

**What is Incremental Aggregation**

Using incremental aggregation, you can apply changes captured from the source to aggregate calculations such as Sum, Min, Max, Average etc... If the source [changes incrementally](http://www.disoln.org/2012/10/change-data-capture-cdc-made-easy-using-mapping-variables.html) and you can capture changes, you can configure the session to process those changes. This allows the Integration Service to update the target incrementally, rather than forcing it to process the entire source and recalculate the same data each time you run the session.

## When to Use Incremental Aggregation

**You can capture new source data** : Use incremental aggregation when you can capture new source data each time you run the session. Use a [change data capture](http://www.disoln.org/2012/10/An-ETL-Framework-for-Change-Data-Capture-CDC.html) mechanism for the same.

**Incremental changes do not significantly change the target** : Use incremental aggregation when the changes do not significantly change the target. If processing the incrementally changed source alters more than half the existing target, the session may not benefit from using incremental aggregation. In this case, drop the table and recreate the target with complete source data.

## **How Incremental Aggregation Works**

When the session runs with incremental aggregation enabled for the first time, it uses the entire source data. At the end of the session, the Integration Service stores aggregate data from that session run in two files, the index file and the data file, in the cache directory specified in the Aggregator transformation properties.

Each subsequent time you run the session with incremental aggregation, you use the incremental source changes in the session. For each input record, the Integration Service checks historical information in the index file for a corresponding aggregate group. If it finds a corresponding group, the Integration Service performs the aggregate operation incrementally, using the aggregate data for that group, and saves the incremental change. If it does, the Integration Service creates a new group and saves the record data.

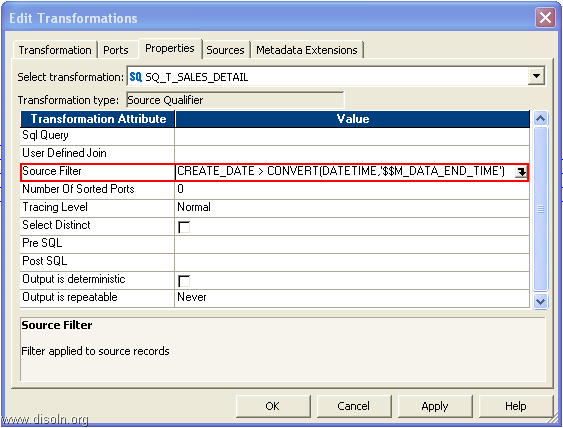
**Note** : Before enabling incremental aggregation, it is important to read [incremental changes](http://www.disoln.org/2012/10/change-data-capture-cdc-made-easy-using-mapping-variables.html) from source to avoid double count.

## **Business Use Case**

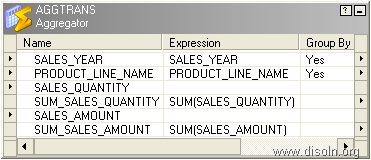
Lets consider an ETL job, which is used to load the Sales Summary Table. The summary table generates yearly sales summary by product line. The table includes the columns 'Sales Year', 'Product Line Name', 'Sales Quantity', 'Sales Amount'

## **Incremental Aggregation Implementation**

Lets create a mapping, which can identify the new sales data from the data source and set the incremental aggregation. New sales data records are identified using the CREATE\_DT column in the source table. The source qualifies of the mapping looks as in below image. The source qualifier is set to read the [changed data](http://www.disoln.org/2012/10/change-data-capture-cdc-made-easy-using-mapping-variables.html) using mapping variables.

[](http://lh3.ggpht.com/--nJlBuFKGDY/U1Rv8eL5chI/AAAAAAAAJUo/bvN3PP07AOw/s1600-h/image%25255B18%25255D.png)

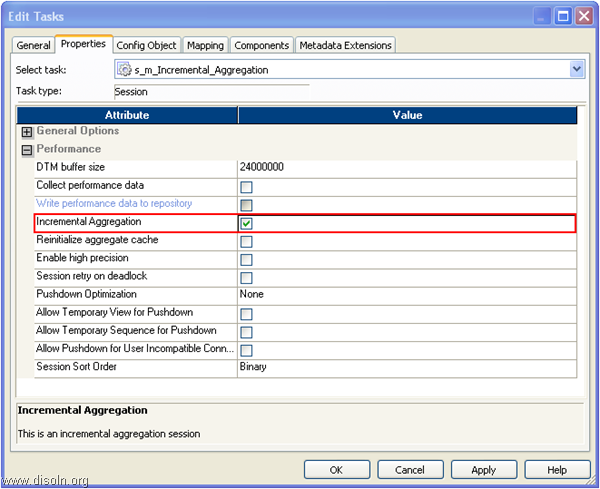
Now do the aggregation calculation using the aggregator transformation as shown in below image.

[](http://lh5.ggpht.com/-_B0AZIsv4iA/U1Rv9_HJVqI/AAAAAAAAJU4/DRyl-iAr6YE/s1600-h/image%25255B21%25255D.png)

Complete the mapping as shown in below image.

[](http://lh6.ggpht.com/-uqe3M2kNbHc/U1Sw9Nt1icI/AAAAAAAAJWA/UR03SEajCG4/s1600-h/image%25255B94%25255D.png)

Create the Workflow and set the incremental aggregation setting in the session property as shown in the image.

[](http://lh3.ggpht.com/-6EjInMkxlgk/U1RwBTc98OI/AAAAAAAAJVY/69Eo31fhrb4/s1600-h/image%25255B51%25255D.png)

**Note** : No need to use an update strategy transformation to implement *Insert* else *Update* logic. You can set the session properties just like '*Insert*' only mapping. When you use the incremental aggregation, Integration Service does the *Insert* or *Update* based on the primary key set in the target table.

## Incremental Aggregation Behind the Scene

Lets understand how incremental aggregator works behind the scene.  For the better understanding lets use the data set from the use case explained above.

### Source data from Day I

On Day 1, all data from the source is read and processed in the mapping.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sales Date** | **Product Line** | **Sales Quantity** | **Sales Amount** | **Create Date** |
| 04-Jan-2014 | Tablet | 1 | $450 | 04-Jan-2014 |
| 03-Feb-2014 | Tablet | 1 | $500 | 03-Feb-2014 |
| 03-Feb-2014 | Computers | 1 | $1,300 | 03-Feb-2014 |
| 13-Mar-2014 | Cell Phone | 2 | $350 | 13-Mar-2014 |

Data from the source is read, summarized and persisted in Aggregator Cache. One row per aggregator group is persisted in the cache.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sales Year** | **Product Line** | **Sales Quantity** | **Sales Amount** | **Note** |
| 2014 | Tablet | 2 | $950 | New In Cache |
| 2014 | Computers | 1 | $1,300 | New In Cache |
| 2014 | Cell Phone | 2 | $350 | New In Cache |

#### 

#### 

### 

### Source data from Day 2

On Day 2, only new data is read from the source and processed in the mapping.

