How to Write Inverse Routine with Expert Routine

Applies to:

Development and support based on SAP BI 7.0

For more information, visit the Business Intelligence homepage.

Summary

The article shows the example of inverse routine and step-by-step process for write an Inverse routine with Expert Routine.

Author: Lakdawala Ashish

Company: Capgemini

Created on: 29 July 2009

Author Bio

Ashish is a SAP BI consultant having around 2 years of experience in SAP BI and currently working in CAPGEMINI. He has extensively worked in SAP BI Development & Production Support projects .His core skill areas are design and development using SAP BI 7.0, SAP BW 3.5, ABAP, VC and BPS/IP.

Table of Contents

Key Concept

If you have defined custom mapping in Expert routine for the SAP Remote Cube, for performance reasons, it's advisable to use an inverse routine. Otherwise the selections cannot be transferred to the source system hence it will bring all the data from Source system and it will filter the records at query level. This hinders the performance of the query based on the remote cube.

Business Scenarios

A query is built on the remote cube to see the total balance based on "Country" level. However, the data at source system is at company code level. To read the data from the source system, you have created an expert routine in transformation to read the master data of company code to populate the "country" code in the target cube. In the query you have created an input variable for "Country". In this case, system will extract all the data from the source and then at query level it filters out the records based on country code selection. Say the source has millions of records, however you want to see the records of only one country code with very minimum data (say in hundreds) if you don't use inverse routine, system will fetch all millions of records and then filters it! This hinders the performance of the query.

Note: Here, though the design is not so complex, we have created expert routine. Because the main aim of this document is to describe how can we use inverse routine with Expert routine?

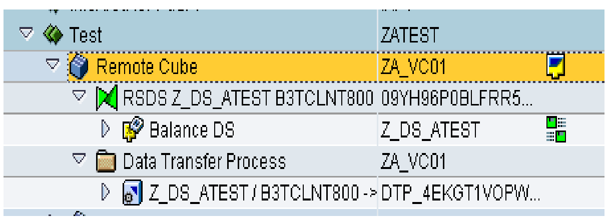
Solution

To resolve the performance issue you need to use the Inverse routine. Inverse routine helps you to pass the desire filter criteria to the source system using query selection parameter. In this case, Inverse Routine will filter the records at the source system itself while extracting the data bases on the Country Code selection values. This will increase your query performance highly.

Step by Step Method Data Model

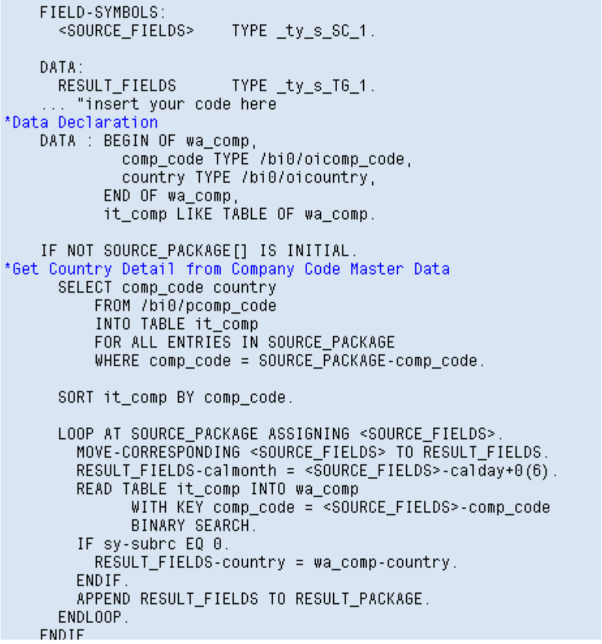
Create a transformation for your remote cube. Within the maintenance of the Transformation, go to Edit menu and Click on Expert Routine. System will give you one information message.

Now in the Expert routine you have written below logic which transfers all source value to target, direct mapping. And for country you have read master data of company code.



