PSA Data Maintenance in SAP BI – Methodology and Techniques

Applies to:

SAP BI 7. For more information, visit the Business Intelligence homepage.

Summary

This article presents suggestions for how and when to delete data periodically from PSA. More importantly, this article shows how to find PSAs that are not being deleted periodically.

Author: Sudhi Karkada

Company: Contractor at Reliant Energy, Inc.

Created on: 03 December 2009

Author Bio

Sudhi Karkada has been an SAP ABAP developer since 1997 and a BW/BI developer since 2002. Specializes in back-end development, performance improvement, custom extractors, and troubleshooting.

Table of Contents

Introduction

In most cases, data that is staged in PSA should be deleted periodically. It helps to delete data in an organized way that will be easy to manage and maintain. How often the data should be deleted and what methods can be employed for the deletion of data will be discussed in this article. An ABAP program is presented that will help in identifying the PSAs whose data is not being deleted periodically.

the data should be deleted from a PSA primarily depends on how often the PSA is loaded.

- Bullet 1
- Bullet 2

Benefits of managing PSA data deletion

1. Recover database space
2. Schedule deletion in such a way as not to interfere with daily data loads and user activity
3. Be able to keep the data stored for a reasonable period of time
4. Easy to monitor, maintain and document

PSA Data Retention Policy

It is a good idea to develop and maintain a document describing the data retention policy for your own organization. Frequency of PSA data deletion should primarily depend upon how often the data is loaded. There are a number of recommendations, some even proposed by SAP. Search on SCN. The idea is to split all PSAs into four or five categories; each category preserving data for certain number of days. For example, consider full daily loads that refresh the data target every time. There is no need to preserve this PSA data for more than a day. For daily delta loads, you might want to preserve PSA data for a few days. Monthly loads may need to preserve PSA data for a month. Yes, every case should be examined and decided upon based on the importance of the data and difficulty of retrieving the exact same data.

