Key Figure Remodeling in BI7.0 using Customer Exit

Applies to:

SAP BI 7.0

Summary

This article describes how to add new key figures to an InfoCube using the Remodeling feature and populate them using Customer Exits.

Author: Suhas Karnik

Company: Wipro Technologies, Bangalore

Created on: 26 June 2008

Author Bio

Suhas Karnik is a BI Consultant and is involved in a maintenance project for a BI implementation. At present he is working with Wipro Technologies in Bangalore.

Table of Contents

Remodeling

Remodeling is a new feature introduced in BI7.0 by which it is possible to alter the structure of the cube without losing its contents. At present, this feature is not available for DSOs and InfoObjects. In remodeling, we have the following options:

- For Characteristics:
  - Add characteristic
  - Remove characteristic
  - Replace characteristic
- For Key Figures:
  - Add Key Figure
  - Remove Key Figure
  - Replace Key Figure

When adding a key figure or characteristic to a cube, the new InfoObject in the cube can be filled using a constant, or using (for characteristics) the value of an existing characteristic in the same dimension.

However, if a new Key Figure is to be added using the value of another key figure or a characteristic, then a Customer Exit needs to be created. This article will describe how to create a Customer Exit to add a Key Figure to a Cube.

Some important precautions to be taken before performing remodeling:

- Remodeling involves conversion of tables. In characteristic remodeling, the dimension tables are converted and while key figure remodeling, the fact tables are converted. There is a chance of the data being lost or corrupted, so a taking a backup of the InfoCube is strongly recommended.
- All Transformations from and to the InfoCube are deactivated. So are their dependent DTPs. Also, the InfoCube is locked for any loads or changes. Therefore, it should be ensured that no loads are scheduled to or from the Cube while the Remodeling is in progress. Additionally, the transformations and DTPs have to be reactivated after the process is over. Since remodeling involves a change to the structure of the cube, some remapping may have to be done.
- Queries based on the InfoCube are rendered invalid, so the queries must also be changed.

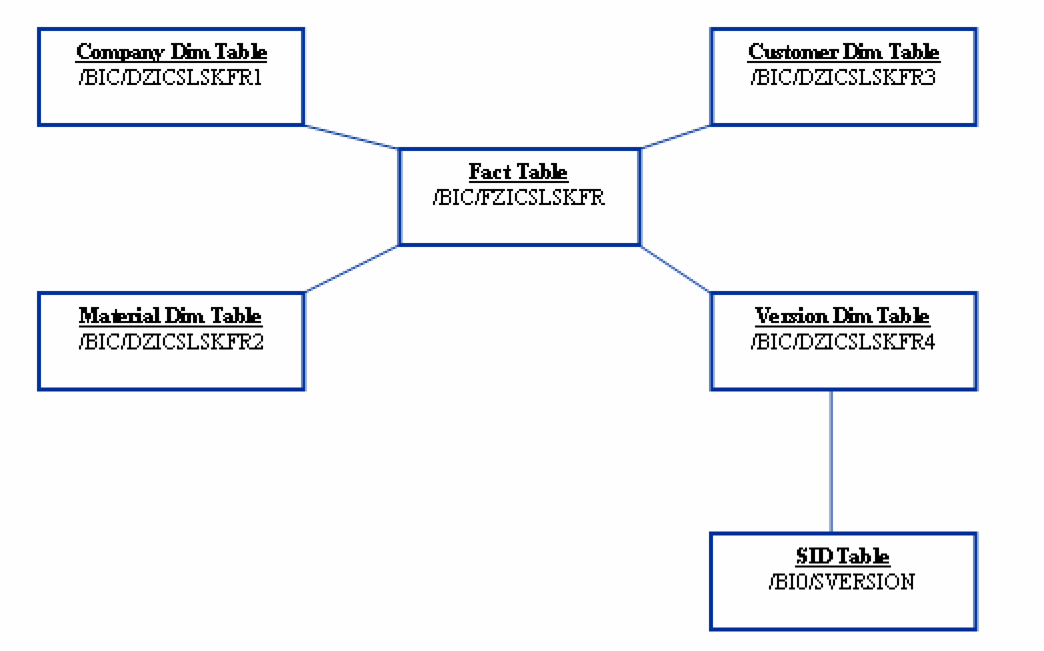
Remodeling cannot be used to add a new dimension or remove it. Nor can it be used to delete a characteristic when doing so will make the corresponding dimension empty. Unit fields cannot be removed during remodeling.

Getting Started

Consider a scenario wherein an InfoCube contains the Sales Quantity, Revenue and Contribution Margin key figures across the dimensions Company, Material, Customer and Version. Some of the data for revenue in the cube is Plan data and some of it is Actual data.

The data is currently stored as an Account Based Model and needs to be converted into a Key Figure based model. In other words, the data for Plan and Actual Revenues is being stored in a single column and the type of record (Plan, Actual) is denoted by the Version column. Now, that column needs to be split into two columns, one for Plan data and one for Actual data, both being filled up by Customer Exits.

The Extended Star Schema for the Cube is as below.



The SID tables for Material, Company and Customer have not been shown here for reasons of space. The procedure to fill up the value in the new Actual Data column is as follows.

First it must be ascertained whether the original value in the Revenue column of the old fact table is Actual or Plan. To do this, first the “Dim ID” for the Version dimension is found. This is, of course, a field in the fact table and so is readily available to the Customer Exit at execution. Generally this DIM ID follows the naming convention KEY_<Dimension technical name>. This technical name is displayed in the InfoCube modeling screen against the name of the dimension.

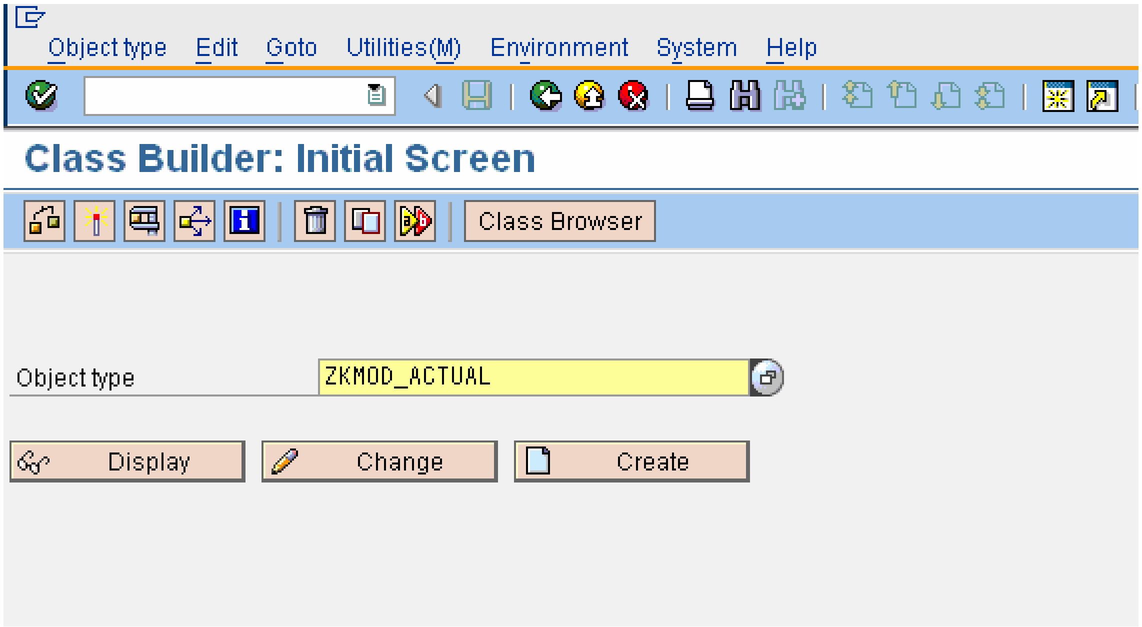
This obtained Dimension ID value is looked up in the Dimension Table to obtain the SID value for the characteristic. The Dimension table follows the naming convention /BIC/D<Dimension technical name>.

