

# SAP BW Process Chain - Customizing PSA Deletion and Cube Compression



## Applies to:

SAP BI 2004s or SAP BI 7.x. For more information, visit the [EDW homepage](#).

## Summary

This document explains the customization of PSA Deletion and Cube Compression process elements in BW Process Chain.

**Author:** Balajee Sivakumar

**Created on:** 31 January 2011

## Author Bio

Balajee Sivakumar has more than 5 Years of experience in SAP BW/BI. He has done end to end design covering the SAP BW and SAP BW Integrated Planning. He was involved in SAP BW and BW - BO End to end scenario creation for the new features being as a part of the SAP BW Research Team (Universe Designer, Xcelsius, Data Federator, Data Services and BO Explorer) in SAP Labs.

## Table of Contents

The Scenario .....	3
PSA Deletion: .....	3
Cube Compression: .....	3
Creating the Process Chain .....	4
Creating the ABAP Report .....	5
Writing the Code .....	5
Explanation of the code .....	8
Executing the Report and Changes in the Process Element .....	10
Enhancements that can be done .....	11
Related contents .....	12
Disclaimer and Liability Notice .....	13

## The Scenario

PSA delete and Cube Compression processes are widely used in any maintenance process chain as the data in the data target keeps growing on a long run and there need to be a methodology in place to handle this.

### PSA Deletion:

With this function you can delete requests from the PSA. This reduces the volume of data in the PSA. It is a good idea to delete incorrect requests or deltas for a data target, to which you do not want to load any more delta.

### Cube Compression:

When you load data into the Info Cube, entire requests can be inserted at the same time. Each of these requests has its own request ID, which is included in the fact table in the packet dimension. This makes it possible to pay particular attention to individual requests. One advantage of the request ID concept is that you can subsequently delete complete requests from the Info Cube.

However, the request ID concept can also cause the same data record (all characteristics agree, with the exception of the request ID) to appear more than once in the fact table. This unnecessarily increases the volume of data, and reduces performance in reporting, as the system has to perform aggregation using the request ID every time you execute a query.

Using compressing, you can eliminate these disadvantages, and bring data from different requests together into one single request

I faced a requirement where in the end user wanted to change the "Number of days older" option in the PSA delete and Cube Compression processes in all their process chains used for maintenance.

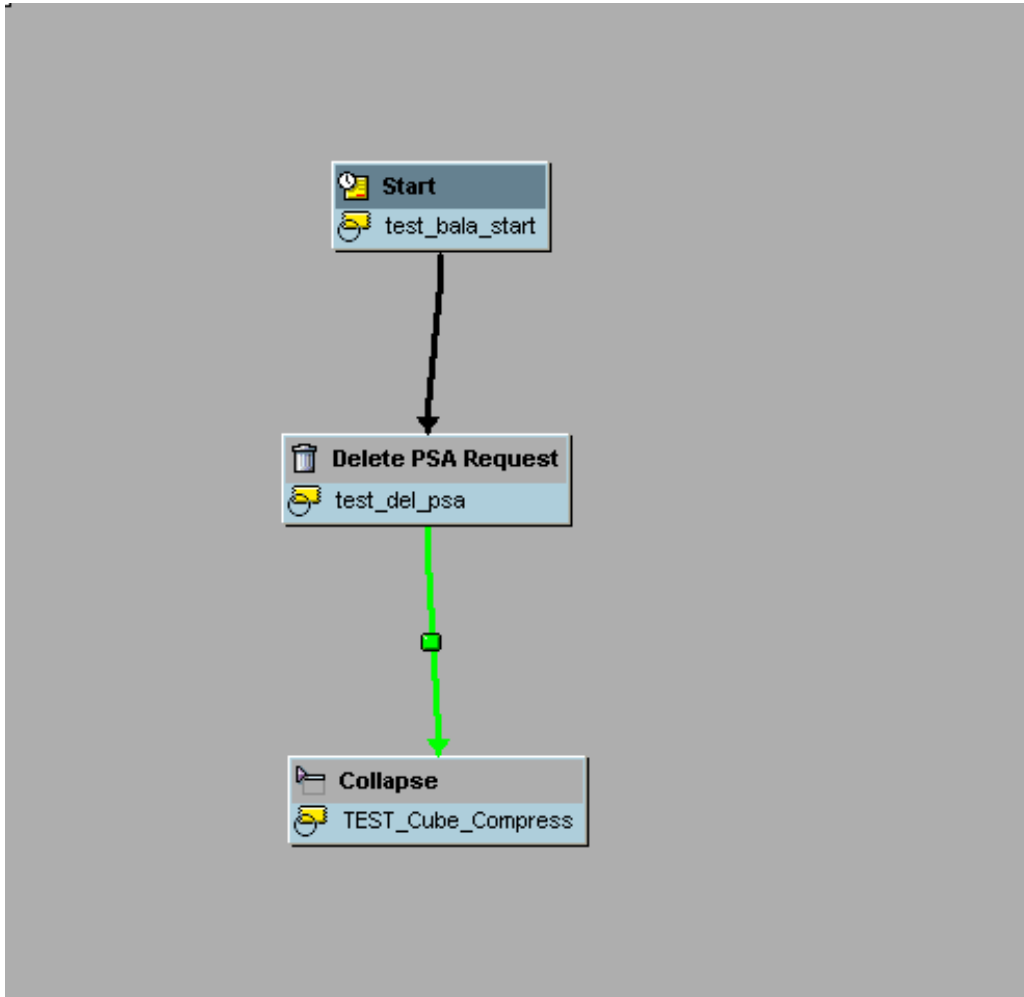
The actual methodology is to go to the Process Chain transaction RSPC and then edit the corresponding process elements and then reactivate the process chain.