PSA Data deletion scheduling

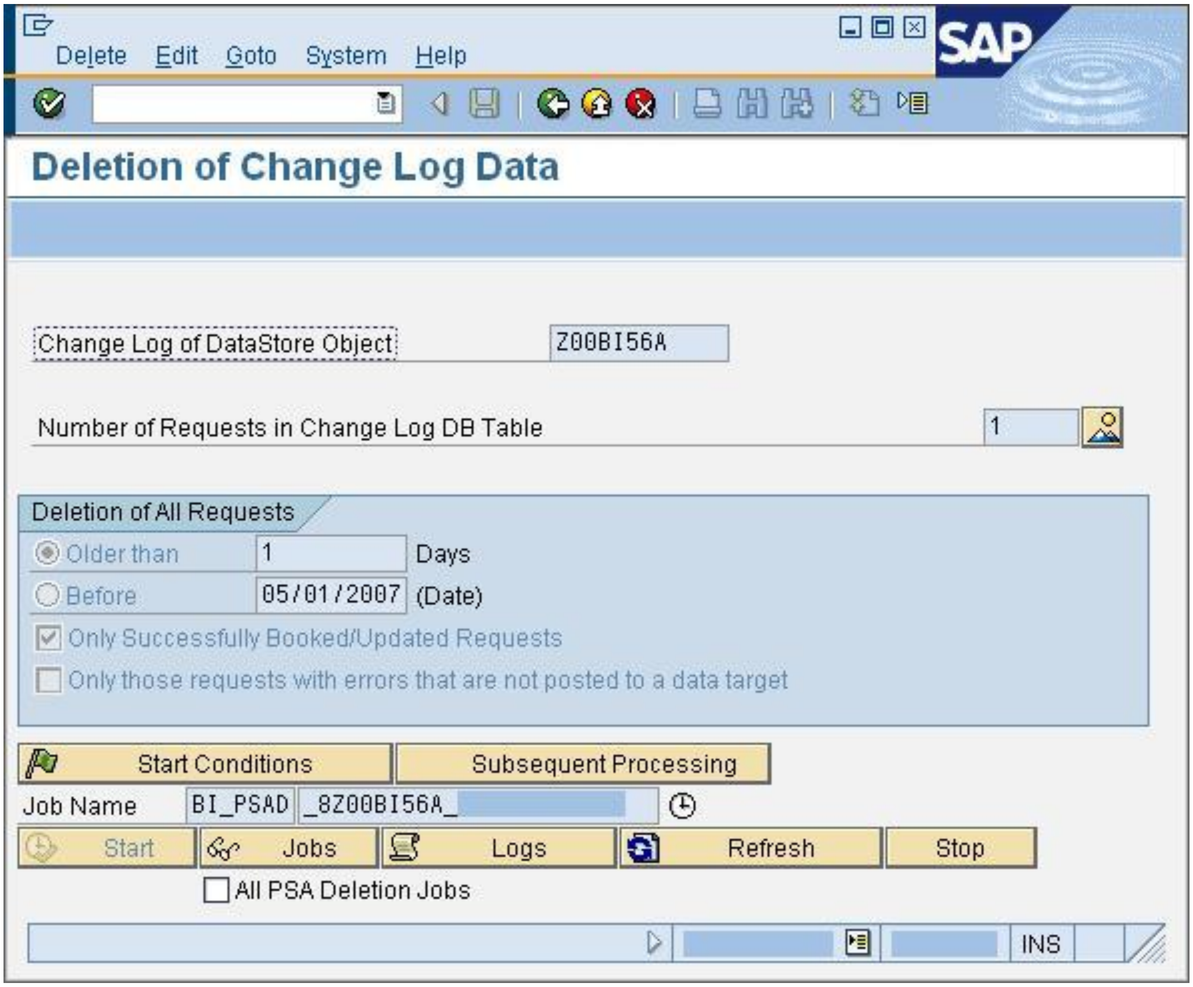
Ideal time to perform PSA data deletion is when data loads are not running and user activity is minimal. However, finding such window will become increasingly difficult as the BI installation matures. If that is your case, you may begin PSA deletion after the daily loads are complete. You may schedule PSAs with huge amounts of data for weekends.

Regardless of when data deletion is scheduled, you have to determine how they are scheduled. You don't want to burden the database with multiple large PSA deletions simultaneously. Process chains help you accomplish this. Create multiple PSA deletion processes. One will contain a list of PSAs with large amounts of data and another with a list of remaining PSAs. Then you can run them in parallel within the same process chain.

Methods of deleting PSA data

Scheduling Direct Deletion Jobs

This method involves deleting PSA data directly without using process chains. You access PSA section in the workbench (RSA1OLD in BI 7), search for your PSA, and choose "Delete PSA Data…" or "Delete Change Log Data…" from the context menu. In the resulting screen, you specify how long to keep and the job name for scheduling the job.



Deletion of Change Log Data

Change Log of DataStore Object: Z00BI56A

Number of Requests in Change Log DB Table: 1

Deletion of All Requests

● Older than 1 Days

○ Before 05/01/2007 (Date)

