

Dynamic Update and Deletion from DSO and Table



Applies To

S/W Component	Release	Level	Highest Support Package	Description
SAP_ABA	640	13	SAPKA64013	Cross-Application Component
SAP_BASIS	640	13	SAPKB64013	SAP Basis Component
PI_BASIS	2005_1_640	10	SAPKIPYJ6A	Basis Plug-In (PI_BASIS) 2005_1_640
ST-PI	2005_1_640	3	SAPKITLQH3	SAP Solution Tools Plug-In
SAP_BW	350	13	SAPKW35013	SAP_BW 350
BI_CONT	353	5	SAPKIBIFP5	Business Intelligence Content
ST-A/PI	01G_BCO640	0	-	Service tools for other App./NetWeaver 04

Summary

Report to perform deletion / update of any database table records at ease. This report fetches the selection criteria based on the data type of the field. As long as the field is a characteristic, it will be picked up dynamically.

Apart from table level update, it also performs deletion from DSO Active Table / Change Log / PSA table. Selections in this scenario are based on the KPI vs. Characteristic option.

Author: Praveen Gudishetty

Company: RevTek Consulting Services

Created on: 01 January 2008

Author Bio

Praveen Gudishetty is a business intelligence architect at RevTek Consulting Services.

Table of Contents

Description of the Report: ZBW_DELETE_DSO_CONTENTS	3
Code Sample: ZBW_DELETE_DSO_CONTENTS	6
Parameter Attributes	55
Execution Snapshots: Deletion from Change Log	56
Execution Snapshot: Update a Database Table	58
Execution Snapshot: Deletion from Change Log and Active Table	60
Disclaimer and Liability Notice	62

Description of the Report: ZBW_DELETE_DSO_CONTENTS

1. This report is basically segregated into two chunks.
 - a. Execution based on DSO.
 - b. Execution based on Table.
2. Execution of DSO is segregated into three phases:
 - a. Execution to delete data records from Active Table.
 - b. Execution to delete data records from Change Log Table.
 - c. Execution to delete data records from PSA Table.
3. Execution of Table is segregated into two phases:
 - a. Execution to delete data records from the table.
 - b. Execution to update the data records in the table based on the selection criteria.
4. Parameter Selection Options:

Selection Text		
Parameter Name	Parameter Text	Description
P_ACT	Delete Active Table Data.	Flag to delete from Active Table
P_ATNAME	Active Table Tech. Name.	Active Table Name
P_BDAYS	Delete Any PSA Before Days.	Number of data prior to current date which needs to be deleted from the PSA.
P_CHECK	Printing Spool Data.	Flag to print detailed level of data
P_CHGLOG	Change Log Tech. Name.	Change Log Technical Name
P_COUNT	Commit Record Counter.	Commit level of data records
P_FORCE	Force Generate Report.	Flag enables force generation of the report, if it already exists.
P_LOG	Delete Change Log Data.	Flag enables deletion from Change Log Table.
P_ODS	ODS Tech. Name.	DSO Technical Name
P_PSA	Delete PSA Table Data.	Flag to delete the PSA Data
P_RDTAB	Delete Table Data Records	Flag to delete data records from the table specified.
P_RNONE	No Update / Delete.	No operations on the table.
P_RUTAB	Update Table Data Records.	Flag to Update the data records of the table specified.
P_TEST	Test Run.	Flag enables test run. Doesn't delete or update any data records.
P_UTNAME	Table Name.	Table Name.

5. Possible execution runs:
 - a. At any time only one of the following can happen:
 - i. Deletion / Update from Table
 - ii. Deletion from Change Log & Active Table / PSA.
 - iii. Operation on Table / DSO.

6. Create the Table ZDEL_DSO_REPORT:

Transp. Table: ZDEL_DSO_REPORT Active
 Short Text: Table maintenance for deletion report

Attributes | Delivery and Maintenance | Fields | Entry help/check | Currency/Quantity Fields

Field	Key	Initi...	Data element	Data T...	Length	Deci...	Short Text
DSO_NAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RSDODSOBJECT	CHAR	30		0 ODS Object
DELETION_REPORT	<input type="checkbox"/>	<input type="checkbox"/>	REPID	CHAR	40		0 ABAP Program Name
UNAME	<input type="checkbox"/>	<input type="checkbox"/>	XUBNAME	CHAR	12		0 User Name in User Master Record
DELETION_DATE	<input type="checkbox"/>	<input type="checkbox"/>	SYDATUM	DATS	8		0 Date and Time, Current (Application Server) Date
DELETION_TIME	<input type="checkbox"/>	<input type="checkbox"/>	SYUZEIT	TIMS	6		0 Date and Time, Current Application Server Time
ACTIVE_TABLE	<input type="checkbox"/>	<input type="checkbox"/>	TABNAME	CHAR	30		0 Table Name
CHANGELOG_TABLE	<input type="checkbox"/>	<input type="checkbox"/>	TABNAME	CHAR	30		0 Table Name
DEL_ACTIVE_TABLE	<input type="checkbox"/>	<input type="checkbox"/>	CHECKBOX	CHAR	1		0 Checkbox in Reporting
DEL_CHANGE_LOG	<input type="checkbox"/>	<input type="checkbox"/>	CHECKBOX	CHAR	1		0 Checkbox in Reporting
DEL_TABLE	<input type="checkbox"/>	<input type="checkbox"/>	CHECKBOX	CHAR	1		0 Checkbox in Reporting
UPDATE_TABLE	<input type="checkbox"/>	<input type="checkbox"/>	CHECKBOX	CHAR	1		0 Checkbox in Reporting

Data Browser: Table ZDEL_DSO_REPORT Select Entries 3

DSO_NAME	DELETION_REPORT	UNAME	DELETION_DATE	DELETION_TIME	ACTIVE_TABLE	CHANGELOG_TABLE	DEL_ACTIVE_TABLE	DEL_CHANGE_LOG	DEL_TABLE	UPDATE_TABLE
/BIC/AZSD_S0100	TAB_48IZLZ8MSZ650GZMGVUV7ZZZ0	PGUDISHETTY	01/20/2008	17.42.20						X
/BIC/B0000550000	TAB_48G1DU6F8V1ERLF9A0INLWCMJ6	PGUDISHETTY	01/20/2008	12.53.04					X	
ZSD_S05	DEL_48JYDLVYN34BI04T09VL5XCI	PGUDISHETTY	01/20/2008	21.58.57	/BIC/AZSD_S0500	/BIC/B0000550000	X	X		

7. Table: ZDEL_DSO_REPORT keeps track of current version of the reports which are already generated.

8. Message Class: ZBW:

Message class: ZBW Active

Attributes | Messages

Message No.	Message short text	Self-explanatory
030	Message 030: No Selection Restriction's available.	<input checked="" type="checkbox"/>
031	Message 031: Report must be executed in background only.	<input checked="" type="checkbox"/>
032	Message 032: Pattern objects like "*" must not be used for Selection's.	<input checked="" type="checkbox"/>
033	Message 033: Pattern objects like "%" must not be used for Selection's.	<input checked="" type="checkbox"/>
034	Message 034: Change Log determined, not in validity period.	<input checked="" type="checkbox"/>
035	Message 035: Change Log could not be found.	<input checked="" type="checkbox"/>
036	Message 036: Deletion Counter less than 1.	<input checked="" type="checkbox"/>
037	Message 037: No Data Store Technical Name specified.	<input checked="" type="checkbox"/>
038	Message 038: No data deleted as flags are not set.	<input checked="" type="checkbox"/>
039	Message 039: Combination of PSA and Active / Change Log not allowed.	<input checked="" type="checkbox"/>
040	Message 040: PSA data cannot be deleted without # of days parameter.	<input checked="" type="checkbox"/>
041	Message 041: You cannot use "Exclude" Option in Selection Criteria.	<input checked="" type="checkbox"/>
042	Message 041: No data records deleted as fetch failed.	<input checked="" type="checkbox"/>
043	Message 043: Deletion from table ZDEL_DSO_REPORT error.	<input checked="" type="checkbox"/>
044	Message 044: Update table name is missing.	<input checked="" type="checkbox"/>
045	Message 045: Either DSO / Update Table must be choosen.	<input checked="" type="checkbox"/>
046	Message 046: Either one of DSO / Update Table must be choosen not both.	<input checked="" type="checkbox"/>
047	Message 047: Update String is empty.	<input checked="" type="checkbox"/>

Message#	Message Description
030	Message 030: No Selection Restriction's available.
031	Message 031: Report must be executed in background only.
032	Message 032: Pattern objects like "*" must not be used for Selection's.
033	Message 033: Pattern objects like "%" must not be used for Selection's.
034	Message 034: Change Log determined, not in validity period.
035	Message 035: Change Log could not be found.
036	Message 036: Deletion Counter less than 1.
037	Message 037: No Data Store Technical Name specified.
038	Message 038: No data deleted as flags are not set.
039	Message 039: Combination of PSA and Active / Change Log not allowed.
040	Message 040: PSA data cannot be deleted without # of days parameter.
041	Message 041: You cannot use "Exclude" Option in Selection Criteria.
042	Message 041: No data records deleted as fetch failed.
043	Message 043: Deletion from table ZDEL_DSO_REPORT error.
044	Message 044: Update table name is missing.
045	Message 045: Either DSO / Update Table must be choosen.
046	Message 046: Either one of DSO / Update Table must be choosen not both.
047	Message 047: Update String is empty.