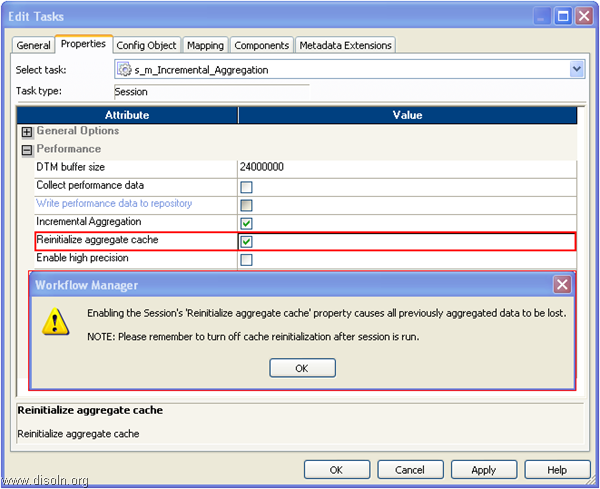
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sales Date** | **Product Line** | **Sales Quantity** | **Sales Amount** | **Create Date** |
| 14-Mar-2014 | Tablet | 1 | $450 | 14-Mar-2014 |
| 14-Mar-2014 | Tablet | 1 | $500 | 14-Mar-2014 |
| 14-Mar-2014 | Video Game | 1 | $300 | 14-Mar-2014 |

Aggregator Cache is updated with the new values and new aggregator groups are inserted.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sales Year** | **Product Line** | **Sales Quantity** | **Sales Amount** | **Note** |
| 2014 | Tablet | 4 | $1,900 | Update In Cache |
| 2014 | Computers | 1 | $1,300 | No Change In Cache |
| 2014 | Cell Phone | 2 | $350 | No Change In Cache |
| 2014 | Video Game | 1 | $300 | New In Cache |

## Reinitializing the Aggregate Cache Files

Based on the use case we discussed here, we need to reset the aggregate cache file for every new year. You can reset the cache file using the settings shown in below image. You get a warning message about clearing the persisted aggregate values, but can be ignored.

[](http://lh6.ggpht.com/-XUZsGtiEmXI/U1RwDb5Y1XI/AAAAAAAAJVo/jWqgTPPQCLs/s1600-h/image%25255B64%25255D.png)

After you run a session that reinitializes the aggregate cache, edit the session properties to disable the Reinitialize Aggregate Cache option. If you do not clear Reinitialize Aggregate Cache, the Integration Service overwrites the aggregate cache each time you run the session.

# Update Huge Tables Using Oracle MERGE

One of the issues we come across during the ETL design is "Update Large Tables".  This is a very common ETL scenarion especially when you treat with large volume of data like loading an [SCD Type 2 Dimension](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html).  We discussed about [a design approach](http://www.disoln.org/2012/07/design-approach-to-update-larger-tables.html)for this scenarion in one of our prior articles. Here in this updated article lets discuss a different approach to update Larger tables using Informatica Mapping.

## **High level Design Approach.**

1. Use Database JOIN to identify the records to be updated.
2. Insert the records into TEMP table, which is identified for UPDATE.
3. Use post session SQL to update the target table.

[](http://3.bp.blogspot.com/-PqxAZDSaxxE/UAy_FLyOp-I/AAAAAAAAElM/6TbOEmdsGPw/s1600/update-large-table-design.png)

## **Design Assumption.**

1. Source and Target tables are relational table.
2. Both source and target table is on the same database.
3. Tables are accessible using a single database user.

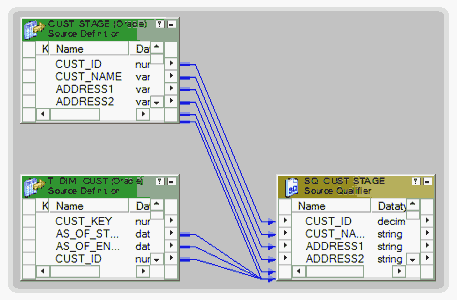
## **Informatica Implementation.**

For the demonstration purpose lets consider the Customer Dimension table T\_DIM\_CUST, which has 100 M records. Each load we are expecting to update 100 K Records records in the Dimension table.

Lets start with the mapping building. As the first step, lets OUTER Join the source table CUST\_STAGE and target table T\_DIM\_CUST. Use the SQL below as the SQL override in source qualifier.

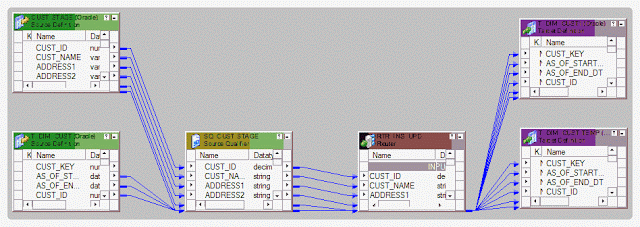
*SELECT  
--Columns From Source Tables  
CUST\_STAGE.CUST\_ID,  
CUST\_STAGE.CUST\_NAME,  
CUST\_STAGE.ADDRESS1,*

CUST\_STAGE.ADDRESS2,  
CUST\_STAGE.CITY,  
CUST\_STAGE.STATE,  
CUST\_STAGE.ZIP,  
--Columns from Target Tables.  
--If any column from T\_DIM\_CUST has NULL value, record to be set as INSERT else UPDATE  
T\_DIM\_CUST.CUST\_ID,  
T\_DIM\_CUST.AS\_OF\_START\_DT,  
T\_DIM\_CUST.AS\_OF\_END\_DT  
T\_DIM\_CUST.CUST\_NAME,  
T\_DIM\_CUST.ADDRESS1,  
T\_DIM\_CUST.ADDRESS2,  
T\_DIM\_CUST.CITY,  
T\_DIM\_CUST.STATE,  
T\_DIM\_CUST.ZIP  
FROM CUST\_STAGE  
--Outer Join is Used  
LEFT OUTER JOIN T\_DIM\_CUST  
ON CUST\_STAGE.CUST\_ID = T\_DIM\_CUST.CUST\_ID  
AND T\_DIM\_CUST.AS\_OF\_END\_DT = TO\_DATE('12-31-4000','MM-DD-YYYY')

[](http://1.bp.blogspot.com/-3VfV84xlkIc/UAzn6MuKsyI/AAAAAAAAElg/Gio0sgxB0-I/s1600/Informatica-SQL-Override.png)

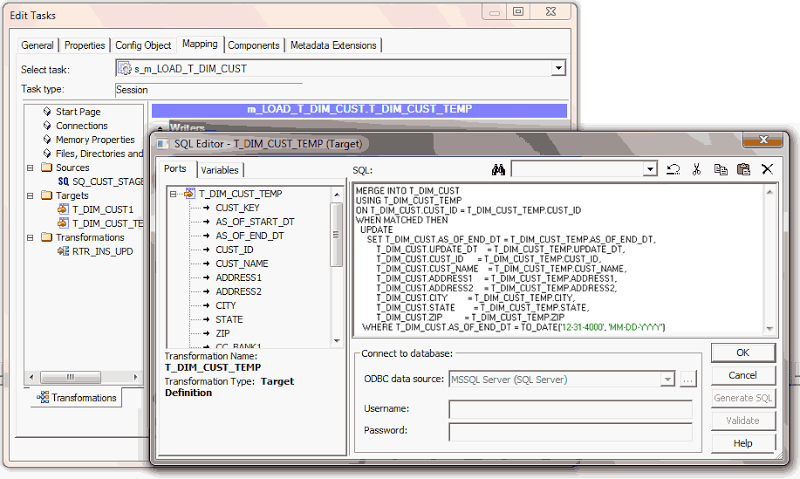
Now using a Router Transformation, route the records to INSERT/UPDATE path. Records identified as INSERT will be mapped to T\_DIM\_CUST and identified as UPDATE will be mapped to T\_DIM\_CUST\_TEMP.  
  
