

# SAP BI – Compression Status for Infocubes



## Applies to:

Consultants accustomed with SAP BI (3.5 / 7.0). For more information, visit the [EDW homepage](#).

## Summary

Document explains ABAP code to get the compression status for all the Infocubes in SAP BI system which is never compressed. It helps the consultants to decide when to compress the InfoCube data based on the Data volume.

**Author:** Suraj Tigga

**Company:** Capgemini Consulting India Pvt. Ltd.

**Created on:** 27 June 2011

## Author Bio



Suraj Tigga is a Senior SAP BI / ABAP consultant at Capgemini Consulting, India. Suraj joined Capgemini Consulting in 2008 and has worked on multiple SAP BI implementation and support projects.

## Table of Contents

Scenario.....	3
Step by Step Solution.....	3
ABAP Code (Compression Status – Cubes).....	3
Execution .....	5
ABAP Code.....	7
Related Content.....	13
Disclaimer and Liability Notice.....	14

## Scenario

Data loaded in InfoCube is identified with the request ID associated with them. However, the request ID concept can also cause the same data records to appear more than once in the fact table. This unnecessarily increases the volume data, and reduces performance in reporting, as the system has to aggregate using the request ID every time a query is executed. Compression of InfoCube would eliminate these disadvantages and bring data from different requests together into one single request.

Document explains the ABAP code which gives the Compression status details of Cubes in the SAP BI systems which are never compressed and have more than 50000 records.

## Step by Step Solution

### Advantages Compression:

- Compression transfers the data from the F fact table to the E fact table while eliminating the request information in the Infocube. It aggregates records with equal keys from different requests. Compression results in reducing the memory consumption.
- Compression improves performance tuning as it removes the redundant data.
- Accelerates loading into Fact table. Faster updating of the Fact table indexes. Accelerates aggregate rollups, since the Fact table is the source of data for rollup.

### Disadvantages Compression:

- Individual requests cannot be accessed and deleted any more after compression.
- If one deletes requests that have been rolled up and compressed in the aggregates, these aggregates have to be dropped and rebuilt completely. This would take a long time depending on the size of the aggregate and associated fact table.
- Compression is critical, as the compressed data can no longer be deleted from the InfoCube using its request IDs. One must be absolutely certain that the data loaded into the InfoCube is correct.

**ABAP Code (Compression Status – Cubes):** Explains the ABAP Code which gives a list of InfoCubes in the SAP BI systems which are never compressed and have more than 50000 records.

### ABAP Code (Compression Status – Cubes)

**Step 1:** Go to transaction SE38 and create ABAP Code 'YBI\_CUBE\_COMPRESSION\_STATUS'. Below is the selection screen code:

```

Report      YBI_CUBE_COMPRESSION_STATUS      Active
-----
50  ***** SELECTION SCREEN *****
51  [ ] SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001 .
52  | * Blocking Mode
53  [ ] SELECTION-SCREEN BEGIN OF LINE.
54  | PARAMETERS: p_all RADIOBUTTON GROUP gp1 MODIF ID op1.
55  | SELECTION-SCREEN COMMENT (30) text-002.
56  | PARAMETERS: p_sel RADIOBUTTON GROUP gp1 .
57  | SELECTION-SCREEN COMMENT 36(15) text-003.
58  | SELECT-OPTIONS : s_cube FOR rsdcube-infocube MODIF ID op2.
59  | SELECTION-SCREEN END OF LINE.
60  [ ] SELECTION-SCREEN END OF BLOCK b1.