☑ Only Successfully Booked/Updated Requests

□ Only those requests with errors that are not posted to a data target

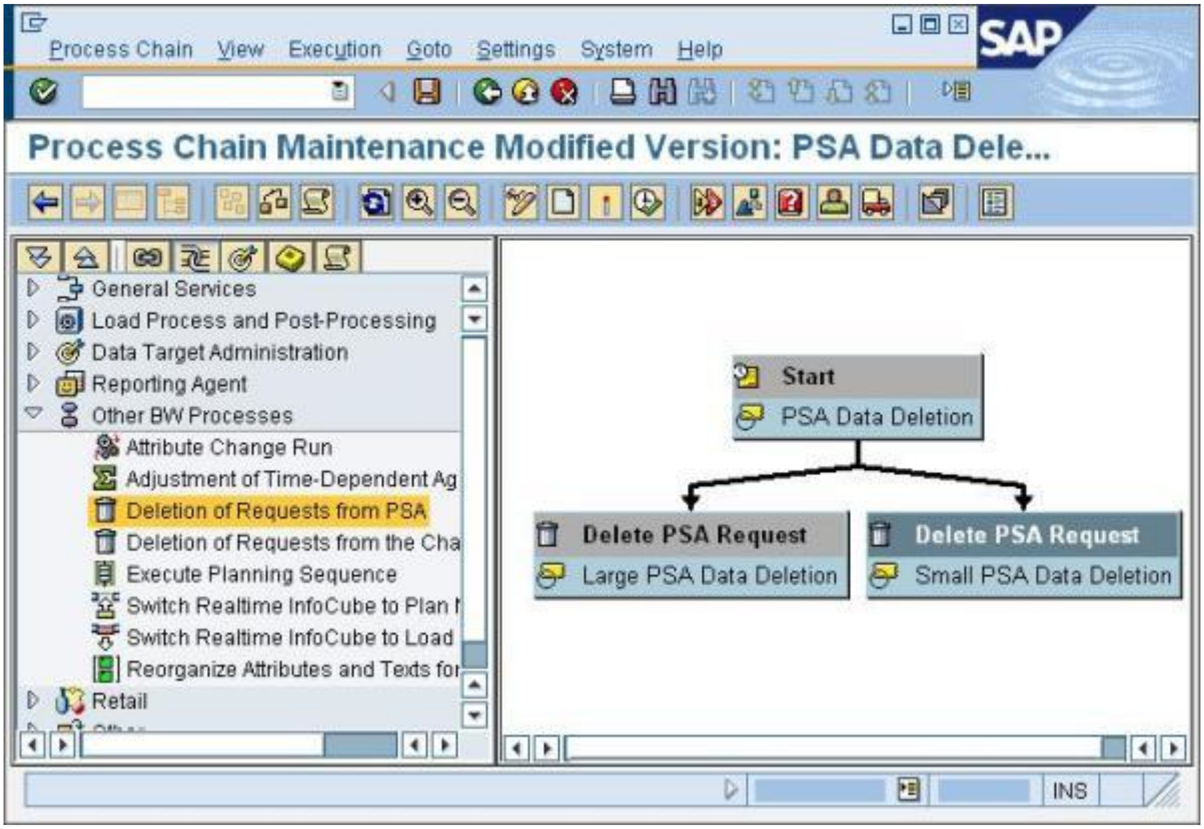
Job Name: BI_PSAD _8Z00BI56A_

□ All PSA Deletion Jobs

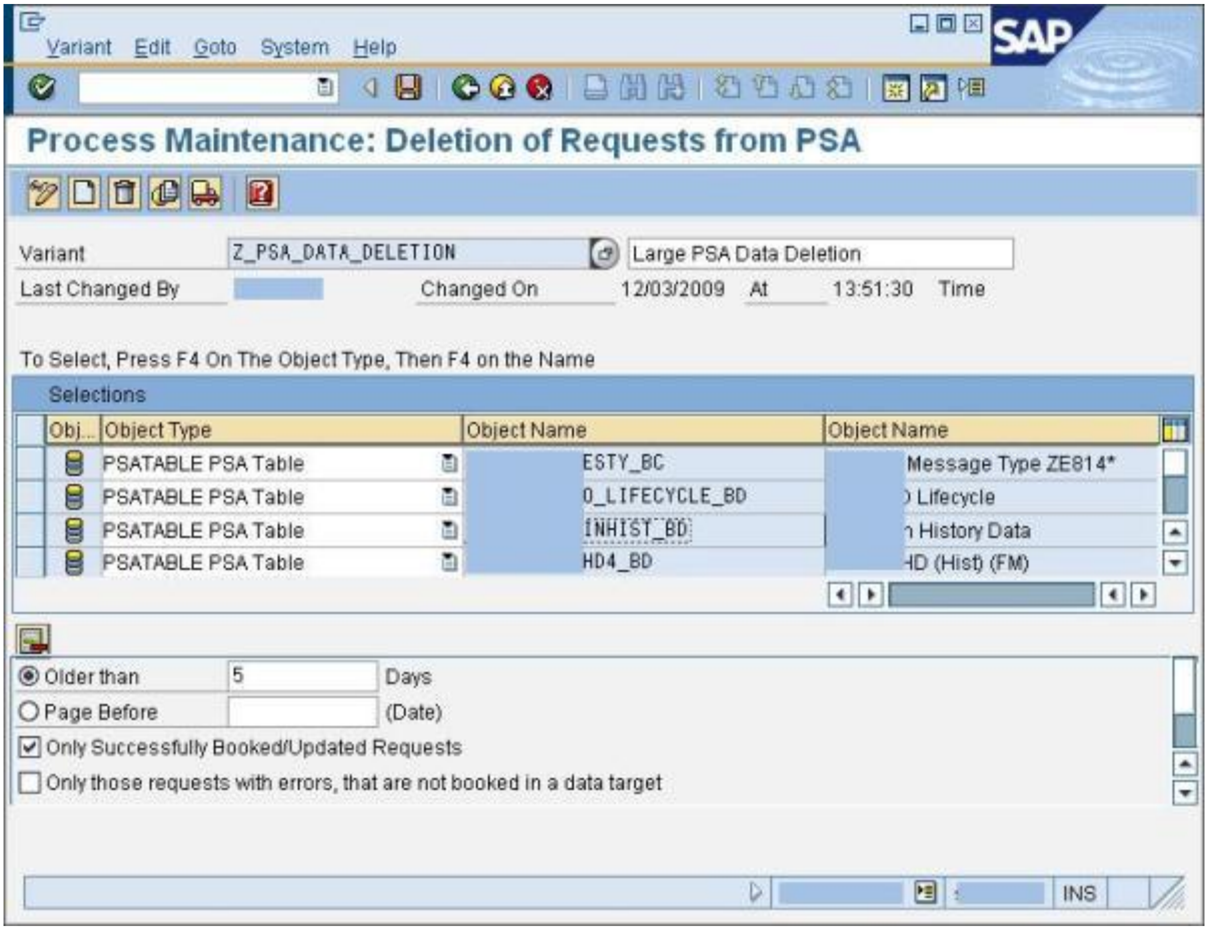
This method of deleting PSA data is hard to monitor and manage. It is not easy to make sure that deletion jobs don’t overlap.

Scheduling PSA data deletion via process chains

It is not only recommended, but also very convenient to schedule PSA data deletion using process chains. You can choose which PSAs should be scheduled in sequence and which can be scheduled in parallel.


Process Chain
View
Execution
Goto
Settings
System
Help
SAP
Process Chain Maintenance Modified Version: PSA Data Dele...
General Services
Load Process and Post-Processing
Data Target Administration
Reporting Agent
Other BW Processes
Attribute Change Run
Adjustment of Time-Dependent Ag
Deletion of Requests from PSA
Deletion of Requests from the Cha
Execute Planning Sequence
Switch Realtime InfoCube to Plan I
Switch Realtime InfoCube to Load
Reorganize Attributes and Texts for
Retail
Start
PSA Data Deletion
Delete PSA Request
Large PSA Data Deletion
Delete PSA Request
Small PSA Data Deletion
INS

PSAs listed in the process below will be processed one at a time in a sequential manner, but with just one background job.



Process Maintenance: Deletion of Requests from PSA

Variant	Z_PSA_DATA_DELETION	Large PSA Data Deletion			
Last Changed By	[redacted]	Changed On	12/03/2009	At	13:51:30 Time

To Select, Press F4 On The Object Type, Then F4 on the Name

Selections

Obj...	Object Type	Object Name	Object Name
	PSATABLE PSA Table	[redacted] ESTY_BC	[redacted] Message Type ZE814*
	PSATABLE PSA Table	[redacted] 0_LIFECYCLE_BD	[redacted]) Lifecycle
	PSATABLE PSA Table	[redacted] INHIST_BD	[redacted] 1 History Data
	PSATABLE PSA Table	[redacted] HD4_BD	[redacted] HD (Hist) (FM)

● Older than 5 Days

○ Page Before (Date)

☑ Only Successfully Booked/Updated Requests

□ Only those requests with errors, that are not booked in a data target

Needless to say, this is the method to manage PSA data deletion.

