

Analysis Process Designer (APD) Illustrated step-by-step implementation Part 2 (using Routine Transformation).



Applies to:

SAP NetWeaver BW. For more information, visit the [EDW homepage](#).

Summary

The document provides an insight into APD development using Real-time business scenario. This is a beginner's guide to APD which provides a step-by-step detailed APD development process. This document provides details on Routine implementation in APD.

Author: Shyam Uthaman

Company: Accenture Services Pvt. Ltd.

Created on: 14 April 2011

Author Bio



Shyam Uthaman is working as SAP-BI Consultant in Accenture Services Private He is working simultaneously on multiple projects for different clients in Accenture.

Table of Contents

Analysis Process Designer	3
Introduction & Use.....	3
Integration	3
Restrictions	4
Functions.....	4
Versioning	4
Transport Connection.....	4
Illustrated APD Implementation	5
Business Requirement	5
Creating a New Design Screen.....	5
Selecting a Data Source	6
Target DSO Creation	9
Selecting a Data Target	11
Selecting a Transformation	12
Result Verification	23
Related Content	26
Disclaimer and Liability Notice.....	27

Analysis Process Designer

Introduction & Use

In SAP BW, data from various databases from systems available in the company are collected, consolidated, managed and prepared for evaluation purposes. There is often further, valuable potential in this data.

It deals with completely new information that is displayed in the form of meaningful connectivity between data but that is too well hidden or complex to be discovered by simple observation or intuition.

The Analysis Process Designer (APD) makes it possible to find and identify these hidden or complex relationships between data in a simple way. Various data transformations are provided for this purpose, such as statistical and mathematical calculations, and data cleansing or structuring processes.

The analysis results are saved in BW data targets or in a CRM system. They are available for all decision and application processes and thus can be decisive (strategically, tactically, and operatively).

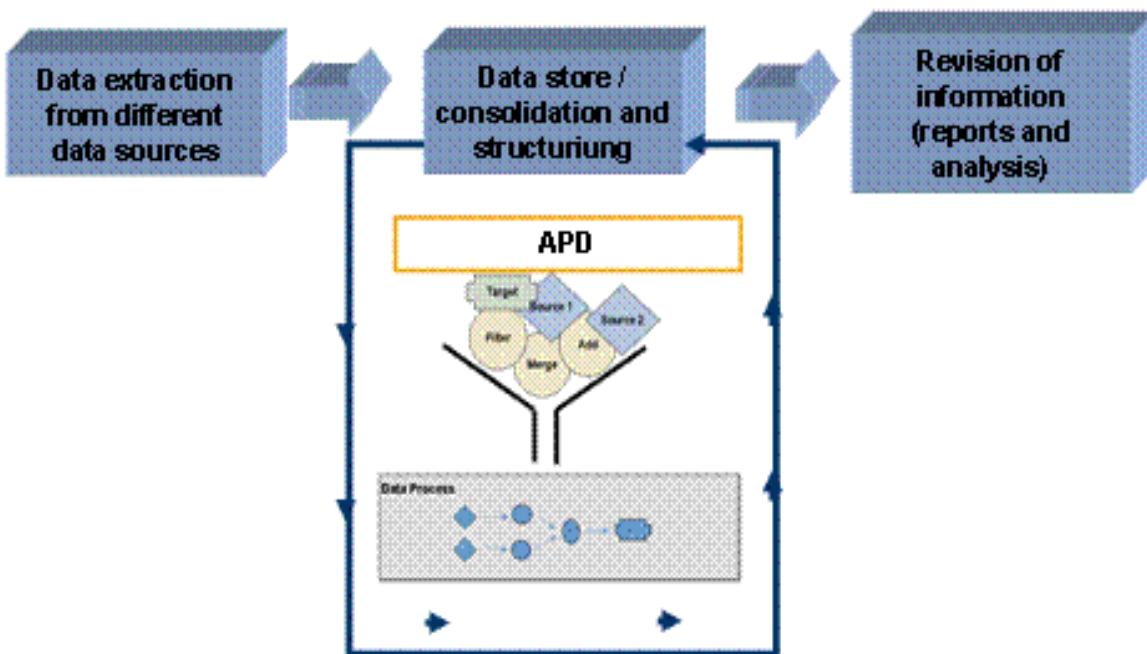
Examples of analysis processes include the calculation of ABC classes, determination of frequency distribution or of scoring information.

Integration

The analysis process designer is the application environment for the SAP data mining solution. The following data mining functions are integrated into the APD:

- Creating and changing data mining models
- Training data mining models with various BW data (data mining model as data target in the analysis process)
- Execution of data mining methods such as prediction with decision tree, with cluster model and integration of data mining models from third parties (data mining model as a transformation in the analysis process)
- Visualization of data mining models

The APD is integrated into the Administrator Workbench:



Restrictions

Integration into the Administrator Workbench has the following restrictions:

- The node texts are not language dependent.
- You can only integrate an analysis process into a process chain using the process type ABAP program. To do this, choose the ABAP report `RSAN_PROCESS_EXECUTE`.
- Analysis processes are not displayed in the data flow display.
- The where-used list only functions from the analysis process to other objects and from data mining models to the analysis process, but not from other objects such as InfoProviders.

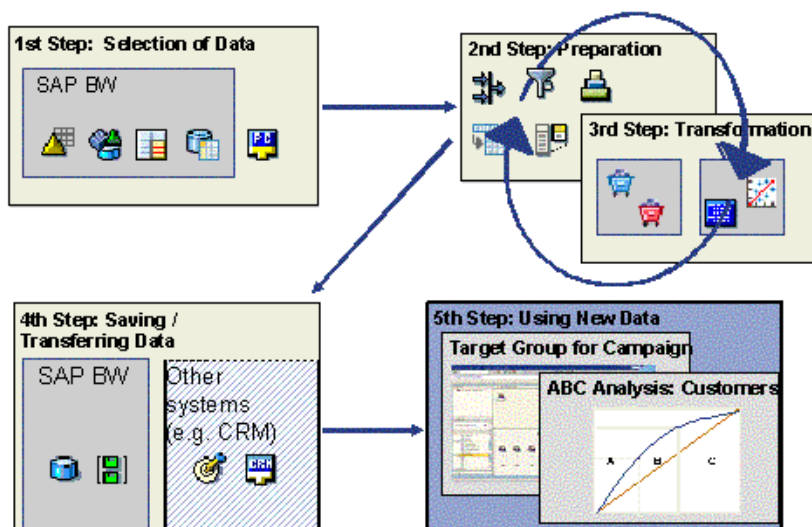
Functions

The analysis process designer is a workbench with an intuitive, graphic user interface for the creation, execution and monitoring of analysis processes. Analysis process can be created using Drag & Drop. Data from different data sources in the BW system can be combined, transformed and prepared for analysis in several individual steps so that it can then be saved into targets in the BW system (transactional ODS object or InfoObjects with attributes) or in a CRM system. Various are available.

- [Data sources](#)
- [Transformations](#)
- [Data Targets](#)

Various additional functions support you during modeling and executing an analysis process, as well as during interpretation of the analysis results.

The following graphic shows the various steps in the Analysis Process Designer.



First select a data target that contains the desired data. Then this data is prepared and then transformed. This transformed data is then saved in a BW object or in another system. For analysis, you can display the data in a query in the Business Explorer.

Versioning

Analysis processes are integrated into the versioning concept (active, inactive version, content version and delivery).

Transport Connection

Analysis processes are connected to the transport system for the SAP BW as TLOGO objects

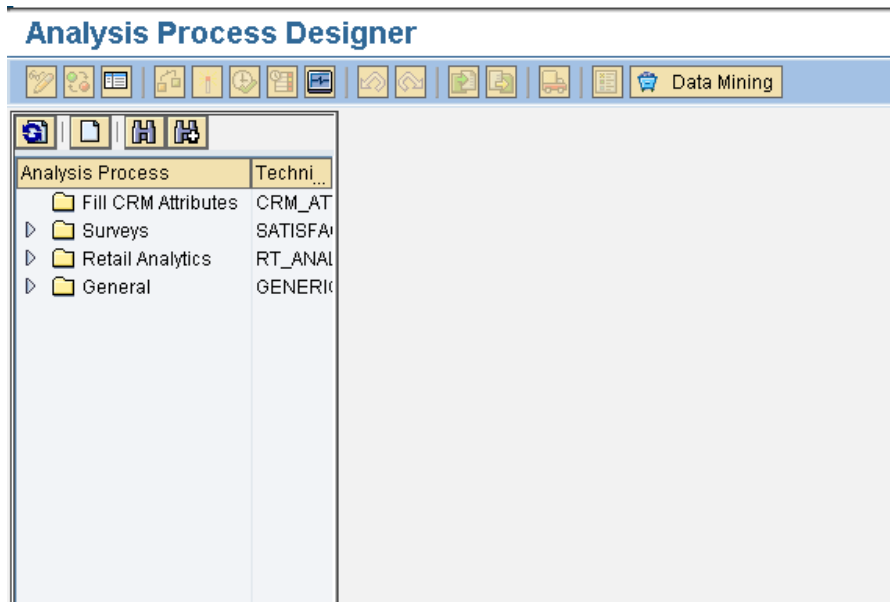
Illustrated APD Implementation

Business Requirement

To create an APD to load from query to DSO but the date field of the DSO should always reflect the last date of the current month.