```

## SAP BI Compression Status - Cube

Compression Status (Infocubes)

All Infocubes       Specific Cube

Compression status for all InfoCubes      Compression status for specific cube (Selection Screen)

**Step 2:** Use Function Module 'RSD\_CUBE\_MULTI\_GET\_ONLY\_DB' to get the details of the InfoCube:

## ABAP Editor: Display Include YINCL\_SUBROUTINE

```

18 IF p_all IS NOT INITIAL.
19   * Get list of all active infocubes in the system
20   CALL FUNCTION 'RSD_CUBE_MULTI_GET_ONLY_DB'
21     EXPORTING
22       i_read_all      = rs_c_true
23       i_objvers      = rs_c_objvers-active
24     IMPORTING
25       e_t_cube       = l_t_cube
26     EXCEPTIONS
27       illegal_input  = 1
28       OTHERS         = 2.
29
30 ELSE.
31   * Compression status for specific cubes
32   LOOP AT s_cube.
33     wa_cube-cube = s_cube-low.
34     APPEND wa_cube TO t_cube.
35     CLEAR wa_cube.
36   ENDOLOOP.
37   CALL FUNCTION 'RSD_CUBE_MULTI_GET_ONLY_DB'
38     EXPORTING
39       i_t_infocube    = t_cube
40       i_objvers      = rs_c_objvers-active
41     IMPORTING
42       e_t_cube       = l_t_cube
43     EXCEPTIONS
44       illegal_input  = 1
45       OTHERS         = 2.
46 ENDIF.
47

```

Get the details of the InfoCube.

**Step 3:** Use Function Module 'RSDU\_INFOCUBE\_TABLE\_SIZES' to get the table size, number of rows in fact table, exact Cube density.

```

Include      YINCL_SUBROUTINE      Active
51
52 LOOP AT l_t_cube INTO l_s_cube.
53 * get cube layout information
54 CALL FUNCTION 'RSDU_INFOCUBE_TABLE_SIZES'
55 EXPORTING
56     i_infocube = l_s_cube-infocube
57     i_check   = rs_c_true
58 IMPORTING
59     e_factrows = l_factrows
60     e_t_tabsize = l_t_tabsize
61     e_density  = l_density.
62
63 * give details only for non-empty cubes
64 IF l_factrows = 0. CONTINUE. ENDIF.
65
66 * count non-empty cubes
67 l_counter = l_counter + 1.
68 l_pdensity = l_density.
69
70 * Store cube table infos
71 LOOP AT l_t_tabsize INTO l_s_tabsize.
72 IF l_s_tabsize-tablnm+5(1) EQ 'F'.
73     wa_final-cube = l_s_cube-infocube.
74     wa_final-record = l_s_tabsize-rows.
75     wa_final-percent = l_s_tabsize-percent.
76     IF l_s_tabsize-percent EQ '100'.
77         wa_final-status = 'Cube never compressed'.
78     ELSEIF l_s_tabsize-percent EQ '0'.
79         wa_final-status = 'Cube is Completely compressed'.
80     ELSEIF l_s_tabsize-percent LT '100'.
81         wa_final-status = 'Cube is partially compressed'.
82     ENDIF.
83     APPEND wa_final TO t_final.
84     CLEAR wa_final.
85 ENDIF.
86 ENDOLOOP.
87 ENDOLOOP.
88

```

## Execution

### Scenario 1: Execution for all Cubes:

**Step 1:** Execute the ABAP Code with option 'All Cubes':

### SAP BI Compression Status - Cube

Compression Status (Infocubes)

