Inactive Objects Checker in BI 7

Applies to:

SAP BI 7.0 For more information, visit the EDW homepage.

Summary

Program to list out inactive objects in BI 7 systems

Author: Nageswara Reddy M

Company: Mahindra Satyam Computer Services Ltd.

Created on: 31 August, 2010

Author Bio

Nageswara Reddy is currently with Mahindra Satyam Computer Services Ltd. He has an overall experience of 6 years in SAP- ABAP. He has been working in SAP-BI 7.0 for 2 years now.

Table of Contents

1. Introduction

Generally some objects become inactive in the BW systems during processes like Transports, Upgrades etc. This can result in the various solutions/products not functioning normally as they should – i.e. it could lead to failure of data loads, failure in execution of queries etc.

There is no standard program or transaction code in SAP, that could read out the inactive objects in BW systems. The following program has been developed to list out inactive objects category-wise.

The program can be used to list out the inactive objects before and after the Transport or Upgrade or any similar activity. These lists can be saved to excel sheets. Now the two lists can be compared to identify the objects that have become inactive during that particular activity. Such inactive objects can then be activated to avoid any possible errors.

The program has to identify and report the list of inactive objects in the below categories.

1. MultiProviders
2. Cubes
3. DataStore Objects
4. InfoObjects
5. Transformations
6. DTPs
7. Process Chains
8. Aggregation Levels
9. DataSources

2. Details of the Database tables related to the Objects.

The basic information i.e. the database tables that contain the status of the respective objects are given below:

MultiProviders

The database table rsdcube gives the list of inactive multiproviders. The inactive multiproviders have the Object Status as ‘INA’ and these are cubes of cubetype equal to ‘M’.

Cubes

The database table rsdcube gives the list of inactive cubes. The inactive cubes have the Object Status as ‘INA’ and these are cubes of cubetype not equal to ‘M’. As the cubetype in this case is not equal to ‘M’, all the other type of cubes like Standard, Aggregate, Remote and Virtual InfoProvider are covered in this type.

DataStore Objects

The database table rsdodsoloc gives the list of inactive DSOs. The inactive DSOs have the Object Status as ‘INA

InfoObjects

The database table rsdiobj gives the list of inactive infoobjects. The inactive infoobjects have the Object Status as ‘INA’.

Transformations

The database table rstran gives the list of inactive transformations. The inactive transformations have the Object Status as ‘INA’. The list of inactive transformations is sorted on the Target data. The list gives Source of the transformation, Target of the transformation and the Transformation ID.

DTPs

The database table rsbkdtpstat gives the list of inactive DTPs. The inactive DTPs have the Object Status as 'INA'. The list of inactive DTPs is sorted on the Target data. The list gives Source of the DTP, Target of the DTP, DTP ID and DTP Text.

Process Chains

The basic table rspcchain gives the list of all Process chains. A small logic in the program filters all the inactive chains. These inactive chains are listed in the order of Chain text and Chain id.

Aggregation Levels

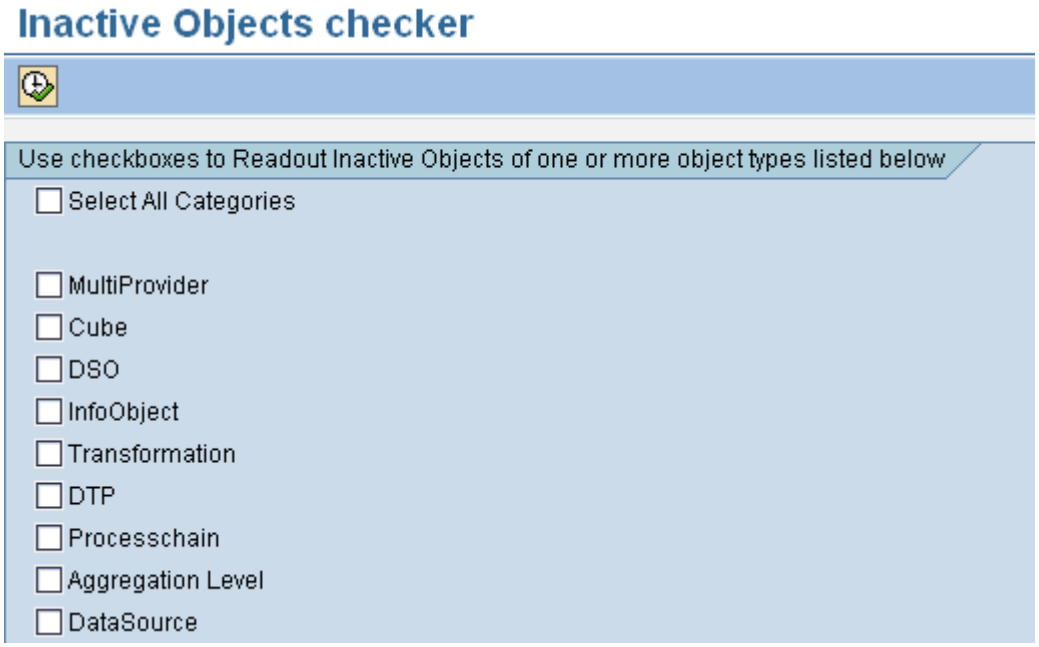
The database table rspls_alvl gives the list of inactive Aggregation Levels. The inactive aggregation levels have the Object Status as 'INA'.