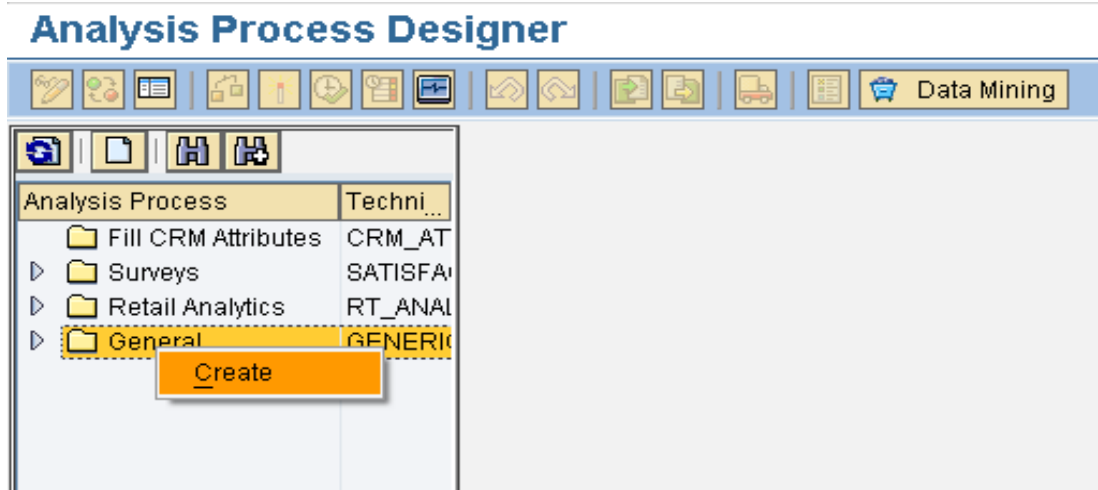
Creating a New Design Screen

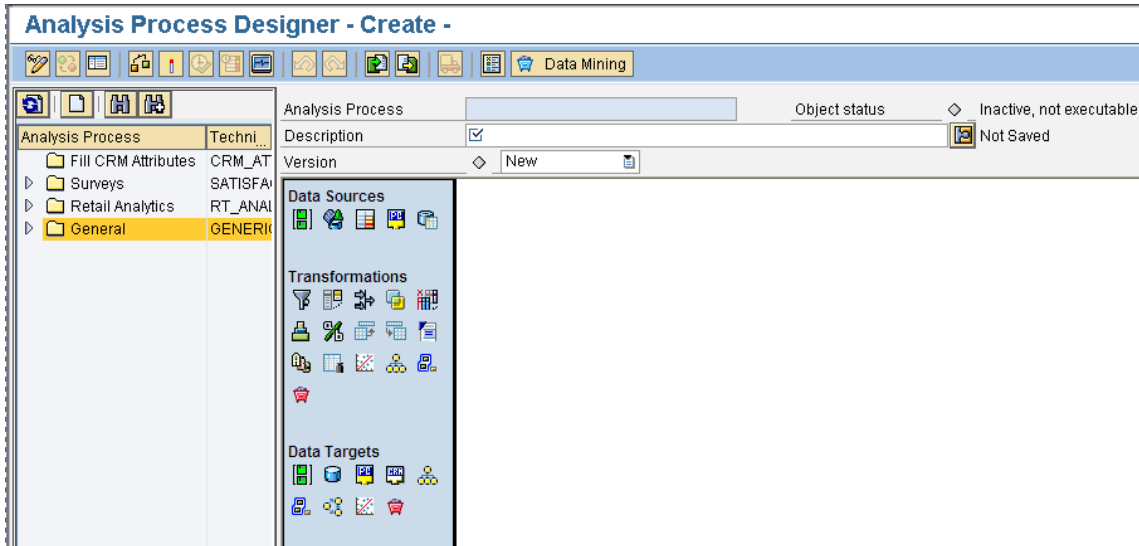
To implement an APD, first use the RSANWB transaction to reach the following screen.



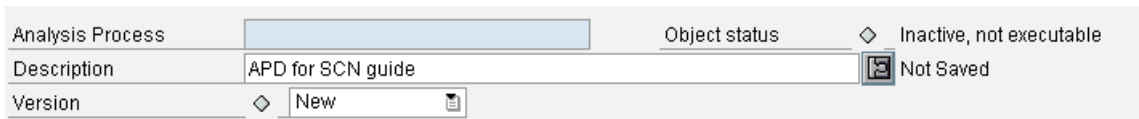
The left panel of the screen displays all the components that can be used to build an analysis process, and the right panel displays the design area where the analysis process is built. Here we will use the 'General' Analysis Process. Right Click on 'General' and select 'Create' as shown below.

You will get the following screen





Give the APD a meaningful description.

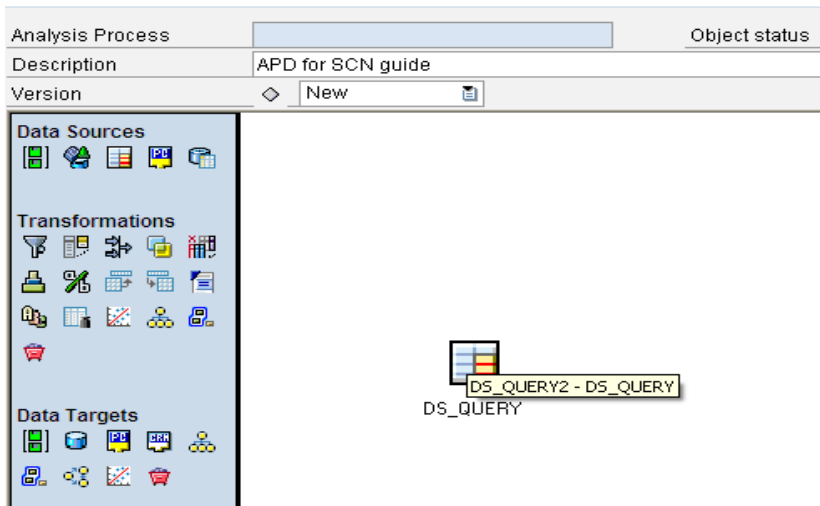


Selecting a Data Source

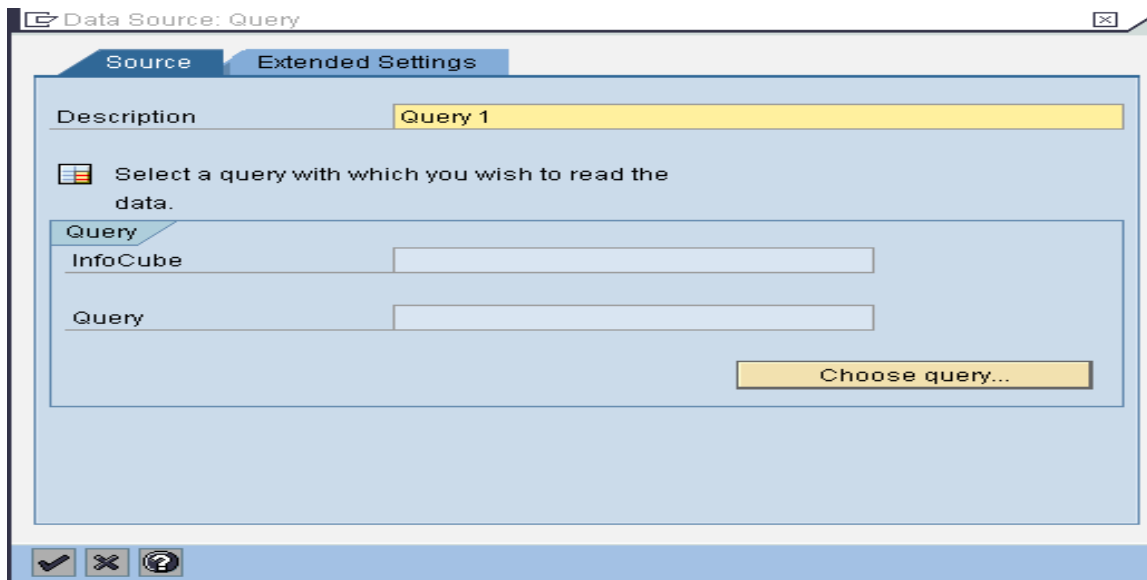
The components from which data is read for further processing in an analysis process are categorized as data sources. An analysis process can use master data, BEx queries, InfoProviders, flat Files, or database tables as a data source. (As shown below).



Our requirement states that the data source will be a query

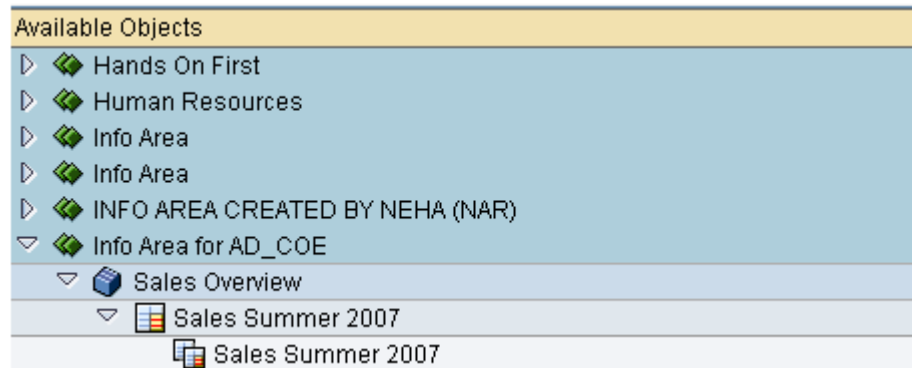


The following window pops up as soon as you drop the query in the pane. Click on choose query to select the query you wish to use as a source.



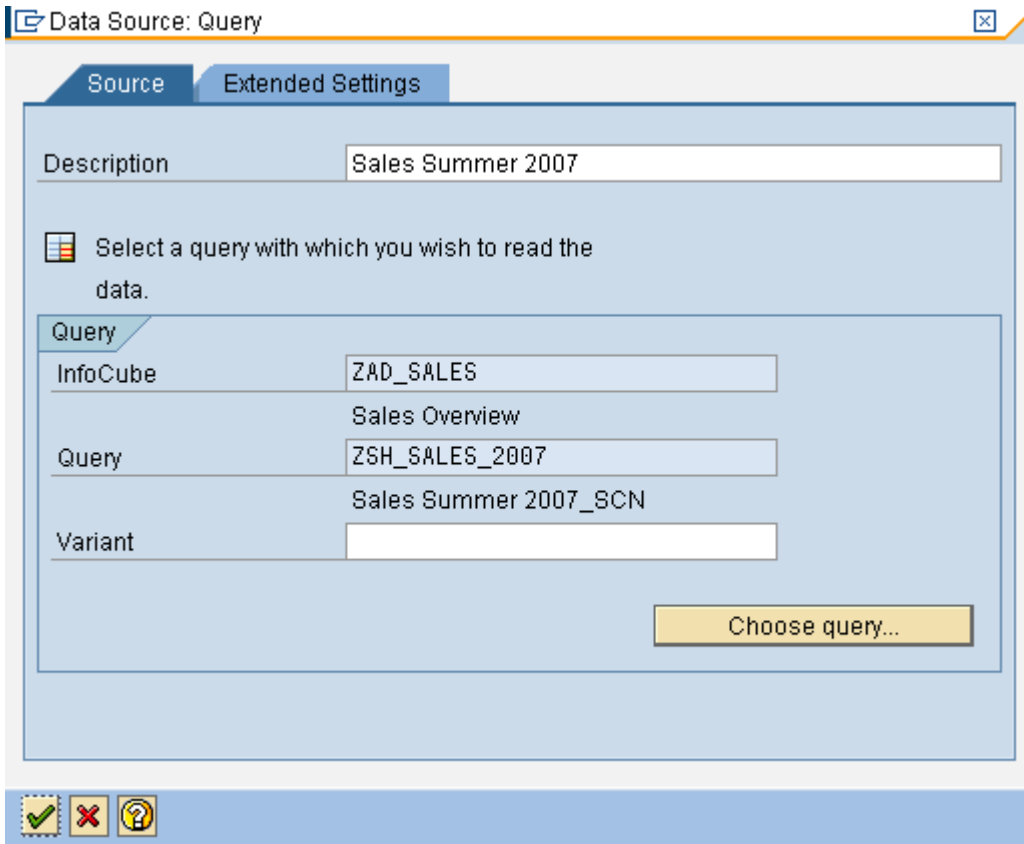
You will get the following window with the list of all info areas in the system.