All Infocubes       Specific Cube       to

**Step 2:** Compression Status for all InfoCube is displayed in the ALV list:

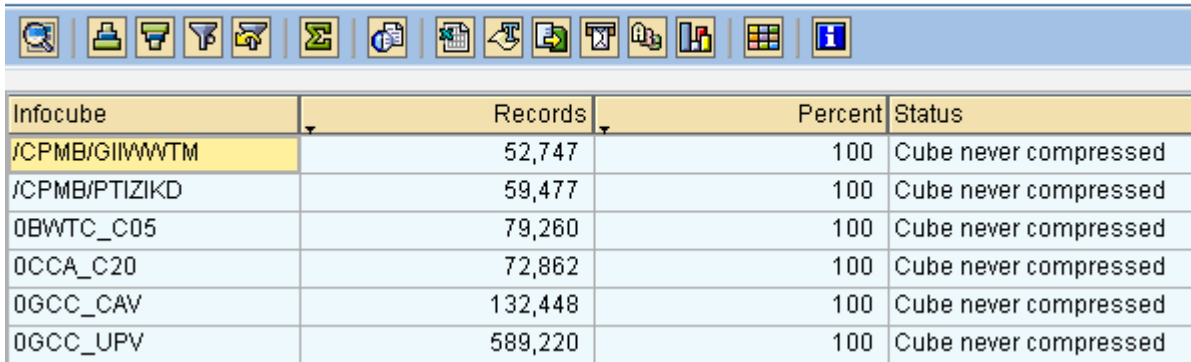
### SAP BI Compression Status - Cube

Infocube	Records	Percent	Status
/CPMB/ATIKOOY	864	100	Cube never compressed
/CPMB/ATIRCJO	384	100	Cube never compressed
/CPMB/DPIBURX	384	100	Cube never compressed
/CPMB/DPID6HA	864	100	Cube never compressed
/CPMB/E2IH2JH	43	100	Cube never compressed
/CPMB/E2IHJYZ	1,233	100	Cube never compressed
/CPMB/E2I185Q	1,106	100	Cube never compressed

Above list can be downloaded to excel sheet.

Filter the ALV output for records GT '50000' and percent EQ '100':

## SAP BI Compression Status - Cube



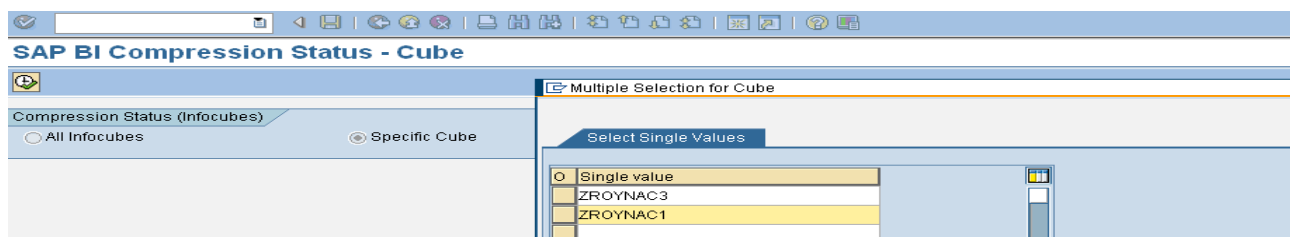
Infocube	Records	Percent	Status
/CPMB/GIIVWWTM	52,747	100	Cube never compressed
/CPMB/PTIZIKD	59,477	100	Cube never compressed
0BWTC_C05	79,260	100	Cube never compressed
0CCA_C20	72,862	100	Cube never compressed
0GCC_CAV	132,448	100	Cube never compressed
0GCC_UPV	589,220	100	Cube never compressed

Above InfoCube should be compressed.

Percent	Status
Percent = 100	Cube never Compressed
Percent = 0	Cube is Completely Compressed
Percent > 0 & Percent < 100	Cube is Partially Compressed

### Scenario 2: Execution for Specific Cube:

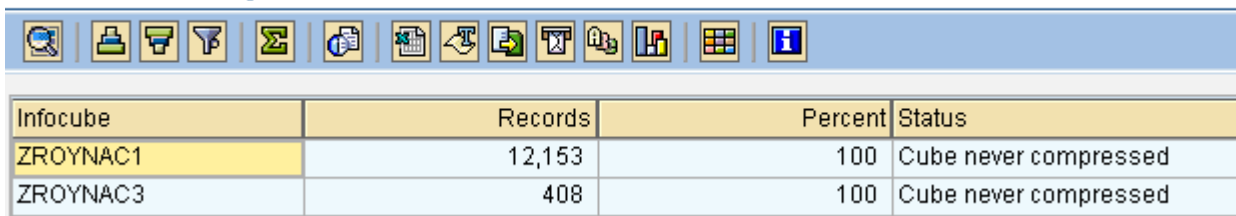
Step 1: Execute the Code for Specific Cubes:



The screenshot shows the SAP BI Compression Status - Cube interface. The 'Compression Status (Infocubes)' section has 'Specific Cube' selected. A dialog box titled 'Multiple Selection for Cube' is open, showing a list of cubes: ZROYNAC3 and ZROYNAC1.