DataSources

The database table rsds gives the list of inactive DataSources. The inactive datasources have the Object Status as 'INA'.

3. Code in the program

The screenshot below gives the selection screen of the program. The user may be interested in one category of the objects or in multiple categories. The user should also be able to select all the categories at a time.



There is a Check box for each category of objects. The program has a Subroutine defined for each category when the user selects the check boxes, only the respective subroutines are executed.

As explained in section 2, in a few cases a SELECT statement on the database table may not be sufficient because it may give wrong data (in case of DTPs) or redundant data (in case of Process Chains). In such cases, some extra code with a little logic may be required to identify the inactive objects.

The code is given in Appendix:

4. Appendix

```
REPORT  z_inactv_objs LINE-SIZE 200.

DATA: checkbox_mark TYPE xfeld.
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.

SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS: x_call AS CHECKBOX USER-COMMAND com1.
SELECTION-SCREEN COMMENT (25) text-002 FOR FIELD x_call.
SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN SKIP.

PARAMETERS: x_c1  AS CHECKBOX USER-COMMAND com2, " MultiProvider
            x_c2  AS CHECKBOX USER-COMMAND com2, " Cube
            x_c3  AS CHECKBOX USER-COMMAND com2, " DSO
            x_c4  AS CHECKBOX USER-COMMAND com2, " InfoObject
            x_c5  AS CHECKBOX USER-COMMAND com2, " Transformation
            x_c6  AS CHECKBOX USER-COMMAND com2, " DTP
            x_c7  AS CHECKBOX USER-COMMAND com2, " Processchain
            x_c8  AS CHECKBOX USER-COMMAND com2, " Aggregation Level
            x_c9  AS CHECKBOX USER-COMMAND com2. " DataSource
SELECTION-SCREEN END OF BLOCK b1.

DATA: l_lines TYPE i.

AT SELECTION-SCREEN ON x_call.
  PERFORM at_call.

AT SELECTION-SCREEN OUTPUT.
  PERFORM at_sel_scr_output.

START-OF-SELECTION.

* Get Inactive Multiproviders
  PERFORM get_inactiv_multipro.

* Get Inactive Cubes
  PERFORM get_inactiv_cube.

* Get Inactive DataStore Objects
  PERFORM get_inactiv_dso.

* Get Inactive InfoObjects
  PERFORM get_inactiv_iobj.

* Get Inactive Transformations
  PERFORM get_inactiv_transf.

* Get Inactive DTPs
  PERFORM get_inactiv_dtp.

* Get Inactive Process Chains
  PERFORM get_inactiv_pchain.
```

```
* Get Inactive Aggregation Level
  PERFORM get_inactiv_aggrlvl.

* Get Inactive DataSources
  PERFORM get_inactiv_dtasrc.

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_MULTIPRO
*&---------------------------------------------------------------------*
FORM get_inactiv_multipro .
  TYPES: BEGIN OF ty_mltpro,
         infocube TYPE rsinfocube,
         END OF ty_mltpro.

  DATA: lt_mltpro TYPE  SORTED TABLE OF ty_mltpro WITH UNIQUE KEY infocube,
        la_mltpro TYPE ty_mltpro.

  CHECK x_c1 IS NOT INITIAL.

  SELECT infocube
    FROM rsdcube
    INTO TABLE lt_mltpro
   WHERE objvers = 'M'
     AND objstat = 'INA'
     AND cubetype = 'M'.

  IF sy-subrc = 0.
    DESCRIBE TABLE lt_mltpro LINES l_lines.

    FORMAT COLOR COL_NEGATIVE.
    WRITE: l_lines, 'Inactive Multiproviders found', /.
    FORMAT COLOR OFF.