Test	ZATEST
Remote Cube	ZA_VC01
RSDS Z_DS_ATEST B3TCLNT800	09YH96P0BLFRR5...
Balance DS	Z_DS_ATEST
Data Transfer Process	ZA_VC01
Z_DS_ATEST / B3TCLNT800 ->	DTP_4EKGT1VOPW...

Expert Routine Code



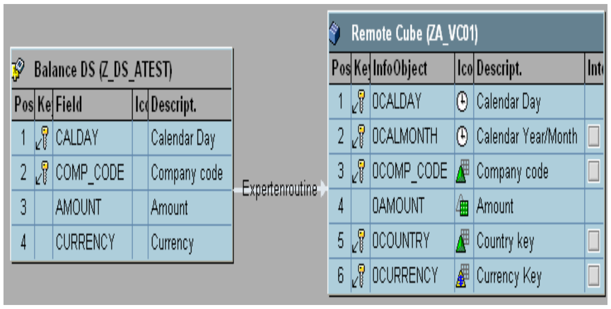
```
    FIELD-SYMBOLS:
      <SOURCE_FIELDS>    TYPE _ty_s_SC_1.

    DATA:
      RESULT_FIELDS      TYPE _ty_s_TG_1.
    ... "insert your code here
*Data Declaration
    DATA : BEGIN OF wa_comp,
             comp_code TYPE /bi0/oicomp_code,
             country TYPE /bi0/oicountry,
           END OF wa_comp,
           it_comp LIKE TABLE OF wa_comp.

    IF NOT SOURCE_PACKAGE[] IS INITIAL.
*Get Country Detail from Company Code Master Data
      SELECT comp_code country
          FROM /bi0/pcomp_code
          INTO TABLE it_comp
          FOR ALL ENTRIES IN SOURCE_PACKAGE
          WHERE comp_code = SOURCE_PACKAGE-comp_code.

      SORT it_comp BY comp_code.

      LOOP AT SOURCE_PACKAGE ASSIGNING <SOURCE_FIELDS>.
        MOVE-CORRESPONDING <SOURCE_FIELDS> TO RESULT_FIELDS.
        RESULT_FIELDS-calmonth = <SOURCE_FIELDS>-calday+0(6).
        READ TABLE it_comp INTO wa_comp
              WITH KEY comp_code = <SOURCE_FIELDS>-comp_code
              BINARY SEARCH.
        IF sy-subrc EQ 0.
          RESULT_FIELDS-country = wa_comp-country.
        ENDIF.
        APPEND RESULT_FIELDS TO RESULT_PACKAGE.
      ENDLOOP.
    ENDIF
```



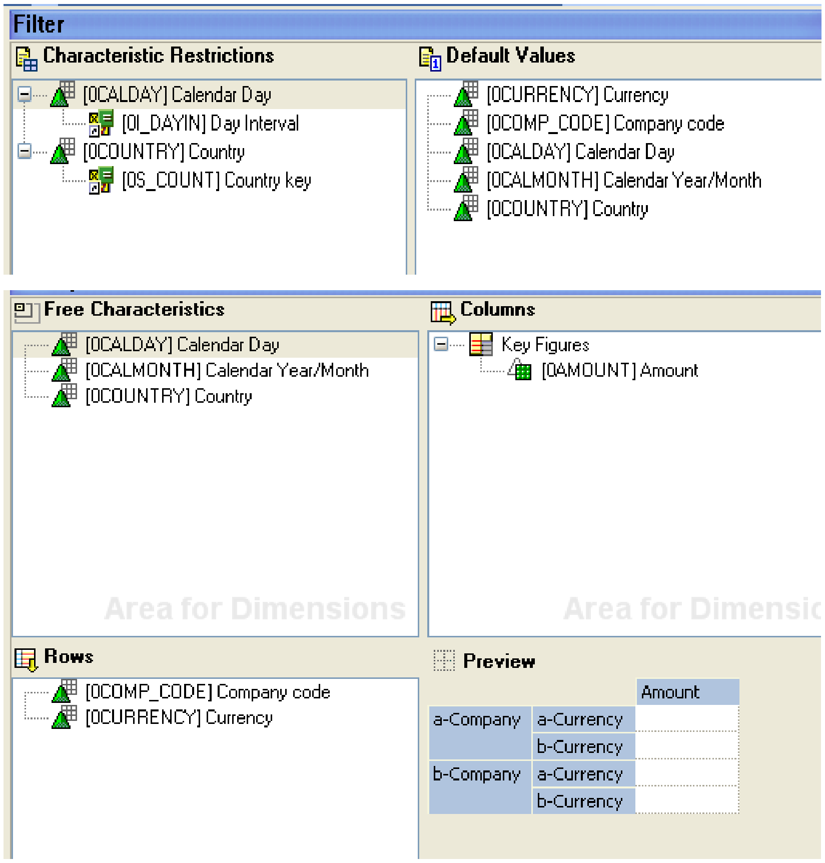
Balance DS (Z_DS_ATEST)