Once this is done, the SID value is looked up in the SID table to get the characteristic value. For this step, a standard function module is available which will do the lookup.

After the lookup is done, the value of the Version characteristic is available. The new key figure is filled with the existing Revenue key figure, based on whether Version value is for Actual or Plan.

Creating the Exit Class and Method

The remodeling exit must be written as a method of a class. To begin, open the class builder using transaction SE24.

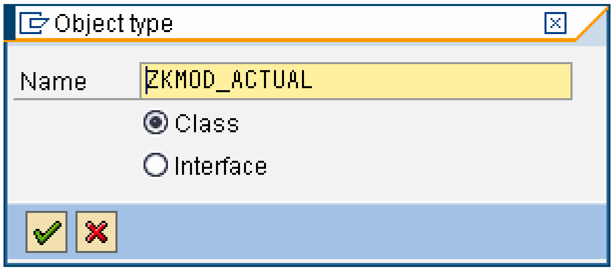


Class Builder: Initial Screen

Object type ZKMOD_ACTUAL

Enter the name of a new class. This name should start with Y or Z. After entering then name, click *Create*.

As this will be a concrete class, not an interface, select *Class* in the following dialog box.



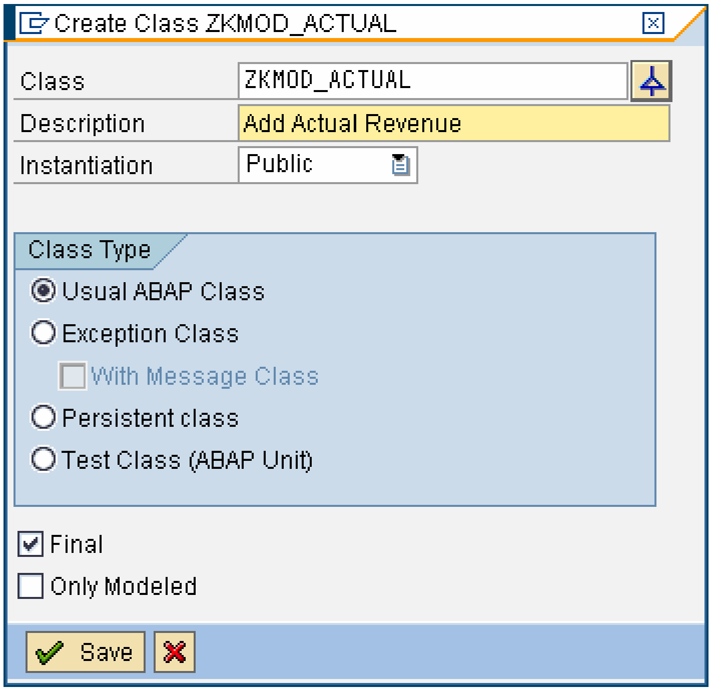
Object type

Name ZKMOD_ACTUAL

● Class

○ Interface

Select the radio button *Usual ABAP Class* and the check box *Final*.



Create Class ZKMOD_ACTUAL

Class ZKMOD_ACTUAL

Description Add Actual Revenue

Instantiation Public

Class Type

● Usual ABAP Class

○ Exception Class

□ With Message Class

○ Persistent class

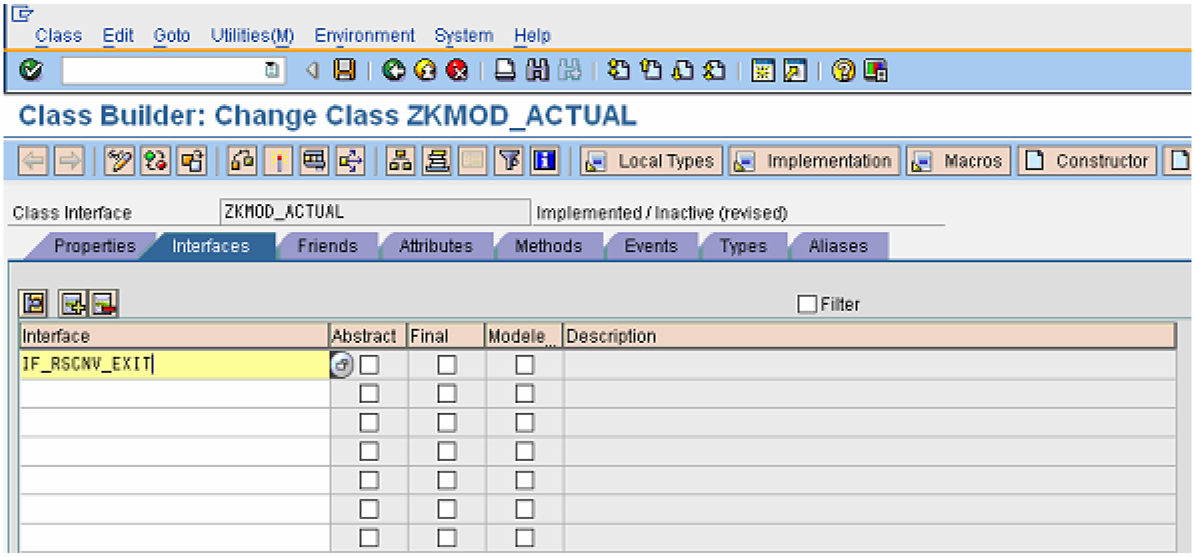
○ Test Class (ABAP Unit)

☑ Final

□ Only Modeled

The Class Builder main screen will come up. Navigate to the *Interfaces* tab. Here, the interface *IF_RSCNV_EXIT* must be entered. An interface is an abstract structure that defines how a class interacts with other classes. An interface can specify a set of methods, and classes that implement the interface will have to provide concrete implementations of those methods.