This document covers an alternative approach by executing an ABAP report to change the 'Days' value without even editing the Process chain.

## Creating the Process Chain

Go to the transaction code: RSPC and create a demo process chain with two following process elements as follows,

- Deletion of Requests from PSA
- Compression of the InfoCube



The names of the process variants are "test\_del\_psa" and "test\_cube\_compress" respectively as shown in the screenshot above.

## Creating the ABAP Report

Go to transaction SE38 and try to create a new ABAP report. Select the options while creating new report, as mentioned in the screenshot below

Title	Test Program for Demo		
Original language	EN	English	
Created	11/15/2010	BSIVAKUMAR03	
Last changed by	11/26/2010	BSIVAKUMAR03	
Status	Active		

Attributes	
Type	Executable program
Status	Test Program
Application	Business Information Warehouse
Authorization Group	
Package	\$TMP Temporary Objects (never transported!)
Logical database	
Selection screen	
<input type="checkbox"/> Editor lock	<input checked="" type="checkbox"/> Fixed point arithmetic
<input checked="" type="checkbox"/> Unicode checks active	<input type="checkbox"/> Start using variant

Save

## Writing the Code

Paste the following code for the ABAP report

```

*&-----*
*& Report  ZTEST
*&
*&-----*
*&
*&
*&-----*

REPORT  ZTEST.

PARAMETERS:

        Del_PSA TYPE c length 50,
        Cmp_cube TYPE c length 50.

write Del_PSA.

```

```

write Cmp_cube.

PERFORM CUBECOMPRESS.

PERFORM PSADELETE.

*****PSADELETE*****
FORM PSADELETE.

data:

l_t_variant type table of rspcvariant,
l_s_variant TYPE rspcvariant,
l_s1_variant TYPE rspcvariant,
l_s2_variant type rspc_variant,
l_type TYPE rspc_type,
l_variant TYPE rspc_variant,
l_t_variantt Type rspc_t_variantt,
l_s_variantt like line of l_t_variantt,
l_r_variant TYPE REF TO cl_rspc_variant.

l_type = 'PSADELETE'.
l_variant = 'TEST_DEL_PSA'.

SELECT * FROM rspcvariant into table l_t_variant WHERE type      = 'PSADELETE'
              AND variante = 'TEST_DEL_PSA'
              AND objvers  = rs_c_objvers-active.

loop at l_t_variant into l_s_variant.

if l_s_variant-FNAM = 'DEL_OLDER_DAYS'.

l_s_variant-low = Del_PSA.

modify l_t_variant from l_s_variant.

endif.

endloop.

CALL METHOD cl_rspc_variant=>create
EXPORTING
    i_type      = l_type
    i_variant   = l_variant
    i_objvers   = rs_c_objvers-active
    i_no_transport = rs_c_true
    i_lock      = rs_c_true
RECEIVING
    r_r_variant = l_r_variant
EXCEPTIONS
    locked      = 1
    OTHERS      = 2.

CALL METHOD l_r_variant->save

```

```

EXPORTING
  i_t_rspcvariant = l_t_variant
EXCEPTIONS
  failed          = 1
  OTHERS         = 2.

CALL METHOD l_r_variant->free.

ENDFORM.
*****END***PSADELETE*****
*
*****CUBECOMPRESS*****

form CUBECOMPRESS.

data:

l_t_var type table of rspcvariant,
l_s_var TYPE rspcvariant,
l_s1_var TYPE rspcvariant,
l_s2_var type rspc_variant,
l_type1 TYPE rspc_type,
l_var TYPE rspc_variant,
*l_t_variantt Type rspc_t_variantt,
*l_s_variantt like line of l_t_variantt,
l_r_var  TYPE REF TO cl_rspc_variant.

l_type1 = 'COMPRESS'.
l_var = 'TEST_CUBE_COMPRESS'.

SELECT * FROM rspcvariant into table l_t_var WHERE type      = 'COMPRESS'
              AND variante = 'TEST_CUBE_COMPRESS'
              AND objvers  = rs_c_objvers-active.

loop at l_t_var into l_s_var.

if l_s_var-FNAM = 'DAYS_RNR_NOT_PROC'.

l_s_var-low = Cmp_cube.

modify l_t_var from l_s_var.

endif.

endloop.

CALL METHOD cl_rspc_variant=>create
EXPORTING
  i_type      = l_type1
  i_variant   = l_var
  i_objvers   = rs_c_objvers-active

```

```

        i_no_transport = rs_c_true
        i_lock          = rs_c_true
RECEIVING
        r_r_variant    = l_r_var
EXCEPTIONS
        locked         = 1
        OTHERS         = 2.

CALL METHOD l_r_var->save
EXPORTING
        i_t_rspcvariant = l_t_var
EXCEPTIONS
        failed          = 1
        OTHERS         = 2.

CALL METHOD l_r_var->free.

ENDFORM.
*****END*****CUBECOMPRESS*****
*****

```

## Explanation of the code

PARAMETERS:

```

        Del_PSA TYPE c length 50,
        Cmp_cube TYPE c length 50.