Pos	Ke	Field	Ic	Descript.
1	⚷	CALDAY		Calendar Day
2	⚷	COMP_CODE		Company code
3		AMOUNT		Amount
4		CURRENCY		Currency

Remote Cube (ZA_VC01)

Pos	Key	InfoObject	Ico	Descript.	Int
1	⚷	0CALDAY	🕒	Calendar Day	
2	⚷	0CALMONTH	🕒	Calendar Year/Month	□
3	⚷	0COMP_CODE	▲	Company code	□
4		0AMOUNT	▦	Amount	
5	⚷	0COUNTRY	▲	Country key	□
6	⚷	0CURRENCY	▲	Currency Key	□

Query Design:



Filter

Characteristic Restrictions

- [0CALDAY] Calendar Day
  - [0I_DAYIN] Day Interval
- [0COUNTRY] Country
  - [0S_COUNT] Country key

Default Values

- [0CURRENCY] Currency
- [0COMP_CODE] Company code
- [0CALDAY] Calendar Day
- [0CALMONTH] Calendar Year/Month
- [0COUNTRY] Country

Free Characteristics

- [0CALDAY] Calendar Day
- [0CALMONTH] Calendar Year/Month
- [0COUNTRY] Country

Columns

- Key Figures
  - [0AMOUNT] Amount

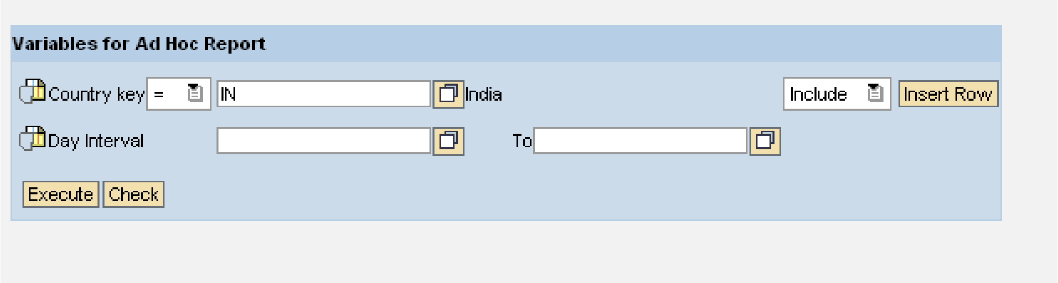
Rows

- [0COMP_CODE] Company code
- [0CURRENCY] Currency

Preview

		Amount
a-Company	a-Currency	
	b-Currency	
b-Company	a-Currency	
	b-Currency	

Case 1: Without Inverse Routine.



Variables for Ad Hoc Report

Country key	=	IN	India	Include
Day Interval			To	

As you can see even here you have restricted the country code value eq ‘IN’. Result package contains country codes other than ‘IN’ also. Unnecessary system fetches all the data of the source though the query will show 2 records related to “IN”. This will consume time and degrade the performance of the Query.