Use T\_DIM\_CUST\_CUST\_ID, which is the column from the target table to identify the records to be inserted/updated. If it is NULL, record will be set for insert else record will be set for update. Below is the Router Group Filter Condition and you can see how the mapping looks like in the below image (Below mapping image has not any transformation logic in it).

* INSERT : IIF(ISNULL( T\_DIM\_CUST\_*CUST\_ID* ), TRUE, FALSE)
* UPDATE : IIF(*NOT*ISNULL( T\_DIM\_CUST\_*CUST\_ID* ), TRUE, FALSE)

[](http://4.bp.blogspot.com/-6YxLSsYmdJA/UAztV0C4hqI/AAAAAAAAEls/AEnpj4A7JGk/s1600/Informatica-Huge-update.png)

Now the mapping development is complete,  during the [session configuration process](http://www.disoln.org/2012/09/understand-informatica-powercenter-Workflow-Designer.html), add the below SQL as part of the Post session SQL statement as shown below. This MERGE INTO SQL will update the records in T\_DIM\_CUST table with the values from T\_DIM\_CUST\_TEMP.

***MERGE******INTO****T\_DIM\_CUST****USING****T\_DIM\_CUST\_TEMP****ON****T\_DIM\_CUST.CUST\_ID = T\_DIM\_CUST\_TEMP.CUST\_ID****WHEN******MATCHED******THEN******UPDATE******SET****T\_DIM\_CUST.AS\_OF\_END\_DT = T\_DIM\_CUST\_TEMP.AS\_OF\_END\_DT,  
         T\_DIM\_CUST.UPDATE\_DT    = T\_DIM\_CUST\_TEMP.UPDATE\_DT,  
         T\_DIM\_CUST.CUST\_ID      = T\_DIM\_CUST\_TEMP.CUST\_ID,  
         T\_DIM\_CUST.CUST\_NAME    = T\_DIM\_CUST\_TEMP.CUST\_NAME,  
         T\_DIM\_CUST.ADDRESS1     = T\_DIM\_CUST\_TEMP.ADDRESS1,  
         T\_DIM\_CUST.ADDRESS2     = T\_DIM\_CUST\_TEMP.ADDRESS2,  
         T\_DIM\_CUST.CITY         = T\_DIM\_CUST\_TEMP.CITY,  
         T\_DIM\_CUST.STATE        = T\_DIM\_CUST\_TEMP.STATE,  
         T\_DIM\_CUST.ZIP          = T\_DIM\_CUST\_TEMP.ZIP****WHERE****T\_DIM\_CUST.AS\_OF\_END\_DT = TO\_DATE('12-31-4000', 'MM-DD-YYYY')*

[](http://lh6.ggpht.com/-XrZpT1Q-R0o/UPYujODnyxI/AAAAAAAAG6w/50X94a9OdcE/Post-Session-SQL_thumb%25255B4%25255D.png?imgmax=800?anchor=)

That is all we need.

# Update Without Update Strategy for Better Session Performance

You might have come across an ETL scenario, where you need to update a huge table with few records and occasional inserts. The straight forward approach of using LookUp transformation to identify the Inserts, Update and Update Strategy to do the Insert or Update may not be right for this particular scenario, mainly because of the LookUp transformation may not perform better and start degrading as the lookup table size increases.

In this article lets talk about a design, which can take care of the scenario we just spoke.

## **The Theory**

When you configure an Informatica PowerCenter session, you have several options for handling database operations such as insert, update, delete.

### **Specifying an Operation for All Rows**

During session configuration, you can select a single database operation for all rows using the Treat Source Rows As setting from the 'Properties' tab of the session.

* 1. **Insert** :- Treat all rows as inserts.
  2. **Delete** :- Treat all rows as deletes.
  3. **Update** :- Treat all rows as updates.
  4. **Data Driven** :- Integration Service follows instructions coded into Update Strategy flag rows for insert, delete, update, or reject.

### **Specifying Operations for Individual Target Rows**

Once you determine how to treat all rows in the session, you can also set options for individual rows, which gives additional control over how each rows behaves. Define these options in the Transformations view on Mapping tab of the session properties.

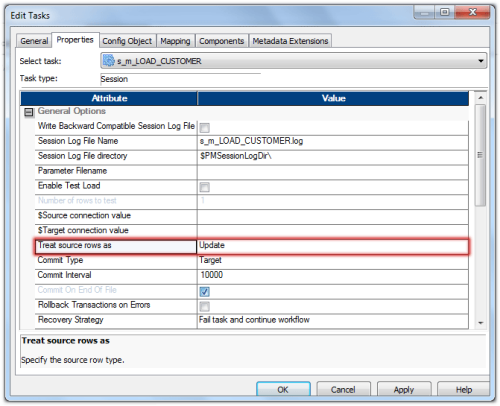
* 1. **Insert** :- Select this option to insert a row into a target table.
  2. **Delete** :- Select this option to delete a row from a table.
  3. **Update** :- You have the following options in this situation:
     + *Update as Update* :- Update each row flagged for update if it exists in the target table.
     + *Update as Insert* :- Insert each row flagged for update.
     + *Update else Insert*:- Update the row if it exists. Otherwise, insert it.
  4. **Truncate Table** :- Select this option to truncate the target table before loading data.

## **Design and Implementation**

Now we understand the properties we need to use for our design implementation.

We can create the mapping just like an 'INSERT' only mapping, with out LookUp, Update Strategy Transformation. During the session configuration lets set up the session properties such that the session will have the capability to both insert and update.

First set Treat Source Rows As property as shown in below image.



Now lets set the properties for the target table as shown below.  Choose the properties Insert and Update else Insert.



Thats all we need to set up the session for update and insert without update strategy.

# [MD5()]Use CHECKSUM Number to Make the Lookup Transformation Simple

Lookup is one of the frequently used transformations in Informatica PowerCenter. This transformation is considered to be relatively simple, but at times this can become complex because of the amount of data to be cached, number of columns used in the lookup condition, checks on columns fetched from lookup table etc... In this article lets see how we can take care of couple of these issues using CHECKSUM number.

## What is Checksum

A checksum is a value used to verify the integrity of a file or a data. Checksums are typically used to compare two sets of data to make sure they are the same. If the checksums don't match those of the original file or data, the data may have been altered.

## Informatica Implementation

Lets consider one of the complex lookup scenario, which we use in [SCD Type 2 mapping](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html) for our demonstration purpose. You can check out the complete [SCD Type 2 Implementation](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html) from one of our previous article.  
  