Report Output:

## SAP BI Compression Status - Cube



Infocube	Records	Percent	Status
ZROYNAC1	12,153	100	Cube never compressed
ZROYNAC3	408	100	Cube never compressed

## ABAP Code

**Report: YBI\_CUBE\_COMPRESSION\_STATUS**

```

*&-----*
*& Report  YBI_CUBE_COMPRESSION_STATUS
*&
*&-----*
*& Get the Compression Status for the Infocubes (Selected) and all the
*& cubes present in the system
*&-----*

```

```
REPORT  ybi_cube_compression_status.
```

```
TABLES: rsdcube.
```

```
* TYPE POOLS
```

```
TYPE-POOLS: slis ,
             sscr ,
             rsd, rsdu.
```

```
***** DATA DECLARATION *****
```

```
DATA:
```

```
* Selection screen data for restriction
  wa_restrict      TYPE sscr_restrict ,
* Auxiliary objects for filling RESTRICT
  wa_optlist       TYPE sscr_opt_list,
  wa_sscr          TYPE sscr_ass.
```

```
DATA: l_t_cube     TYPE rsd_t_cube,
      l_s_cube     TYPE rsd_s_cube,
      l_factrows   TYPE i,
      l_s_tabsize  TYPE rsdu_s_tabsize,
      l_t_tabsize  TYPE rsdu_t_tabsize,
      l_density    TYPE f,
      l_pdensity   TYPE p DECIMALS 1,
      l_counter    TYPE i,
      l_odd        TYPE i.
```

```
TYPES: BEGIN OF typ_final ,
       cube     TYPE rsd_tablnm ,
       record   TYPE i ,
       percent  TYPE i ,
       status(60) TYPE c ,
       END OF typ_final.
```

```
DATA : wa_final TYPE typ_final ,
      t_final  TYPE STANDARD TABLE OF typ_final.
```

```
DATA : t_fieldcat_final TYPE slis_t_fieldcat_alv ,
      wa_layout_final   TYPE slis_layout_alv ,
      t_event_final     TYPE slis_t_event .
```

```
***** SELECTION SCREEN *****
```

```
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001 .
```

```
* Blocking Mode
```

```

SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS: p_all RADIOBUTTON GROUP gp1 MODIF ID op1.
SELECTION-SCREEN COMMENT (30) text-002.
PARAMETERS: p_sel RADIOBUTTON GROUP gp1 .
SELECTION-SCREEN COMMENT 36(15) text-003.
SELECT-OPTIONS : s_cube FOR rsdcube-infocube MODIF ID op2.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN END OF BLOCK b1.

***** INCLUDE (For Subroutines)*****
INCLUDE yincl_subroutine.

***** AT SELECTION SCREEN OUPUT *****
AT SELECTION-SCREEN OUTPUT.
  LOOP AT SCREEN.
  * Customer Blocking type is selected
    IF p_all EQ 'X' AND screen-group1 EQ 'OP2'.
      screen-input = '0'.
      MODIFY SCREEN.
    ENDIF.
  ENDLLOOP.

***** INITIALISATION *****
INITIALIZATION.
* Restricting the Customer selection to only EQ
wa_optlist-name = 'OBJECTKEY1'.
wa_optlist-options-eq = 'X' .
APPEND wa_optlist TO wa_restrict-opt_list_tab.
wa_sccr-kind = 'S'.
wa_sccr-name = 'S_CUBE'.
wa_sccr-sg_main = 'I'.
wa_sccr-sg_addy = space.
wa_sccr-op_main = 'OBJECTKEY1'.
APPEND wa_sccr TO wa_restrict-ass_tab.

* Call the function module
CALL FUNCTION 'SELECT_OPTIONS_RESTRICT'
  EXPORTING
    program          = sy-repid
    restriction      = wa_restrict
  *   DB              = ' '
  EXCEPTIONS
    too_late         = 1
    repeated         = 2
    selopt_without_options = 3
    selopt_without_signs = 4
    invalid_sign     = 5
    empty_option_list = 6
    invalid_kind     = 7
    repeated_kind_a  = 8
    OTHERS           = 9
.

***** START OF SELECTION *****
START-OF-SELECTION.
* Get the Cube Compression Status for all the Cubes Present in the system
PERFORM sub_get_all_compression.