Find your query inside your InfoArea in which your InfoProvider (on which the query was built) exists.



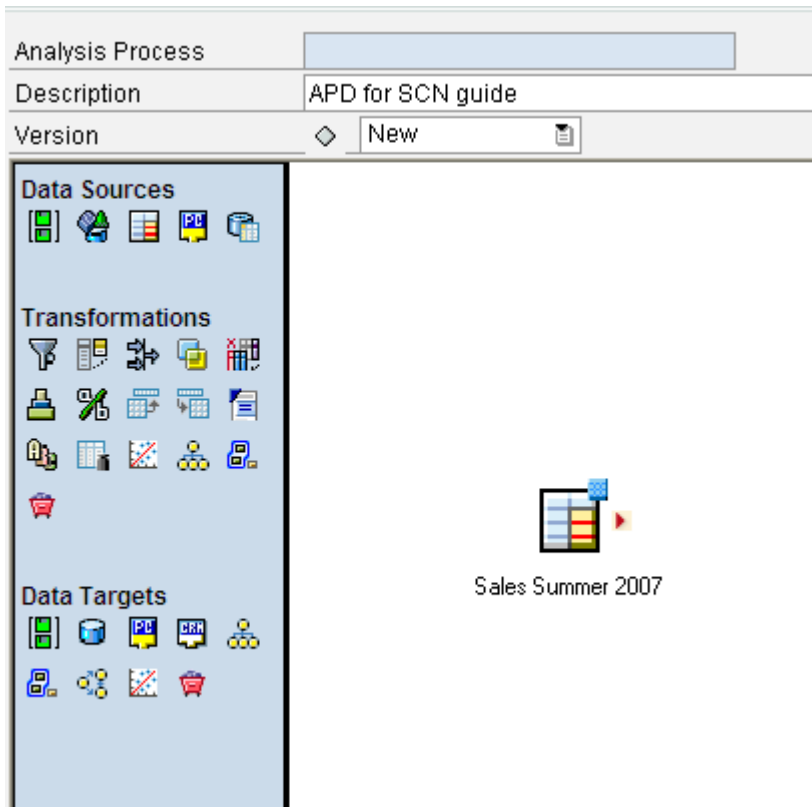
Double click on the query name.

You will be redirected back to the previous window which now shows the details of your query. Change the Description if you require, but we will keep the same description for simplicity.

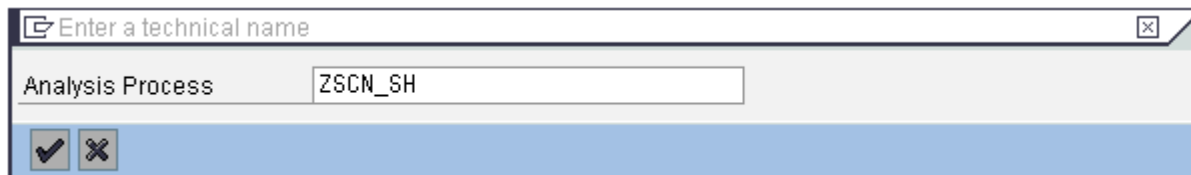


Click on Ok.

The Data source has been configured in the APD as you can see in the image below.



Now we need a Target DSO to store the data. We will be creating the required Direct-Update DSO. We will save the APD before going to RSA1 transaction again. Give the APD a technical name and press Ok.

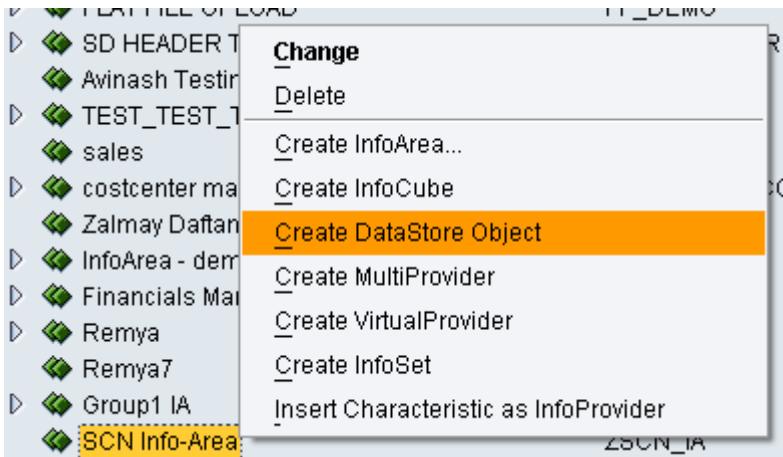


It will be saved.

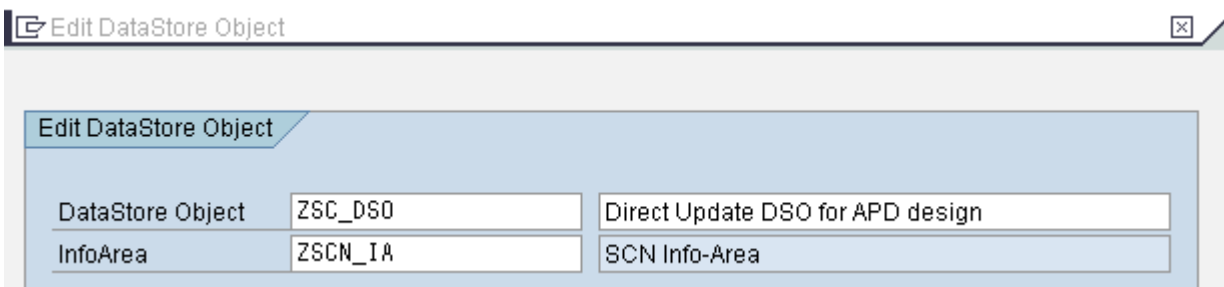
Target DSO Creation

The data target to be used here is a Direct-Update DSO (as it is the only SO type that supports APD). We will be creating one using the following steps:

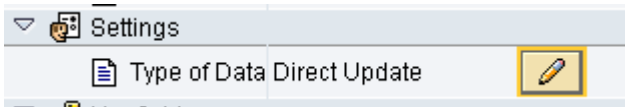
Right click on your InfoArea. Select Create DataStore Object from the context menu.



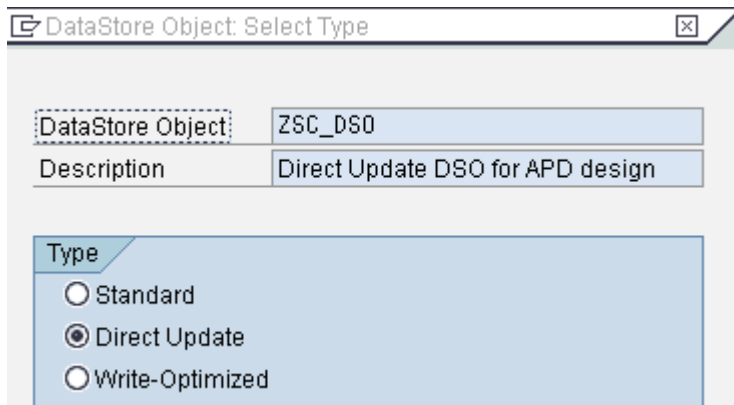
You will get the following window. Enter the technical name and the description of the DSO.



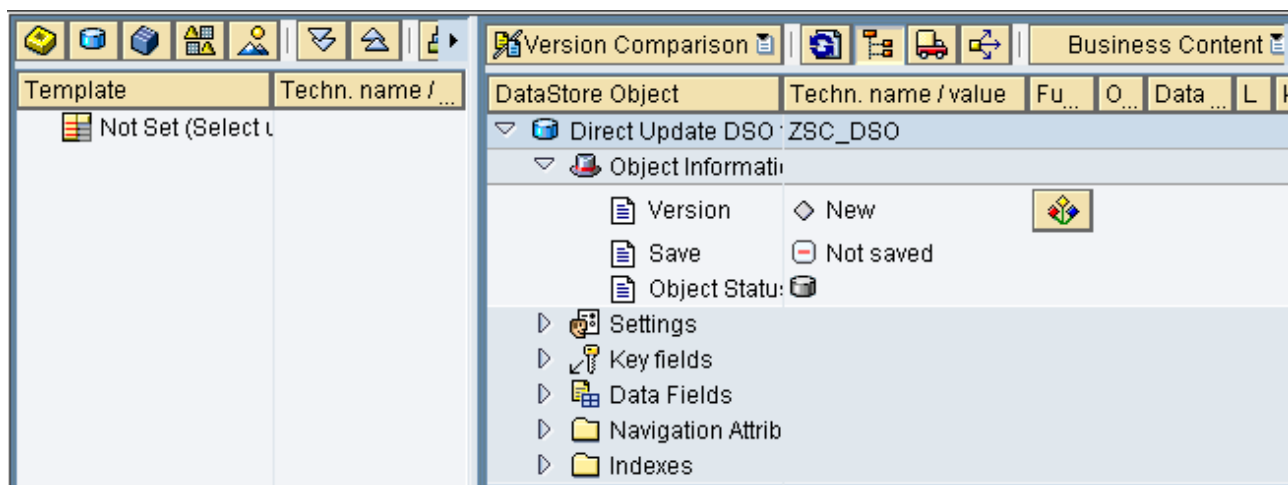
Now, on the DSO edit screen, edit the 'Type of Direct Update' in the Settings.