- Execution of the second report is possible only in background. It doesn't allow the report to be run in foreground.

Code Sample: ZBW_DELETE_DSO_CONTENTS

```

*&-----*
*& Report      :  ZBW_DELETE_DSO_CONTENTS                               *
*&-----*
*& Purpose    :  Report to delete Change Log / Active Table / PSA Data *
*&              and also delete / update table records depending of   *
*&              selection options.                                     *
*& Author     :  Praveen Gudishetty                                     *
*& Date       :  01/01/2008                                           *
*& Change Log:                                                         *
*& 01/01/2008 PGUDISHETTY                                           *
*&              Initial Creation.                                     *
*&-----*

REPORT  ZBW_DELETE_DSO_CONTENTS MESSAGE-ID ZBW.
TABLES: RSDODSO, RSTSODS, ZDEL_DSO_REPORT.
TYPE-POOLS ABAP.

***** Parameter Declaration.
SELECTION-SCREEN BEGIN OF BLOCK 6 WITH FRAME TITLE TEXT-006.
PARAMETERS: P_ODS      TYPE RSDODSOBJECT.
* Change Log Options.
SELECTION-SCREEN BEGIN OF BLOCK 1 WITH FRAME TITLE TEXT-001.
PARAMETERS: P_CHGLOG   TYPE RSODSNAME.
PARAMETERS : P_LOG     AS   CHECKBOX.
SELECTION-SCREEN END OF BLOCK 1.
* Active Table Options.
SELECTION-SCREEN BEGIN OF BLOCK 2 WITH FRAME TITLE TEXT-002.
PARAMETERS : P_ATNAME  TYPE TABNAME.
PARAMETERS : P_ACT     AS   CHECKBOX.
SELECTION-SCREEN END OF BLOCK 2.
* PSA Table Options.
SELECTION-SCREEN BEGIN OF BLOCK 3 WITH FRAME TITLE TEXT-003.
PARAMETERS : P_PSA     AS   CHECKBOX.
PARAMETERS : P_BDAYS   TYPE I.
SELECTION-SCREEN END OF BLOCK 3.
SELECTION-SCREEN END OF BLOCK 6.

SELECTION-SCREEN BEGIN OF BLOCK 7 WITH FRAME TITLE TEXT-007.
* Update Table Options.
SELECTION-SCREEN BEGIN OF BLOCK 4 WITH FRAME TITLE TEXT-004.
PARAMETERS : P_RUTAB   RADIOBUTTON GROUP TAB,
              P_RDTAB   RADIOBUTTON GROUP TAB,
              P_RNONE   RADIOBUTTON GROUP TAB DEFAULT 'X'.
PARAMETERS : P_UTNAME  TYPE TABNAME.
SELECTION-SCREEN END OF BLOCK 4.
SELECTION-SCREEN END OF BLOCK 7.

```

* Global Parameter Options.

SELECTION-SCREEN BEGIN OF BLOCK 5 WITH FRAME TITLE TEXT-005.

PARAMETERS : P_FORCE AS CHECKBOX DEFAULT 'X'.

PARAMETERS : P_COUNT TYPE I DEFAULT '1000'.

PARAMETERS : P_CHECK AS CHECKBOX.

PARAMETERS : P_TEST AS CHECKBOX.

SELECTION-SCREEN END OF BLOCK 5.

***** Pool Type's Declaration.

TYPE-POOLS: RSDRD,
 RSDQ,
 RSDDS,
 RSENG,
 SSCR,
 RRSM.

***** Type Declaration.

TYPES: RSD_S_IOBJ TYPE RSDIOBJV,
 RSD_IOBJNM TYPE RSD_S_IOBJ-IOBJNM,
 BEGIN OF RSDRD_S_SELOPT,
 IOBJNM TYPE RSD_IOBJNM,
 SELNM(4) TYPE C,
 KEYFL TYPE RS_BOOL,
 END OF RSDRD_S_SELOPT,
 BEGIN OF TYPE_INFOBJECT,
 INFOBJECT TYPE RSIOBJNM,
 FIELDNM TYPE RSDIOBJFIELDNM,
 L_CHAR TYPE CHAR6,
 END OF TYPE_INFOBJECT,
 BEGIN OF TYPE_DEL_SEL,
 L_TYPE_RANGE TYPE RSDRD_T_RANGE,
 INFOBJECT TYPE RSIOBJNM,
 FIELDNM TYPE RSDIOBJFIELDNM,
 L_CHAR TYPE CHAR6,
 END OF TYPE_DEL_SEL,
 BEGIN OF TYPE_REQUEST,
 RNR TYPE RSREQUNR,
 RECORDS TYPE RSNUMRECORD,
 DATUM TYPE SYDATUM,
 UZEIT TYPE SYUZEIT,
 TSTATUS TYPE RSSTATUS,
 REQRECORDS TYPE RSNUMRECORD,
 END OF TYPE_REQUEST.

***** Table Type Declaration.

TYPES: RSDRD_T_ABAPSOURCE TYPE STANDARD TABLE OF ABAPSOURCE
 WITH DEFAULT KEY,

```

RSDRD_T_SELOPT      TYPE STANDARD TABLE OF RSDRD_S_SELOPT
                    WITH DEFAULT KEY,
TYPE_T_INFOBJECT    TYPE STANDARD TABLE OF TYPE_INFOBJECT,
TYPE_T_DEL_SEL      TYPE STANDARD TABLE OF TYPE_DEL_SEL,
TYPE_T_REQUEST      TYPE STANDARD TABLE OF TYPE_REQUEST,
TYPE_DFIES_TAB      TYPE STANDARD TABLE OF DFIES,
TYPE_H_REQUEST      TYPE HASHED TABLE OF TYPE_REQUEST
                    WITH UNIQUE KEY RNR,
TYPE_H_DSO_RPT      TYPE HASHED TABLE OF ZDEL_DSO_REPORT
                    WITH UNIQUE KEY DSO_NAME.

```

***** Data Type Declaration:

```

DATA: WA_RSDODSO      TYPE RSDODSO,
      WA_RSTSODS      TYPE RSTSODS,
      APOSTOPHE       TYPE C VALUE ' ',
      DEL_GEN_RPT_NAME TYPE SY-REPID,
      CHANGELOG_ODSNAME TYPE RSODSNAME,
      ODS_ACT_TABNAME  TYPE TABNAME,
      ODS_ACT_TTYPE_NAME TYPE TTYPE_NAME.

```

***** Structure Data Type Declaration.

```

DATA: WA_TYPE_INFOBJECT TYPE TYPE_INFOBJECT,
      WA_RSREQDONE      TYPE TYPE_REQUEST,
      WA_DSO_RPT        TYPE ZDEL_DSO_REPORT,
      WA_DFIES          TYPE DFIES.

```

***** Internat Table Data Type Declaration.

```

DATA: DEL_SUB_PROGRAM  TYPE RSDRD_T_ABAPSOURCE,
      DEL_SELECTION_OPTIONS TYPE RSDRD_T_SELOPT,
      T_TYPE_INFOBJECT TYPE TYPE_T_INFOBJECT,
      T_H_REQUEST      TYPE TYPE_H_REQUEST,
      T_RSREQDONE      TYPE TYPE_T_REQUEST,
      T_H_DSO_RPT      TYPE TYPE_H_DSO_RPT,
      T_DFIES_TAB      TYPE TYPE_DFIES_TAB.