In  normal [SCD Type 2 mapping](http://www.disoln.org/2012/08/slowly-changing-dimension-type-2-implementation-using-informatica.html) we use a Lookup to fetch all the columns (attributes) from the dimension tables and compare the attributes with the records from the source.  If there is a changed value in any of the attributes, the record will be set for update else insert.  If you are loading a dimension with hundreds of attribute, you can imagine the complexity involved in the lookup in terms of number of columns in the lookup, lookup cache size and comparison of the attributes.

Lets consider the Customer Dimension load, we will be identifying the update based on any change in any of the columns CUST\_NAME, ADDRESS1, ADDRESS2, CITY, STATE or ZIP else the record will be inserted into the Dimension Table.

## Datamodel needs

We need to create an additional database table column to store the CHECKSUM number. Checksum number created by Informatica is 32 character hexadecimal value. So add the column below to the Dimension Table.

* CHK\_SUM\_NB VARCHAR2(32)

This Column in the target table will be populated for every record inserted into the Dimension Table, using the function provided in the next section.

## Checksum number generation

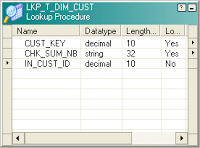
Informatica provides the function **MD5()** for Checksum generation. This function returns a unique 32 character hexadecimal value.  Below expression generates the checksum number for the Customer Dimension.

* MD5(CUST\_NAME || ADDRESS1 || ADDRESS2 || CITY || STATE || TO\_CHAR(ZIP))

Note : This value needs to be populated for every record inserted into the Dimension Table by the mapping.

## Lookup Transformation

Now we can create a simple lookup transformation to fetch the checksum number (CHK\_SUM\_NB, which is already stored in the Dimension table T\_DIM\_CUST) and the surrogate key (CUST\_KEY) from the dimension table.

[](http://4.bp.blogspot.com/-6UXfHkVXphU/UCnReqQw0OI/AAAAAAAAEy4/TkPwrLRI43U/s1600/checksum-lookup.PNG)

You can see the lookup is really simple, with one input and two output ports.

Now we just need a simple expression to identify the records for INSERT and UPDATE as below

IIF(ISNULL(CUST\_KEY),'INSERT',

IIF(NOT ISNULL(CUST\_KEY) AND CHK\_SUM\_NB <> NEW\_CHK\_SUM\_NB, 'UPDATE'))

Note : NEW\_CHK\_SUM\_NB is the Checksum for the incoming record from source table. Calculate the Checksum in an Expression Transformation using **MD5()**function. MD5(CUST\_NAME || ADDRESS1 || ADDRESS2 || CITY || STATE || TO\_CHAR(ZIP))

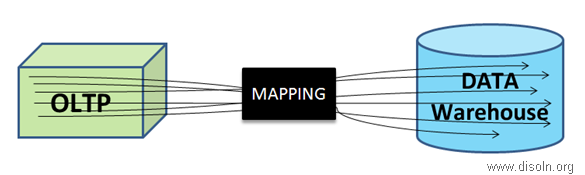
## Design Advantages

* Lookup Transformation made simple and the mapping as well.
* Less number of columns in the lookup cache, hence improved lookup performance.
* Avoid complex expression logic to make the mapping simple.

# SOFT and HARD Deleted Records and Change Data Capture in Data Warehouse

In our couple of prior  articles we spoke about [change data capture](http://www.disoln.org/2012/12/Change-Data-Capture-CDC-Implementation-Using-CHECKSUM-Number.html), different [techniques to capture change data](http://www.disoln.org/2012/10/change-data-capture-cdc-made-easy-using-mapping-variables.html) and a [change data capture frame work](http://www.disoln.org/2012/10/An-ETL-Framework-for-Change-Data-Capture-CDC.html) as well. In this article we will deep dive into different aspects for change data in Data Warehouse including soft and hard deletions in source systems.

**Revisiting Change Data Capture (CDC)**

When we talk about Change Data Capture (CDC) in DW, we mean to capture those changes that have happened at the source side so far after we have run our job last time. In Informatica we call our ETL code as ‘Mapping’, because we MAP the source data (OLTP) into the target data (DW) and the purpose of running the ETL codes is to keep the source and target data in sync, along with some transformations in between, as per the business rules.  
[](http://lh5.ggpht.com/-DktAzQHKc5I/UqUZ5dkOIfI/AAAAAAAAI-g/GFxlEwdA_8s/s1600-h/image%25255B92%25255D.png)

Now, data may get changed at source in three different ways.

* *NEW* transactions happened at source.
* *CORRECTIONS* happened on old transactional values or measured values.
* *INVALID* transactions removed from source.

Usually in our ETL we take care of the 1st and 2nd case(Insert/Update Logic); the 3rd change is not captured in DW unless it is specifically instructed in the requirement specification. But when it’s especially amended, we need to devise convenient ways to track the transactions that were removed i.e., to track the deleted records at source and accordingly DELETE those records in DW.

One thing to make clear is that Purging might be enabled at your OLTP, i.e OLTP keeping data for a fixed historical period of time, but that is a different scenario. Here we are more interested about what was DELETED at Source because the transactions was NOT valid.

## **Effects in DW for Source Data Deletion**

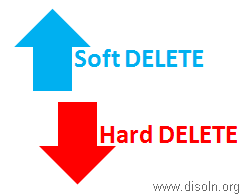
DW tables can be divided into three categories as related to the deleted source data.

1. When the DW table load nature is '*Truncate & Load*' or '*Delete & Reload*', we don't have any impact, since the requirement is to keep the exact snapshot of the source table at any point of time.
2. When the DW table *does not track history on data changes* and deletes are allowed against the source table. If a record is deleted in the source table, it is also deleted in the DW.
3. When the DW table *tracks history on data changes* and deletes are allowed against the source table. The DW table will retain the record that has been deleted in the source system, but this record will be either expired in DW based on the change captured date or 'Soft Delete' will be applied against it.

## **Types of Data Deletion**

Academically, deleting records from DW table is forbidden, however, it’s a common practice in most DWs when we face this kind of situations. Again, if we are deleting records from DW, it has to be done after proper discussions with Business. If your Business requires DELETION, then there are two ways.

* **Logical Delete** :- In this case, we have a specific flag in the source table as STATUS which would be having the values as ‘ACTIVE’ or ‘INACTIVE’. Some OLTPs keep the field name as ACTIVE with the values as ‘I’, ‘U’ or ‘D’, where ‘D’ means that the record is deleted or the record is INACTIVE. This approach is quite safe and also known as **Soft DELETE**.

[](http://lh6.ggpht.com/-9vbcriJm-v8/UqUZ6b_T23I/AAAAAAAAI-s/k03a5SG-guI/s1600-h/image%25255B91%25255D.png)

* **Physical Delete** :- In this case the record related to invalid transactions are fully deleted from the source table by issuing DML statement. This is usually done after thorough discussing with Business Users and related business rules are strictly followed. This is also known as **Hard DELETE**.