```



```

***** END-OF-SELECTION *****
END-OF-SELECTION.
* Display Output
PERFORM sub_display.

```

### Include: YINCL\_SUBROUTINE

```

*&-----*
*& Include          YINCL_SUBROUTINE
*&-----*
*&-----*
*& Form SUB_GET_ALL_COMPRESSION
*&-----*
*      Get all the compression Cube details
*-----*
FORM sub_get_all_compression .

TYPES : BEGIN OF typ_cube ,
        cube TYPE rs_char30,
        END OF typ_cube.

DATA : wa_cube TYPE typ_cube ,
        t_cube  TYPE STANDARD TABLE OF typ_cube.

IF p_all IS NOT INITIAL.
* Get list of all active infocubes in the system
CALL FUNCTION 'RSD_CUBE_MULTI_GET_ONLY_DB'
  EXPORTING
    i_read_all      = rs_c_true
    i_objvers       = rs_c_objvers-active
  IMPORTING
    e_t_cube       = l_t_cube
  EXCEPTIONS
    illegal_input  = 1
    OTHERS         = 2.

ELSE.
* Compression status for specific cubes
LOOP AT s_cube.
  wa_cube-cube = s_cube-low.
  APPEND wa_cube TO t_cube.
  CLEAR wa_cube.
ENDLOOP.
CALL FUNCTION 'RSD_CUBE_MULTI_GET_ONLY_DB'
  EXPORTING
    i_t_infocube   = t_cube
    i_objvers       = rs_c_objvers-active
  IMPORTING
    e_t_cube       = l_t_cube
  EXCEPTIONS
    illegal_input  = 1
    OTHERS         = 2.
ENDIF.

```

```

* loop over the cubes and get information
l_counter = 0.

LOOP AT l_t_cube INTO l_s_cube.
* get cube layout information
CALL FUNCTION 'RSDU_INFOCUBE_TABLE_SIZES'
  EXPORTING
    i_infocube = l_s_cube-infocube
    i_check    = rs_c_true
  IMPORTING
    e_factrows = l_factrows
    e_t_tablsize = l_t_tablsize
    e_density  = l_density.

* give details only for non-empty cubes
IF l_factrows = 0. CONTINUE. ENDIF.

* count non-empty cubes
l_counter = l_counter + 1.
l_pdensity = l_density.

* Store cube table infos
LOOP AT l_t_tablsize INTO l_s_tablsize.
IF l_s_tablsize-tablnm+5(1) EQ 'F'.
  wa_final-cube = l_s_cube-infocube.
  wa_final-record = l_s_tablsize-rows.
  wa_final-percent = l_s_tablsize-percent.
  IF l_s_tablsize-percent EQ '100'.
    wa_final-status = 'Cube never compressed'.
  ELSEIF l_s_tablsize-percent EQ '0'.
    wa_final-status = 'Cube is Completely compressed'.
  ELSEIF l_s_tablsize-percent LT '100'.
    wa_final-status = 'Cube is partially compressed'.
  ENDIF.
  APPEND wa_final TO t_final.
  CLEAR wa_final.
ENDIF.
ENDLOOP.
ENDLOOP.

ENDFORM.                    " SUB_GET_ALL_COMPRESSION
*&-----*
*&      Form  SUB_DISPLAY
*&-----*
*      Display Records
*-----*

FORM sub_display .
* Intialising the ALV
PERFORM sub_init_alv.
* Fieldcatalogue
PERFORM sub_fieldcat.
* Event
PERFORM sub_event.
* Call ALV FM