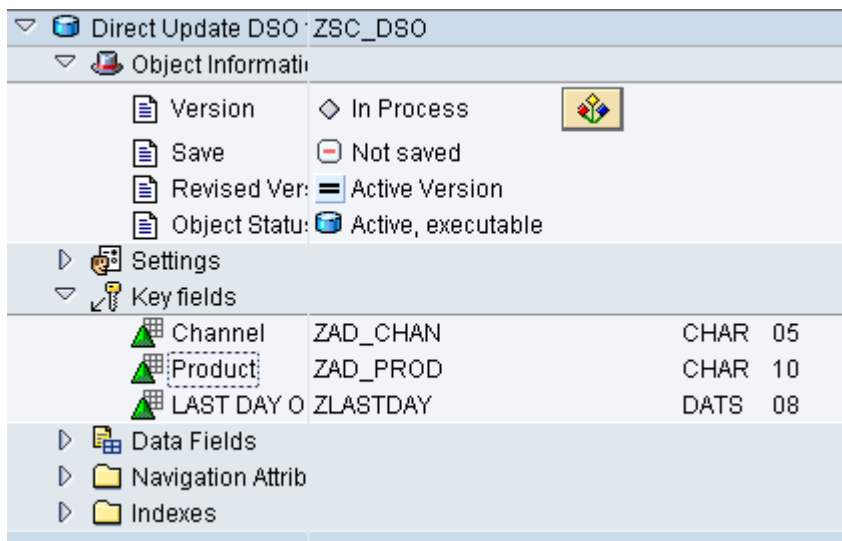
On the popup, Select Direct Update Radio Button and press OK.



You will see the following screen:



Now I have added the some key fields and Data fields to the DSO and then saved and activated it as seen below. Note that Key field ZLASTDAY is the field on which our Routine will be applied.



Selecting a Data Target

The data processed and generated in an analysis process can be saved into different targets, such as direct update DSOs, master data attributes, flat files, and so on (as shown below). The data generated in an analysis process can also be fed to data mining models.



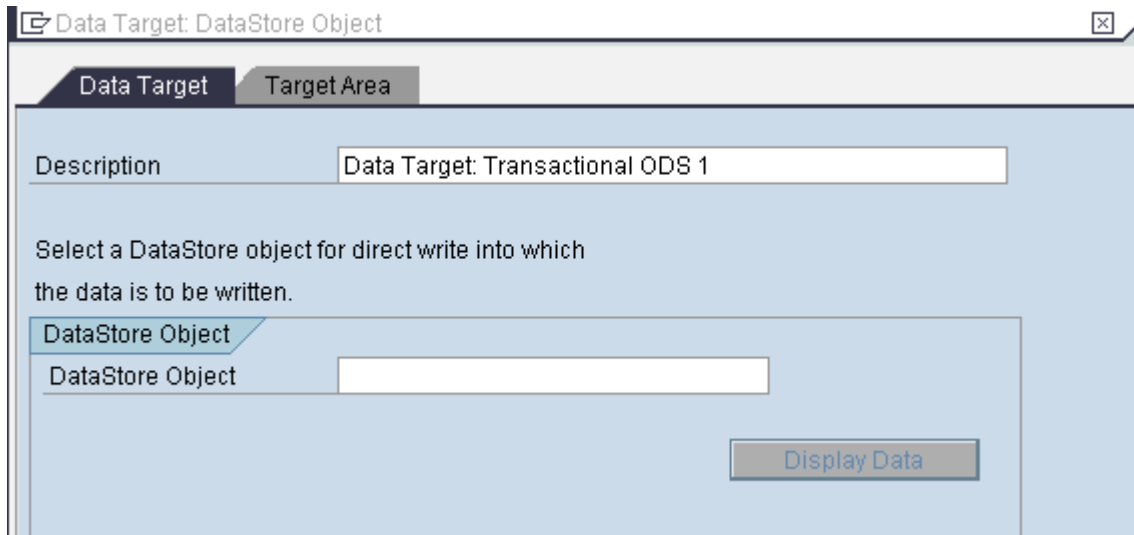
Here, we will be using the Direct-Update DSO that we created earlier to act as a Data Target.

To do this, Drag and drop the DSO icon to the Design pane.

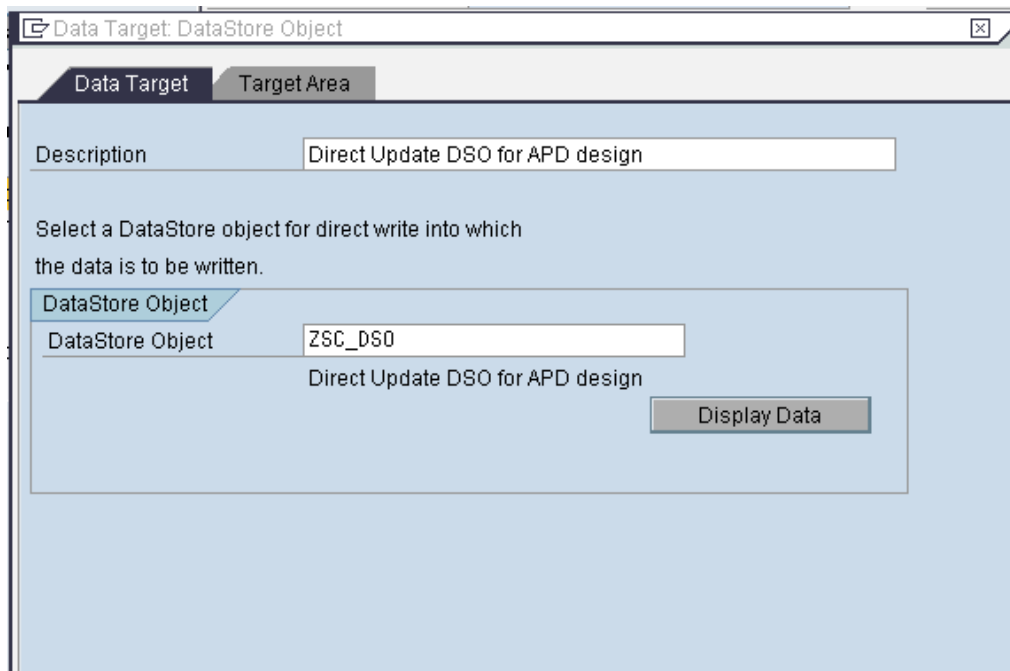


As soon as you drop the DSO on to the pane, you will get the following pop-up window.

Here we have to enter your target DSO name in the text field and also specify a description for it.



The below image shows the fully filled details of the window described above.

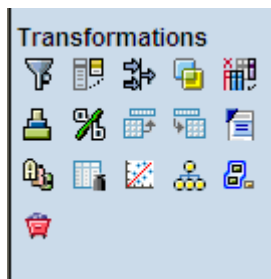


Press OK.

You will be redirected back to the design pane.

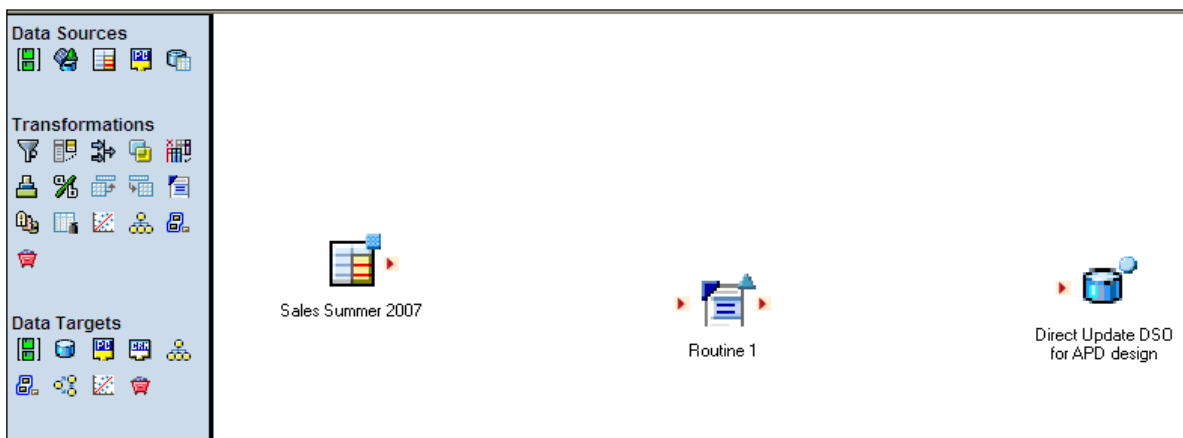
Selecting a Transformation

Different analysis functions used on the data are grouped under Transformations. These different transformations range from simple filter functions to more complex data mining functions (as shown below).



In our example, we will be using a Routine in between the source and the output.

To do this, drag and drop the Routine Icon into the design pane as shown below.



Now, to apply the routine, we need to supply it with the query output.
Drag and drop the connection from the query output to the routine input as shown below.
Blue dotted lines indicate that the connection is still incomplete.