Identifying PSAs that are not being managed

Even if checklists and QA reviews are put in place to make sure every new project is implemented according to company standards, it is often forgotten to schedule PSA data deletion jobs. There is no simple way to catch all the PSAs whose data is not deleted periodically.

The following ABAP program comes in handy to determine which PSAs are not being managed.

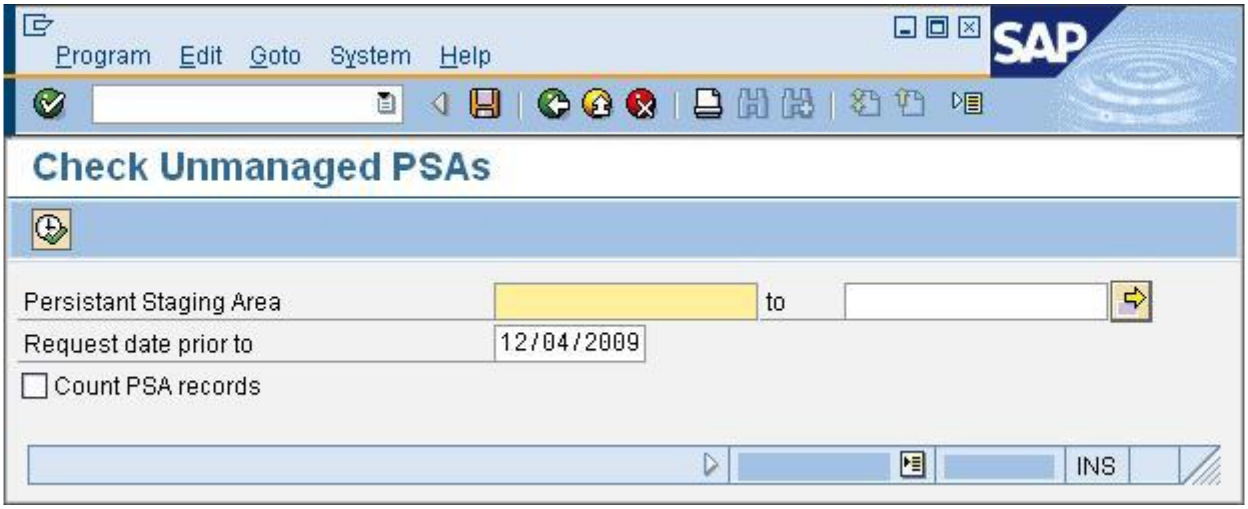
```
REPORT  zbw_check_unmanaged_psas.
*-----------------------------------------------------------------------*
* Author:             Sudhi Karkada                                      *
* Title:              Display PSA statistics                             *
* Transaction Code:   None                                               *
*-----------------------------------------------------------------------*
* Function:           To see which PSA is not being deleted              *
*                     periodically and to see which PSA table is         *
*                     occupying significant space.                       *
*-----------------------------------------------------------------------*

TABLES:     rstsods.
TYPE-POOLS: slis.
TYPES: BEGIN OF ty_display,
         tabname     TYPE rstsods-odsname,
         count        TYPE rscewcount,
         odsname(20) TYPE c,
         records     TYPE nrows,
       END   OF ty_display,

       BEGIN OF ty_rsreqicods,
         tabname     TYPE rsreqicods-tabname,
         timestamp   TYPE rsreqicods-timestamp,
         req_date    TYPE sy-datum,
       END   OF ty_rsreqicods,

       BEGIN OF ty_rstsods,
         odsname     TYPE rstsods-odsname,
         odstech     TYPE rstsods-odsname_tech,
         dateto      TYPE rstsods-dateto,
       END   OF ty_rstsods.
DATA:
  wa_display    TYPE ty_display,
  wa_rsreqicods TYPE ty_rsreqicods,
  wa_rstsods    TYPE ty_rstsods,
  t_display     TYPE STANDARD TABLE OF ty_display     INITIAL SIZE 0,
  t_rsreqicods  TYPE STANDARD TABLE OF ty_rsreqicods  INITIAL SIZE 0,
  t_rstsods     TYPE STANDARD TABLE OF ty_rstsods     INITIAL SIZE 0.

SELECT-OPTIONS:
  s_tabnm FOR rstsods-odsname.
PARAMETERS:
 p_date LIKE sy-datum OBLIGATORY DEFAULT sy-datum,
 p_cnt  TYPE c AS CHECKBOX.

START-OF-SELECTION.
  DATA: lc_timestamp(14) TYPE c,
        l_timestamp      TYPE rsreqicods-timestamp.
```

```
CONCATENATE p_date '000000' INTO lc_timestamp.
l_timestamp = lc_timestamp.

SELECT tabname
       timestamp
INTO   TABLE t_rsreqicods
FROM   rsreqicods
WHERE  timestamp < l_timestamp
  AND  typ = 'O'
  AND  tabname IN s_tabnm.

IF sy-subrc <> 0.
  WRITE: / 'No matching records found.'(001).
  EXIT.
ENDIF.

LOOP AT t_rsreqicods INTO wa_rsreqicods.
  lc_timestamp = wa_rsreqicods-timestamp.
  wa_rsreqicods-req_date = lc_timestamp(8).
  MODIFY t_rsreqicods FROM wa_rsreqicods TRANSPORTING req_date.
ENDLOOP.

SELECT odsname
       odsname_tech
