

Working with Data Flow Involving 2 Infosources in BW 7.0 – A Scenario



Applies to:

SAP BW release 7.0 and above.

Summary

This document intends to explain the concept of using 2 Infosources while creating a data flow between a source and target. The central transformation concept is also explained using an example scenario. Further the advantages of using such a data flow are explained and a possible error scenario is also covered as a part of this document.

Author: Ramachandran Srinivasan

Company: Cognizant Technology Solutions Ltd.,

Created on: 23 September 2011

Author Bio



Ramachandran Srinivasan is working for Cognizant Technology Solutions as an SAP BI Consultant.

Table of Contents

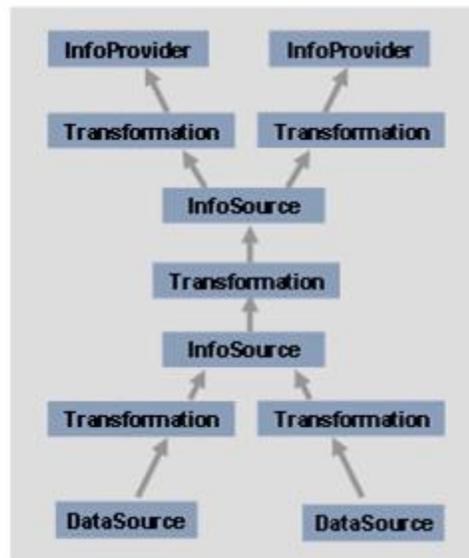
The concept	3
Data Flow with Two Infosources:	3
How this is done:	3
The advantage:	4
A Scenario:.....	4
The Central Transformation:	7
Related Content	8
Disclaimer and Liability Notice.....	9

The concept

Some of us would have come across the kind of developments where we would use 2 Infosources in a data flow between a source and a data target. This type of flow is used if your data flow not only contains two different sources, but the data is to be written to multiple identical (or almost identical) targets. The required business rules are executed in the central transformation so that you only have to modify the one transformation in order to change the business rules. You can connect sources and targets that are independent of this transformation.

This document explains the uses of this type of data flow and a sample scenario which explains the advantages and possible issues with this type of data flow.

Data Flow with Two Infosources:



(Picture taken from help.sap.com)

How this is done:

First let us take an example data flow where data load is being done from 2 or more DSOs to 2 or more cubes. As stated earlier, in this scenario all the sources would have similar structures and the same is the case with the target info providers.

The steps involved in creating such a data flow are as follows:

1. Create a first level infosource as a copy of one of the source DSOs, so that it has the same structure as that of the source.
2. Create transformations between the data sources and the first level infosource. Note that all the mappings in these transformations are direct mapping.
3. Create a second level infosource which is a copy of the data target, so that it has the same structure as that of the target info provider.
4. Create transformations from the second level infosource to the data targets which in our case are the cubes. Note that all the mappings in these transformations are direct mapping.
5. Now create a central transformation that would contain all the rules for data to be transformed from the 2 sources to 2 targets.
6. Create a DTP from each source to each data target.

The advantage:

The main advantage of using such data flow is its scalability. If a source needs to be added, then all that we need to do is create a transformation between the source and the First infosource and do a direct mapping for all the rules.

The same is the case for adding a target.

Any change in the rules would have to be effected at only one place and that is the central transformation. This contributes to the maintainability of the system.

A Scenario:

Let us consider a company which extensively uses FI module and in SAP BI there is currently one DSO ZFIGLDSO and one cube ZFIGLCUB. Let us assume that the extractor runs twice every day and brings about a million records to the DSO during every run. Since the data in the cube has been increasing rapidly over the years the company has decided to split the data year wise and store them in individual cubes for each year. Once in 3 years the data older than 3 years are archived.