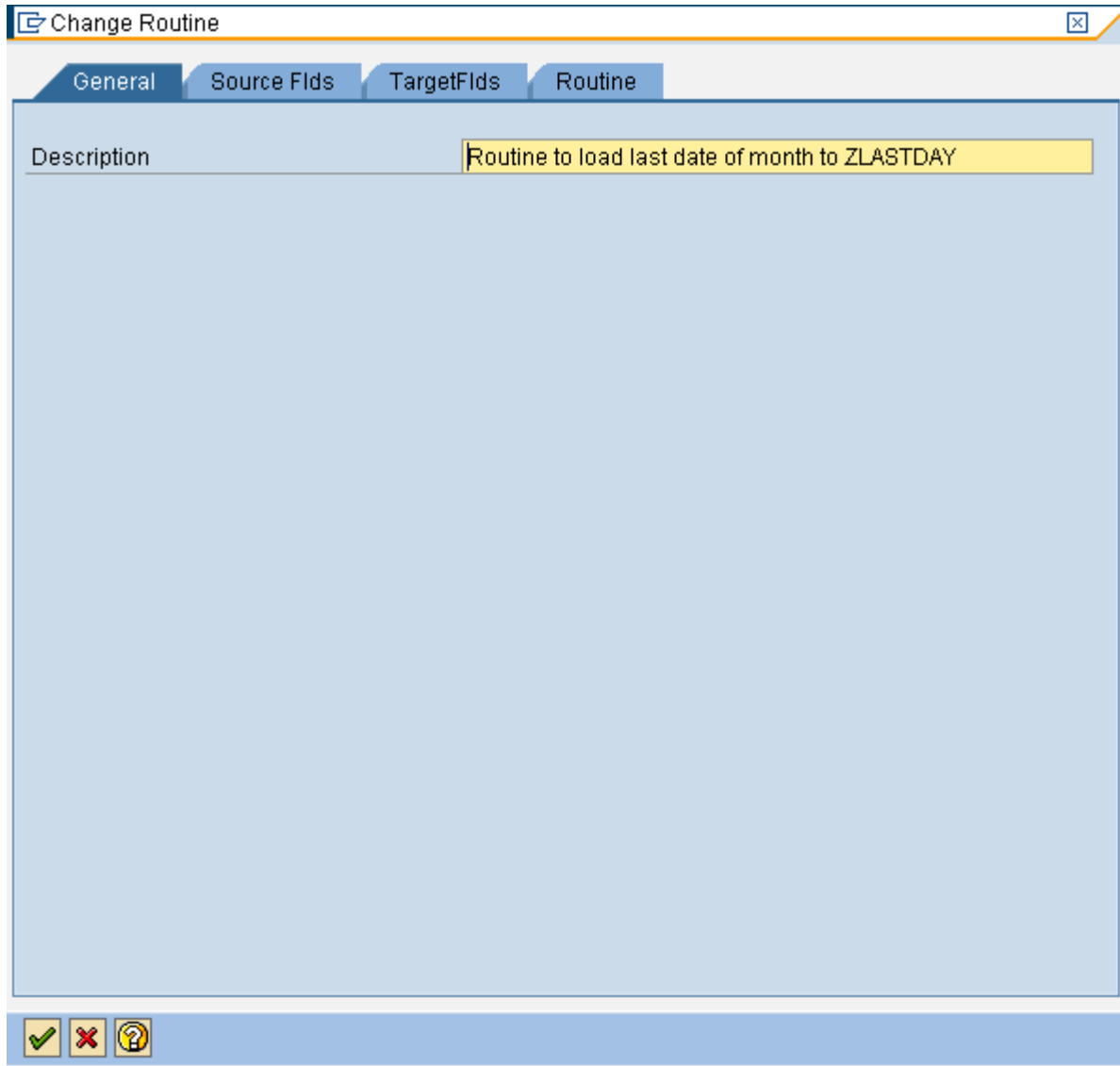


The solid black line (as shown below) indicates that the connection is complete.



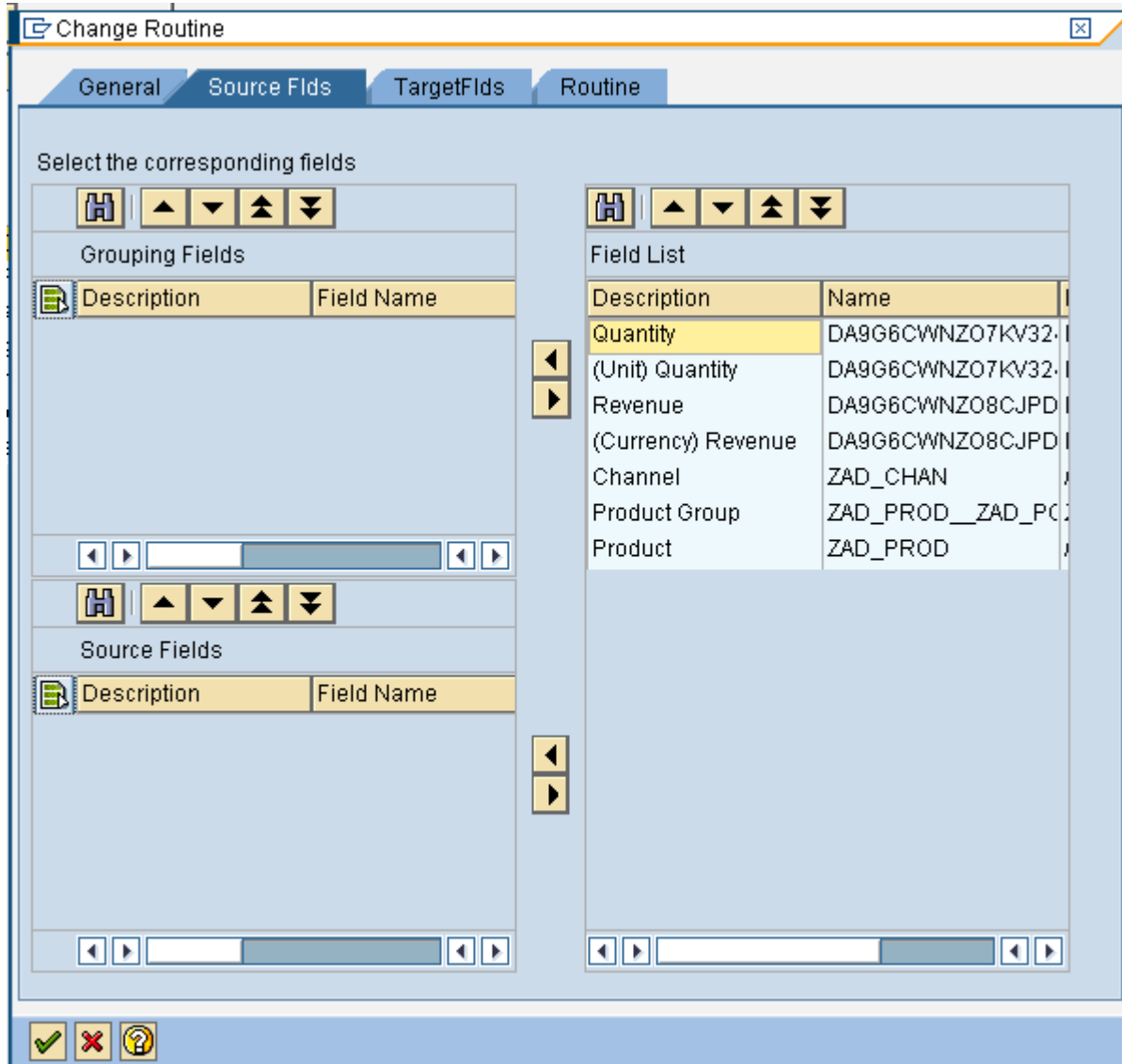
Now, double click on the Routine .

The following window opens.



Give in a meaningful description to the routine as shown above.

Now go on to the 'Source Fields' Tab

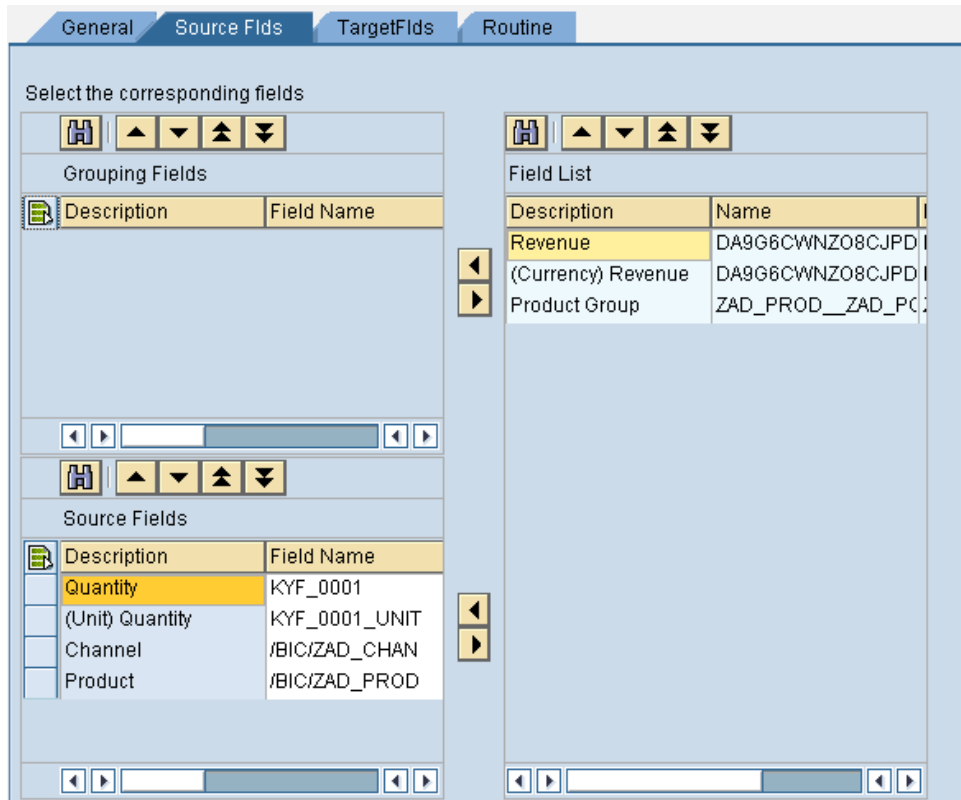


Now we need to select in the Fields which we require to transfer from query to DSO.