```

```

PERFORM sub_alv_append.
ENDFORM.          " SUB_DISPLAY
*&-----*
*&      Form  SUB_INIT_ALV
*&-----*
*      Initialise
*-----*
FORM sub_init_alv .
* ALV initialisation
  CALL FUNCTION 'REUSE_ALV_BLOCK_LIST_INIT'
    EXPORTING
      i_callback_program = sy-repid.
ENDFORM.          " SUB_INIT_ALV
*&-----*
*&      Form  SUB_FIELDCAT
*&-----*
*      Fieldcatalogue
*-----*
FORM sub_fieldcat .
* Definition of Fieldcatalogue
  DATA : wl_fieldcat_final TYPE slis_fieldcat_alv.
* Clearing work area of fieldcatalogue
  CLEAR wl_fieldcat_final.
* Plant
  wl_fieldcat_final-col_pos = '1'.
  wl_fieldcat_final-fieldname = 'CUBE'.
  wl_fieldcat_final-reptext_ddic = text-005.
  wl_fieldcat_final-outputlen = '20'.
  APPEND wl_fieldcat_final TO t_fieldcat_final.

* Clearing work area of fieldcatalogue
  CLEAR wl_fieldcat_final.
* Plant
  wl_fieldcat_final-col_pos = '2'.
  wl_fieldcat_final-fieldname = 'RECORD'.
  wl_fieldcat_final-reptext_ddic = text-006.
  wl_fieldcat_final-outputlen = '20'.
  APPEND wl_fieldcat_final TO t_fieldcat_final.

* Clearing work area of fieldcatalogue
  CLEAR wl_fieldcat_final.
* Plant
  wl_fieldcat_final-col_pos = '3'.
  wl_fieldcat_final-fieldname = 'PERCENT'.
  wl_fieldcat_final-reptext_ddic = text-007.
  wl_fieldcat_final-outputlen = '20'.
  APPEND wl_fieldcat_final TO t_fieldcat_final.

* Clearing work area of fieldcatalogue
  CLEAR wl_fieldcat_final.
* Plant
  wl_fieldcat_final-col_pos = '4'.
  wl_fieldcat_final-fieldname = 'STATUS'.
  wl_fieldcat_final-reptext_ddic = text-008.
  wl_fieldcat_final-outputlen = '60'.

```

```

APPEND wl_fieldcat_final TO t_fieldcat_final.

ENDFORM.                " SUB_FIELDCAT
*&-----*
*&      Form  SUB_EVENT
*&-----*
*      TOP OF PAGE
*-----*
FORM sub_event .

* Work area declaratoin for EVENT table
DATA: wl_event_final LIKE LINE OF t_event_final.
* TOP OF PAGE 1 population
  wl_event_final-form = 'TOP_OF_PAGE1'.
  wl_event_final-name = slis_ev_top_of_list.
  APPEND wl_event_final TO t_event_final.

ENDFORM.                " SUB_EVENT
FORM top_of_page1.
  WRITE: 'Compression Status'.
ENDFORM.
*&-----*
*&      Form  SUB_ALV_APPEND
*&-----*
*      ALV Display
*-----*
FORM sub_alv_append .

* Populate the Layout table
  wa_layout_final-box_tabname = 'T_FINAL'.

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
  EXPORTING
    is_layout           = wa_layout_final
    it_fieldcat         = t_fieldcat_final
    it_events           = t_event_final
  TABLES
    t_outtab           = t_final
  EXCEPTIONS
    program_error      = 1
    maximum_of_appends_reached = 2
    OTHERS              = 3.

ENDFORM.                " SUB_ALV_APPEND

```

## Related Content

For more information, visit the [EDW 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.