    LOOP AT lt_mltpro INTO la_mltpro.
      WRITE:/ la_mltpro-infocube.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE: 'No Inactive Multiproviders found'.
    FORMAT COLOR OFF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_MULTIPRO

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_CUBE
*&---------------------------------------------------------------------*
FORM get_inactiv_cube .
  TYPES: BEGIN OF ty_cube,
         infocube TYPE rsinfocube,
         END OF ty_cube.

  DATA: lt_cube TYPE  SORTED TABLE OF ty_cube WITH UNIQUE KEY infocube,
        la_cube TYPE ty_cube.

  CHECK x_c2 IS NOT INITIAL.
```

```
  SELECT infocube
    FROM rsdcube
    INTO TABLE lt_cube
   WHERE objvers = 'M'
     AND objstat = 'INA'
     AND cubetype NE 'M'.

  IF sy-subrc = 0.
    DESCRIBE TABLE lt_cube LINES l_lines.

    FORMAT COLOR COL_NEGATIVE.
    WRITE:/ l_lines, 'Inactive Standard Cubes found', /.
    FORMAT COLOR OFF.

    LOOP AT lt_cube INTO la_cube.
      WRITE:/ la_cube-infocube.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE:/ 'No Inactive Standard Cubes found'.
    FORMAT COLOR OFF.
  ENDIF.

ENDFORM.                    " GET_INACTIV_CUBE

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_DSO
*&---------------------------------------------------------------------*
FORM get_inactiv_dso .
  TYPES: BEGIN OF ty_odso,
         dso TYPE rsdodsobject,
         END OF ty_odso.

  DATA: lt_odso    TYPE  SORTED TABLE OF ty_odso WITH UNIQUE KEY dso,
        la_odso    TYPE ty_odso,
        lt_rsdodso TYPE TABLE OF ty_odso.

  CHECK x_c3 IS NOT INITIAL.

  SELECT odsobject
    FROM rsdodso
    INTO TABLE lt_rsdodso
   WHERE objvers = 'M'.

  IF sy-subrc = 0.
    SELECT odsobject
      FROM rsdodsoloc
      INTO TABLE lt_odso
       FOR ALL ENTRIES IN lt_rsdodso
     WHERE odsobject = lt_rsdodso-dso
       AND objstat = 'INA'.

    IF sy-subrc = 0.
      DESCRIBE TABLE lt_odso LINES l_lines.

      FORMAT COLOR COL_NEGATIVE.
```

```
      WRITE:/ l_lines, 'Inactive DSOs found', /.
      FORMAT COLOR OFF.

      LOOP AT lt_odso INTO la_odso.
        WRITE:/ la_odso-dso.
      ENDLOOP.
    ELSE.
      FORMAT COLOR COL_POSITIVE.
      WRITE:/ 'No Inactive DSOs found'.
      FORMAT COLOR OFF.
    ENDIF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_DSO

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_IOBJ
*&---------------------------------------------------------------------*
FORM get_inactiv_iobj .
  TYPES: BEGIN OF ty_iobj,
         iobjnm TYPE rsdiobjnm,
         END OF ty_iobj.

  DATA: lt_iobj TYPE SORTED TABLE OF ty_iobj WITH UNIQUE KEY iobjnm,
        la_iobj TYPE ty_iobj.

  CHECK x_c4 IS NOT INITIAL.

  SELECT iobjnm
    FROM rsdiobj
    INTO TABLE lt_iobj
   WHERE objvers = 'M'
     AND objstat = 'INA'.

  IF sy-subrc = 0.
    DESCRIBE TABLE lt_iobj LINES l_lines.

    FORMAT COLOR COL_NEGATIVE.
    WRITE:/ l_lines, 'Inactive InfoObjects found', /.
    FORMAT COLOR OFF.

    LOOP AT lt_iobj INTO la_iobj.
      WRITE:/ la_iobj-iobjnm.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE:/ 'No Inactive InfoObjects found'.
    FORMAT COLOR OFF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_IOBJ

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_TRANSF