       dateto
INTO   TABLE t_rstsods
FROM   rstsods
WHERE  odsname IN (  SELECT DISTINCT tabname
                    FROM   rsreqicods
                    WHERE  timestamp < l_timestamp
                      AND  typ = 'O'
                      AND  tabname IN s_tabnm ).

SORT t_rstsods BY odsname dateto.

LOOP AT t_rsreqicods INTO wa_rsreqicods.
  CLEAR wa_display.
  wa_display-tabname = wa_rsreqicods-tabname.
  wa_display-count   = 1.

  READ TABLE t_rstsods WITH KEY odsname = wa_rsreqicods-tabname
    TRANSPORTING NO FIELDS
    BINARY SEARCH.
  IF sy-subrc = 0.
    LOOP AT t_rstsods INTO wa_rstsods FROM sy-tabix.
      IF wa_rstsods-odsname <> wa_rsreqicods-tabname.
        EXIT.
      ENDIF.

      IF wa_rstsods-dateto >= wa_rsreqicods-req_date.
        wa_display-odsname = wa_rstsods-odstech.
        EXIT.
      ENDIF.
    ENDLOOP.
  ENDIF.
```

```
    COLLECT wa_display INTO t_display.
  ENDLOOP.

  LOOP AT t_display INTO wa_display.
    IF wa_display-odsname <> ''.
*&     Does the table really exist in the DB?
      SELECT tabname
      INTO   wa_display-odsname
      FROM   dd02l
      UP TO  1 ROWS
      WHERE  tabname = wa_display-odsname.
      ENDSELECT.

      IF sy-subrc = 0.
        IF p_cnt = 'X'.
          SELECT COUNT(*)
          INTO   wa_display-records
          FROM   (wa_display-odsname).
        ENDIF.
      ELSE.
*&       Table doesn't exist. Put paranthesis to indicate so.
        CLEAR wa_display-records.
        CONCATENATE '(' wa_display-odsname ')'
               INTO wa_display-odsname
          SEPARATED BY space.
      ENDIF.

      MODIFY t_display FROM wa_display.
    ENDIF.
  ENDLOOP.

  IF p_cnt = 'X'.
    SORT t_display BY records DESCENDING.
  ELSE.
    SORT t_display BY count DESCENDING.
  ENDIF.

END-OF-SELECTION.
  DATA: wa_fc TYPE slis_fieldcat_alv,
        t_fc  TYPE STANDARD TABLE OF slis_fieldcat_alv INITIAL SIZE 0.

  wa_fc-tabname    = 'T_DISPLAY'.

  wa_fc-col_pos    = 1.
  wa_fc-fieldname  = 'TABNAME'.
  wa_fc-seltext_s  = wa_fc-seltext_m = 'PSA Name'(005).
  wa_fc-outputlen  = 30.
  APPEND wa_fc TO t_fc.

  wa_fc-col_pos    = 2.
  wa_fc-fieldname  = 'COUNT'.
  wa_fc-seltext_s  = wa_fc-seltext_m = 'Num Requests'(006).
  wa_fc-outputlen  = 12.
  APPEND wa_fc TO t_fc.
```

```
wa_fc-col_pos    = 3.
wa_fc-fieldname  = 'ODSNAME'.
wa_fc-seltext_s  = wa_fc-seltext_m = 'Table Name'(007).
wa_fc-outputlen  = 20.
APPEND wa_fc TO t_fc.

IF p_cnt = 'X'.
  wa_fc-col_pos    = 4.
  wa_fc-fieldname  = 'RECORDS'.
  wa_fc-seltext_s  = wa_fc-seltext_m = 'Record Count'(008).
  wa_fc-outputlen  = 20.
  APPEND wa_fc TO t_fc.
ENDIF.

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
  EXPORTING
    it_fieldcat   = t_fc[]
  TABLES
    t_outtab      = t_display[]
  EXCEPTIONS
    program_error = 1
    OTHERS         = 2.
IF sy-subrc <> 0.
  WRITE: / 'ALV Display Error:'(003), sy-subrc.
ENDIF.
```