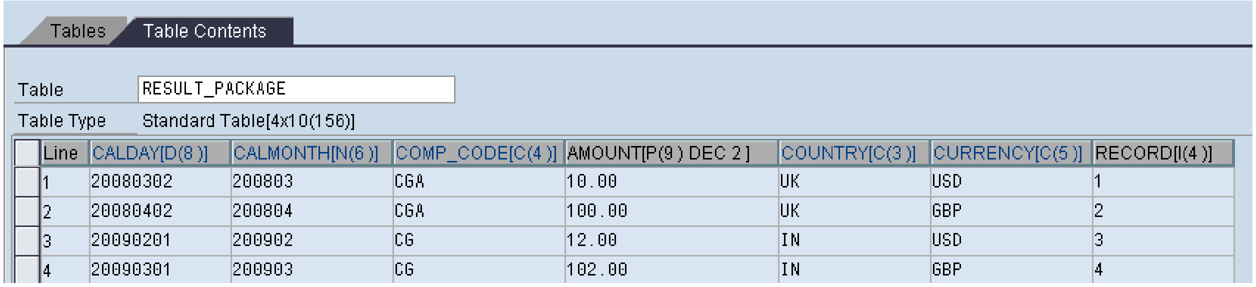
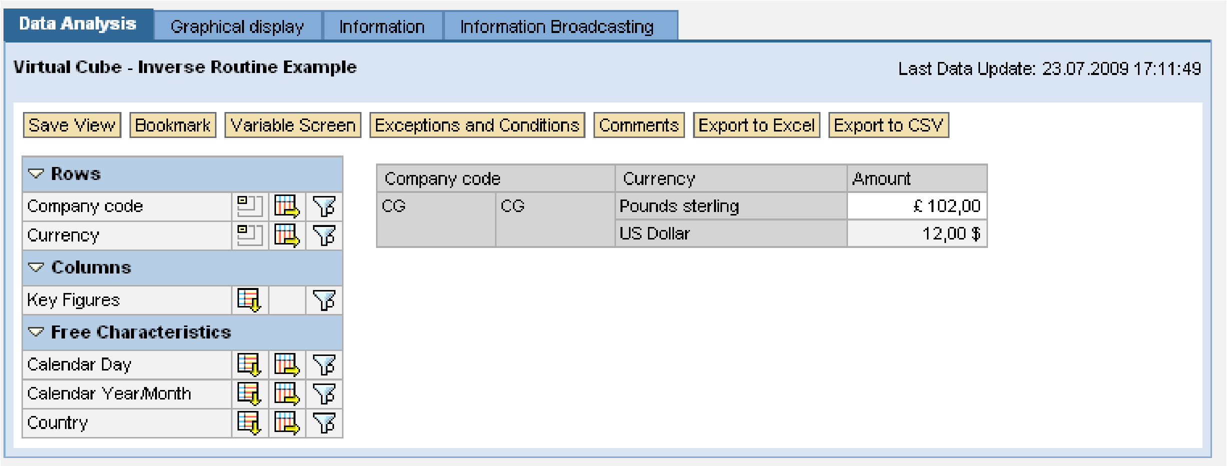


Table RESULT_PACKAGE

Table Type Standard Table[4x10(156)]

Line	CALDAY[D(8)]	CALMONTH[N(6)]	COMP_CODE[C(4)]	AMOUNT[P(9) DEC 2]	COUNTRY[C(3)]	CURRENCY[C(5)]	RECORD[I(4)]
1	20080302	200803	CGA	10.00	UK	USD	1
2	20080402	200804	CGA	100.00	UK	GBP	2
3	20090201	200902	CG	12.00	IN	USD	3
4	20090301	200903	CG	102.00	IN	GBP	4