```

***** Internat Table Data Type Declaration.

```

FIELD-SYMBOLS: <FS_REQUEST> TYPE TYPE_REQUEST.

```

***** Constant Declaration:

```

CONSTANTS: C_COMMA      TYPE CHAR1 VALUE ',',
           C_KEYFIGURE   TYPE CHAR3 VALUE 'KYF',
           C_S          TYPE CHAR1 VALUE 'S',
           C_P          TYPE CHAR1 VALUE 'P',
           C_QUOTE      TYPE CHAR1 VALUE ' ',
           C_STAR       TYPE CHAR1 VALUE '*',
           C_PERCENT    TYPE CHAR1 VALUE '%',
           C_PSA       TYPE CHAR5 VALUE 'REQU%',

```



```

C_LINE          TYPE ABAPSOURCE VALUE
'=====',
C_CHAR          TYPE CHAR1 VALUE 'C'.

```

* Make the Change Log / Active Table Display Attribute.

AT SELECTION-SCREEN OUTPUT.

LOOP AT SCREEN.

IF SCREEN-NAME = 'P_CHGLOG' OR SCREEN-NAME = 'P_ATNAME' .

SCREEN-INPUT = '0'.

SCREEN-OUTPUT = '1'.

SCREEN-ACTIVE = '1'.

MODIFY SCREEN.

ENDIF.

* IF SCREEN-NAME = 'P_ODS'.

* SCREEN-REQUIRED = '1'.

* MODIFY SCREEN.

* ENDIF.

ENDLOOP.

* Selection Screen entries for parameter P_ODS on F4.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR P_ODS.

CALL FUNCTION 'RSDC_ODSO_F4_DYNP_UPDATE'

EXPORTING

I_REPID = SY-REPID

I_DYNNR = SY-DYNNR

I_FNAME = 'P_ODS'

* I_FNAME_TXT =

* I_FNAME_ICON =

* I_OBJVERS = RS_C_OBJVERS-ACTIVE

* I_DISPLAY = RS_C_FALSE

IMPORTING

E_ODSOBJECT = P_ODS.

* E_TXTLG =.

AT SELECTION-SCREEN.

IF P_ODS IS NOT INITIAL.

* Determine ODS Change Log.

CHANGELOG_ODSNAME = P_ODS.

CONCATENATE '8' CHANGELOG_ODSNAME '_AA' INTO CHANGELOG_ODSNAME.

SELECT SINGLE * FROM RSTSODS INTO WA_RSTSODS

WHERE ODSNAME = CHANGELOG_ODSNAME.

IF SY-SUBRC <> 0.

* If ChangeLog couldnt be found.

CLEAR: P_CHGLOG, P_ATNAME, P_ODS.

MESSAGE ID 'ZBW' TYPE 'E' NUMBER '035'.

STOP.

```

ELSE.
  IF SY-DATUM >= WA_RSTSODS-DATEFROM AND SY-DATUM <= WA_RSTSODS-DATETO.
* Determined Change log in validity period.
    P_CHGLOG = WA_RSTSODS-ODSNAME_TECH.
    ELSE.
* If the current date is not in validity period.
    MESSAGE ID 'ZBW' TYPE 'E' NUMBER '034'.
    STOP.
    ENDIF.
  ENDIF.
* End of Determine ODS Change Log.

* Determine the ODS Active Table.
  CALL METHOD CL_RSD_ODSO=>GET_TABLNM
    EXPORTING
      I_ODSOBJECT   = P_ODS
    IMPORTING
      E_TABLNM      = ODS_ACT_TABNAME
      E_TTYPENAME   = ODS_ACT_TTYPENAME
    EXCEPTIONS
      NAME_ERROR    = 1
      INPUT_INVALID = 2.
  IF SY-SUBRC <> 0.
    MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
      WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4
      RAISING FAILED.
  ENDIF.
  P_ATNAME = ODS_ACT_TABNAME.

* End of Determine the ODS Active Table.
ELSEIF P_RNONE IS INITIAL.
  IF P_UTNAME IS INITIAL.
    MESSAGE ID 'ZBW' TYPE 'E' NUMBER '044'.
    STOP.
  ENDIF.
ENDIF.

START-OF-SELECTION.
* Check to ensure the counter is always greater than ZERO.
IF P_COUNT < 1.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '036'.
  STOP.
ENDIF.

* Check and ensure either DSO / Update table is checked.
IF P_ODS IS INITIAL AND P_RNONE IS NOT INITIAL.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '045'.

```

```

STOP.
ELSEIF P_ODS IS NOT INITIAL AND P_RNONE IS INITIAL.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '046'.
  STOP.
ENDIF.

```

```

* Check to ensure that the ODS Technical Name is entered.
IF P_ODS IS INITIAL AND P_RNONE IS NOT INITIAL.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '037'.
  STOP.
ENDIF.

```

```

IF P_ODS IS NOT INITIAL.
* Perform operation's on the DSO.
* Check to ensure Update Table is entered.
  IF P_RNONE IS INITIAL AND P_UTNAME IS INITIAL.
    MESSAGE ID 'ZBW' TYPE 'E' NUMBER '044'.
    STOP.
  ENDIF.

```

```

* Check if any of the flags are checked or not.
IF P_LOG IS INITIAL AND P_ACT IS INITIAL AND P_PSA IS INITIAL.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '038'.
ENDIF.

```

```

*Combination of PSA and Active / Change Log not allowed.
IF ( P_LOG IS NOT INITIAL OR P_ACT IS NOT INITIAL ) AND
  P_PSA IS NOT INITIAL.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '039'.
  STOP.
ENDIF.

```

```

IF P_LOG IS NOT INITIAL OR P_ACT IS NOT INITIAL.
* Generate the Selection Screen for the Deletion of Change Log Sub
* Program.

```

```

  PERFORM GENERATE_SELECTION_SCREEN USING      P_CHGLOG
                                                P_ODS
                                                P_ATNAME
                                                P_LOG
                                                P_ACT
                                                P_COUNT
                                                P_CHECK
                                                P_TEST
                                                P_FORCE
                                                CHANGING DEL_GEN_RPT_NAME
                                                DEL_SUB_PROGRAM
                                                DEL_SELECTION_OPTIONS.

```

```

ELSEIF P_PSA IS NOT INITIAL.
* If PSA is to be deleted.
* Check and ensure P_BDAYS is not NULL.
  IF P_BDAYS IS INITIAL.
    MESSAGE ID 'ZBW' TYPE 'E' NUMBER '040'.
    STOP.
  ENDIF.
  PERFORM DELETE_PSA      USING    P_PSA
                                P_BDAYS
                                P_CHGLOG
                                P_TEST
                                P_CHECK
                                P_COUNT.

  ENDIF.
ELSEIF P_RNONE IS INITIAL.
* Perform operations in the Update of the table given.
* Generate the Selection Screen for the Update / Delete Data Records of
* table.
  PERFORM GENERATE_TABLE_SEL_SCREEN
                                USING    P_RUTAB
                                P_RDTAB
                                P_UTNAME
                                P_COUNT
                                P_CHECK
                                P_TEST
                                P_FORCE
                                CHANGING DEL_GEN_RPT_NAME
                                DEL_SUB_PROGRAM
                                DEL_SELECTION_OPTIONS.

  ENDIF.

*&-----*
*& Form      :   GET_STRUCTURE
*& Purpose   :   Sub-routine to fetch the structure of the dynamic table
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*&-----*
FORM GET_STRUCTURE TABLES   IFC
                                USING    P_TABLE TYPE RSODSNAME.
  DATA : IDetails TYPE ABAP_COMPDESCR_TAB,
          XDetails TYPE ABAP_COMPDESCR.
  DATA : REF_TABLE_DES TYPE REF TO CL_ABAP_STRUCTDESCR.
  DATA : XFC TYPE LVC_S_FCAT.

* Get the structure of the table.
REF_TABLE_DES ?=
  CL_ABAP_TYPEDESCR=>DESCRIBE_BY_NAME( P_TABLE ).