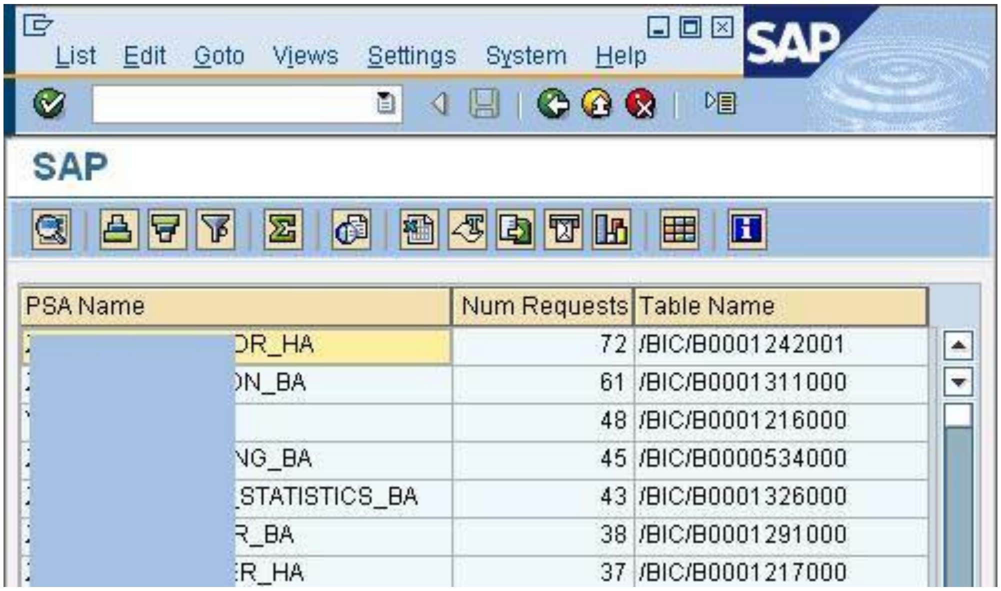
Program selection screen



Check Unmanaged PSAs

Persistant Staging Area		to	
Request date prior to	12/04/2009		
□ Count PSA records			