## **ETL Perspective on Deletion**

When we have ‘Soft DELETE’ implemented at the source side, it becomes very easy to track the invalid transactions and we can tag those transactions in DW accordingly. We just need to filter the records from source using that STATUS field and issue an UPDATE in DW for the corresponding records. Few things to be kept in mind in this case.

If only ACTIVE records are supposed to be used in ETL processing, we need to add specific filters while fetching source data.

Sometimes INACTIVE records are pulled into the DW and moved till the ETL Data Warehouse level. While pushing the data into Exploration Data Warehouse, only the ACTIVE records are sent for reporting purpose.

For ‘Hard DELETE’, if Audit Table is maintained at source systems for what are transactions were deleted, we can source the same, i.e. join the Audit table and the Source table based on NK and logically delete them in DW too. But it becomes quite cumbersome and costly when no account is kept of what was deleted at all. In these cases, we need to use different ways to track them and update the corresponding records in DW.

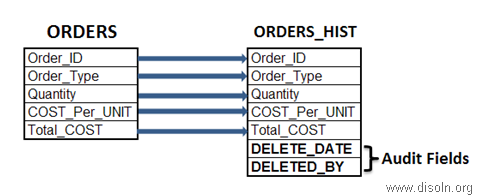
## **Deletion in Data Warehouse : Dimension Vs Fact**

In most of the cases, we see only the transactional records to be deleted from source systems. DELETION of Data Warehouse records are a rare scenario.

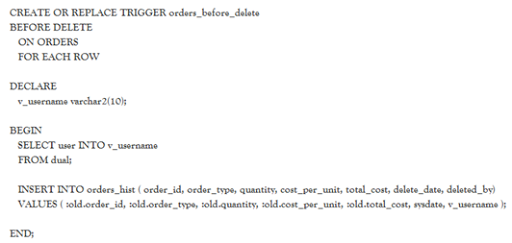
### **Deletion in Dimension Tables**

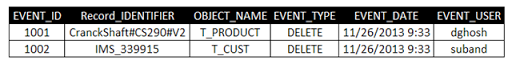
If we have DELETION enabled for Dimensions in DW, it's always safe to keep a copy of the OLD record in some AUDIT table, as it helps to track any defects in future. A simple DELETE trigger should work fine; since DELETION hardly happens, this trigger would not degrade the performance much.

Let's take this ORDERS table into consideration. Along with this, we can have a History table for ORDERS, e.g. ORDERS\_Hist, which would store the DELETED records from ORDERS.

[](http://lh6.ggpht.com/-VSVjiVHQcFo/UqUZ7b07aLI/AAAAAAAAI-8/PS7FeRyBs3U/s1600-h/image%25255B90%25255D.png)

The below Trigger will work fine to achieve this.

[](http://lh5.ggpht.com/-zlS_Mp-rY1Q/UqUZ8IwD62I/AAAAAAAAI_Q/3xkoyA-DMKI/s1600-h/image%25255B89%25255D.png)

The AUDIT Fields will convey when this particular record was deleted and by which user. But this table needs to be created for each and every DW table where we want to keep the audit of what was DELETED. If the entire record is not need and only fields involved in Natural Key(NK) may work, we can have a consolidated table for all the Dimensions.  
[](http://lh5.ggpht.com/-GlAd4xbpLTI/UqUZ9OR79qI/AAAAAAAAI_g/2KtdrlDdjYk/s1600-h/image%25255B88%25255D.png)

Here the Record\_IDENTIFIER field contains the values of all the columns involved in the Natural Key(NK) separated by '#' of the table mentioned in the OBJECT\_NAME field.

Sometimes, we face a situation in DW where a FACT table record contains a Surrogate Key(SK) from a Dimension but the Dimension table doesn't own it anymore. In those cases, the FACT table record becomes orphan and it will hardly be able to appear in any report since we always use the INNER JOIN between Dimensions and Fact while retrieving data in the reporting layer, and there it misses the Referential Integrity(RI).

Suppose, we want to track the orphan records from the SALES Fact table in respect of Product Dimension. We can use the query as below.  
[](http://lh3.ggpht.com/-xEhcjXQo2qc/UqUZ_xMiu-I/AAAAAAAAI_s/nG0_H1IcPZ0/s1600-h/image%25255B51%25255D.png)

So, the above query will provide only the Orphan records, BUT certainly it cannot provide you the records DELETED from the PRODUCT\_Dimension. So, one feasible solution could be while populating the EVENT table with the SKs from PRODUCT\_Dimension that are being DELETED, provided we don't reuse our Surrogate Keys. So, when we have both the SKs and the NKs from the PRODUCT\_Dimension in the EVENT table for DELETED entries, we can achieve a better compliance over the Data Warehouse data.

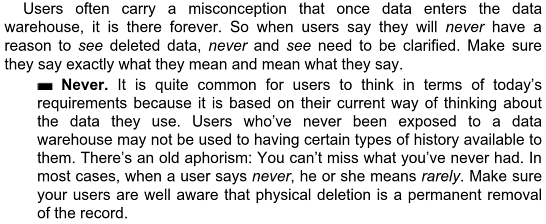
Another useful but least used approach is enabling the [audit](http://docs.oracle.com/cd/B28359_01/server.111/b28337/tdpsg_auditing.htm#TDPSG50000) for any table for DELETE in an Oracle DB using queries like the following.

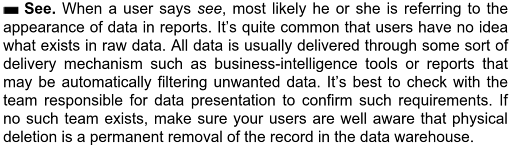
Audit DELETE on SCHEMA.TABLE;

The table [DBA\_AUDIT\_STATEMENT](http://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_3055.htm) will contain all the related details related to this deletion, example the user who issued the, exact DML statement and so on, but this cannot provide you with the record that was deleted. Since this approach cannot directly provide you information on which record was deleted, it’s not so useful in our current discussion, so I would like to keep aloof from the topic here.

### Deletion in Fact Tables

Now, this was all about DELETION in DW Dimension tables. Regarding FACT data DELETION, I would like to cite an extract of what [Ralph Kimball](http://www.kimballgroup.com/) has to say on Physical Deletion of Facts from DW.

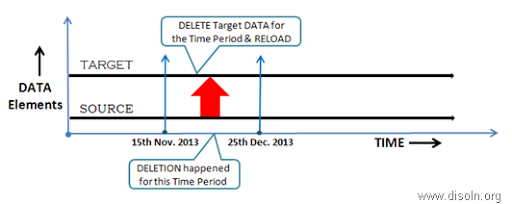




## Change Data Capture & Apply for 'Hard DELETE' in Source

Again, whether we should track the DELETED records from source or not depends on the type of table and its Load Nature. I will share few genuine scenarios that are usually faced in any DW and discuss about the solutions accordingly.