```

```

IDetails[] = REF_TABLE_DES->COMPONENTS[].

LOOP AT IDetails INTO XDetails.
  CLEAR XFC.
  XFC-FIELDNAME = XDetails-NAME .
  XFC-DATATYPE = XDetails-TYPE_KIND.
  XFC-INTTYPE = XDetails-TYPE_KIND.
  XFC-INTLEN = XDetails-LENGTH.
  XFC-DECIMALS = XDetails-DECIMALS.
  APPEND XFC TO IFC.
ENDLOOP.

ENDFORM.                "GET_STRUCTURE

*&-----*
*& Form      :   GEN_REPORT_NAME
*& Purpose   :   Used to generate the name of the program.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*
FORM GEN_REPORT_NAME CHANGING E_REPNM   LIKE SY-REPID.

DATA: L_UNIQUEID TYPE RSSG_UNI_IDC25.

* Get Unique Name
CALL FUNCTION 'RSS_SYSTEM_GET_UNIQUE_ID'
  IMPORTING
    E_UNI_IDC25 = L_UNIQUEID
  EXCEPTIONS
    OTHERS      = 8.
IF SY-SUBRC <> 0.
*   Generation Failed
  PERFORM X_MESSAGE USING RS_C_TRUE 'GEN_REPORT_NAME-1-'.
ENDIF.

CONCATENATE 'DEL_' L_UNIQUEID INTO E_REPNM.

ENDFORM.                "GEN_REPORT_NAME

*&-----*
*& Form      :   X_MESSAGE
*& Purpose   :   Populate the exception message.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*

```

```
FORM X_MESSAGE USING I_ABEND TYPE RS_BOOL
                   I_FORM TYPE C.
```

```
DATA: L_REPID LIKE SY-REPID.
```

```
L_REPID = SY-REPID.
```

```
CALL FUNCTION 'RRMS_X_MESSAGE'
```

```
EXPORTING
```

```
  I_ABEND = I_ABEND
```

```
  I_PROGRAM = L_REPID
```

```
  I_TEXT = I_FORM.
```

```
ENDFORM.                "x_message
```

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

```
*& Form      :  GENERATE_SELECTION_SCREEN
```

```
*& Purpose   :  Populates Selection Screen and processes the Selection
*&           :  Screen.
```

```
*& Author    :  Praveen Gudishetty
```

```
*& Date      :  01 / 01 / 2008
```

```
*-----*
```

```
FORM GENERATE_SELECTION_SCREEN USING      RV_CHG_LOG      TYPE RSODSNAME
                                           RV_ODS          TYPE RSDODSOBJECT
                                           RV_ATNAME       TYPE TABNAME
                                           RVC_LOG          TYPE CHECKBOX
                                           RVC_ACT         TYPE CHECKBOX
                                           RV_COUNT         TYPE I
                                           RV_CHECK        TYPE CHECKBOX
                                           RV_TEST         TYPE CHECKBOX
                                           RVC_FORCE        TYPE CHECKBOX
                                           CHANGING RV_DEL_GEN_RPT_NAME TYPE
                                           SY-REPID
                                           RT_DEL_SUB_PROGRAM TYPE
                                           RSDRD_T_ABAPSOURCE
                                           RT_DEL_SELECTION_OPTIONS TYPE
                                           RSDRD_T_SELOPT.
```

```
DATA: INFO_LINE      LIKE ABAPSOURCE,
      TEXT_QUESTION  TYPE ABAPSOURCE VALUE
```

```
      'Do you still want to continue?'
```

```
      ANSWER         TYPE CHAR1.
```

```
DATA: INI_SUB_PROGRAM TYPE RSDRD_T_ABAPSOURCE.
```

```
DATA: TEMP_EXECUTE TYPE CHAR1 VALUE 'X'.
```

```
* Check for the generated report.
```

```
SELECT SINGLE * FROM ZDEL_DSO_REPORT
```

```
  INTO WA_DSO_RPT
```

```
  WHERE DSO_NAME = RV_ODS.
```

```
IF SY-SUBRC = 0 AND RVC_FORCE = 'X'.
```

```

* Force Deletion.
* Update the deletion report and delete the generated report.
  DELETE REPORT WA_DSO_RPT-DELETION_REPORT.

  DELETE FROM ZDEL_DSO_REPORT WHERE DSO_NAME = RV_ODS.
  IF SY-SUBRC <> 0.
    MESSAGE ID 'ZBW' TYPE 'E' NUMBER '043'.
    STOP.
  ENDIF.
  COMMIT WORK.
ELSEIF SY-SUBRC = 0 AND RVC_FORCE <> 'X'.
* Execute the report which is already generated.
* Check if the same global parameters are chosen or not.
  IF WA_DSO_RPT-DEL_CHANGE_LOG = RVC_LOG AND
    WA_DSO_RPT-DEL_ACTIVE_TABLE = RVC_ACT AND
    WA_DSO_RPT-CHANGELOG_TABLE = RV_CHG_LOG AND
    WA_DSO_RPT-ACTIVE_TABLE = RV_ATNAME.
    CLEAR: TEMP_EXECUTE.
    SUBMIT (WA_DSO_RPT-DELETION_REPORT)
      VIA SELECTION-SCREEN AND
      RETURN.
  ELSE.
* Same global parameters not used.
  CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
      TITLEBAR                = 'Different Global Parameters'
*      DIAGNOSE_OBJECT          = ' '
      TEXT_QUESTION            = TEXT_QUESTION
      TEXT_BUTTON_1            = 'YES'
*      ICON_BUTTON_1           = ' '
      TEXT_BUTTON_2            = 'No'
*      ICON_BUTTON_2           = ' '
*      DEFAULT_BUTTON          = '1'
*      DISPLAY_CANCEL_BUTTON   = 'X'
*      USERDEFINED_F1_HELP     = ' '
*      START_COLUMN            = 25
*      START_ROW                = 6
*      POPUP_TYPE              =
*      IV_QUICKINFO_BUTTON_1   = ' '
*      IV_QUICKINFO_BUTTON_2   = ' '
    IMPORTING
      ANSWER                   = ANSWER
*      TABLES                  =
*      PARAMETER                =
    EXCEPTIONS
      TEXT_NOT_FOUND           = 1
      OTHERS                    = 2

```

```

      IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
      ENDIF.

      IF ANSWER = '1'.
        CLEAR: TEMP_EXECUTE.
        SUBMIT (WA_DSO_RPT-DELETION_REPORT)
              VIA SELECTION-SCREEN AND
              RETURN.
      ELSE.
        CLEAR: TEMP_EXECUTE.
        STOP.
      ENDIF.
    ENDIF.
  ENDIF.

  IF TEMP_EXECUTE = 'X'.
* Generate the name of the report.
    PERFORM GEN_REPORT_NAME CHANGING RV_DEL_GEN_RPT_NAME.

* Header
    INFO_LINE = '* Generated Header'.
    APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
    PERFORM GEN_HEADER USING    RV_DEL_GEN_RPT_NAME
                          CHANGING RT_DEL_SUB_PROGRAM.

* Type Pool Declaration.
    INFO_LINE = '* Type Pool Declaration.'.
    APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
    PERFORM GEN_TYPE_POOL CHANGING RT_DEL_SUB_PROGRAM.

* Table Declaration.
    INFO_LINE = '* Generated TABLES'.
    APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
    CLEAR: INFO_LINE.
    CONCATENATE 'TABLES: ' RV_CHG_LOG C_COMMA SPACE RV_ATNAME '.'
                INTO INFO_LINE.
    APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .

* DATA/TYPES block
    INFO_LINE = '* Generated DATA/TYPES '.
    APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
    PERFORM GEN_TYPES_DATA_BLOCK CHANGING RT_DEL_SUB_PROGRAM .

* Selection Screen Option Parameters.

```



```

INFO_LINE = '* Generated SELECTION-SCREEN'.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.
PERFORM FETCH_SELECTION_OPTIONS USING      RV_ODS
                                           RV_ATNAME
                                           RV_CHG_LOG
                                           RV_DEL_GEN_RPT_NAME
                                           RVC_LOG
                                           RVC_ACT
                                           RV_COUNT
                                           RV_CHECK
                                           RV_TEST
                                           CHANGING RT_DEL_SUB_PROGRAM
                                           INI_SUB_PROGRAM.

```

```

APPEND LINES OF INI_SUB_PROGRAM TO RT_DEL_SUB_PROGRAM.

```

* Initialization Event.

```

PERFORM GENERATE_INITIALIZATION CHANGING RT_DEL_SUB_PROGRAM.

```

* Create the Selection Where Clause.

```

APPEND '* Process the report:' TO RT_DEL_SUB_PROGRAM.

```

```

APPEND 'START-OF-SELECTION.'