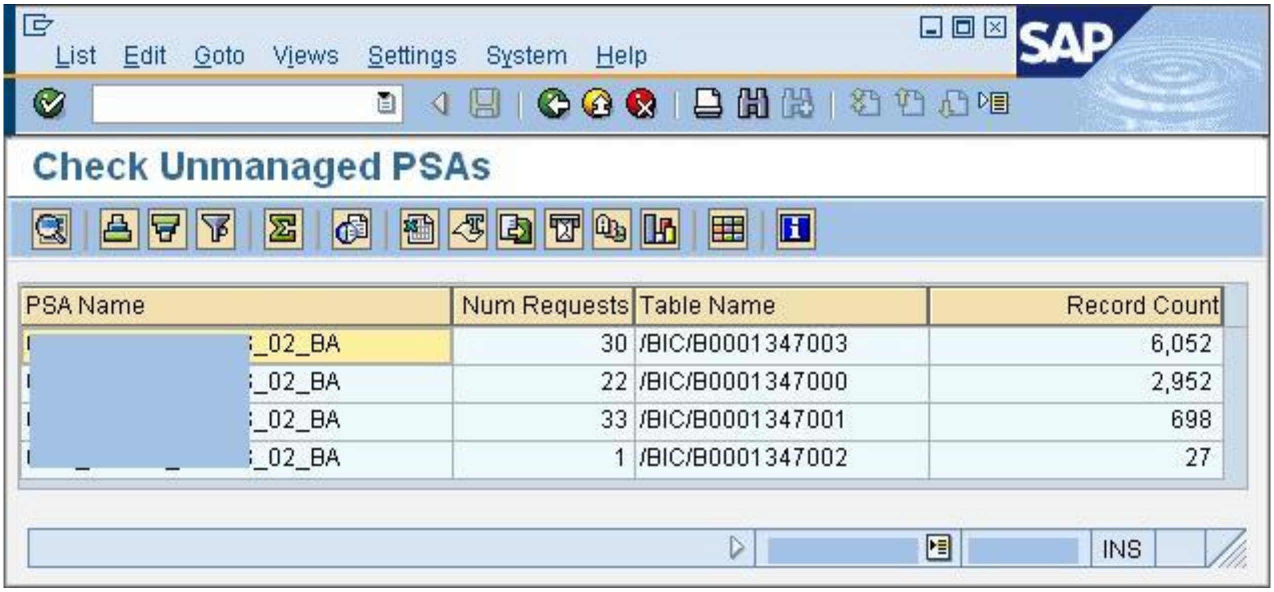
Leave "Count PSA records" check box unchecked if your intention is to just see the request counts for each PSA. If you check this box and do not supply any PSA name, then the program may run for quite a long time. When you run the program without changing the default values, results look like this:



SAP

PSA Name	Num Requests	Table Name
[redacted]DR_HA	72	/BIC/B0001242001
[redacted]ON_BA	61	/BIC/B0001311000
[redacted]	48	/BIC/B0001216000
[redacted]NG_BA	45	/BIC/B0000534000
[redacted]_STATISTICS_BA	43	/BIC/B0001326000
[redacted]R_BA	38	/BIC/B0001291000
[redacted]ER_HA	37	/BIC/B0001217000

When you choose to output record counts by setting the checkbox, results look like this:



Check Unmanaged PSAs

PSA Name	Num Requests	Table Name	Record Count
[redacted]_02_BA	30	/BIC/B0001347003	6,052
[redacted]_02_BA	22	/BIC/B0001347000	2,952
[redacted]_02_BA	33	/BIC/B0001347001	698
[redacted]_02_BA	1	/BIC/B0001347002	27

Related Content

Deleting Requests from PSA and Change Log tables in BI

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.