### 1. Records are DELETED from SOURCE for a known Time Period, no Audit Trail was kept.

In this case, the ideal solution is to DELETE the entire records’ set in DW for the Target table and pull the source records once again for the time period. This will bring the DW in sync with Source and DELETED records also will not be available in DW.  
[](http://lh4.ggpht.com/-zpgPts-69zg/UqUaAn2YWFI/AAAAAAAAI_8/CXcw8tUC_DM/s1600-h/image%25255B87%25255D.png)

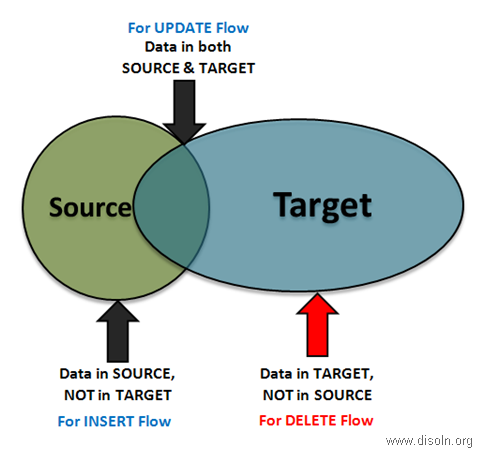
Usually time period is mentioned in terms of Ship\_DATE or Invoice\_DATE or Event\_DATE, i.e. a DATE type field from the actual dataset of the source table is used, and hence the way we can filter the records for Extraction from source table using WHERE clause, we can do the same in DW table as well.

Obviously, in this case we are NOT able to capture the 'Hard DELETE' from the Source i.e., we cannot track the History of DATA, but we would be able to bring the Source and DW in sync at the least. Again, this approach is recommended only when the situation occurs once in a while and not on regular basis.

### 2. Records are DELETED from SOURCE on regular basis with NO Timeframe, no Audit Trail was kept.

The possible solution in this case would be to implement FULL Outer JOIN between the Source and the Target table. The tables should be joined on the fields involved in the Natural Key(NK). This approach will help us to track all three kinds of changes to source data in one shot.

The logic can be better explained with the help of a Venn diagram.

[](http://lh4.ggpht.com/-rQBzV98olM0/UqUaBsaYesI/AAAAAAAAJAM/uaHAaFoxEHc/s1600-h/image%25255B71%25255D.png)

Out of the Joiner (kept in FULL Outer Join mode),

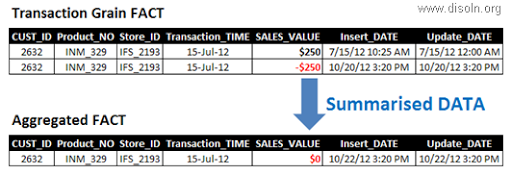
* Records that have values for the NK fields only from the Source and not from the Target, they should go for the INSERT flow. These are all new records coming from source.
* Records that have values for the NK fields from both the Source and the Target, they should go for the UPDATE flow. These are already existing records of Source.
* Records that have values for the NK fields only from Target, will go for the DELETE flow. These are the records that were somehow DELETED from Source table.

Now, what we do with those DELETED records from Source, i.e. apply 'Soft DELETE' or 'Hard DELETE' in DW, depends on our requirement specification and business scenarios.

But this approach is having severe disadvantage in terms of ETL Performance. Whenever we go for a FULL Outer JOIN between Source and Target, we are using the entire data set from both the ends and this will obviously obstruct the smooth processing of ETL when data volume increases.

### 3. Records are DELETED from SOURCE, Audit Trail was kept.

Even though I'm mentioning it a DELETION, it's NOT the kind of Physical DELETION that we discussed previously. This is mainly related to incorrect transactions in Legacy Systems, e.g. Mainframes, which usually send data in flat files.

[](http://lh5.ggpht.com/-FxnyDdpbRDM/UqUaCsYZEZI/AAAAAAAAJAg/r3fRJRA_rvI/s1600-h/image%25255B86%25255D.png)

When some old transactions become invalidated, source team sends those transactions related records again to DW but with inverted measures, i.e. the sales figure are same as the old ones but they are negative. So, DW contains both the old set of records and the newly arrived records, but the aggregated measures become NULL in the aggregated FACT table, thus diminishing the impact of those invalid transactions in DW to NULL.

Only disadvantage of this approach is, Aggregated FACT contains the correct data at the summarized level, but the transactional FACT dual set of records, which together represent the real scenario, i.e. at first the transaction happened(with the older record) and then it became invalid(with the newer record)

# Performance Tuning for Transformation in Informatica

**Joiner Transformation** -

* Always prefer to perform joins in the database if possible, as database joins are faster than joins created in Informatica joiner transformation.
* Sort the data before joining if possible, as it decreases the disk I/O performed during joining.
* Make the table with less no of rows as master table.

**Lookup Transformation** –

* Create an index for the column in a lookup table which is used in lookup condition. Since the lookup table will be queried for looking up the matching data, adding an index would increase the performance.
* If possible, instead of using lookup transformation use join in the database. As database joins are faster, performance will be increased.
* Delete unnecessary columns from the lookup table and keep only the required columns. This will bring down the overhead of fetching the extra columns from the database.

**Filter Transformation** –

* Use filter transformation as early as possible inside the mapping. If the unwanted data can be discarded early in the mapping, it would increase the throughput.'
* Use source qualifier to filter the data. You can also use source qualifier[SQL](http://www.guru99.com/sql.html)override to filter the records, instead of using filter transformation.

**Aggregator Transformation**

* Filter the data before aggregating it. If you are using filter transformation in the mapping, then filter the data before using aggregator as it will reduce the unnecessary aggregation operation.
* Limit the no of ports used in the aggregator transformation. This will reduce the volume of data that aggregator transformation stores inside the cache.

**Source Qualifier Transformation**

* Bring only the required columns from the source. Most of the times not all the columns of the source table are required, so bring only the required fields by deleting the unnecessary columns.
* Avoid using order by clause inside the source qualifier[SQL](http://www.guru99.com/sql.html)override. The order by clause requires additional processing and performance can be increased by avoiding it.

### **Informatica Pipeline Partitioning Explained**

Each mapping contains one or more pipelines. A pipeline consists of a source qualifier, all the transformations and the target. When the Integration Service runs the session, it can achieve higher performance by partitioning the pipeline and performing the extract, transformation, and load for each partition in parallel. Basically a partition is a pipeline stage that executes in a single reader, transformation, or writer thread.

The number of partitions in any pipeline stage equals the number of threads in the stage. **By default, the Integration Service creates one partition in every pipeline stage.** If we have the Informatica Partitioning option, we can configure multiple partitions for a single pipeline stage.

Setting partition attributes includes partition points, the number of partitions, and the partition types. In the session properties we can add or edit partition points. When we change partition points we can define the partition type and add or delete partitions (number of partitions).

We can set the following attributes to partition a pipeline:

1. **Partition point:**

Partition points mark thread boundaries and divide the pipeline into stages. A stage is a section of a pipeline between any two partition points. The Integration Service redistributes rows of data at partition points. When we add a partition point, we increase the number of pipeline stages by one. Increasing the number of partitions or partition points increases the number of threads.

We cannot create partition points at Source instances or at Sequence Generator transformations.