```

```

                                TO RT_DEL_SUB_PROGRAM.

```

```

LOOP AT T_TYPE_INFOOBJECT INTO WA_TYPE_INFOOBJECT.

```

```

CONCATENATE 'IF ' WA_TYPE_INFOOBJECT-L_CHAR
            ' IS NOT INITIAL.'

```

```

                                INTO INFO_LINE SEPARATED BY SPACE..

```

```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE 'T_TYPE_DEL_SEL-FIELDNM      = ' SPACE C_QUOTE
            WA_TYPE_INFOOBJECT-FIELDNM C_QUOTE ' .'

```

```

                                INTO INFO_LINE.

```

```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE 'T_TYPE_DEL_SEL-INFOOBJECT = ' SPACE C_QUOTE
            WA_TYPE_INFOOBJECT-INFOOBJECT C_QUOTE ' .'

```

```

                                INTO INFO_LINE.

```

```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE 'T_TYPE_DEL_SEL-L_CHAR      = ' SPACE C_QUOTE
            WA_TYPE_INFOOBJECT-L_CHAR C_QUOTE ' .'

```

```

                                INTO INFO_LINE.

```

```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE 'LOOP AT ' WA_TYPE_INFOOBJECT-L_CHAR ' .'

```

```

            INTO INFO_LINE SEPARATED BY SPACE.

```

```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.
APPEND ' CLEAR: L_S_RANGE.' TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-SIGN = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-SIGN.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-OPTION = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-OPTION.' INTO INFO_LINE .
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-LOW = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-LOW.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-HIGH = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-HIGH.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

APPEND: ' APPEND L_S_RANGE TO T_TYPE_DEL_SEL-L_TYPE_RANGE.'
      TO RT_DEL_SUB_PROGRAM,
      ' ENDLLOOP.' TO RT_DEL_SUB_PROGRAM,
      ' APPEND T_TYPE_DEL_SEL.' TO RT_DEL_SUB_PROGRAM,
      ' CLEAR: T_TYPE_DEL_SEL.' TO RT_DEL_SUB_PROGRAM,
      ' ENDIF.' TO RT_DEL_SUB_PROGRAM.

ENDLOOP.

```

* Create the WHERE CLAUSE.

```

APPEND: '* Create the WHERE Clause' TO RT_DEL_SUB_PROGRAM,
      ' CLEAR: L_SEP, L_T_WHERE.' TO RT_DEL_SUB_PROGRAM,
      ' ' TO RT_DEL_SUB_PROGRAM,
      'LOOP AT T_TYPE_DEL_SEL ASSIGNING <L_SX_CHA>.'
      TO RT_DEL_SUB_PROGRAM,
      ' IF NOT L_SEP IS INITIAL.' TO RT_DEL_SUB_PROGRAM,
      ' APPEND L_SEP TO L_T_WHERE.' TO RT_DEL_SUB_PROGRAM,
      ' ENDIF.' TO RT_DEL_SUB_PROGRAM,
      ' ' TO RT_DEL_SUB_PROGRAM,
      ' CLEAR L_T_CONDITION.' TO RT_DEL_SUB_PROGRAM,
      ' PERFORM BUILD_CONDITION_ODSTA ' TO RT_DEL_SUB_PROGRAM,
      ' IN PROGRAM ZBW_DELETE_DSO_CONTENTS'
      TO RT_DEL_SUB_PROGRAM,
      ' USING <L_SX_CHA>' TO RT_DEL_SUB_PROGRAM,
      ' CHANGING L_T_CONDITION' TO RT_DEL_SUB_PROGRAM,

```

```

        '          C_T_MSG.' TO RT_DEL_SUB_PROGRAM,
        ' ' TO RT_DEL_SUB_PROGRAM.
CONCATENATE 'APPEND' C_QUOTE '(' C_QUOTE 'TO L_T_WHERE.'
          INTO INFO_LINE SEPARATED BY SPACE.
APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM,
        ' LOOP AT L_T_CONDITION ASSIGNING <L_CONDITION>.'
          TO RT_DEL_SUB_PROGRAM,
        ' APPEND <L_CONDITION>-LINE TO L_T_WHERE.'
          TO RT_DEL_SUB_PROGRAM,
        ' ENDLLOOP.' TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'APPEND' C_QUOTE ') ' C_QUOTE 'TO L_T_WHERE.'
          INTO INFO_LINE SEPARATED BY SPACE.
APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'L_SEP = ' C_QUOTE 'AND' C_QUOTE '.' INTO INFO_LINE.
APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM,
        ' ENDLLOOP.' TO RT_DEL_SUB_PROGRAM.

* Execute only if selection options are given.
APPEND: 'IF L_T_WHERE IS NOT INITIAL.' TO RT_DEL_SUB_PROGRAM.
* Perform the delete based on WHERE Clause.
APPEND '* Perform the delete based on WHERE Clause.'
          TO RT_DEL_SUB_PROGRAM.

IF RVC_LOG = 'X'.
* Delete Change Log.
APPEND:
        ' PERFORM DELETE_TABLE IN PROGRAM ZBW_DELETE_DSO_CONTENTS'
          TO RT_DEL_SUB_PROGRAM,
        ' USING P_CHGLOG' TO RT_DEL_SUB_PROGRAM,
        ' P_COUNT' TO RT_DEL_SUB_PROGRAM,
        ' L_T_WHERE ' TO RT_DEL_SUB_PROGRAM,
        ' P_TEST. ' TO RT_DEL_SUB_PROGRAM.

ENDIF.

IF RVC_ACT = 'X'.
* Delete Active Table.
APPEND:
        ' PERFORM DELETE_TABLE IN PROGRAM ZBW_DELETE_DSO_CONTENTS'
          TO RT_DEL_SUB_PROGRAM,
        ' USING P_ATNAME' TO RT_DEL_SUB_PROGRAM,
        ' P_COUNT' TO RT_DEL_SUB_PROGRAM,
        ' L_T_WHERE ' TO RT_DEL_SUB_PROGRAM,
        ' P_TEST. ' TO RT_DEL_SUB_PROGRAM.

ENDIF.
APPEND: ' ELSE.' TO RT_DEL_SUB_PROGRAM.

```

```

* WHERE Clause is NULL then raise exception.
  CONCATENATE '          MESSAGE ID ' SPACE C_QUOTE 'ZBW' C_QUOTE
    SPACE ' TYPE ' C_QUOTE 'E' C_QUOTE SPACE ' NUMBER ' SPACE
    C_QUOTE '030' C_QUOTE '.' INTO INFO_LINE.
  APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM,
    ' STOP.' TO RT_DEL_SUB_PROGRAM,
' ENDIF.' TO RT_DEL_SUB_PROGRAM.

* Insert the deletion report.
  CLEAR: WA_DSO_RPT.
  WA_DSO_RPT-DSO_NAME = RV_ODS.
  WA_DSO_RPT-DELETION_REPORT = RV_DEL_GEN_RPT_NAME.
  WA_DSO_RPT-UNAME = SY-UNAME.
  WA_DSO_RPT-DELETION_DATE = SY-DATUM.
  WA_DSO_RPT-DELETION_TIME = SY-UZEIT.
  WA_DSO_RPT-ACTIVE_TABLE = RV_ATNAME.
  WA_DSO_RPT-CHANGELOG_TABLE = RV_CHG_LOG.
  WA_DSO_RPT-DEL_ACTIVE_TABLE = RVC_ACT.
  WA_DSO_RPT-DEL_CHANGE_LOG = RVC_LOG.

  INSERT INTO ZDEL_DSO_REPORT VALUES WA_DSO_RPT.
  COMMIT WORK.

* Execute the dynamic report.
  INSERT REPORT RV_DEL_GEN_RPT_NAME FROM RT_DEL_SUB_PROGRAM.
  SUBMIT (RV_DEL_GEN_RPT_NAME)
    VIA SELECTION-SCREEN
    AND RETURN.

  ELSE.
* Execute the generated report which exists.

  ENDIF.

*   DELETE REPORT  RV_DEL_GEN_RPT_NAME.

ENDFORM.          "GENERATE_SELECTION_SCREEN

*&-----*
*& Form      :   FETCH_SELECTION_OPTIONS
*& Purpose   :   Fetches the Selection Option Objects.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*&-----*