Virtual Cube - Inverse Routine Example

Last Data Update: 23.07.2009 17:11:49

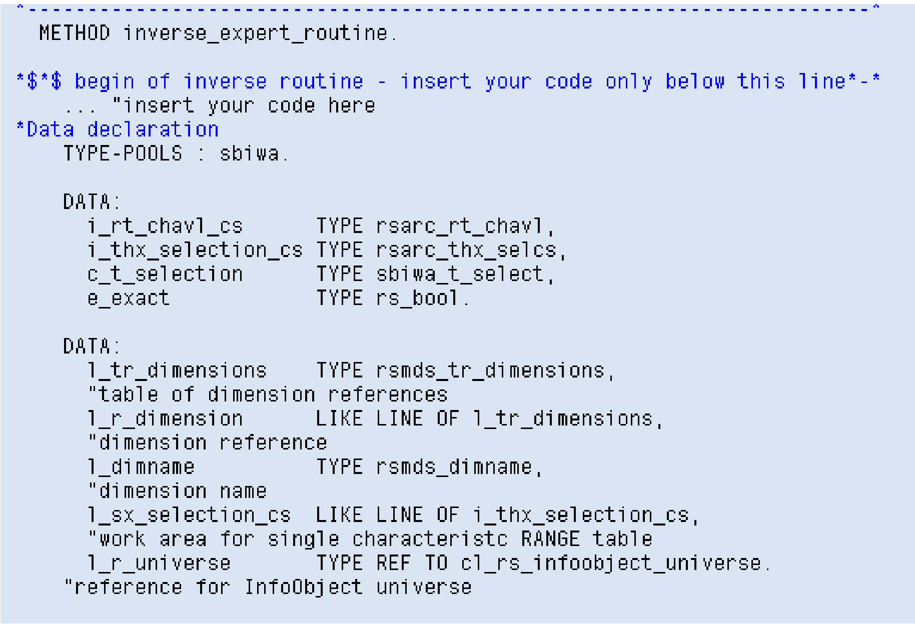
Rows
- Company code
- Currency

Columns
- Key Figures

Free Characteristics
- Calendar Day
- Calendar Year/Month
- Country

Company code		Currency	Amount
CG	CG	Pounds sterling	£ 102,00
		US Dollar	12,00 $

Case 2: After Including Inverse Routine.



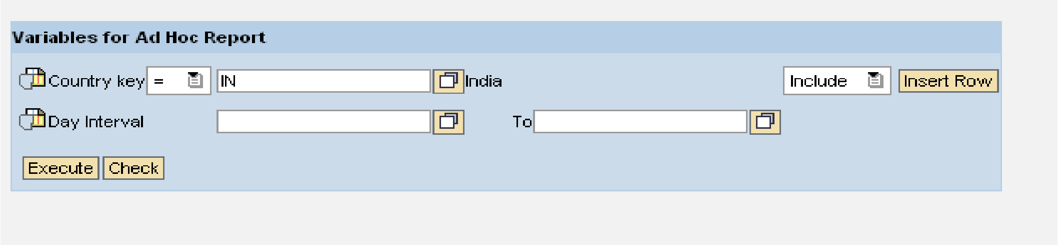
```
*-----------------------------------------------------------------------*
  METHOD inverse_expert_routine.

*$*$ begin of inverse routine - insert your code only below this line*-*
    ... "insert your code here
*Data declaration
    TYPE-POOLS : sbiwa.

    DATA:
      i_rt_chavl_cs      TYPE rsarc_rt_chavl,
      i_thx_selection_cs TYPE rsarc_thx_selcs,
      c_t_selection      TYPE sbiwa_t_select,
      e_exact            TYPE rs_bool.

    DATA:
      l_tr_dimensions    TYPE rsmds_tr_dimensions,
      "table of dimension references
      l_r_dimension      LIKE LINE OF l_tr_dimensions,
      "dimension reference
      l_dimname          TYPE rsmds_dimname,
      "dimension name
      l_sx_selection_cs  LIKE LINE OF i_thx_selection_cs,
      "work area for single characteristc RANGE table
      l_r_universe       TYPE REF TO cl_rs_infoobject_universe.
    "reference for InfoObject universe
```

Now you add the inverse routine as given above and again run the same query for Country Key = ‘IN’.