1. **Number of partitions:**

A partition is a pipeline stage that executes in a single thread. If we purchase the Partitioning option, we can set the number of partitions at any partition point. When we add partitions, we increase the number of processing threads, which can improve session performance. We can define up to 64 partitions at any partition point in a pipeline. When we increase or decrease the number of partitions at any partition point, the Workflow Manager increases or decreases the number of partitions at all partition points in the pipeline. The number of partitions remains consistent throughout the pipeline. The Integration Service runs the partition threads concurrently.

1. **Partition types:**

The Integration Service creates a default partition type at each partition point. If we have the Partitioning option, we can change the partition type. The partition type controls how the Integration Service distributes data among partitions at partition points.

We can define the following partition types here: Database partitioning, Hash auto-keys, Hash user keys, Key range, Pass-through, Round-robin.

* + **Database partitioning:**

The Integration Service queries the database system for table partition information. It reads partitioned data from the corresponding nodes in the database.

* + **Pass-through:**

The Integration Service processes data without redistributing rows among partitions. All rows in a single partition stay in the partition after crossing a pass-through partition point. Choose pass-through partitioning when we want to create an additional pipeline stage to improve performance, but do not want to change the distribution of data across partitions.

* + **Round-robin:**

The Integration Service distributes data evenly among all partitions. Use round-robin partitioning where we want each partition to process approximately the same numbers of rows i.e. load balancing.

* + **Hash auto-keys:**

The Integration Service uses a hash function to group rows of data among partitions. The Integration Service groups the data based on a partition key. The Integration Service uses all grouped or sorted ports as a compound partition key. We may need to use hash auto-keys partitioning at Rank, Sorter, and unsorted Aggregator transformations.

* + **Hash user keys:**

The Integration Service uses a hash function to group rows of data among partitions. We define the number of ports to generate the partition key.

* + **Key range:**

The Integration Service distributes rows of data based on a port or set of ports that we define as the partition key. For each port, we define a range of values. The Integration Service uses the key and ranges to send rows to the appropriate partition. Use key range partitioning when the sources or targets in the pipeline are partitioned by key range.

### **Points to consider while using Informatica partitions**

* We cannot create a partition key for hash auto-keys, round-robin, or pass-through types partitioning
* If you have bitmap index defined upon the target and you are using pass-through partitioning to, say Update the target table - the session might fail as bitmap index creates serious locking problem in this scenario
* Partitioning considerably increases the total DTM buffer memory requirement for the job. Ensure you have enough free memory in order to avoid memory allocation failures
* When you do pass-through partitioning, Informatica will try to establish multiple connection requests to the database server. Ensure that database is configured to accept high number of connection requests
* As an alternative to partitioning, you may also use native database options to increase degree of parallelism of query processing. For example in Oracle database you can either specify PARALLEL hint or alter the DOP of the table in subject.
* If required you can even combine Informatica partitioning with native database level parallel options - e.g. you create 5 pass-through pipelines, each sending query to Oracle database with PARALLEL hint.

### PARTITIONING

* A pipeline consists of a source qualifier and all the transformations and Targets that receive data from that source qualifier.
* When the Integration Service runs the session, it can achieve higher Performance by partitioning the pipeline and performing the extract, Transformation, and load for each partition in parallel.

A partition is a pipeline stage that executes in a single reader, transformation, or Writer thread. The number of partitions in any pipeline stage equals the number of Threads in the stage. By default, the Integration Service creates one partition in every pipeline stage.

**PARTITIONING ATTRIBUTES**

**1. Partition points**

* By default, IS sets partition points at various transformations in the pipeline.
* Partition points mark thread boundaries and divide the pipeline into stages.
* A stage is a section of a pipeline between any two partition points.

**2. Number of Partitions**

* we can define up to 64 partitions at any partition point in a pipeline.
* When we increase or decrease the number of partitions at any partition point, the Workflow Manager increases or decreases the number of partitions at all Partition points in the pipeline.
* increasing the number of partitions or partition points increases the number of threads.
* The number of partitions we create equals the number of connections to the source or target. For one partition, one database connection will be used.

[clip_image006](http://lh5.ggpht.com/_MbhSjEtmzI8/TapQ-kl8ZOI/AAAAAAAAALY/YbKFtydQ-50/s1600-h/clip_image0065.gif)[clip_image007](http://lh3.ggpht.com/_MbhSjEtmzI8/TapQ_WYbjjI/AAAAAAAAALg/CSTesptpjWQ/s1600-h/clip_image0075.gif)[clip_image011](http://lh5.ggpht.com/_MbhSjEtmzI8/TapRAVP1rVI/AAAAAAAAALo/yGRRj-aZ8vc/s1600-h/clip_image0115.gif)[clip_image027](http://lh3.ggpht.com/_MbhSjEtmzI8/TapRBWxTPVI/AAAAAAAAALw/5-VEZuT22uY/s1600-h/clip_image0275.gif)

**3. Partition types**

* The Integration Service creates a default partition type at each partition point.
* If we have the Partitioning option, we can change the partition type. This option is purchased separately.
* The partition type controls how the Integration Service distributes data among partitions at partition points.

[clip_image007[1]](http://lh4.ggpht.com/_MbhSjEtmzI8/TapRCYM8H7I/AAAAAAAAAL4/ar-Lv-o43MM/s1600-h/clip_image00712.gif)**PARTITIONING TYPES**

**1. Round Robin Partition Type**

* In round-robin partitioning, the Integration Service distributes rows of data evenly to all partitions.
* Each partition processes approximately the same number of rows.
* Use round-robin partitioning when we need to distribute rows evenly and do not need to group data among partitions.

**2. Pass-Through Partition Type**

* In pass-through partitioning, the Integration Service processes data without Redistributing rows among partitions.
* All rows in a single partition stay in that partition after crossing a pass-Through partition point.
* Use pass-through partitioning when we want to increase data throughput, but we do not want to increase the number of partitions.

**3. Database Partitioning Partition Type**

* Use database partitioning for Oracle and IBM DB2 sources and IBM DB2 targets only.
* Use any number of pipeline partitions and any number of database partitions.
* We can improve performance when the number of pipeline partitions equals the number of database partitions.

**Database Partitioning with One Source**

When we use database partitioning with a source qualifier with one source, the Integration Service generates SQL queries for each database partition and distributes the data from the database partitions among the session partitions Equally.

For example, when a session has three partitions and the database has five partitions, 1stand 2nd session partitions will receive data from 2 database partitions each. Thus four DB partitions used. 3rd Session partition will receive Data from the remaining 1 DB partition.

**Partitioning a Source Qualifier with Multiple Sources Tables**

The Integration Service creates SQL queries for database partitions based on the Number of partitions in the database table with the most partitions.

If the session has three partitions and the database table has two partitions, one of the session partitions receives no data.

**4. Hash Auto-Keys Partition Type**

* The Integration Service uses all grouped or sorted ports as a compound Partition key.
* Use hash auto-keys partitioning at or before Rank, Sorter, Joiner, and Unsorted Aggregator transformations to ensure that rows are grouped Properly before they enter these transformations.

**5. Hash User-Keys Partition Type**

* The Integration Service uses a hash function to group rows of data among Partitions.
* we define the number of ports to generate the partition key.
* we choose the ports that define the partition key .