*&---------------------------------------------------------------------*
FORM get_inactiv_transf .
  TYPES: BEGIN OF ty_rstran,
         tranid     TYPE rstranid,
```

```
          sourcename TYPE sobj_name,
          targetname TYPE sobj_name,
          END OF ty_rstran,

          BEGIN OF ty_rstrant,
          tranid TYPE rstranid,
          txtlg  TYPE rstxtlg,
          END OF ty_rstrant.

  DATA: lt_rstran  TYPE TABLE OF ty_rstran,
        la_rstran  TYPE ty_rstran,
        lt_rstrant TYPE HASHED TABLE OF ty_rstrant WITH UNIQUE KEY tranid,
        la_rstrant TYPE ty_rstrant.

  CHECK x_c5 IS NOT INITIAL.

  SELECT tranid sourcename targetname
    FROM rstran
    INTO TABLE lt_rstran
   WHERE objvers = 'M'
     AND objstat = 'INA'.

  IF sy-subrc = 0.
    SELECT tranid txtlg
      FROM rstrant
      INTO TABLE lt_rstrant
       FOR ALL ENTRIES IN lt_rstran
     WHERE langu = 'K'
       AND tranid = lt_rstran-tranid
       AND objvers = 'M'.

    DESCRIBE TABLE lt_rstran LINES l_lines.
    SORT lt_rstran BY sourcename.

    FORMAT COLOR COL_NEGATIVE.
    WRITE:/ l_lines, 'Inactive Transformations found', /.
    FORMAT COLOR COL_HEADING.
    WRITE: text-t53, 43 text-t54, 72 text-t52.
    FORMAT COLOR OFF.

    LOOP AT lt_rstran INTO la_rstran.
      READ TABLE lt_rstrant INTO la_rstrant WITH KEY tranid = la_rstran-tranid.

      WRITE:/ la_rstran-sourcename UNDER text-t53, la_rstran-targetname UNDER text-
t54,
              la_rstran-tranid UNDER text-t52.
      CLEAR: la_rstrant.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE:/ 'No Inactive Transformations found'.
    FORMAT COLOR OFF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_TRANSF

*&---------------------------------------------------------------------*
```

```
*&      Form  GET_INACTIV_DTP
*&---------------------------------------------------------------------*
FORM get_inactiv_dtp .
  TYPES: BEGIN OF ty_dtp,
         dtp TYPE rsbkdtpnm,
         objvers TYPE rsobjvers,
         src TYPE rsbksrcnm,
         tgt TYPE rsbktgtnm,
         text TYPE symsgv,
         END OF ty_dtp.

  DATA: lt_dtp    TYPE TABLE OF ty_dtp,
        la_dtp    TYPE ty_dtp,
        lt_inadtp TYPE TABLE OF ty_dtp,
        la_inadtp TYPE ty_dtp,
        lr_dtp    TYPE REF TO cl_rsbk_dtp,
        l_text    TYPE symsgv,
        lv_active TYPE c.

  CHECK x_c6 IS NOT INITIAL.

  SELECT dtp objvers src tgt
    FROM rsbkdtp
    INTO TABLE lt_dtp
   WHERE objvers IN ('M', 'A').

  IF lt_dtp IS NOT INITIAL.
    SORT lt_dtp BY dtp ASCENDING objvers DESCENDING.
    DELETE ADJACENT DUPLICATES FROM lt_dtp COMPARING dtp.

    LOOP AT lt_dtp INTO la_dtp.
      CALL METHOD cl_rsbk_dtp=>factory
        EXPORTING
          i_dtp   = la_dtp-dtp
        RECEIVING
          r_r_dtp = lr_dtp.

      CALL METHOD lr_dtp->if_rso_tlogo_maintain~is_active
        RECEIVING
          r_is_active = lv_active.

      IF lv_active IS INITIAL.
        TRY.
            CALL METHOD lr_dtp->get_text
              EXPORTING
                i_langu   = sy-langu
              RECEIVING
                r_dtptext = l_text.
          CATCH cx_rs_access_error .
        ENDTRY.
        la_dtp-text = l_text.

        MODIFY lt_dtp FROM la_dtp.
      ELSE.
        DELETE lt_dtp.
      ENDIF.
```

```