The interface only needs to specify the signature of the methods; any other class that needs to interact with this class only needs to know the interface being implemented by the class.



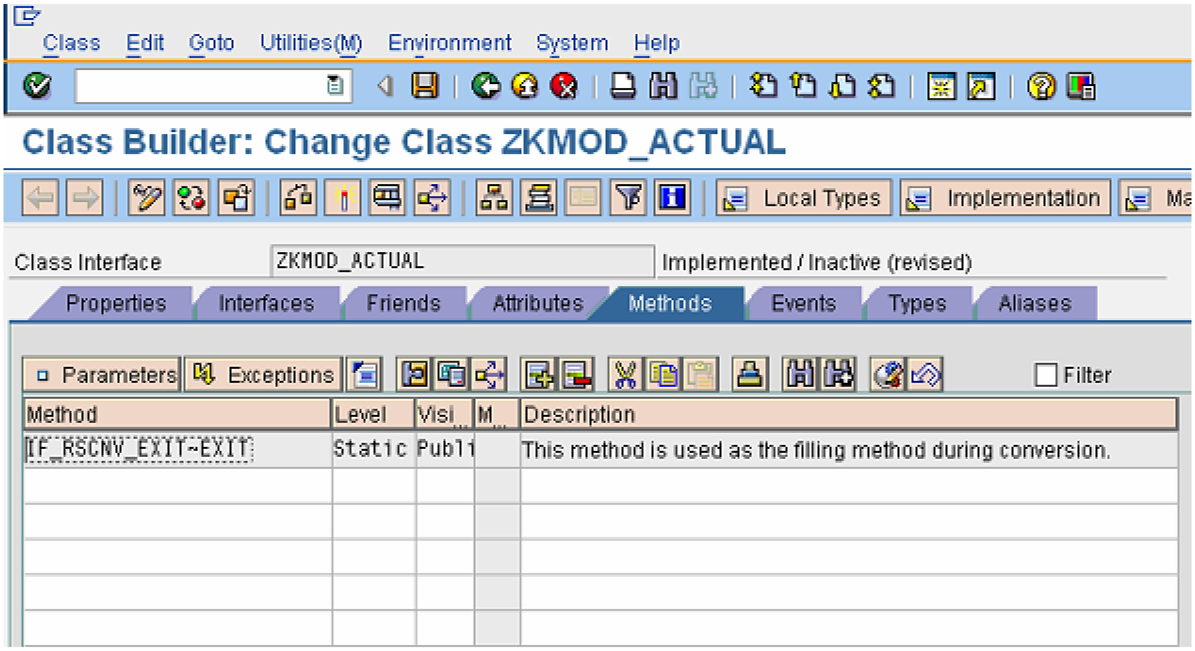
Class Builder: Change Class ZKMOD_ACTUAL

Class Interface ZKMOD_ACTUAL Implemented / Inactive (revised)

Interface	Abstract	Final	Modele...	Description
IF_RSCNV_EXIT	□	□	□	
	□	□	□	
	□	□	□	
	□	□	□	
	□	□	□	
	□	□	□	
	□	□	□	

Here, *IF_RSCNV_EXIT* is the interface that a class must implement for the Remodeling functionality to call it during Remodeling. The method *EXIT* (seen later) is declared in this interface and the Remodeling process will expect that method to be present in the Remodeling class.

After entering the interface name in the *Interfaces* tab, navigate to the *Methods* tab.



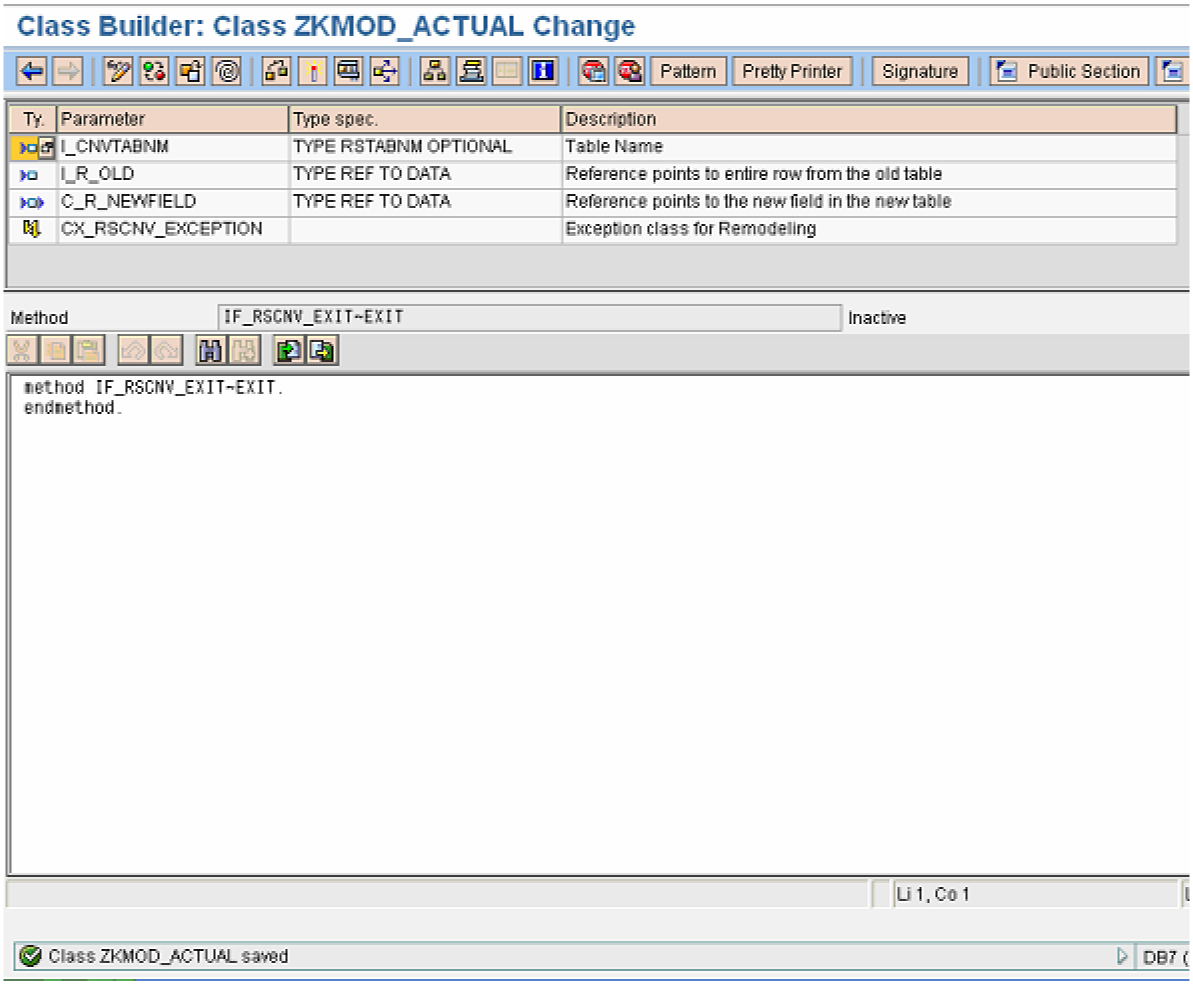
Class Builder: Change Class ZKMOD_ACTUAL

Class Interface ZKMOD_ACTUAL Implemented / Inactive (revised)

Method	Level	Visi...	M...	Description
IF_RSCNV_EXIT~EXIT	Static	Publi		This method is used as the filling method during conversion.