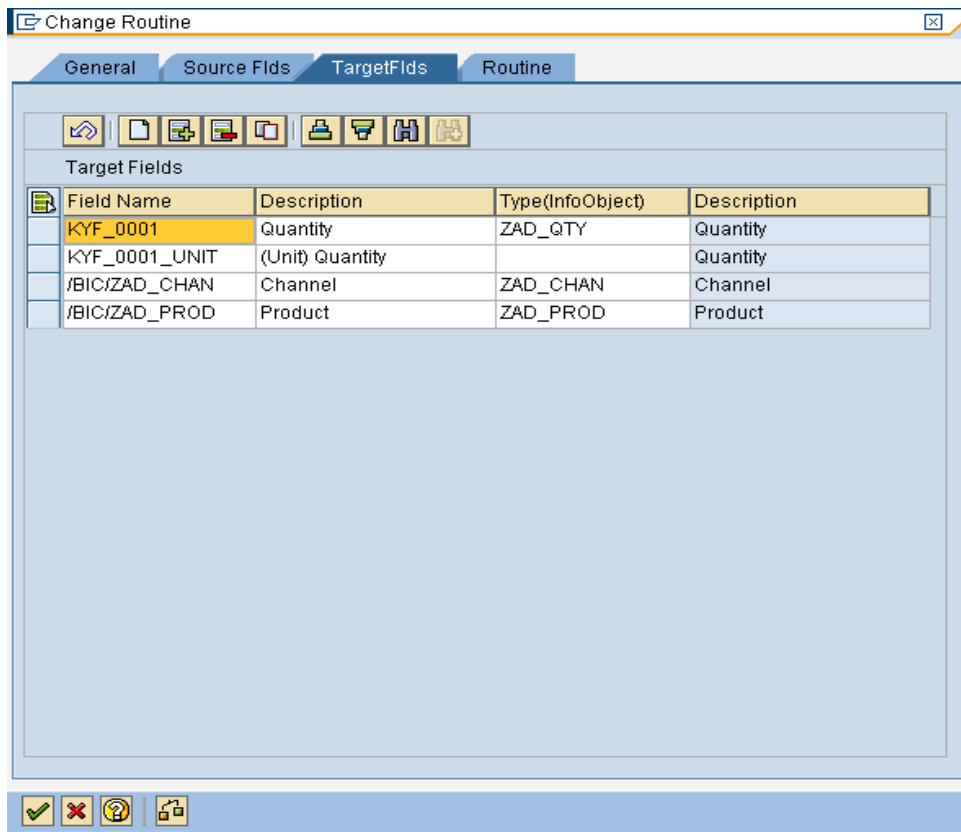
This will be done by moving the required fields from the 'Field List' pane to the 'Source Flds' pane using the

 button.

As we can see below, the fields required have been successfully transferred.



Now move on to the 'Target Flds' Tab



We can see here that the Quantity unit has no 'Type' .

We will give it as 0UNIT and Press enter.

The description will come automatically.

The result is shown below.

Field Name	Description	Type(InfoObject)	Description
KYF_0001	Quantity	ZAD_QTY	Quantity
KYF_0001_UNIT	(Unit) Quantity	0UNIT	Unit of Measure
/BIC/ZAD_CHAN	Channel	ZAD_CHAN	Channel
/BIC/ZAD_PROD	Product	ZAD_PROD	Product

Now, we will add a new field to hold the date of the last day of the month.

We will give it the type info-object ZLASTDAY that was used in the DSO also.

To do that, click on the Create button (Circled in Red) below.

Field Name	Description	Type(InfoObject)	Description
KYF_0001	Quantity	ZAD_QTY	Quantity
KYF_0001_UNIT	(Unit) Quantity	0UNIT	Unit of Measure
/BIC/ZAD_CHAN	Channel	ZAD_CHAN	Channel
/BIC/ZAD_PROD	Product	ZAD_PROD	Product

The screenshot below shows the new row after entering the new details.

Field Name	Description	Type(InfoObject)	Description
KYF_0001	Quantity	ZAD_QTY	Quantity
KYF_0001_UNIT	(Unit) Quantity	0UNIT	Unit of Measure
/BIC/ZAD_CHAN	Channel	ZAD_CHAN	Channel
/BIC/ZAD_PROD	Product	ZAD_PROD	Product
ZLASTDAY	Last Day of the month	ZLASTDAY	LAST DAY OF THE MONTH

Now move to the 'Routine' tab.

You will see that the source and target fields are already declared here.

Now we need to write the code to fill in ZLASTDAY with Last Date of the month.

```

1  REPORT RSAN_WB_ROUTINE_TEMP_REPORT .
2
3  TYPES: BEGIN OF y_source_fields ,
4         KYF_0001 TYPE /BIC/OIZAD_QTY ,
5         KYF_0001_UNIT TYPE MEINS ,
6         /BIC/ZAD_CHAN TYPE /BIC/OIZAD_CHAN ,
7         /BIC/ZAD_PROD TYPE /BIC/OIZAD_PROD ,
8     END OF y_source_fields .
9  TYPES: yt_source_fields TYPE STANDARD TABLE OF y_source_fie
10
11 TYPES: BEGIN OF y_target_fields ,
12         KYF_0001 TYPE /BIC/OIZAD_QTY ,
13         KYF_0001_UNIT TYPE /BIC/OIUNIT ,
14         /BIC/ZAD_CHAN TYPE /BIC/OIZAD_CHAN ,
15         /BIC/ZAD_PROD TYPE /BIC/OIZAD_PROD ,
16         ZLASTDAY TYPE /BIC/OIZLASTDAY ,
17     END OF y_target_fields .
18 TYPES: yt_target_fields TYPE STANDARD TABLE OF y_target_fie
19 *----- Begin of type definitions -----
20
21 *TYPES: ...
22
23 *----- End of type definitions -----
24 FORM compute_data_transformation

```

To fill in ZLASTDAY with Last Date of the month, we will call a standard Function Module:

SLS_MISC_GET_LAST_DAY_OF_MONTH within the program.

The complete code is as given below:

Note: The highlighted part shows the code introduced by me. Rest of the code is auto-generated as we provide the Source and Target Field Declarations in their Respective Tabs

```

REPORT RSAN_WB_ROUTINE_TEMP_REPORT .

TYPES: BEGIN OF y_source_fields ,
        KYF_0001 TYPE /BIC/OIZAD_QTY ,
        KYF_0001_UNIT TYPE MEINS ,
        /BIC/ZAD_CHAN TYPE /BIC/OIZAD_CHAN ,
        /BIC/ZAD_PROD TYPE /BIC/OIZAD_PROD ,
    END OF y_source_fields .
TYPES: yt_source_fields TYPE STANDARD TABLE OF y_source_fields .

TYPES: BEGIN OF y_target_fields ,
        KYF_0001 TYPE /BIC/OIZAD_QTY ,

```

```

        KYF_0001_UNIT TYPE /BI0/OIUNIT ,
        /BIC/ZAD_CHAN TYPE /BIC/OIZAD_CHAN ,
        /BIC/ZAD_PROD TYPE /BIC/OIZAD_PROD ,
        ZLASTDAY TYPE /BIC/OIZLASTDAY ,
    END OF y_target_fields .
TYPES: yt_target_fields TYPE STANDARD TABLE OF y_target_fields .
*----- Begin of type definitions -----
*TYPES: ...
*----- End of type definitions -----
FORM compute_data_transformation
    USING      it_source TYPE yt_source_fields
              ir_context TYPE REF TO if_rsan_rt_routine_context
    EXPORTING et_target TYPE yt_target_fields .
*----- Begin of transformation code -----

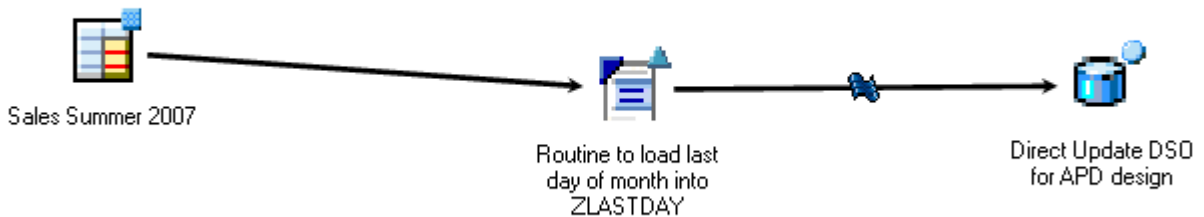
    DATA: ls_source TYPE y_source_fields,
          ls_target TYPE y_target_fields.

    LOOP AT it_source INTO ls_source.
        MOVE-CORRESPONDING ls_source TO ls_target.
        *****Filling Last Day of the month*****
        DATA: GLV_Result type sy-datum.
        CALL FUNCTION 'SLS_MISC_GET_LAST_DAY_OF_MONTH'
            EXPORTING
                day_in          = sy-datum
            IMPORTING
                last_day_of_month = glv_result.
        ls_target-zlastday = glv_result.
        *****
        APPEND ls_target TO et_target.
    ENDLLOOP.

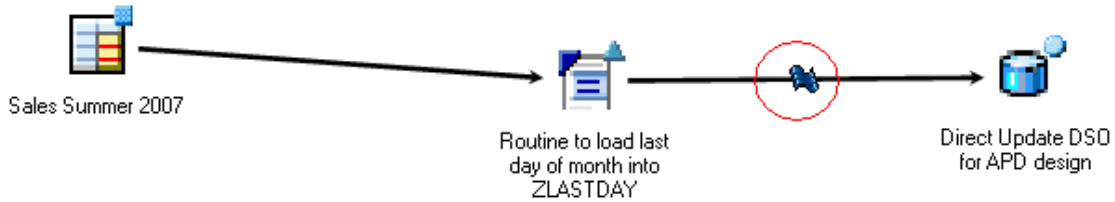
*----- End of transformation code -----
ENDFORM.

```

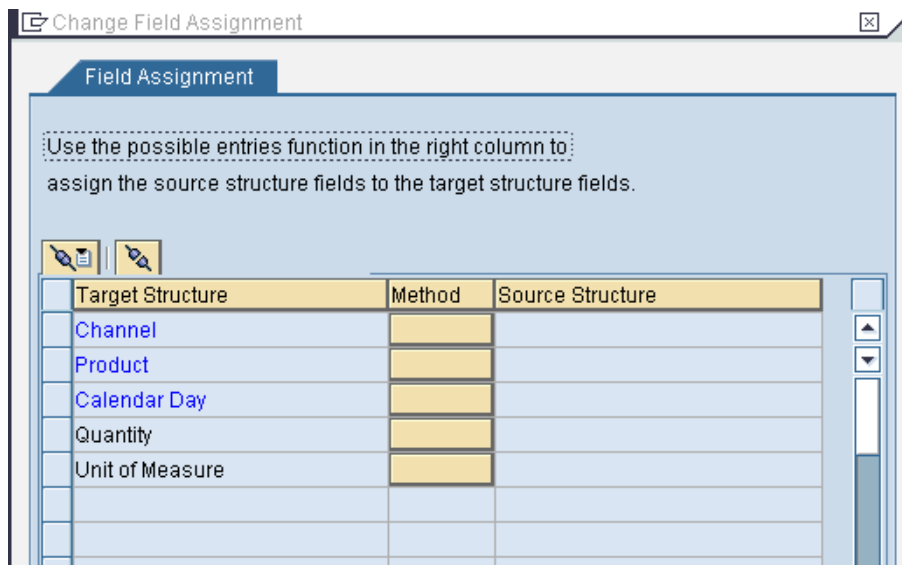
Now create the final connection between the Routine and the DSO as we did earlier.