Variables for Ad Hoc Report

Country key	=	IN	India		Include	Insert Row
Day Interval				To		

As you can see in inverse routine, in I_THX_SELECTION_CS table which is derived from i_r_selset_outbound (query selection condition) you can get all the dimension which are restricted in query.

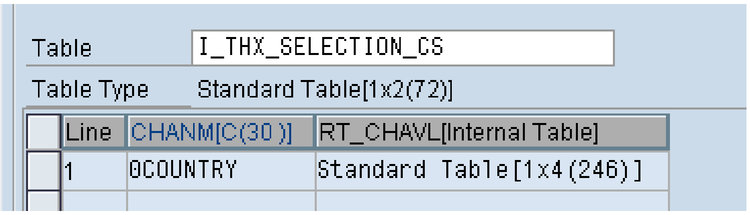


Table I_THX_SELECTION_CS

Table Type Standard Table[1x2(72)]

Line	CHANM[C(30)]	RT_CHAVL[Internal Table]
1	0COUNTRY	Standard Table[1x4(246)]

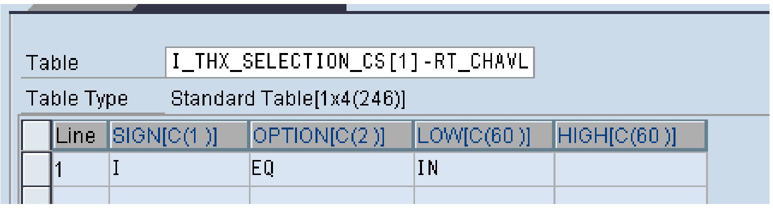


Table I_THX_SELECTION_CS[1]-RT_CHAVL

Table Type Standard Table[1x4(246)]

Line	SIGN[C(1)]	OPTION[C(2)]	LOW[C(60)]	HIGH[C(60)]
1	I	EQ	IN	

Here as per your requirement you have written code to find the company code from country code by using company code master data. And pass this c_r_selset_inbound which is Target selection conditions.

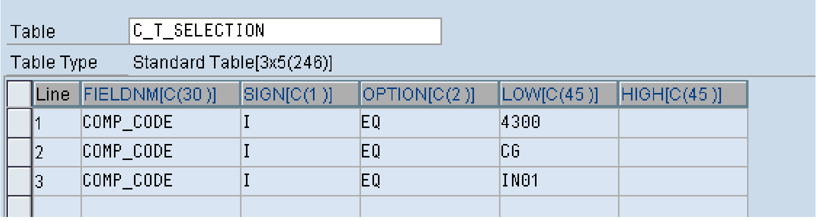


Table C_T_SELECTION

Table Type Standard Table[3x5(246)]

Line	FIELDNM[C(30)]	SIGN[C(1)]	OPTION[C(2)]	LOW[C(45)]	HIGH[C(45)]
1	COMP_CODE	I	EQ	4300	
2	COMP_CODE	I	EQ	CG	
3	COMP_CODE	I	EQ	IN01	

Now you can see that system have fetched only those records whose company code having country key ‘IN’. So this will save query execution time and will increase query performance highly.