A new method, called *IF_RSCNV_EXIT~EXIT* is present. This method is a member of the interface and was automatically added when the interface itself was added.

Double click the method name to enter the method editor.



Class Builder: Class ZKMOD_ACTUAL Change

Ty.	Parameter	Type spec.	Description
	I_CNVTABNM	TYPE RSTABNM OPTIONAL	Table Name
	I_R_OLD	TYPE REF TO DATA	Reference points to entire row from the old table
	C_R_NEWFIELD	TYPE REF TO DATA	Reference points to the new field in the new table
	CX_RSCNV_EXCEPTION		Exception class for Remodeling

Method IF_RSCNV_EXIT~EXIT Inactive

```
method IF_RSCNV_EXIT~EXIT.
endmethod.
```

Li 1, Co 1

Class ZKMOD_ACTUAL saved

DB7 (

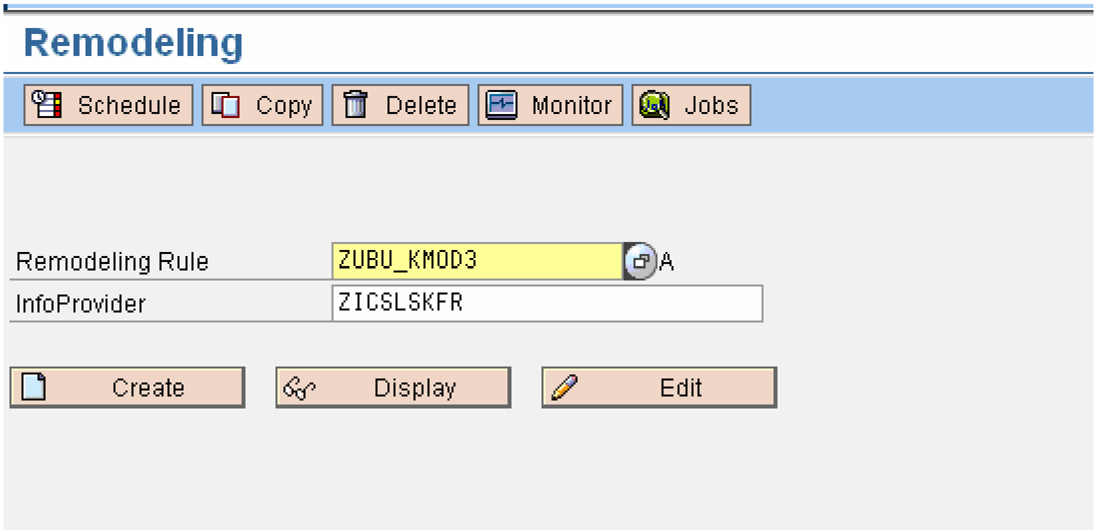
In this editor, enter the remodeling code (given in a later section). Once done, save and activate the method. Go back to the Class Builder Screen and activate the class. This will be the class for the Actual Revenue Key Figure. Similarly create a class for the Planned Revenue Key Figure with the appropriate changes in the code.

The Remodeling process

Once the class is activated, go to the Remodeling screen. This can be done any of these three paths:

- Go to transaction RSMRT
- In RSA1, navigate to the cube that is to be remodeled, right click the cube, and in the menu select *Additional Functions → Remodeling*
- In RSA1, go to the *Administration* tab and click *Remodeling*

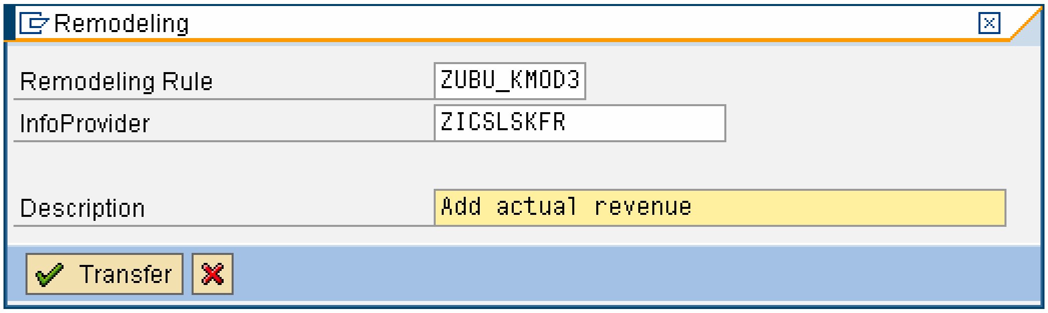
The system will throw a warning. Click OK to continue.