```

```

FORM FETCH_SELECTION_OPTIONS USING   RV_ODS           TYPE RSDODSOBJECT
                                       RV_ATNAME        TYPE TABNAME
                                       RV_CHG_LOG       TYPE RSODSNAME
                                       RV_DEL_GEN_RPT_NAME TYPE
                                       SY-REPID
                                       RVC_LOG         TYPE CHECKBOX
                                       RVC_ACT         TYPE CHECKBOX
                                       RV_COUNT        TYPE I
                                       RV_CHECK        TYPE CHECKBOX
                                       RV_TEST        TYPE CHECKBOX
CHANGING RT_DEL_SUB_PROGRAM TYPE
                                       RSDRD_T_ABAPSOURCE
RT_INI_SUB_PROGRAM TYPE
                                       RSDRD_T_ABAPSOURCE.

DATA: INFO_LINE           TYPE ABAPSOURCE,
      INFO_LINE1(100)    TYPE C,
      WA_INFOOBJECTS     TYPE BAPI6116IO,
      L_CHAR             TYPE CHAR6,
      L_S_SELOPT         TYPE RSDRD_S_SELOPT,
      S_DETAILS          TYPE BAPI6108.

DATA: INFOOBJECTS        TYPE STANDARD TABLE OF BAPI6116IO,
      E_T_SELSCR         TYPE RSDRD_T_ABAPSOURCE.
DATA: TEMP_RV_COUNT(10)  TYPE N.

INFO_LINE = 'AT SELECTION-SCREEN OUTPUT.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

TEMP_RV_COUNT = RV_COUNT.
* Fetch the preliminary selections.
CONCATENATE '           P_LOG AS CHECKBOX DEFAULT ' SPACE C_QUOTE
            RVC_LOG C_QUOTE ',' INTO INFO_LINE.
APPEND: 'PARAMETERS: ' TO RT_DEL_SUB_PROGRAM,
        INFO_LINE      TO RT_DEL_SUB_PROGRAM.

CONCATENATE '           P_CHGLOG TYPE RSODSNAME DEFAULT ' SPACE
            C_QUOTE RV_CHG_LOG C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE      TO RT_DEL_SUB_PROGRAM.

CONCATENATE '           P_ACT AS CHECKBOX DEFAULT ' SPACE C_QUOTE
            RVC_ACT C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE      TO RT_DEL_SUB_PROGRAM.

CONCATENATE '           P_ATNAME TYPE TABNAME DEFAULT' SPACE C_QUOTE
            RV_ATNAME C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE      TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE '          P_COUNT TYPE I DEFAULT' SPACE C_QUOTE
          TEMP_RV_COUNT C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE          TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE '          P_CHECK AS CHECKBOX DEFAULT' SPACE C_QUOTE
          RV_CHECK C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE          TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE '          P_TEST AS CHECKBOX DEFAULT' SPACE C_QUOTE
          RV_TEST C_QUOTE '.' INTO INFO_LINE.
APPEND:  INFO_LINE          TO RT_DEL_SUB_PROGRAM.

```

* Change Log Declaration.

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_LOG' C_QUOTE '.'
          INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE '.'
          INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Delete Change Log'
          C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

*

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_CHGLOG' C_QUOTE '.'
          INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE '.'
          INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Change Log Table Name:'
          C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTXt TO L_T_SELTXtS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

*

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-NAME = ' C_QUOTE 'P_ACT' C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-KIND = ' C_QUOTE C_P C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-TEXT = ' C_QUOTE 'Delete Active Table'
      C_QUOTE ' .' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTXt TO L_T_SELTXtS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

*

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-NAME = ' C_QUOTE 'P_ATNAME' C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-KIND = ' C_QUOTE C_P C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-TEXT = ' C_QUOTE 'Active Table Name:'
      C_QUOTE ' .' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTXt TO L_T_SELTXtS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

*
*

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXt-NAME = ' C_QUOTE 'P_COUNT' C_QUOTE ' .'
      INTO INFO_LINE.

```

```

APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-KIND = ' C_QUOTE C_P C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-TEXT = ' C_QUOTE 'Delete Record Counter:'
      C_QUOTE ' .' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX-T TO L_T_SELTX-TS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
*

*

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-NAME = ' C_QUOTE 'P_CHECK' C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-KIND = ' C_QUOTE C_P C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-TEXT = ' C_QUOTE 'Print Spool'
      C_QUOTE ' .' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX-T TO L_T_SELTX-TS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
*

*

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-NAME = ' C_QUOTE 'P_TEST' C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-T-KIND = ' C_QUOTE C_P C_QUOTE ' .'
      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```



```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Test Run'
           C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
*
CALL FUNCTION 'BAPI_ODSO_GETDETAIL'
  EXPORTING
    OBJVERS          = RS_C_OBJVERS-ACTIVE
    ODSOBJECT        = RV_ODS
* IMPORTING
*   DETAILS          =
*   RETURN           =
  TABLES
    INFOOBJECTS      = INFOOBJECTS
*   NAVIGATIONATTRIBUTES =
*   INDEXES          =
*   INDEXESINFOBJECTS =

.

SORT INFOOBJECTS BY IOBJTP.
LOOP AT INFOOBJECTS INTO WA_INFOOBJECTS WHERE IOBJTP <> C_KEYFIGURE.
  CLEAR: L_CHAR.
  L_CHAR      = SY-TABIX.
  SHIFT L_CHAR LEFT DELETING LEADING SPACE.
  CONCATENATE RSDQ_C_VARNM-CHAR L_CHAR INTO L_CHAR.
  CLEAR: INFO_LINE.

* Fetch the InfoObject Properties.
CALL FUNCTION 'BAPI_IOBJ_GETDETAIL'
  EXPORTING
    VERSION          = RS_C_OBJVERS-ACTIVE
    INFOOBJECT       = WA_INFOOBJECTS-INFOOBJECT
  IMPORTING
    DETAILS          = S_DETAILS
*   RETURN           =
*   TABLES
*   COMPOUNDS        =
*   ATTRIBUTES       =
*   NAVIGATIONATTRIBUTES =
*   ATRNAVINFOPROVIDER =
*   HIERARCHYCHARACTERISTICS =
*   ELIMINATION      =

```

```

CLEAR: INFO_LINE.
CONCATENATE RV_ATNAME '-' S_DETAILS-FIELDNM INTO INFO_LINE1.
CONCATENATE 'SELECT-OPTIONS: ' L_CHAR ' FOR '
  INFO_LINE1 '.' INTO INFO_LINE SEPARATED BY SPACE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE L_CHAR C_QUOTE '.'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_S C_QUOTE '.'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE S_DETAILS-TEXTSHORT
  C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: WA_TYPE_INFOOBJECT.
WA_TYPE_INFOOBJECT-INFOOBJECT = WA_INFOOBJECTS-INFOOBJECT.
WA_TYPE_INFOOBJECT-FIELDNM = S_DETAILS-FIELDNM.
WA_TYPE_INFOOBJECT-L_CHAR = L_CHAR.
APPEND WA_TYPE_INFOOBJECT TO T_TYPE_INFOOBJECT.

ENDLOOP.
* Function to super impose the field titles.
APPEND: ' CALL FUNCTION ''SELECTION_TEXTS_MODIFY''' TO
  RT_INI_SUB_PROGRAM,
  ' EXPORTING ' TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE '          PROGRAM          = ' C_QUOTE
  RV_DEL_GEN_RPT_NAME C_QUOTE INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

APPEND: ' TABLES ' TO RT_INI_SUB_PROGRAM,
  '          SELTEXTS          = L_T_SELTXS' TO
  RT_INI_SUB_PROGRAM,
  ' EXCEPTIONS ' TO RT_INI_SUB_PROGRAM,

```

```

'          PROGRAM_NOT_FOUND          = 1 ' TO
                                           RT_INI_SUB_PROGRAM,
'          PROGRAM_CANNOT_BE_GENERATED = 2 ' TO
                                           RT_INI_SUB_PROGRAM,
'          OTHERS                      = 3. ' TO
                                           RT_INI_SUB_PROGRAM.

