Reading data directly from InfoCube using ABAP code

Introduction:

This document explains the process of reading data directly from an InfoCube using ABAP code.

Business Scenario:

Suppose that you are loading data into an InfoProvider and for calculating an InfoObject you need to apply certain logic that depends on the data residing in another InfoCube. For example, you are building an InfoCube Z_DOS for calculating the Days of Supply of Materials at the Product Group level. For several reasons you have decided to calculate the Days of Supply in the back-end and store in the InfoCube, instead of calculating it in the front-end (using restricted and calculated key figures). The definition of Days of Supply varies from Vendor to Vendor. For example, for Vendor A, Days of Supply for a given day is defined as "Average Monthly Inventory for the past 2 week divided by Average Monthly Sales for the past 2 weeks", where as for Vendor B, it is defined as "Average Monthly Inventory for Past 30 days divided by Average Monthly Sales for Past 30 days".

Reading data from an ODS is simple and straight forward because the ODS is pretty close to a flat database table. However, if you have to read data from an InfoCube directly, you have to keep the following things in mind:

1. Behind an InfoCube there are several tables, including Fact tables and Dimension tables. Apart from the key figures, the fact table has the DIMIDs for regular dimensions and SIDs for line item dimensions.
2. If the cube is physically partitioned, then in addition to the key figures and DIMIDs/SIDs, the SIDs of the characteristic on which the cube is physically partitioned is also stored in the fact table. Hence some special code might be required to take advantage of the performance benefits of the physical partitioning.
3. If the InfoCube is partially compressed then the data will be residing in two fact tables, the E and F fact tables.

The following is the general algorithm to read a characteristic or a key figure for a given set of values of characteristics:

Step 1:
Find the SIDs of the characteristic values by which you are reading data from the InfoCube. These SIDs are independent of the InfoCube.

Step 2:
If the characteristic is a part of a regular dimension, then find the list of DIMIDs for the SID. You can have several DIMIDs for a given SID.
If the dimension is a Line Item dimension, then fill the list of DIMIDs with the SIDs.

Step 3:
Based on the filter characteristic values if the values of characteristic by which the cube is physically partitioned can be derived then find the SIDs of the physical partitioning characteristic. For example, if one of the filter characteristics is Calendar Week and the Cube is physically partitioned by Calendar Month then the calendar month values should be derived based on the calendar week values.

Step 4:
Read the required field from the E and F tables and merge the result set.

If the required field is a key figure (for example, find total sales for a set of materials and weeks) then the resultant values from E and F tables have to be added together.
If the required field is a characteristic (for example, find the customers that have purchased a set of materials in a set of weeks) then based on the DIMIDs/SIDs from the E and F fact tables you need to calculate the characteristic values.