Remodeling Rule	ZUBU_KMOD3
InfoProvider	ZICSLSKFR

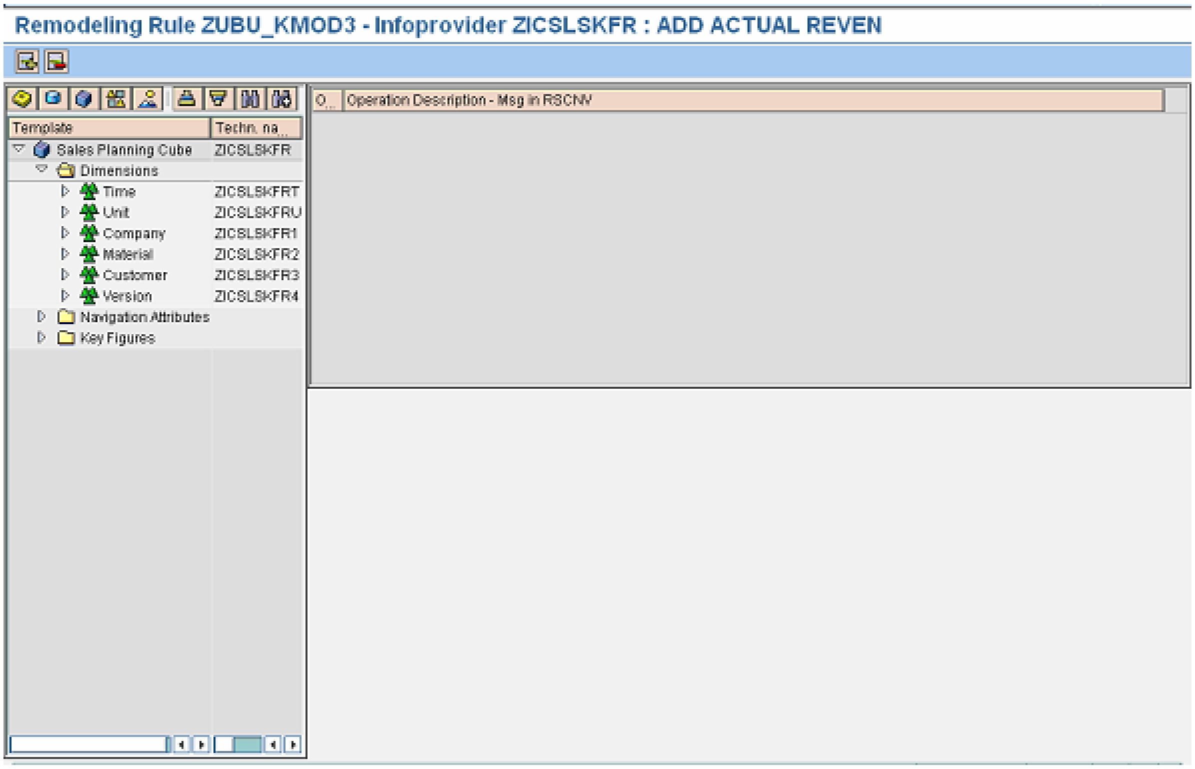
Enter the name of the Info cube and also a name for the Remodeling Rule. If you had entered the Remodeling screen by the option in the Info Cube’s right-click menu, the Info Cube name will be automatically entered.

Click *Create*.



Remodeling Rule	ZUBU_KMOD3
InfoProvider	ZICSLSKFR
Description	Add actual revenue

Enter a description for the Remodeling Rule and click *Transfer*.

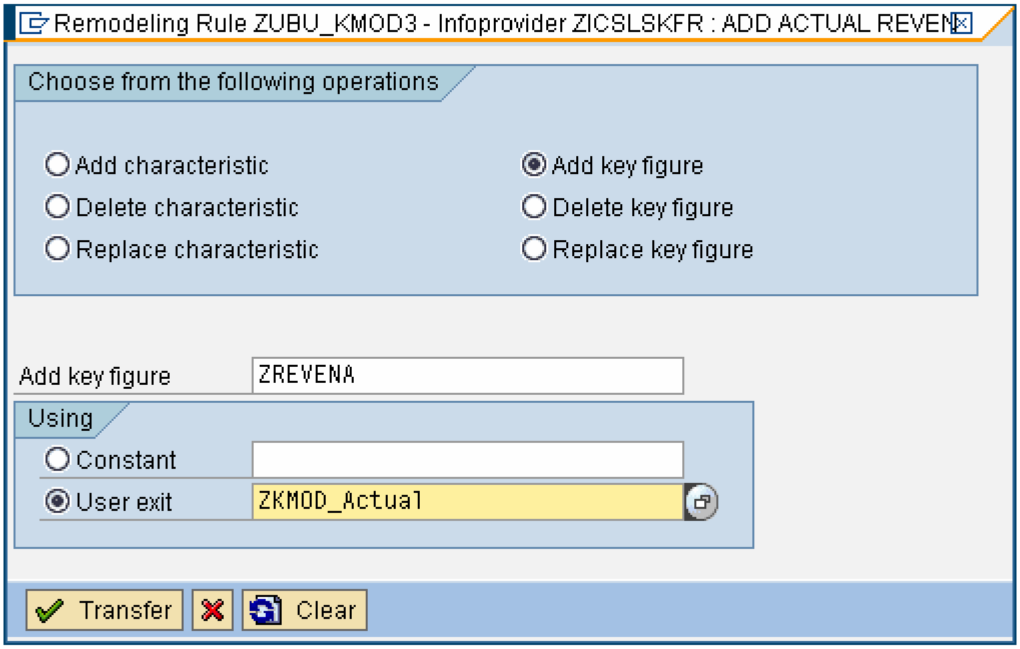


Remodeling Rule ZUBU_KMOD3 - Infoprovider ZICSLSKFR : ADD ACTUAL REVEN

Template	Techn. na...
Sales Planning Cube	ZICSLSKFR
Dimensions	
Time	ZICSLSKFRT
Unit	ZICSLSKFRU
Company	ZICSLSKFR1
Material	ZICSLSKFR2
Customer	ZICSLSKFR3
Version	ZICSLSKFR4
Navigation Attributes	
Key Figures	

This will enter the screen to create the Remodeling rules. Create a new rule here by clicking the *Add* button on the toolbar or the menu path *Edit → Add an operation*.

The following dialog will pop up



Remodeling Rule ZUBU_KMOD3 - Infoprovider ZICSLSKFR : ADD ACTUAL REVEN

Choose from the following operations

○ Add characteristic ● Add key figure
○ Delete characteristic ○ Delete key figure
○ Replace characteristic ○ Replace key figure