```

Here we are trying to get the values for Number of days for PSA delete and Compress cube process elements of our process chain.

```
PERFORM CUBECOMPRESS.
```

```
PERFORM PSADELETE.
```

There are two routines written inside the report for cube compress and PSA delete separately.

```

l_type = 'PSADELETE'.
l_variant = 'TEST_DEL_PSA'.

SELECT * FROM rspcvariant into table l_t_variant WHERE type      = 'PSADELETE'
                AND variante = 'TEST_DEL_PSA'
                AND objvers  = rs_c_objvers-active.

loop at l_t_variant into l_s_variant.
if l_s_variant-FNAM = 'DEL_OLDER_DAYS'.
l_s_variant-low = Del_PSA.
modify l_t_variant from l_s_variant.

endif.
endloop.

```

In PSA Deletion routine, we are first getting the existing values set for our variant 'TEST\_DEL\_PSA'. And after that we are updating the values with the user entered days value (DEL\_PSA). In this case, the process



variant name is hard coded but the code can be further enhanced and the variant name can also be got from the user directly.

```
CALL METHOD cl_rspc_variant=>create
  EXPORTING
    i_type          = l_type
    i_variant       = l_variant
    i_objvers       = rs_c_objvers-active
    i_no_transport  = rs_c_true
    i_lock          = rs_c_true
  RECEIVING
    r_r_variant     = l_r_variant
  EXCEPTIONS
    locked          = 1
    OTHERS         = 2.
```

```
CALL METHOD l_r_variant->save
  EXPORTING
    i_t_rspcvariant = l_t_variant
  EXCEPTIONS
    failed          = 1
    OTHERS         = 2.
```

```
CALL METHOD l_r_variant->free.
```

After this, we are creating and saving the variant again to commit out changes to the backend.

At the end, we are unlocking the process variant and bringing it back to display mode from change mode.

```
l_type1 = 'COMPRESS'.
l_var = 'TEST_CUBE_COMPRESS'.

SELECT * FROM rspcvariant into table l_t_var WHERE type      = 'COMPRESS'
              AND variante = 'TEST_CUBE_COMPRESS'
              AND objvers  = rs_c_objvers-active.

loop at l_t_var into l_s_var.

if l_s_var-FNAM = 'DAYS_RNR_NOT_PROC'.

l_s_var-low = Cmp_cube.

modify l_t_var from l_s_var.

endif.

endloop.
```

In Cube Compress routine, we are first getting the existing values set for our variant 'TEST\_CUBE\_COMPRESS'. And after that we are updating the values with the user entered days value (COMPRESS).

```
CALL METHOD cl_rspc_variant=>create
  EXPORTING
    i_type          = l_type1
    i_variant       = l_var
    i_objvers       = rs_c_objvers-active
    i_no_transport  = rs_c_true
```

```

        i_lock          = rs_c_true
RECEIVING
        r_r_variant    = l_r_var
EXCEPTIONS
        locked         = 1
        OTHERS        = 2.

CALL METHOD l_r_var->save
EXPORTING
        i_t_rspcvariant = l_t_var

EXCEPTIONS
        failed         = 1
        OTHERS        = 2.

CALL METHOD l_r_var->free.

```

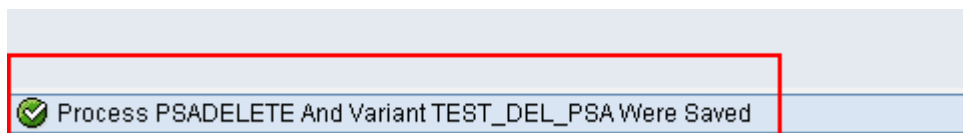
After this, we are creating and saving the variant again to commit out changes to the backend.  
At the end, we are unlocking the process variant and bringing it back to display mode from change mode.

### Executing the Report and Changes in the Process Element

Execute the report "ZTEST" by providing the following values,

DEL_PSA	14
CMP_CUBE	14

After giving the values and executing the Report, the following message is displayed,



So from this message it is clear that both the required process variants are saved properly with the new 'No: of days' value.

Now if the user checks the process variants in the sample process chain, we can observe that the "No:of days" value is changed appropriately,

Process Variant " TEST\_DEL\_PSA" - Delete PSA



## Related contents

[Business Intelligence: Overview](#)

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.