```

```
APPEND: ' LOOP AT SCREEN.' TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE '      IF SCREEN-NAME = ' C_QUOTE 'P_CHGLOG' C_QUOTE ' OR
  SCREEN-NAME = ' C_QUOTE 'P_ATNAME' C_QUOTE 'OR' INTO INFO_LINE.
```

```
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE 'SCREEN-NAME = ' C_QUOTE 'P_CHECK' C_QUOTE ' OR '
  INTO INFO_LINE.
```

```
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE 'SCREEN-NAME = ' C_QUOTE 'P_LOG' C_QUOTE ' OR
  SCREEN-NAME = ' C_QUOTE 'P_ACT' C_QUOTE '.' INTO INFO_LINE.
```

```
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE '      SCREEN-INPUT = ' C_QUOTE '0' C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE '      SCREEN-OUTPUT = ' C_QUOTE '1' C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE '      SCREEN-ACTIVE = ' C_QUOTE '1' C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM,
```

```
      '      MODIFY SCREEN.' TO RT_INI_SUB_PROGRAM,
```

```
      '      ENDIF.' TO RT_INI_SUB_PROGRAM,
```

```
      '      ENLOOP.' TO RT_INI_SUB_PROGRAM.
```

```
ENDFORM.                "fetch_selection_options
```

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

```
*& Form      :  GENERATE_INITIALIZATION
```

```
*& Purpose   :  Generates the Initialization Section of the report..
```

```
*& Author    :  Praveen Gudishetty
```

```
*& Date      :  01 / 01 / 2008
```

```
*-----*
```

```
FORM GENERATE_INITIALIZATION CHANGING RT_DEL_SUB_PROGRAM TYPE
```

```
      RSDRD_T_ABAPSOURCE.
```

```

DATA: L_STRING          TYPE ABAPSOURCE.
L_STRING = 'INITIALIZATION.'.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.
CLEAR: L_STRING.
CONCATENATE 'SET TITLEBAR ' C_QUOTE 'SELECTION' C_QUOTE
  ' OF PROGRAM ' C_QUOTE 'SAPLRSDRD' C_QUOTE ' WITH ' INTO L_STRING.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.

CONCATENATE C_QUOTE 'Selection Options for deletion' C_QUOTE ' '
                                                    INTO
  L_STRING SEPARATED BY SPACE.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.
ENDFORM.                "generate_initialization
*&-----*
*& Form GEN_TYPE_POOL
*&-----*
*      text
*-----*
FORM GEN_TYPE_POOL CHANGING RT_DEL_SUB_PROGRAM TYPE RSDRD_T_ABAPSOURCE.
DATA: L_STRING          TYPE ABAPSOURCE.
L_STRING = 'TYPE-POOLS: ABAP, RSDQ, RSDRD.'.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.

APPEND: 'TYPES: BEGIN OF TYPE_DEL_SEL,' TO RT_DEL_SUB_PROGRAM,
  '      L_TYPE_RANGE TYPE RSDRD_T_RANGE,' TO RT_DEL_SUB_PROGRAM,
  '      INFOBJECT   TYPE RSIOBJNM,' TO RT_DEL_SUB_PROGRAM,
  '      FIELDNM     TYPE RSIOBJFIELDNM,' TO RT_DEL_SUB_PROGRAM,
  '      L_CHAR      TYPE CHAR6,' TO RT_DEL_SUB_PROGRAM,
  '      END OF TYPE_DEL_SEL.' TO RT_DEL_SUB_PROGRAM,
  'TYPES: TYPE_T_DEL_SEL TYPE STANDARD TABLE OF TYPE_DEL_SEL.'
  TO RT_DEL_SUB_PROGRAM.
* Declare the Types.
CLEAR: L_STRING.

ENDFORM.                "GEN_TYPE_POOL

*&-----*
*& Form      : GEN_HEADER
*& Purpose   : Used to generate the Header Details of the program.
*& Author    : Praveen Gudishetty
*& Date      : 01 / 01 / 2008
*-----*
FORM GEN_HEADER USING I_REPNM TYPE SY-REPID
  CHANGING E_T_REPORT TYPE RSDRD_T_ABAPSOURCE.

DATA:
  L_STRING TYPE ABAPSOURCE.

```

```

APPEND:
  '*-----' TO E_T_REPORT,
  '* generated REPORT (RSDRD)' TO E_T_REPORT,
  '* (should be deleted after use)' TO E_T_REPORT.
CONCATENATE
  '* Generated by..:' SY-UNAME INTO L_STRING SEPARATED BY ' '.
APPEND L_STRING TO E_T_REPORT.
CONCATENATE
  '* Date/Time.....:' SY-DATUM '/' SY-UZEIT
  INTO L_STRING SEPARATED BY ' '.
APPEND L_STRING TO E_T_REPORT.
APPEND
  '*-----' TO E_T_REPORT.
CONCATENATE 'REPORT' I_REPNM ' '.
  INTO L_STRING SEPARATED BY SPACE.
APPEND L_STRING TO E_T_REPORT.

ENDFORM.                "GEN_HEADER

*&-----*
*& Form      :   GEN_TYPES_DATA_BLOCK
*& Purpose   :   Used to generate the Data Declaration of the program.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*
FORM GEN_TYPES_DATA_BLOCK CHANGING C_T_REPORT TYPE RSDRD_T_ABAPSOURCE.
APPEND:
'DATA:'
' L_INTERACTIVE      TYPE RS_BOOL, ' TO C_T_REPORT,
' L_ANSWER(1)        TYPE C, ' TO C_T_REPORT,
' L_TYPE(1)          TYPE C, ' TO C_T_REPORT,
' L_PARALLEL         TYPE I, ' TO C_T_REPORT,
' L_NO_OF_ROWS       TYPE I, ' TO C_T_REPORT,
' L_NO_OF_ROWS_C(10) TYPE C, ' TO C_T_REPORT,
' L_TITEL            TYPE RS_CHAR72, ' TO C_T_REPORT,
' L_TEXT1            TYPE RS_CHAR72, ' TO C_T_REPORT,
' L_TEXT2            TYPE RS_CHAR72, ' TO C_T_REPORT,
' L_S_RANGE          TYPE RSDRD_S_RANGE, ' TO C_T_REPORT,
' L_S_SELTXT         TYPE RSDQ_S_SELTEXT, ' TO C_T_REPORT,
' L_SX_SEL           TYPE RSDRD_SX_SEL, ' TO C_T_REPORT,
' L_T_MSG            TYPE RS_T_MSG, ' TO C_T_REPORT,
' L_THX_SEL          TYPE RSDRD_THX_SEL, ' TO C_T_REPORT,
' L_T_SELTXTS        TYPE RSDQ_T_SELTEXT ' TO C_T_REPORT,
'                   WITH HEADER LINE, ' TO C_T_REPORT,
' T_TYPE_DEL_SEL     TYPE TYPE_T_DEL_SEL with header line, '
'                   TO C_T_REPORT,
' L_SEP(10)          TYPE C, ' TO C_T_REPORT,

```

```

' L_T_WHERE          TYPE RSDRD_T_ABAPSOURCE, '          TO C_T_REPORT,
' L_T_CONDITION      TYPE RSDRD_T_ABAPSOURCE, '          TO C_T_REPORT,
' C_T_MSG            TYPE RS_T_MSG, '                    TO C_T_REPORT,
' L_S_MSG            TYPE RS_S_MSG. '                    TO C_T_REPORT,

'FIELD-SYMBOLS: '          TO C_T_REPORT,
' <L_SX_CHA>         TYPE TYPE_DEL_SEL, '                TO C_T_REPORT,
' <L_CONDITION>     TYPE ABAPSOURCE. '                  TO C_T_REPORT.

ENDFORM.                  " GEN_TYPES_DATA_BLOCK

```

```

*&-----*
*& Form      :   DELETE_TABLE
*& Purpose   :   Used to Delete the data from the table specified using
*&           :   the Selection Options provided.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*
FORM DELETE_TABLE USING   RVP_CHGLOG TYPE RSODSNAME
                        RVP_COUNT  TYPE I
                        RV_SLOG    TYPE RSDRD_T_ABAPSOURCE
                        RVP_TEST   TYPE CHECKBOX.

DATA: TABLE_NAME      TYPE RSODSNAME,
      V_COUNTER        TYPE I,
      TOTAL_COUNT(20) TYPE N.

DATA: IFC              TYPE LVC_T_FCAT,
      IT_CHANGELOG_TABLE TYPE REF TO DATA.

FIELD-SYMBOLS: <FS_IT_ACTIVE_TABLE> TYPE STANDARD TABLE,
               <FS_IT_CHANGELOG_TABLE> TYPE STANDARD TABLE,
               <FS_SLOG>              TYPE ABAPSOURCE.

V_COUNTER = 0.

* Check if it is a Test Run or not.
IF RVP_TEST <> 'X'.
* Do not allow this report to be run in foreground
IF SY-BATCH = SPACE.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '031'.
  STOP.
ENDIF.

* Get the structure of the Dynamic Table
TABLE_NAME = RVP_CHGLOG.
PERFORM GET_STRUCTURE TABLES IFC
      USING TABLE_NAME .

* Create the table type for the Change Log.