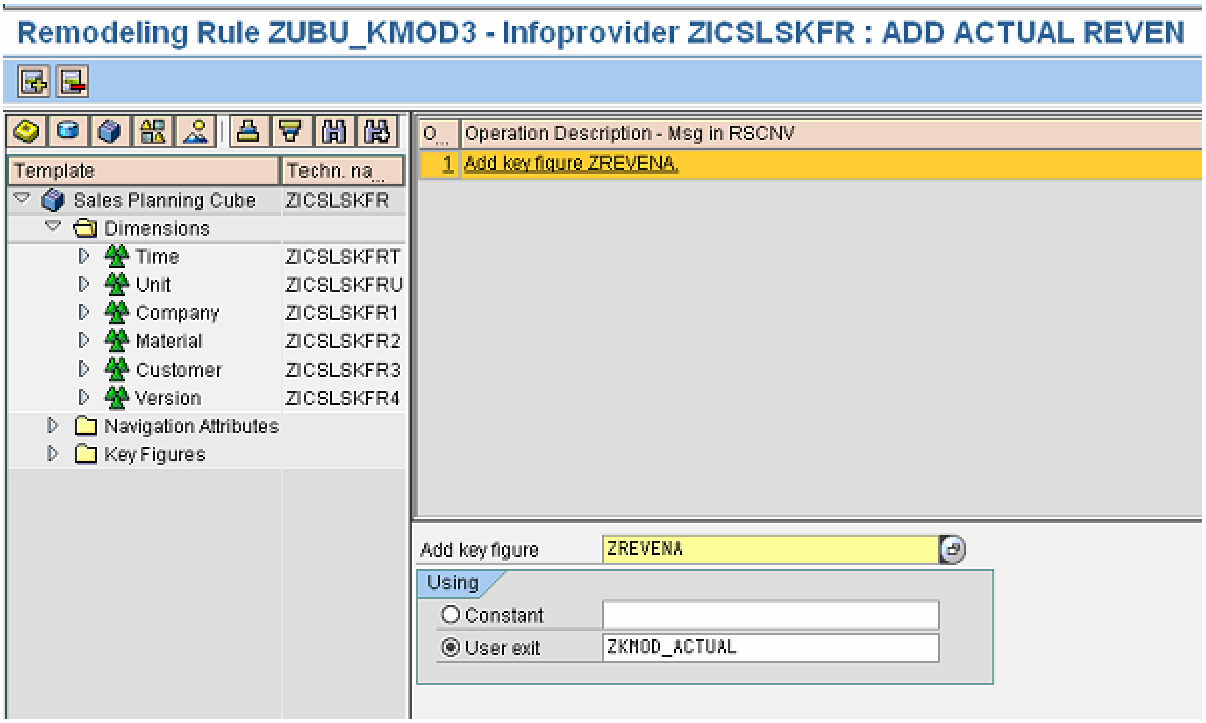
Add key figure ZREVENA

Using

○ Constant
● User exit ZKMOD_Actual

Since the actual revenue key figure is going to be added, select that option and enter the key figure name ZREVENA. Select the User Exit radio button and enter the name of the Remodeling class.

Once done, this will add the rule to the sequence, as shown.



Remodeling Rule ZUBU_KMOD3 - Infoprovider ZICSLSKFR : ADD ACTUAL REVEN

Template	Techn. na...
Sales Planning Cube	ZICSLSKFR
Dimensions	
Time	ZICSLSKFRT
Unit	ZICSLSKFRU
Company	ZICSLSKFR1
Material	ZICSLSKFR2
Customer	ZICSLSKFR3
Version	ZICSLSKFR4
Navigation Attributes	
Key Figures	

O...	Operation Description - Msg in RSCNV
1	Add key figure ZREVENA

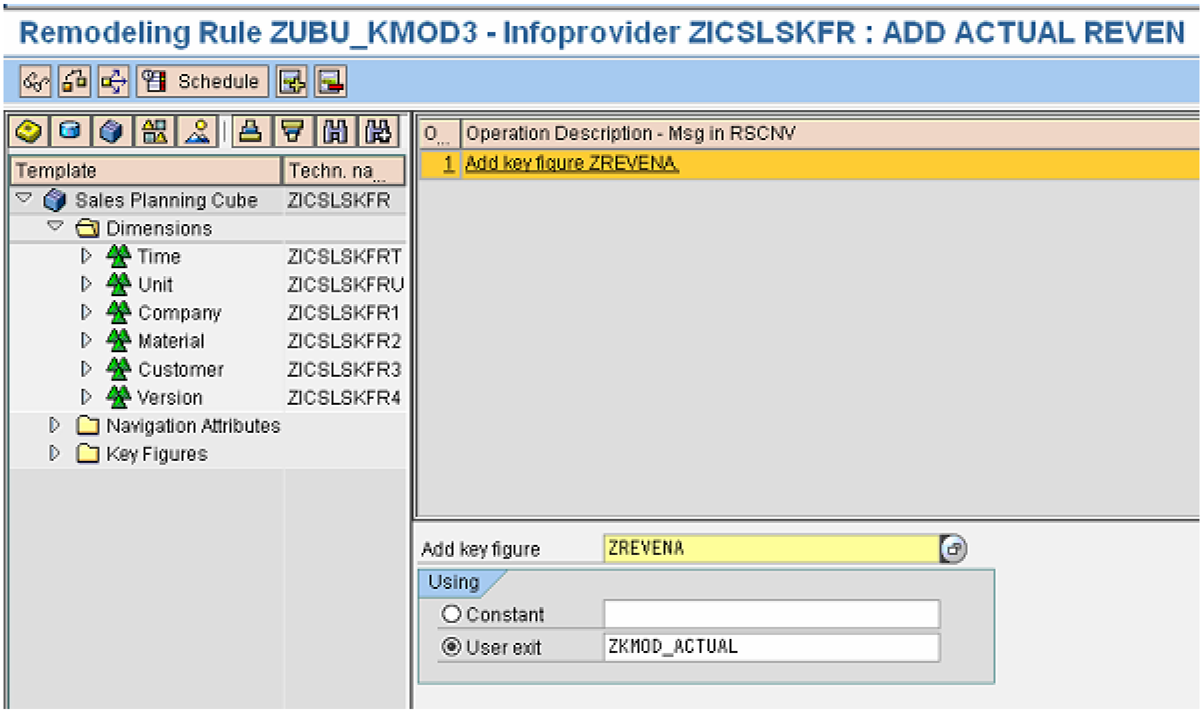
Add key figure: ZREVENA

Using

○ Constant

● User exit: ZKMOD_ACTUAL

Save the work done so far.



Remodeling Rule ZUBU_KMOD3 - Infoprovider ZICSLSKFR : ADD ACTUAL REVEN

Template	Techn. na...
Sales Planning Cube	ZICSLSKFR
Dimensions	
Time	ZICSLSKFRT
Unit	ZICSLSKFRU
Company	ZICSLSKFR1
Material	ZICSLSKFR2
Customer	ZICSLSKFR3
Version	ZICSLSKFR4
Navigation Attributes	
Key Figures	