        CLEAR: l_text, la_inadtp, la_dtp.
      ENDLOOP.
    ENDIF.

    IF lt_dtp IS NOT INITIAL.
      SORT lt_dtp BY tgt.
      DESCRIBE TABLE lt_dtp LINES l_lines.

      FORMAT COLOR COL_NEGATIVE.
      WRITE:/ l_lines, 'Inactive DTPs found', /.
      FORMAT COLOR COL_HEADING.
      WRITE: text-t53, 48 text-t54, 76 text-t62, 110 text-t61.
      FORMAT COLOR OFF.

      LOOP AT lt_dtp INTO la_dtp.
        WRITE:/ la_dtp-src UNDER text-t53, la_dtp-tgt UNDER text-t54,
                la_dtp-dtp UNDER text-t62, la_dtp-text UNDER text-t61.
      ENDLOOP.

    ELSE.
      FORMAT COLOR COL_POSITIVE.
      WRITE:/ 'No Inactive DTPs found'.
      FORMAT COLOR OFF.
    ENDIF.
  ENDFORM.                    " GET_INACTIV_DTP

  *&---------------------------------------------------------------------*
  *&      Form  GET_INACTIV_PCHAIN
  *&---------------------------------------------------------------------*
  FORM get_inactiv_pchain .

    TYPES: BEGIN OF ty_chaint,
           chain_id TYPE rspc_chain,
           txtlg    TYPE rstxtlg,
           END OF ty_chaint.

    DATA: lt_chain_a TYPE TABLE OF rspcchain,
          lt_chain_m TYPE TABLE OF rspcchain,
          ls_chain   TYPE rspcchain,
          lt_chain   TYPE TABLE OF rspcchain,
          lt_chaint  TYPE TABLE OF ty_chaint,
          ls_chaint  TYPE ty_chaint,
          lt_chain_all     TYPE TABLE OF rspcchain,
          ls_chain_all     TYPE rspcchain.

    CHECK x_c7 IS NOT INITIAL.

    SELECT *
      FROM rspcchain
      INTO TABLE lt_chain
     WHERE objvers IN ('M', 'A').

    lt_chain_all[] = lt_chain[].

    SORT lt_chain BY chain_id.
    DELETE ADJACENT DUPLICATES FROM lt_chain COMPARING chain_id.
```

```
  LOOP AT lt_chain INTO ls_chain.
    LOOP AT lt_chain_all INTO ls_chain_all WHERE chain_id = ls_chain-chain_id.
      IF ls_chain_all-objvers = 'A'.
        APPEND ls_chain_all TO lt_chain_a.
      ELSEIF ls_chain_all-objvers = 'M'.
        ls_chain_all-objvers = 'A'.
        APPEND ls_chain_all TO lt_chain_m.
      ENDIF.
    ENDLOOP.

    SORT lt_chain_m BY type variante lnr.
    SORT lt_chain_a BY type variante lnr.

    IF lt_chain_m[] = lt_chain_a[].
      DELETE lt_chain.
    ENDIF.

    CLEAR: ls_chain, ls_chain_all.
    REFRESH: lt_chain_a, lt_chain_m.
  ENDLOOP.

  IF lt_chain IS NOT INITIAL.
    SELECT chain_id txtlg
      FROM rspcchaint
      INTO TABLE lt_chaint
       FOR ALL ENTRIES IN lt_chain
     WHERE langu = sy-langu
       AND chain_id = lt_chain-chain_id.

    DESCRIBE TABLE lt_chain LINES l_lines.
    FORMAT COLOR COL_NEGATIVE.
    WRITE:/ l_lines, 'Inactive Process Chains found', /.
    FORMAT COLOR COL_HEADING.
    WRITE:/ text-t72, 30 text-t71.
    FORMAT COLOR OFF.

    LOOP AT lt_chain INTO ls_chain.
      READ TABLE lt_chaint INTO ls_chaint WITH KEY chain_id = ls_chain-chain_id.
      FORMAT COLOR COL_NORMAL.
      WRITE:/ ls_chaint-txtlg UNDER text-t71.
      FORMAT COLOR COL_KEY.
      WRITE: ls_chain-chain_id UNDER text-t72.
      FORMAT COLOR OFF.
      CLEAR: ls_chaint, ls_chain.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE:/ 'No inactive Process chains found'.
    FORMAT COLOR OFF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_PCHAIN

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_AGGRLVL