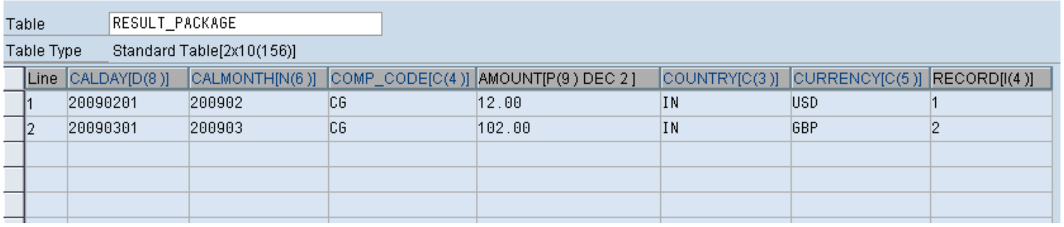
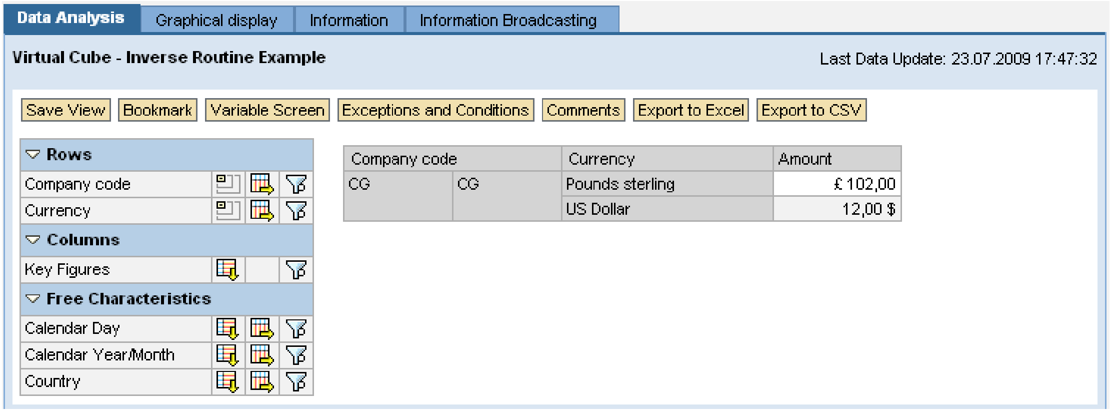


Table RESULT_PACKAGE

Table Type Standard Table[2x10(156)]

Line	CALDAY[D(8)]	CALMONTH[N(6)]	COMP_CODE[C(4)]	AMOUNT[P(9) DEC 2]	COUNTRY[C(3)]	CURRENCY[C(5)]	RECORD[I(4)]
1	20090201	200902	CG	12.00	IN	USD	1
2	20090301	200903	CG	102.00	IN	GBP	2