**6. Key range Partition Type**

* We specify one or more ports to form a compound partition key.
* The Integration Service passes data to each partition depending on the Ranges we specify for each port.
* Use key range partitioning where the sources or targets in the pipeline are Partitioned by key range.
* **Example:** Customer 1-100 in one partition, 101-200 in another and so on. We Define the range for each partition.

# A Quick Guide to Informatica Version Control

Repository Version control in informatica is been around from some years. Still it is not common in industry as it is a paid utility. Here is my understanding of informatica version control for new developers of informatica.

Intention here is to share overview and the people who are in industry from quite a good time can avoid this post.

# What is informatica version control?

You can control the changes over source code. Developers can retain the multiple copies of source code and can give them a label (name). Informatica version control [tracks the code changes](http://etllabs.com/informatica/informatica-versioning/556/). Informatica Version control retains the accountability for a source code change. In simple words you can identify that which developer has done what.

# How to do version control in informatica?

Code version control is gets operated by check-in and check-out options of informatica.

Check out: Check out the object if you want to modify code. Informatica will place write-intent lock to it. No other user can edit the object after check-out.

Check in: Save the object after editing. After check-in your changes will be visible to others. Now object can be edited by other user. Write-intent lock will be released.

Undo Check-out: This option will rollback all the changes. Informatica will restore latest checked-in version to repository.

# Is version control must?

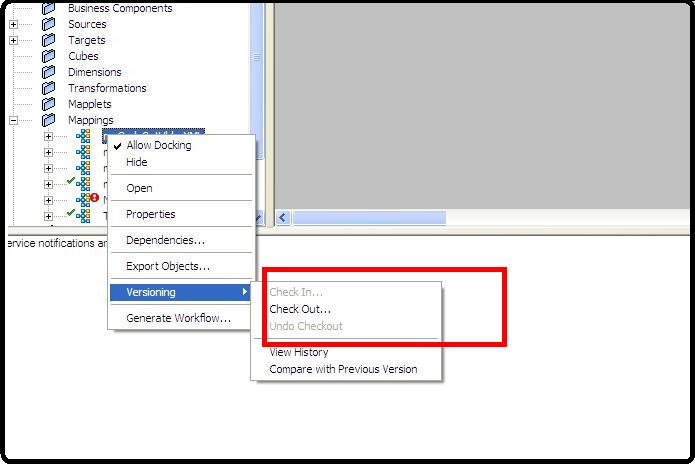
Engineers always debate performance overhead with the capability informatica version control provides. My personal experience with version control in informatica was amazing and i did not notice any performance issue in accessing objects.

Version control is a must feature to be enabled provided org can afford it.

# Which Version control: Built-in or Third Party (SVN, GIT, etc.)?

 Being built-in makes it convenient to use. Third party tool creates many overhead.  One has to export the object in XML format, which further need to be checked in to version control tool.   Someone has to take care of this task regularly which leads to cost overhead.

So it is advisable to use informatica versioning tool.

[](http://etllabs.com/wp-content/uploads/Informatica-version-control.jpg)

Informatica version control options

**Difference between Function and Procedure.**

1. Function is mainly used in the case where it must return a value. Where as a procedure may or may not return a value or may return more than one value using the OUT parameter.

2. Function can be called from SQL statements where as procedure can not be called from the sql statements

3. Functions are normally used for computations where as procedures are normally used for executing business logic.

4. You can have DML (insert,update, delete) statements in a function. But, you cannot call such a function in a SQL query.

5. Function returns 1 value only. Procedure can return multiple values (max 1024).

6. Stored Procedure: supports deferred name resolution. Example while writing a stored procedure that uses table named tabl1 and tabl2 etc. but actually not exists in database is allowed only in during creation but runtime throws error Function wont support deferred name resolution.

7. Stored procedure returns always integer value by default zero. where as function return type could be scalar or table or table values

8. Stored procedure is precompiled execution plan where as functions are not.

9. A procedure may modify an object where a function can only return a value The RETURN statement immediately completes the execution of a subprogram and returns control to the caller.

**What is NoSQL**

NoSQL is not the name of any particular database instead it refers to a broad class of non-relational databases that differ from classical relational database management systems (RDBMS) in some significant aspects, most notably because they do not use SQL as their primary query language, instead providing access by means of Application Programming Interfaces (API).

NoSQL can be considered "Internet age" databases that are being used by Amazon, Facebook, Google and the like to address performance and scalability requirements that cannot be met by traditional relational databases.

One major difference between traditional relational databases and NoSQL is that the latter do not generally provide guarantees for atomicity, consistency, isolation and durability (commonly known as ACID property), although some support is beginning to emerge. Instead of ACID, NoSql databases more or less follow something called "BASE".

The other major difference is, NoSQL databases are generally schema-less - that is records in these databases do not require to conform to a pre-defined storage schema.

**Informatica command line objects/programs:**

Informatica includes the following command line programs:

**1. infacmd -** Use infacmd to access the Informatica application services.

**2. infasetup -** Use infasetup to complete installation tasks such as defining a node or a domain.

**3. pmcmd -** Use pmcmd to manage workflows. You can start, stop, schedule, and monitor workflows using pmcmd.

**4. pmrep -** Use pmrep to complete repository administration tasks such as listing repository objects, creating and editing groups, and restoring and deleting repositories.

**pmcmd example:**

cd C:\Informatica\9.0.1\server\bin\

pmcmd startworkflow -sv PROD\_INTEG -d Domain\_INFORMATICA -u "Administrator" -p "Administrator" -f "FOLDER\_NAME" wf\_m\_JOBS\_ALL

**pmrep example:**

backupproduction.bat

REM This batch file uses pmrep to connect to and back up the repository Production on the server

ServerName

@echo off

echo Connecting to repository Production...

Using pmrep 611

c:\PowerCenter\pmrep\pmrep connect -r Production -n Administrator -x Adminpwd -d MyDomain -h

Machine -o 8080

echo Backing up repository Production...

c:\PowerCenter\pmrep\pmrep backup -o c:\backup\Production\_backup.rep

# What is DTM Process (Data Transformation Manager)?

After the load manager performs validations for the session, it creates the DTM process. The DTM process is the second process associated with the session run. The primary purpose of the DTM process is to create and manage threads that carry out the session tasks.

� The DTM allocates process memory for the session and divide it into buffers. This is also known as buffer memory. It creates the main thread, which is called the master thread. The master thread creates and manages all other threads.

� If we partition a session, the DTM creates a set of threads for each partition to allow concurrent processing.. When Informatica server writes messages to the session log it includes thread type and thread ID.

**Following are the types of threads that DTM creates:**  
  
Master Thread - Main thread of the DTM process. Creates and manages all other threads.  
  
Mapping Thread - One Thread to Each Session. Fetches Session and Mapping Information.  
  
Pre and Post Session Thread - One Thread each to Perform Pre and Post Session Operations.  
  
Reader Thread - One Thread for Each Partition for Each Source Pipeline.  
  
Writer Thread - One Thread for Each Partition if target exist in the source pipeline write to the target.  
  
Transformation Thread - One or More Transformation Thread For Each Partition.