Now, we have to specify the Fields Assignment between the routine and the DSO. Double click on the mapping symbol circled in red.



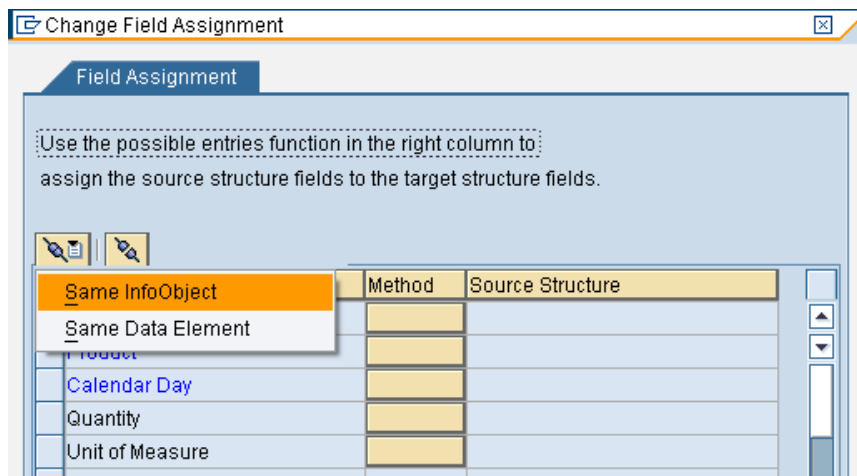
The following window opens up



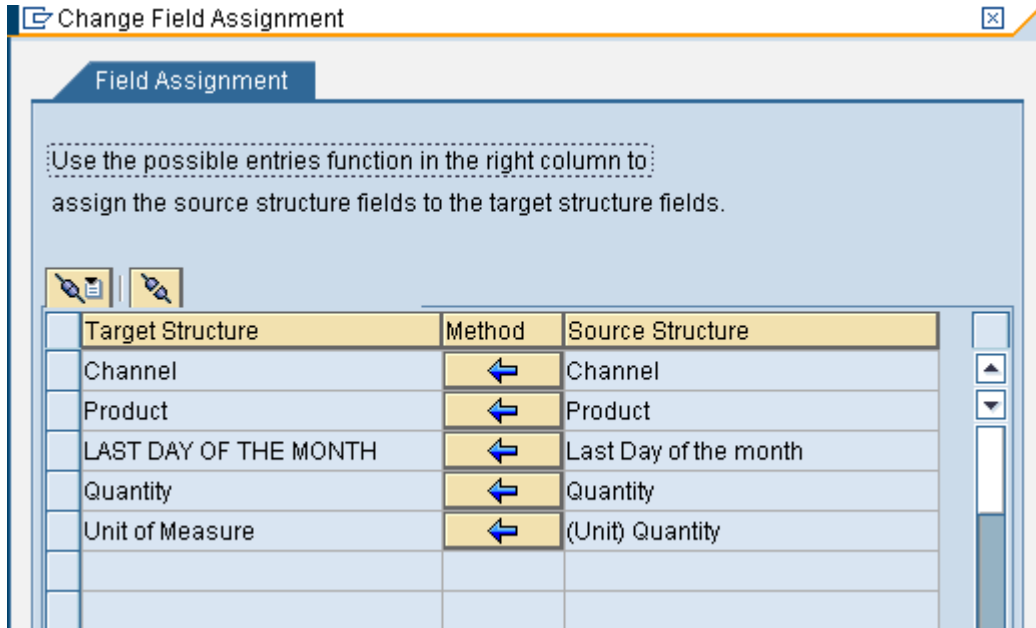
We can automatically map fields which have been created on the same info-object.

Click on the automatic assignment button shown below and select 'Same InfoObject' as shown below.

This reduces a great deal of effort.



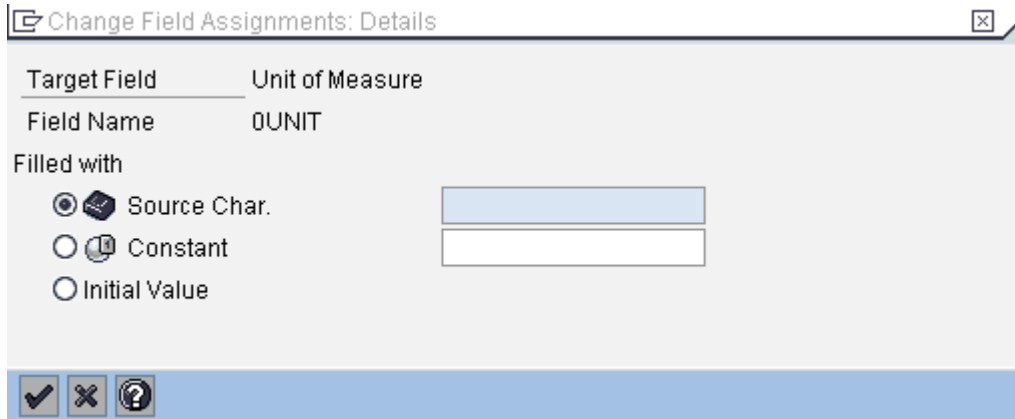
You can see that the automatic mapping has been successful for all the fields.

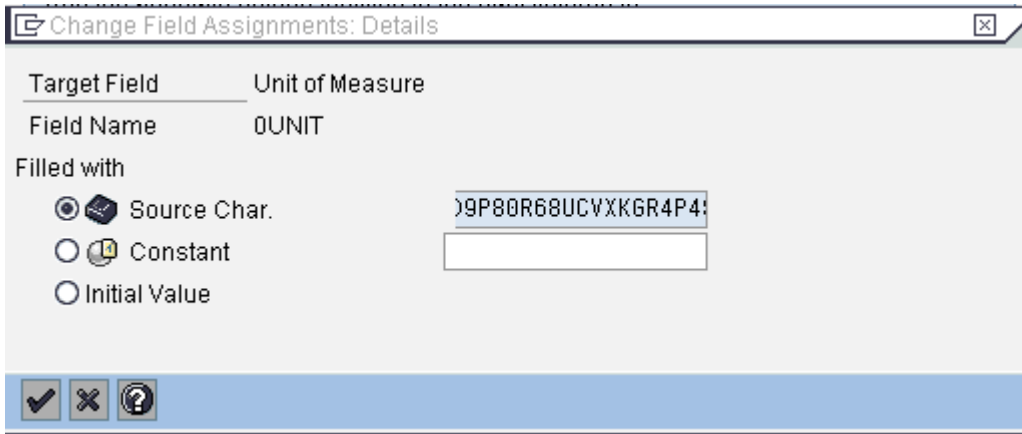


In case your info-objects are different in Source and Target, you can do a custom mapping by clicking the method button.

Here we do not require custom mapping but just to satisfy the curious readers, I will show the Custom Field mapping window which you get after clicking the method Button.

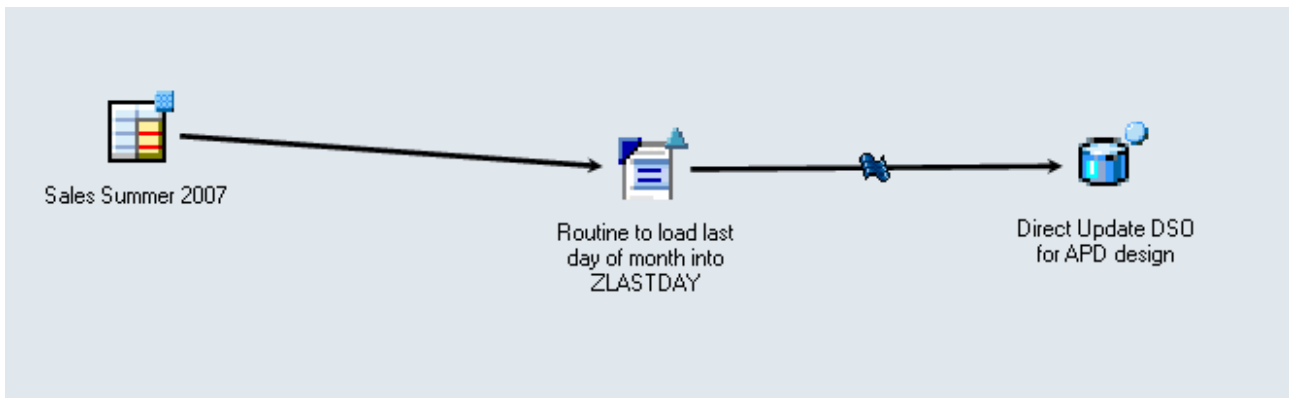
You can map a Source Characteristic manually or assign a constant value to it or even provide it with initial values as shown below.





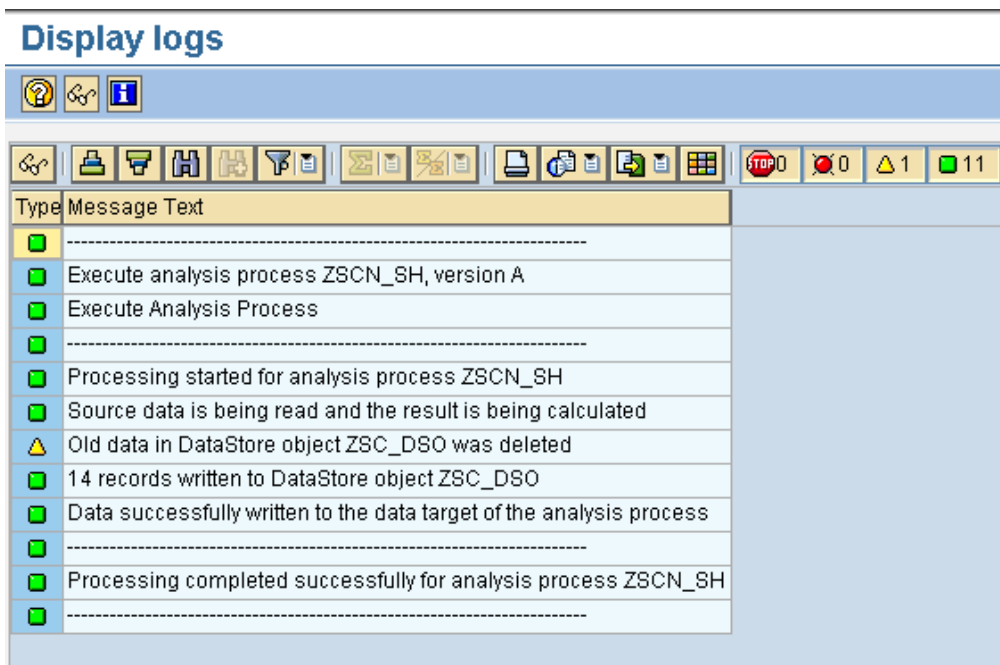
Press OK and come back to the main screen.