In this article you will see example code to read the sales for a given set of materials and a given set of weeks. Following are the assumptions made for this example. The cube name is MYCUBE and the field name is MYFIELD. Material is a Line item dimension and it is the first dimension in the cube. The cube is physically partitioned by Calendar Month.

```
REPORT  ZCUBE_LOOKUP.

************************************************************************
* Variable declarations:
************************************************************************
data: AMT   TYPE  /BI0/OIAMOUNT,
      e_AMT TYPE  /BI0/OIAMOUNT,
      f_AMT TYPE  /BI0/OIAMOUNT.

data: begin of MATERIALS occurs 10.
data: material(18) type c.
data: end   of MATERIALS.

data: mat_sid            like /BI0/SMATERIAL-sid.
ranges: r_material_sids  for /BI0/SMATERIAL-sid.

data: begin of weeks occurs 10.
data: week like scal-week.
data: end   of weeks.

data: week_sid       type RSSID.
data: week_dimid     type RSDIMID.

ranges: r_week_dimid for /BIC/DMYCUBET-dimid.

ranges: r_calmonth_sid for /BI0/SCALMONTH-SID.

data: first_day like sy-datum.
data: last_day like sy-datum.

data: begin of months occurs 10.
data: month like /BI0/SCALMONTH-calmonth.
data: end   of months.


************************************************************************
*Step A:  Prepare test data.
************************************************************************
weeks-week = '200434'.
append weeks.
```

```
weeks-week = '200935'.
append weeks.

MATERIALS-material = '000000000001111111'.
append MATERIALS.

MATERIALS-material = '000000000002222222'.
append MATERIALS.

MATERIALS-material = '000000000003333333'.
append MATERIALS.

MATERIALS-material = '000000000004444444'.
append MATERIALS.

MATERIALS-material = '000000000005555555'.
append MATERIALS.


************************************************************************
*Step B:  Material is a part of a Line item dimension.
*         Identify the material SIDS.  You don't have DIMIDs.
************************************************************************
r_material_sids-sign = 'I'.
r_material_sids-option = 'EQ'.
loop at materials.
  select single sid
    into mat_sid
    from /BI0/SMATERIAL
   where material = materials.
  if sy-subrc = 0.
    r_material_sids-low = mat_sid.
    append r_material_sids.
  endif.
endloop.


************************************************************************
*Step C:  Time is not a Line item dimension.
*         Hence first identify SIDs and then DIMIDs for week.
*         Also identify Months corresponding to Weeks.
************************************************************************
r_week_dimid-sign = 'I'.
r_week_dimid-option = 'EQ'.

loop at weeks.
* Get first day of the week.
  CALL FUNCTION 'WEEK_GET_FIRST_DAY'
    EXPORTING
      WEEK               = weeks-week
    IMPORTING
      DATE               = first_day
    EXCEPTIONS
```

```
      WEEK_INVALID       = 1
      OTHERS             = 2.
  IF SY-SUBRC = 0.
    months-month = first_day+0(6).
    append months.
  ENDIF.

* Get last day of the week.
  last_day = first_day + 6.
  months-month = last_day+0(6).
  append months.

* Identify the Calweek SID from /BI0/SCALWEEK
  select single sid
    into week_sid
    from /BI0/SCALWEEK
   where CALWEEK = weeks-week.
  if sy-subrc = 0.
* Identify the Time dimension
    select dimid
      into week_dimid
      from /BIC/DMYCUBET
     where SID_0CALWEEK = week_sid.
      r_week_dimid-low = week_dimid.
      append r_week_dimid.
    endselect.
  endif.
endloop.

************************************************************************
*Step D:  Find the SIDs for months
************************************************************************
* Find the distinct list of months
sort months by month.
delete adjacent duplicates from months comparing month.

* Find the SIDs for months
r_calmonth_sid-sign = 'I'.
r_calmonth_sid-option = 'EQ'.
loop at months.
  select single sid
    into r_calmonth_sid-low
    from /BI0/SCALMONTH
   where calmonth = months-month.
  if sy-subrc = 0.
    append r_calmonth_sid.
  endif.
endloop.


************************************************************************
*Step E:  Add up Sales for all weeks and all materials
************************************************************************
```

```
if ( materials[] is not initial and
     r_material_sids[] is initial )
     or
   ( weeks[] is not initial and
     r_week_dimid[] is initial ).
* If ALL materials or ALL weeks you are looking for do not exist
* in the cube, then return zero.
  amt = 0.
else.
* Read data from the E table.  Since the E-table is partitioned
* by cal month, specify the filters on month in the WHERE clause.
  SELECT sum( /BIC/MYFIELD )
    into e_AMT
    FROM /BIC/EMYCUBE
   WHERE KEY_MYCUBET in r_week_dimid
     AND KEY_MYCUBE1 in r_material_sids
     and SID_0CALMONTH  in r_calmonth_sid.

* Read data from the F table.  Since the F-table is NOT partitioned
* by cal month, don't specify the filters on month in the WHERE clause.
  SELECT sum( /BIC/MYFIELD )
    into f_AMT
    FROM /BIC/FMYCUBE
   WHERE KEY_MYCUBET in r_week_dimid
     AND KEY_MYCUBE1 in r_material_sids.

  amt = e_AMT + f_AMT.
endif.

write:/ amt.
```

Disclaimer