O...	Operation Description - Msg in RSCNV
1	Add key figure ZREVENA

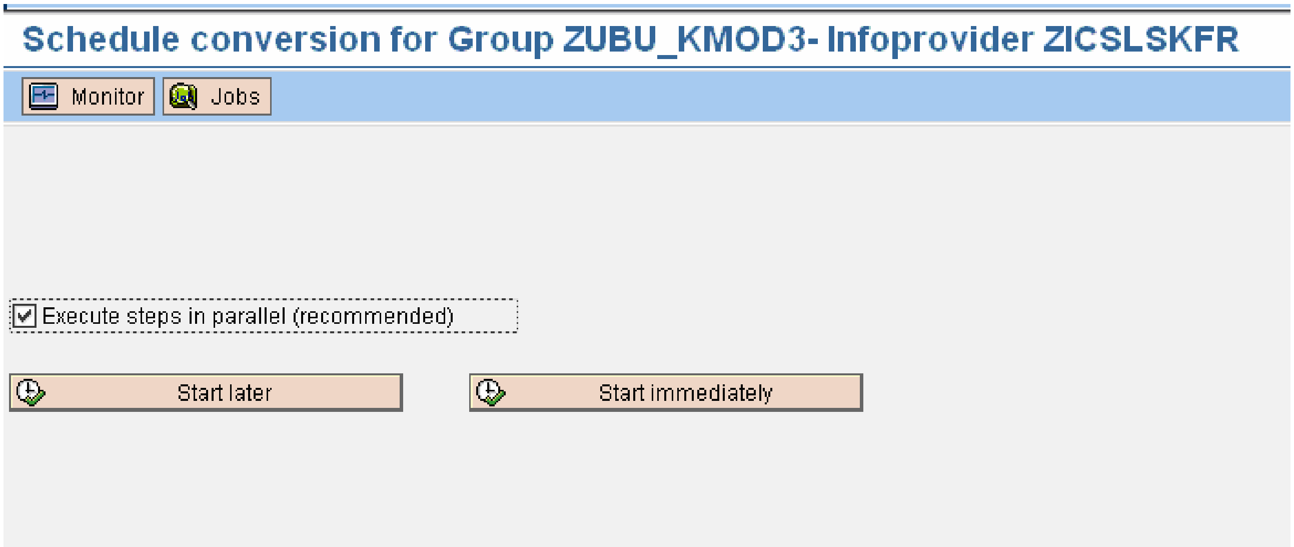
Add key figure: ZREVENA

Using

○ Constant

● User exit: ZKMOD_ACTUAL

Once saved, the Options to check and *Schedule* will appear. *Check* the rule if needed and then click on *Schedule*.

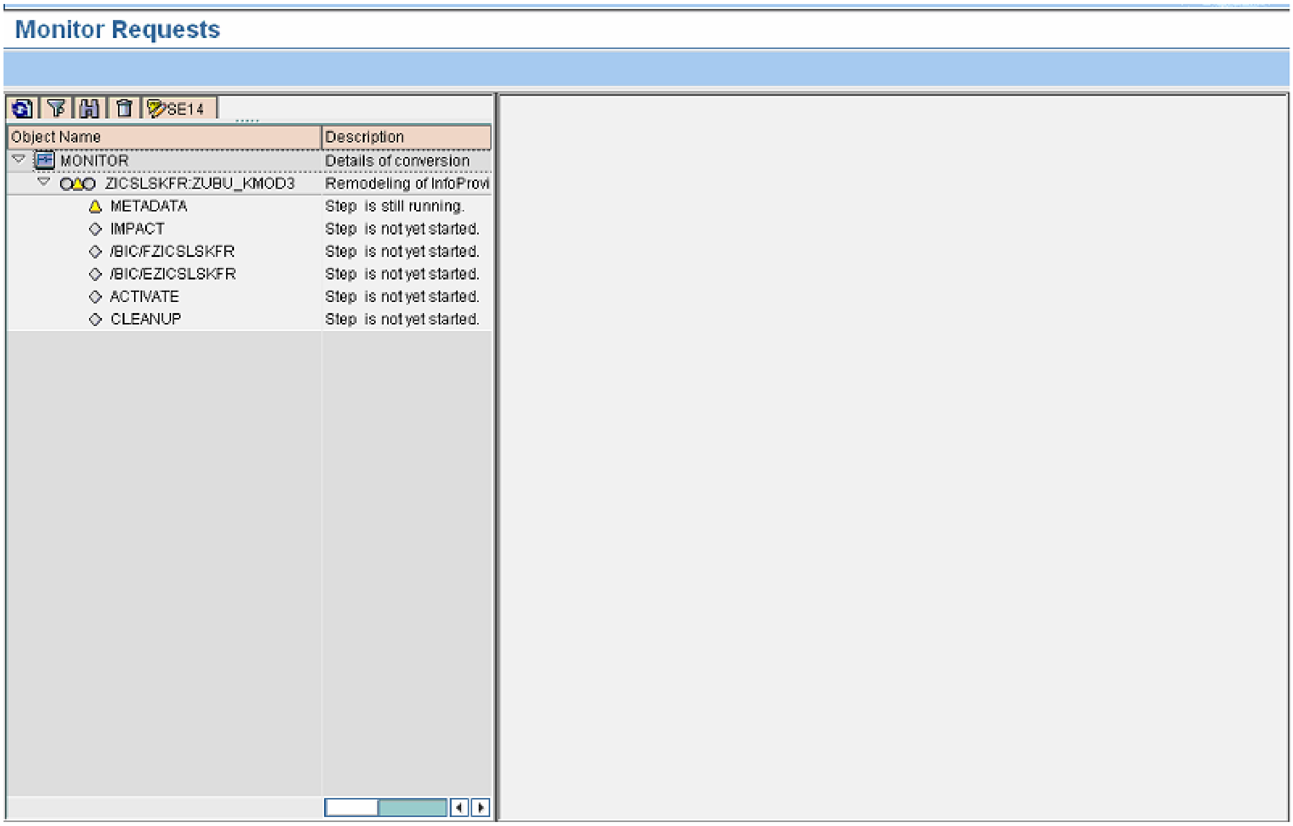


Schedule conversion for Group ZUBU_KMOD3- Infoprovider ZICSLSKFR

☑ Execute steps in parallel (recommended)

The scheduler screen appears. Click *Start Immediately* or schedule a job by selecting *Start Later*. Both will schedule the job in background.

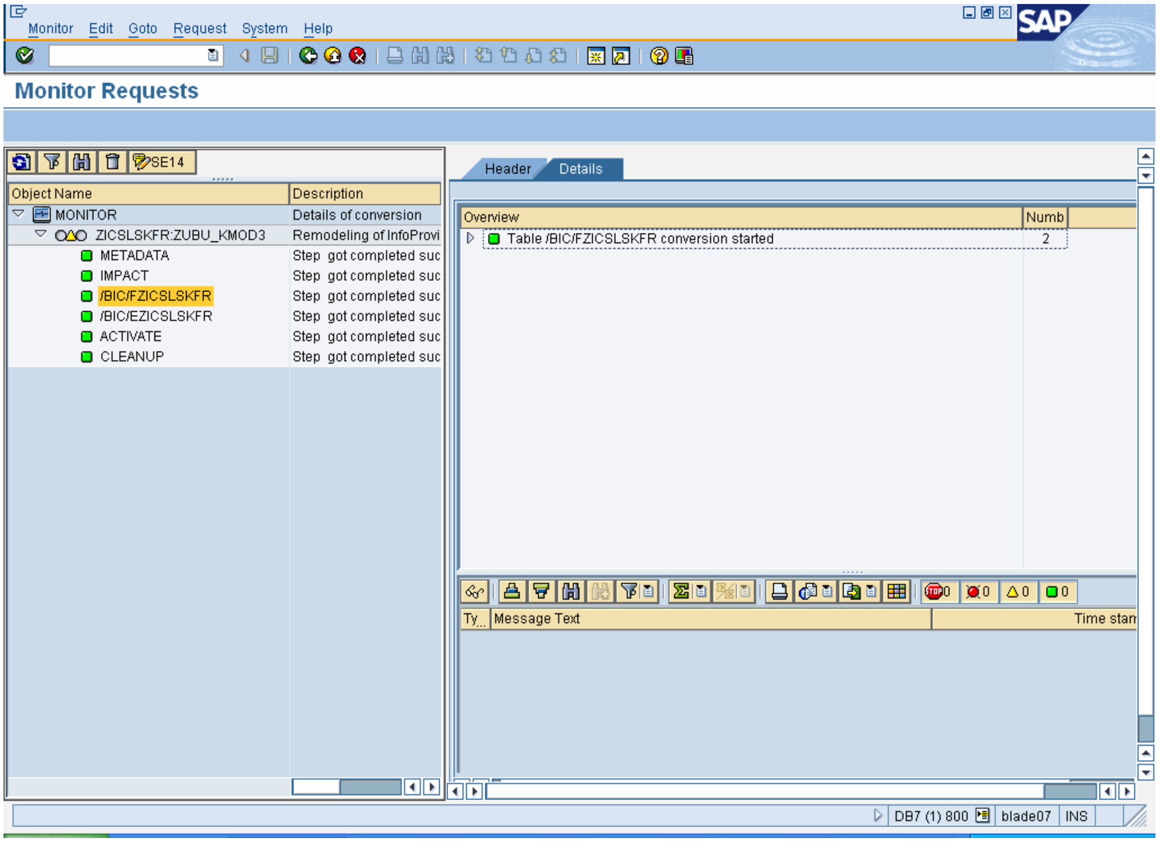
Once scheduled, go to the *Monitor* to view the progress.