*&---------------------------------------------------------------------*
```

```
FORM get_inactiv_aggrlvl .

  TYPES: BEGIN OF ty_aggrlvl,
         aggrlevel TYPE rspls_aggrlevel,
         END OF ty_aggrlvl.

  DATA: lt_aggrlvl TYPE SORTED TABLE OF ty_aggrlvl WITH UNIQUE KEY aggrlevel,
        la_aggrlvl TYPE ty_aggrlvl.

  CHECK x_c8 IS NOT INITIAL.

  SELECT aggrlevel
    FROM rspls_alvl
    INTO TABLE lt_aggrlvl
   WHERE objvers = 'M'
     AND objstat = 'INA'.

  IF lt_aggrlvl IS NOT INITIAL.
    DESCRIBE TABLE lt_aggrlvl LINES l_lines.

    FORMAT COLOR COL_NEGATIVE.
    WRITE:/ l_lines, 'Inactive Aggregation Levels found', /.
    FORMAT COLOR OFF.

    LOOP AT lt_aggrlvl INTO la_aggrlvl.
      WRITE: la_aggrlvl-aggrlevel.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE:/ 'No Inactive Aggregation Levels found'.
    FORMAT COLOR OFF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_AGGRLVL

*&---------------------------------------------------------------------*
*&      Form  GET_INACTIV_DTASRC
*&---------------------------------------------------------------------*
FORM get_inactiv_dtasrc .
  TYPES: BEGIN OF ty_rsds,
         ds     TYPE roosourcer,
         logsys TYPE rsslogsys,
         END OF ty_rsds.

  DATA: lt_rsds TYPE SORTED TABLE OF ty_rsds WITH UNIQUE KEY ds logsys,
        la_rsds TYPE ty_rsds.

  CHECK x_c9 IS NOT INITIAL.

  SELECT datasource logsys
    FROM rsds
    INTO TABLE lt_rsds
   WHERE objvers = 'M'
     AND objstat = 'INA'.

  IF sy-subrc = 0.
    DESCRIBE TABLE lt_rsds LINES l_lines.
```

```
    FORMAT COLOR COL_NEGATIVE.
    WRITE:/ l_lines, 'Inactive DataSources found', /.
    FORMAT COLOR OFF.

    LOOP AT lt_rsds INTO la_rsds.
      WRITE:/ la_rsds-ds, la_rsds-logsys.
    ENDLOOP.
  ELSE.
    FORMAT COLOR COL_POSITIVE.
    WRITE:/ 'No Inactive DataSources found'.
    FORMAT COLOR OFF.
  ENDIF.
ENDFORM.                    " GET_INACTIV_DTASRC

*&---------------------------------------------------------------------*
*&      Form  AT_SEL_SCR_OUTPUT
*&---------------------------------------------------------------------*
FORM at_sel_scr_output .
  DATA: l_option TYPE xfeld.
  CASE checkbox_mark.
    WHEN 'S'.
      l_option = 'X'.
    WHEN 'D'.
      l_option = ' '.
  ENDCASE.
  IF checkbox_mark IS NOT INITIAL.
    x_c1  = l_option. x_c2  = l_option. x_c3  = l_option. x_c4  = l_option. x_c5  = l
_option.
    x_c6  = l_option. x_c7  = l_option. x_c8  = l_option. x_c9  = l_option.
    CLEAR checkbox_mark.
  ENDIF.
ENDFORM.                    " AT_SEL_SCR_OUTPUT

*&---------------------------------------------------------------------*
*&      Form  AT_CALL
*&---------------------------------------------------------------------*
FORM at_call .
  IF sy-ucomm EQ 'COM1'.
    CASE x_call.
      WHEN 'X'.
        checkbox_mark = 'S'.
      WHEN space.
        checkbox_mark = 'D'.
    ENDCASE.
  ELSEIF sy-ucomm EQ 'COM2'.
    CLEAR x_call.
  ENDIF.
ENDFORM.                    " AT_CALL
```

Related Content

http://forums.sdn.sap.com/thread.jspa?messageID=9071294#9071294

http://forums.sdn.sap.com/thread.jspa?messageID=8731432#8731432

http://forums.sdn.sap.com/thread.jspa?messageID=7336569#7336569

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.