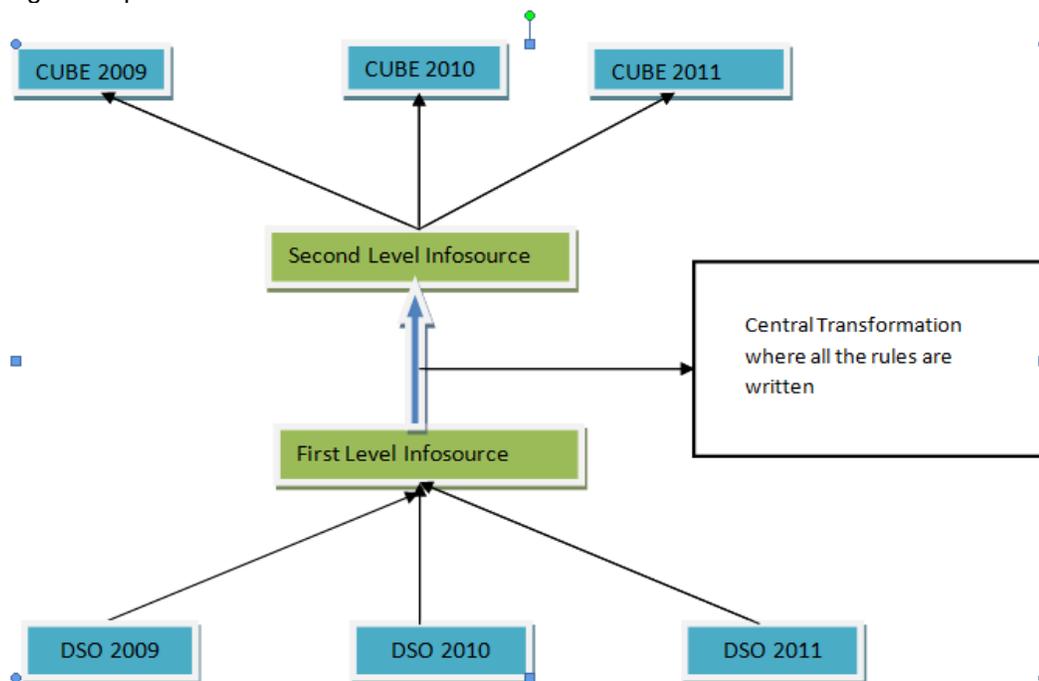
Let us assume the year is 2011.

To build this scenario, we would be creating the following objects:

1. DSOs: 3 DSO for the year 2009, 2010 and 2011.
2. CUBEs: 3 DSO for the year 2009, 2010 and 2011.

We also understand that for all the years, the rules of processing the data from the DSO to the cube would be the same. So we now have a situation where we can use 2 Infosources between the sources and targets and have a central transformation in between the 2 Infosources.

The following diagram depicts the data flow:

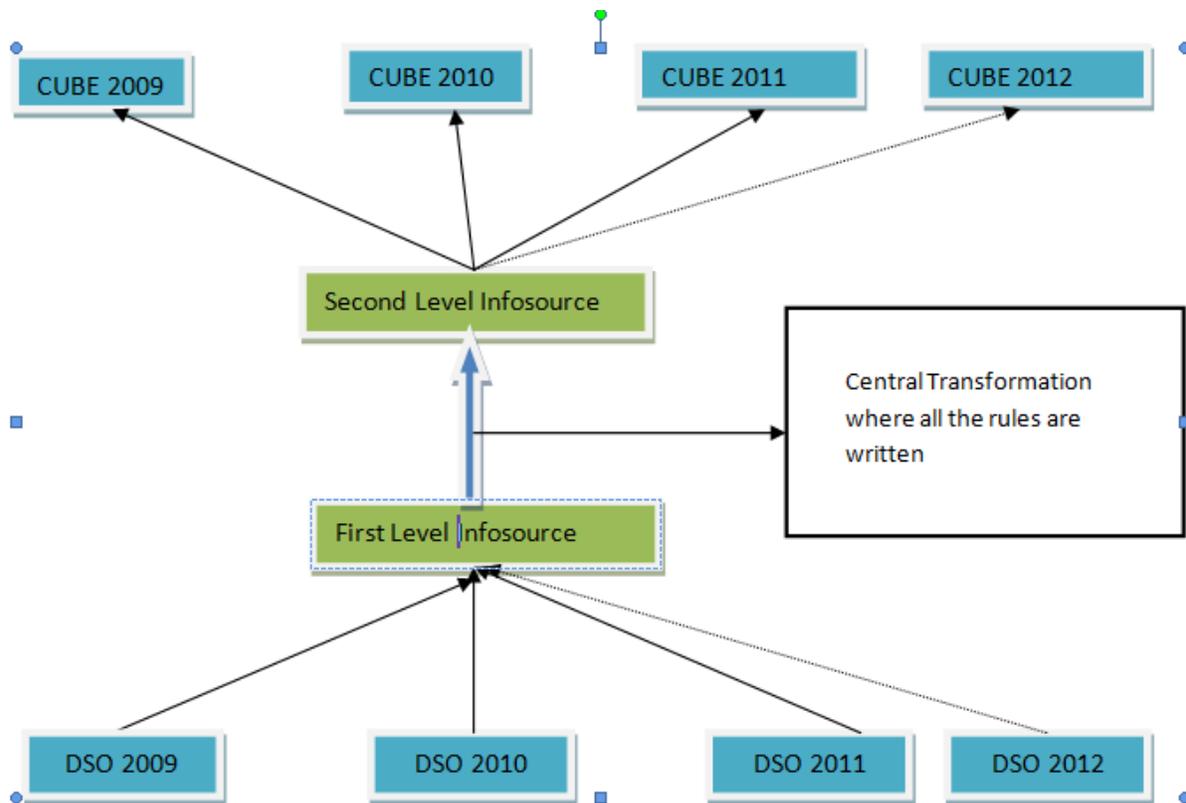


As seen above the following steps are needed to create this data flow:

1. Create 3 DSOs with the same structure as that of ZFIGLDSO.
2. Create a First level infosource which is a copy of the DSO.
3. Create transformation with direct mapping from each DSO to the First level infosource.
4. Create 3 Cubes with the same structure as that of ZFIGLCUB.
5. Create a Second level infosource which is a copy of the cube.
6. Create transformations from with direct mapping from Second level infosource to each cube.

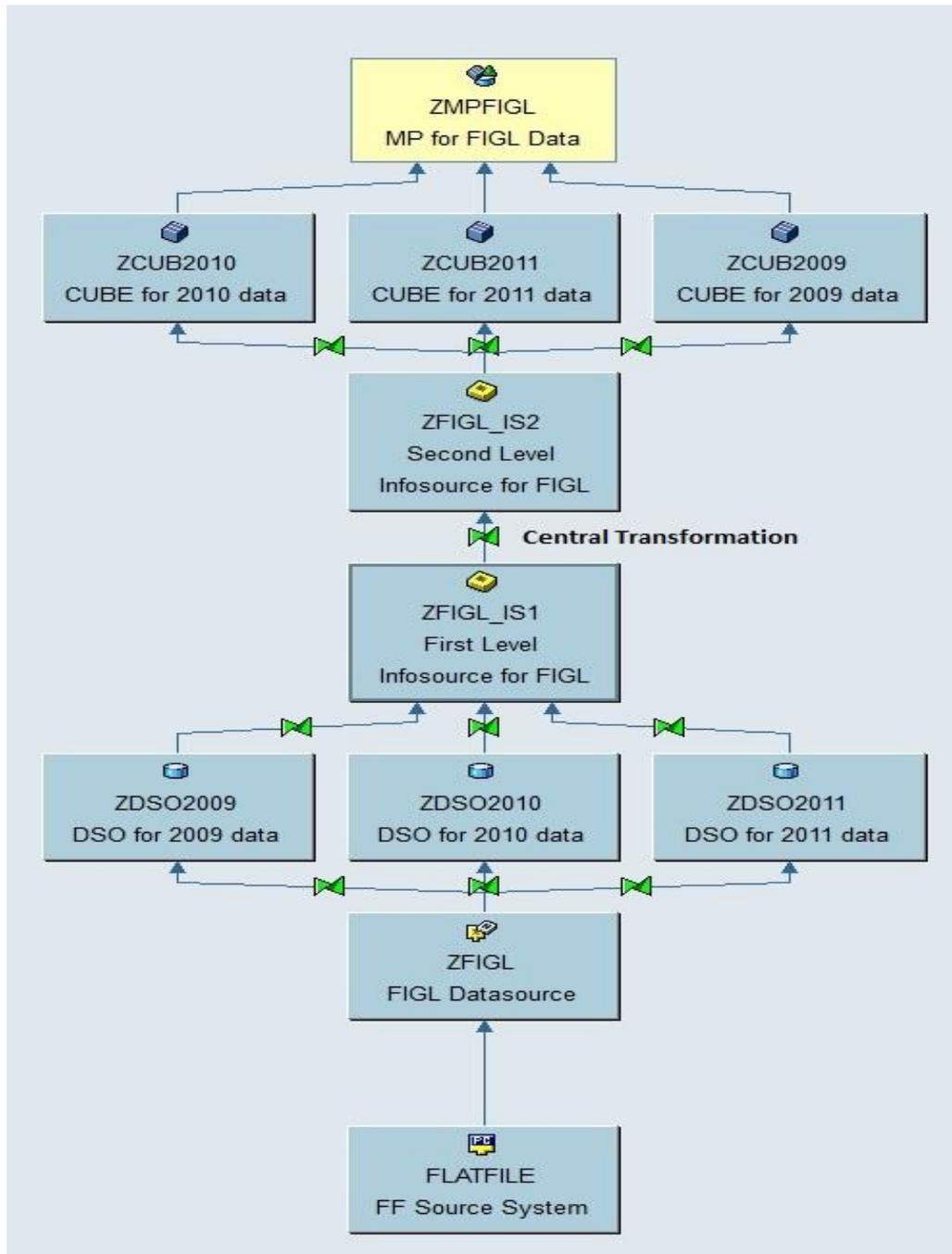
7. Create the Central transformations. This transformation should contain all the rules exactly in the same way as in the transformation from ZFIGLDSO to ZFIGLCUB.
8. Create one DTP from DSO 2009 to CUBE 2009 with the Filter for year 2009. Similar DTPs need to be created for loads from DSOs 2010 and 2011 to cubes 2010 and 2011 respectively.