Monitor Requests

Object Name	Description
MONITOR	Details of conversion
ZICSLSKFR:ZUBU_KMOD3	Remodeling of InfoProvi
METADATA	Step is still running.
IMPACT	Step is not yet started.
/BIC/FZICSLSKFR	Step is not yet started.
/BIC/EZICSLSKFR	Step is not yet started.
ACTIVATE	Step is not yet started.
CLEANUP	Step is not yet started.

The processing will take some time.



Monitor Requests

Object Name	Description
MONITOR	Details of conversion
ZICSLSKFR:ZUBU_KMOD3	Remodeling of InfoProvi
🟩 METADATA	Step got completed suc
🟩 IMPACT	Step got completed suc
🟩 /BIC/FZICSLSKFR	Step got completed suc
🟩 /BIC/EZICSLSKFR	Step got completed suc
🟩 ACTIVATE	Step got completed suc
🟩 CLEANUP	Step got completed suc

Overview	Numb
🟩 Table /BIC/FZICSLSKFR conversion started	2

STOP 0 ● 0 △ 0 🟩 0

Ty...	Message Text	Time stam

As shown, both the F and E tables will be converted. A similar procedure needs to be followed to create and fill the planned revenue key figure. The code for the plan key figure is the same as for the actuals, except for one change, explained in the code section below.

Once both key figures are filled up, a new remodeling sequence can be created to delete the original 0COPAREVEN keyfigure from the Cube.

Exit Code

The code to be entered in the SE24 method editor for actual revenue is as below. The code for planned revenue involves a changing the last condition from `if l_version = ‘A’` to `if l_version = ‘B’`

```
method IF_RSCNV_EXIT~EXIT.
  FIELD-SYMBOLS: <l_s_old> type any,
    <l_fillfield> type any,
    <l_newfield> type any,
    <l_version_dimid> type any.
  data: l_chavl type rschavl,
        l_zpl_line_chavl type rschavl,
        l_version_sid type /BIC/DZICSLSKFR4-SID_0VERSION,
        l_attr type string,
        l_version type C.
* Name of the field in Dim Table will be SID_<InfoObject name>
* Name of the Dim Table will be /BIC/D<dimension name>

* c_r_newfield is a reference to the field in the new fact table. Assigned to a field
symbol
  assign c_r_newfield->* to <l_newfield>.
```

```
* i_r_old is a reference to the whole row from the old fact table.
  assign i_r_old->* to <l_s_old>.

* allows to directly access the COPAREVEN field of the record from old fact table.
  assign component 'COPAREVEN' of structure <l_s_old> to <l_fillfield>.

* assigns the DIM ID of the Version Dimension
  assign component 'KEY_ZICSLSKFR4' of structure <l_s_old> to <l_version_dimid>.
  if not <l_version_dimid> is assigned.
    RAISE EXCEPTION TYPE cx_rscnv_exception
        EXPORTING
        attr1 = 'CL_RSCNV_USER_EXIT_SAMPLE'
        attr2 = 'EXIT-0'
        attr3 = <l_version_dimid>.
        l_ATTR = sy-subrc.
    return.
  endif.

** The fact table will contain the Dimension ID. It needs to be translated into the
** Characteristic Value for Version. This is done by first querying the Dimension
** Table with the DIMID to obtain the SID of the characteristic.

  select single SID_0VERSION from /BIC/DZICSLSKFR4 into l_version_sid where DIMID =
<l_version_dimid>.
  if sy-subrc <> 0.
* Invalid DIMID, raise exception
    RAISE EXCEPTION TYPE cx_rscnv_exception
        EXPORTING
        attr1 = 'CL_RSCNV_USER_EXIT_SAMPLE'
        attr2 = 'EXIT-1'
        attr3 = <l_version_dimid>.
        l_ATTR = sy-subrc.
  endif.

** This function converts the SID to the value of the Characteristic Info Object
  CALL FUNCTION 'RRSI_SID_VAL_CONVERT_COMPLETE'
      EXPORTING
       I_IOBJNM               = '0VERSION'
        I_SID                  = l_version_sid
     IMPORTING
       E_CHAVL                = l_version
     EXCEPTIONS
       NO_VALUE_FOR_SID       = 1
       X_MESSAGE              = 2
       OTHERS                 = 3.
  if sy-subrc <> 0.
      RAISE EXCEPTION TYPE cx_rscnv_exception
        EXPORTING
        attr1 = 'CL_RSCNV_USER_EXIT_SAMPLE'
        attr2 = 'EXIT-1'
        attr3 = <l_version_dimid>.
        l_ATTR = sy-subrc.
  endif.

** Now l_version will contain either 'A' or 'B'. Since this is the exit for
** Actual Revenue, we will fill the key figure if Version is 'A', and make it zero
** If Version is 'B'.
** In the exit for planned revenue key figure, the condition will fill for Version =
** ‘B’ and make it zero if the version is ‘A’

  if l_version = 'A'.
      <l_newfield> = <l_fillfield>.
    else.
      <l_newfield> = 0.
    endif.
endmethod.
```

Post-remodeling steps

The remodeling process deactivates all the DTPs, Transformations and other dependent objects of the InfoCube. Once the remodeling is over, these need to be manually reactivated. Since InfoObjects have been added, replaced or removed, a remapping of the Transformations may be required.

Aggregates existing on the cube, if any, will have to be recreated.

Related Content

Documentation on Remodeling

Characteristic Remodeling in BI 7.0 using Customer Exit

ABAP Interfaces

Copyright

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.