Virtual Cube - Inverse Routine Example

Last Data Update: 23.07.2009 17:47:32

▽ Rows
- Company code
- Currency

▽ Columns
- Key Figures

▽ Free Characteristics
- Calendar Day
- Calendar Year/Month
- Country

Company code		Currency	Amount
CG	CG	Pounds sterling	£ 102,00
		US Dollar	12,00 $

Appendix

Sample source code for Inverse routine within Expert Routine. Just change the code which is written in between “Write Your Code*******” based on your logic.

```
*-----------------------------------------------------------------------*
*       Method inverse_expert_routine
*-----------------------------------------------------------------------*
*
*       This subroutine needs to be implemented only for direct access
*       (for better performance) and for the Report/Report Interface
*       (drill through).
*       The inverse routine should transform a projection and
*       a selection for the target to a projection and a selection
*       for the source, respectively.
*       If the implementation remains empty all fields are filled and
*       all values are selected.
*
*-----------------------------------------------------------------------*
*
*-----------------------------------------------------------------------*
  METHOD inverse_expert_routine.

*$*$ begin of inverse routine - insert your code only below this line*-*
    ... "insert your code here
*Data declaration
    TYPE-POOLS : sbiwa.

    DATA:
      i_rt_chavl_cs      TYPE rsarc_rt_chavl,
      i_thx_selection_cs TYPE rsarc_thx_selcs,
      c_t_selection      TYPE sbiwa_t_select,
      e_exact            TYPE rs_bool.

    DATA:
      l_tr_dimensions    TYPE rsmds_tr_dimensions,
      "table of dimension references
      l_r_dimension      LIKE LINE OF l_tr_dimensions,
      "dimension reference
      l_dimname          TYPE rsmds_dimname,
      "dimension name
      l_sx_selection_cs  LIKE LINE OF i_thx_selection_cs,
      "work area for single characteristc RANGE table
      l_r_universe       TYPE REF TO cl_rs_infoobject_universe.
    "reference for InfoObject universe

    DATA : BEGIN OF wa_comp,
             comp_code TYPE /bi0/oicomp_code,
             country TYPE /bi0/oicountry,
           END OF wa_comp,
           it_comp LIKE TABLE OF wa_comp,
           rsarc_rt_chavl TYPE RANGE OF rschavl.

    DATA:
       l_s_selection      LIKE LINE OF c_t_selection,
       l_s_chavl_cs       LIKE LINE OF i_rt_chavl_cs,
       date TYPE sy-datum.

*Start
    TRY.

*       Transform selection set for outbound
        CALL METHOD i_r_selset_outbound->to_ranges
          CHANGING
            c_t_ranges = i_rt_chavl_cs.

*       Get reference to InfoObject universe
        l_r_universe = cl_rs_infoobject_universe=>get_instance( ).

*Get all dimensions (i.e. fields) from outbound selection which are
*restricted
        l_tr_dimensions = i_r_selset_outbound->get_dimensions( ).


        LOOP AT l_tr_dimensions INTO l_r_dimension.
          CLEAR l_sx_selection_cs.

*         Get dimension name (= field name)
          l_dimname = l_r_dimension->get_name( ).

*         Transform dimension name to InfoObject name
          l_sx_selection_cs-chanm = l_r_universe->dimname_to_iobjnm(
          l_dimname ).

*Project complete outbound selection set to current dimension and
*and convert to RANGE table representation
          CALL METHOD i_r_selset_outbound->to_ranges
            EXPORTING
              i_r_dimension = l_r_dimension
            CHANGING
              c_t_ranges    = l_sx_selection_cs-rt_chavl.
          APPEND l_sx_selection_cs TO i_thx_selection_cs.
        ENDLOOP.

*Now Read the range table for your Infoobject and
*write your Code***********************************************
*Test Logic: -
*From Country we are finding the Company code and then pass
*it to the Taget selection conditions
        READ TABLE i_thx_selection_cs INTO l_sx_selection_cs
                    WITH KEY chanm = '0COUNTRY'.
                    "Your Input Infoobject Name

        l_s_selection-fieldnm = 'COMP_CODE'. "Source fields name
        IF sy-subrc EQ 0.
          rsarc_rt_chavl = l_sx_selection_cs-rt_chavl.

          SELECT comp_code country
              FROM /bi0/pcomp_code
              INTO TABLE it_comp
              WHERE country IN rsarc_rt_chavl.
          SORT it_comp BY comp_code.
          DELETE ADJACENT DUPLICATES FROM it_comp
                        COMPARING comp_code.

          l_s_selection-sign   = 'I'.
          l_s_selection-option = 'EQ'.
          CLEAR l_s_selection-high.
          LOOP AT it_comp INTO wa_comp.
            l_s_selection-low = wa_comp-comp_code.
            APPEND l_s_selection TO c_t_selection.
          ENDLOOP.
        ENDIF.
********End your Code*******************************************

        c_r_selset_inbound =
        i_r_universe_inbound->create_set_from_ranges(
                               i_fieldname_dimension = 'FIELDNM'
                               i_t_ranges            = c_t_selection )
                               .
        c_exact = e_exact.

      CATCH cx_rsmds_input_invalid
            cx_rsmds_input_invalid_type.

*       Should not occur
*       If the exception occurs request all values from source
*       for this routine to be on the save side
        c_r_selset_inbound = cl_rsmds_set=>get_universal_set( ).
        c_exact = rs_c_false. "Inversion is no longer exact

    ENDTRY.
```

Related Information

http://help.sap.com/saphelp_nw70/helpdata/en/45/f1f9b861c706f9e10000000a155369/content.htm

For more information, visit the Business Intelligence homepage.

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.