The advantage of this flow is that, in future if new DSO and Cube are created for the year 2012, it would only require the creation of the objects (DSO and CUBE) and their respective transformations with direct mapping as follows:



Even a change in the processing rules would have to be effected only at one place i.e. the central transformation, instead of doing the same change at 4 different places in a normal scenario.

The following is the screenshot of the sample scenario created:



The Central Transformation:

As can be seen above the central transformation is the place where all the processing rules would be mapped and the routines would be written. While creating the central transformation, an important factor should be considered to avoid any confusion with the data.

The point is:

Like any transformation, for all Key figures the aggregation behavior needs to be set. We all know that in case of loading to an Info cube the aggregation behavior of the Key figures are always set to Summation.

Rule Details	
Description	
Target InfoObjct	0CREDIT_LC Credit amount in local currency
Rule Type	Direct Assignment
Aggregation	Overwrite
Source Fields of Rule:	

Whereas in case of the central transformation between 2 Infosources, the Second level infosource exhibits the behavior of that of a DSO, and hence by default for all the Key figures the Aggregation is set to Overwrite in the central transformation.

Consider our case where we load FIGL data from DSOs to Cubes, in this case, the DSO contains 0AC_DOC_NO (Accounting Document Number) and 0ITEM_NUM (Number of Line item with in Accounting document) and 0COMPANY_CODE as its Key fields.

But since the cube does not require detailed document level data, the accounting document number and the item number are not available in the Cube.

In this case let us assume we have 2 records being sent from the DSO to the Cube:

0AC_DOC_NO	0ITEM_NUM	0COMP_CODE	0CREDIT_LC
12345	1	1122	100
23456	1	1122	50

If in the central transformation, the aggregation type for 0CREDIT_LC is set to overwrite then out of these 2 records only 1 record would go to the cube as a result of overwrite in the central transformation.

0COMP_CODE	0CREDIT_LC
1122	50

To avoid this from happening, the solution is to set the aggregation type as Summation in these types of cases.

Thus it is important to keep this point in mind while designing data flows involving 2 Infosources. This type of data flows are now being used widely due to their scalability and easy maintenance. Hence it is worthwhile to understand the reason and the concept behind the design and build solutions.

Related Content

<http://help.sap.com>

<https://www.sdn.sap.com/irj/sdn/index>

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.