Save and activate the APD.



Now press the execute button  to start the data transfer.

Here, it is seen that 14 records have been successfully loaded to the APD.

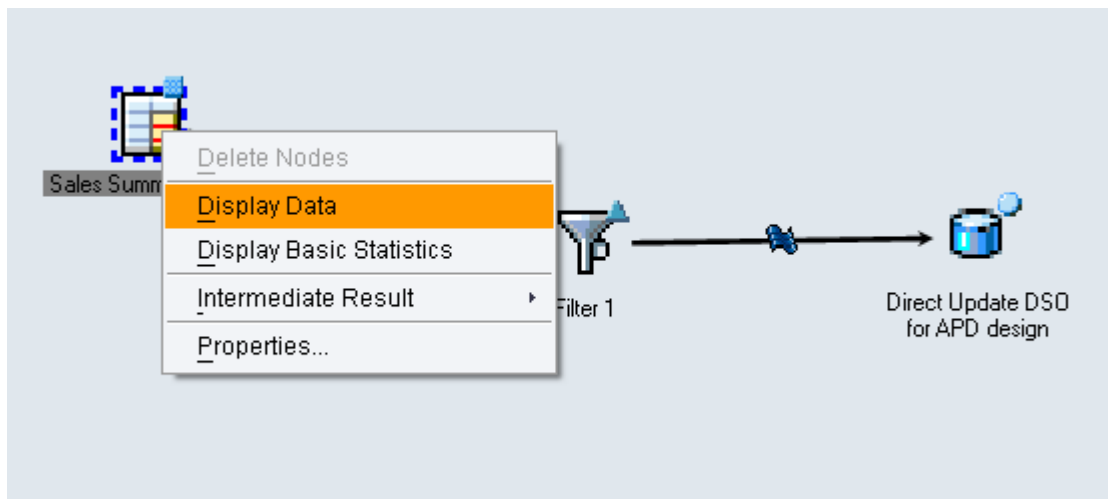


This Display log will show any warnings or errors that occur in APD execution.

Result Verification

Now we will check what the output from the query was and what got loaded into the DSO.

To see the query data, right click on the query and select 'Display Data' from the context menu.



The following output will be shown.

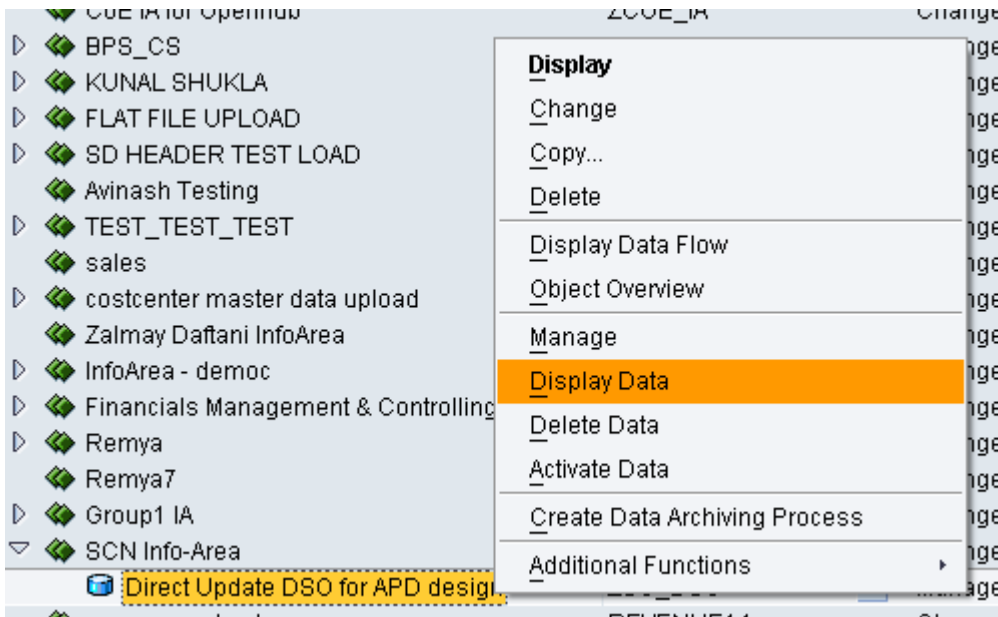
Test Result for Node - Display Data as Table 1

Log On/Off

Quantity	(Unit) Quant..	Revenue (Currency) ...	Channel	Product Gro...	Product
600,000	EA	329.994,00 USD	00001	DS10	PDS01
540,000	EA	351.534,60 USD	00001	DS10	PDS02
630,000	EA	378.623,70 USD	00001	DS10	PDS03
2.550,000	EA	22.644,00 USD	00001	DS20	PDS04
7.560,000	EA	1.776.524,40 USD	00001	DS20	PDS05
7.320,000	EA	292.653,60 USD	00001	DS20	PDS06
5.400,000	EA	86.346,00 USD	00001	DS20	PDS07
7.200,000	EA	165.528,00 USD	00001	DS30	PDS08
90,000	EA	58.589,10 USD	00002	DS10	PDS02
720,000	EA	468.712,80 USD	00003	DS10	PDS02
600,000	EA	360.594,00 USD	00003	DS10	PDS03
4.200,000	EA	37.296,00 USD	00003	DS20	PDS04
900,000	EA	35.982,00 USD	00003	DS20	PDS06
2.550,000	EA	40.774,50 USD	00003	DS20	PDS07

Now we will check the data that has been loaded to the DSO.

Right click on the DSO name and select Display Data.



Give the Field Selection for output and press Execute.

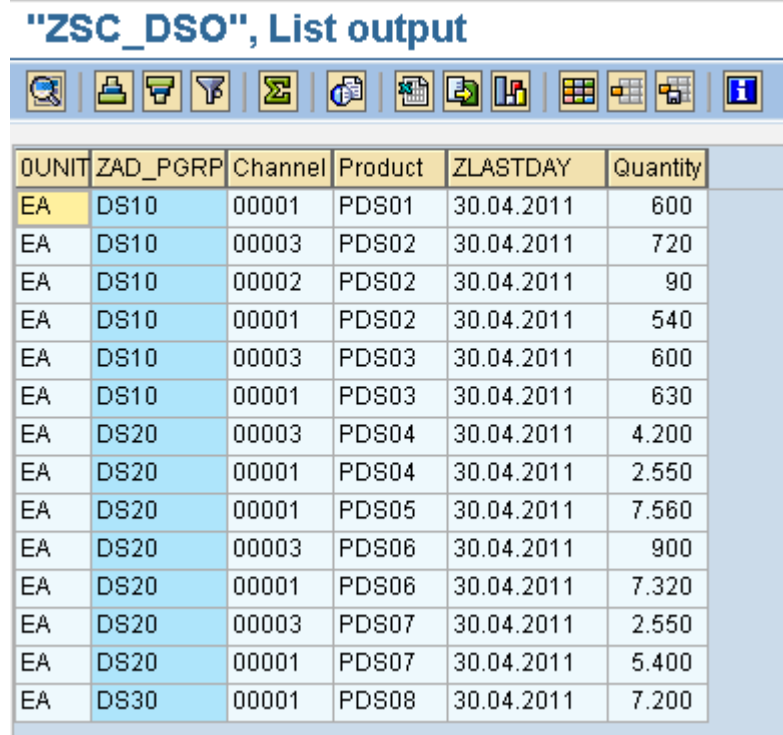
'ZSC_DSO', Select. Screen

Data part			
Update Mode	<input type="text"/>	to	<input type="text"/>
Update Mode(SID)	<input type="text"/>	to	<input type="text"/>
Unit of Measure	<input type="text"/>	to	<input type="text"/>
Unit of Measure(SID)	<input type="text"/>	to	<input type="text"/>
Product Group[Produc	<input type="text"/>	to	<input type="text"/>
Product Group[Produc(SID)	<input type="text"/>	to	<input type="text"/>

Key Part			
Channel	<input type="text"/>	to	<input type="text"/>
Channel(SID)	<input type="text"/>	to	<input type="text"/>
Product	<input type="text"/>	to	<input type="text"/>
Product(SID)	<input type="text"/>	to	<input type="text"/>
LAST DAY OF THE MONT	<input type="text"/>	to	<input type="text"/>
LAST DAY OF THE MONT(SID)	<input type="text"/>	to	<input type="text"/>

We get the following output.

"ZSC_DSO", List output



OUNIT	ZAD_PGRP	Channel	Product	ZLASTDAY	Quantity
EA	DS10	00001	PDS01	30.04.2011	600
EA	DS10	00003	PDS02	30.04.2011	720
EA	DS10	00002	PDS02	30.04.2011	90
EA	DS10	00001	PDS02	30.04.2011	540
EA	DS10	00003	PDS03	30.04.2011	600
EA	DS10	00001	PDS03	30.04.2011	630
EA	DS20	00003	PDS04	30.04.2011	4.200
EA	DS20	00001	PDS04	30.04.2011	2.550
EA	DS20	00001	PDS05	30.04.2011	7.560
EA	DS20	00003	PDS06	30.04.2011	900
EA	DS20	00001	PDS06	30.04.2011	7.320
EA	DS20	00003	PDS07	30.04.2011	2.550
EA	DS20	00001	PDS07	30.04.2011	5.400
EA	DS30	00001	PDS08	30.04.2011	7.200

Here it is seen that only the required fields have been loaded and last Day of the current month (April) – 30.04.2011 is getting loaded along with every row.

This verifies that our routine works fine.

The APD implementation has been completed successfully and the result has been verified successfully.

Related Content

http://help.sap.com/saphelp_erp2004/helpdata/en/49/7e960481916448b20134d471d36a6b/content.htm

<http://forums.sdn.sap.com/thread.jspa?threadID=1933514&tstart=0>

<http://forums.sdn.sap.com/thread.jspa?threadID=1918142&tstart=0>

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.