```

```

CALL METHOD CL_ALV_TABLE_CREATE=>CREATE_DYNAMIC_TABLE
  EXPORTING
    IT_FIELDCATALOG = IFC
  IMPORTING
    EP_TABLE        = IT_CHANGELOG_TABLE.

  ASSIGN IT_CHANGELOG_TABLE->* TO <FS_IT_CHANGELOG_TABLE>.
ENDIF.

WRITE:/ 'Delete from ', RVP_CHGLOG, ' using selection option:'.
* Print the Where Clause:
LOOP AT RV_SLOG ASSIGNING <FS_SLOG>.
  WRITE:/ <FS_SLOG>.
ENDLOOP.

* Fetch the total record count to be deleted:
SELECT COUNT(*) FROM (RVP_CHGLOG)
      INTO TOTAL_COUNT
      WHERE (RV_SLOG).

* Check if it is a Test Run or not.
IF RVP_TEST <> 'X' AND SY-SUBRC = 0.
* Delete from Change log / Active table
DO.
  V_COUNTER = V_COUNTER + 1.
  SELECT * FROM (RVP_CHGLOG)
        INTO CORRESPONDING FIELDS OF TABLE <FS_IT_CHANGELOG_TABLE>
        UP TO RVP_COUNT ROWS
        WHERE (RV_SLOG).

  IF SY-SUBRC <> 0.
    EXIT.
  ENDIF.

  DELETE (RVP_CHGLOG)
    FROM TABLE <FS_IT_CHANGELOG_TABLE>.

  CALL FUNCTION 'DB_COMMIT'.
  WRITE:/ 'Deletion from table ', RVP_CHGLOG, 'SUCCESSFULL Iteration:'
    , V_COUNTER.
  CLEAR: <FS_IT_CHANGELOG_TABLE>.
ENDDO.
ELSEIF SY-SUBRC <> 0 AND RVP_TEST <> 'X' .
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '042'.
ENDIF.
* Print the total records deleted.

```

```

ULINE.
WRITE:/ 'Total record count being deleted: ', TOTAL_COUNT.
ULINE.
ENDFORM.                "DELETE_TABLE

*&-----*
*& Form      :   BUILD_CONDITION_ODSTA
*& Purpose   :   Used to generate the WHERE clause based on the Selection
*&           :   Options.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*&-----*
FORM BUILD_CONDITION_ODSTA USING      I_SX_CHA      TYPE TYPE_DEL_SEL
                                     CHANGING E_T_CODE TYPE RSDRD_T_ABAPSOURCE
                                     C_T_MSG      TYPE RS_T_MSG.

DATA: L_SEP(10)      TYPE C,
      L_LOW          TYPE ABAPSOURCE,
      L_HIGH         TYPE ABAPSOURCE,
      L_S_CONDITION TYPE ABAPSOURCE.

FIELD-SYMBOLS: <L_S_RANGE> TYPE RSDRD_S_RANGE.

CLEAR E_T_CODE.
L_SEP = SPACE.
LOOP AT I_SX_CHA-L_TYPE_RANGE ASSIGNING <L_S_RANGE>.
*   Excluding wird nicht unterstützt
    IF <L_S_RANGE>-SIGN NE 'I'.
        MESSAGE ID 'ZBW' TYPE 'E' NUMBER '041'.
        STOP.
        PERFORM MSG_APPEND
            USING      'E' '100' 'RSDRD' 'BUILD_CONDITION_ODSTA-1-'
                    <L_S_RANGE>-SIGN SPACE
            CHANGING C_T_MSG.
    ENDIF.

    CONCATENATE L_SEP I_SX_CHA-FIELDNM
                INTO L_S_CONDITION
                SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO E_T_CODE.

    WRITE <L_S_RANGE>-LOW TO L_LOW NO-GROUPING LEFT-JUSTIFIED.
    PERFORM CHANGE_QUOTES
        CHANGING L_LOW.
    IF L_LOW <> SPACE.
        CONCATENATE '' L_LOW '' INTO L_LOW.
    ELSE.
        L_LOW = ' ' .

```



```

ENDIF.

IF NOT <L_S_RANGE>-HIGH IS INITIAL.
  WRITE <L_S_RANGE>-HIGH TO L_HIGH NO-GROUPING LEFT-JUSTIFIED.
  PERFORM CHANGE_QUOTES
    CHANGING L_HIGH.
  IF L_HIGH <> SPACE.
    CONCATENATE '' L_HIGH '' INTO L_HIGH.
  ELSE.
    L_HIGH = '' '.
  ENDIF.
ENDIF.

```

* If patterns are used for Selection Option then stop execution.

```

IF L_LOW CS C_STAR OR L_HIGH CS C_STAR.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '032'.
  STOP.
ELSEIF L_LOW CS C_PERCENT OR L_HIGH CS C_PERCENT.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '033'.
  STOP.
ENDIF.

```

* Korrekte Operation auswählen

```

CASE <L_S_RANGE>-OPTION.
  WHEN RS_C_RANGE_OPT-EQUAL.
    CONCATENATE '=' L_LOW
      INTO L_S_CONDITION
      SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO E_T_CODE.

  WHEN RS_C_RANGE_OPT-NOTEQUAL.
    CONCATENATE '<>' L_LOW
      INTO L_S_CONDITION
      SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO E_T_CODE.

  WHEN RS_C_RANGE_OPT-GREATEREQUAL.
    CONCATENATE '>=' L_LOW
      INTO L_S_CONDITION
      SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO E_T_CODE.

  WHEN RS_C_RANGE_OPT-LESSEQUAL.
    CONCATENATE '<=' L_LOW
      INTO L_S_CONDITION
      SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO E_T_CODE.

```

```

WHEN RS_C_RANGE_OPT-GREATER.
  CONCATENATE '>' L_LOW
              INTO L_S_CONDITION
              SEPARATED BY SPACE.
  APPEND L_S_CONDITION TO E_T_CODE.

WHEN RS_C_RANGE_OPT-LESS.
  CONCATENATE '<' L_LOW
              INTO L_S_CONDITION
              SEPARATED BY SPACE.
  APPEND L_S_CONDITION TO E_T_CODE.

WHEN RS_C_RANGE_OPT-BETWEEN.
  CONCATENATE 'BETWEEN' L_LOW
              INTO L_S_CONDITION
              SEPARATED BY SPACE.
  APPEND L_S_CONDITION TO E_T_CODE.
  CONCATENATE 'AND' L_HIGH
              INTO L_S_CONDITION
              SEPARATED BY SPACE.
  APPEND L_S_CONDITION TO E_T_CODE.

WHEN RS_C_RANGE_OPT-NOTBETWEEN.
  CONCATENATE 'NOT BETWEEN' L_LOW
              INTO L_S_CONDITION
              SEPARATED BY SPACE.
  APPEND L_S_CONDITION TO E_T_CODE.
  CONCATENATE 'AND' L_HIGH
              INTO L_S_CONDITION
              SEPARATED BY SPACE.
  APPEND L_S_CONDITION TO E_T_CODE.

WHEN OTHERS.
  PERFORM MSG_APPEND
    USING 'E' '100' 'RSDRD' 'BUILD_CONDITION_ODSTA-3-'
        <L_S_RANGE>-OPTION SPACE
    CHANGING C_T_MSG.
ENDCASE.

L_SEP = 'OR'.
ENDLOOP.
ENDFORM.                "BUILD_CONDITION_ODSTA

*&-----*
*& Form      :   MSG_APPEND
*& Purpose   :   Used to raise exceptions.

```

*& Author : Praveen Gudishetty

*& Date : 01 / 01 / 2008

```
FORM MSG_APPEND USING      I_MSGTY  TYPE RS_S_MSG-MSGTY
                           I_MSGNO  TYPE RS_S_MSG-MSGNO
                           I_MSGV1  TYPE C
                           I_MSGV2  TYPE C
                           I_MSGV3  TYPE C
                           I_MSGV4  TYPE C
                           CHANGING C_T_MSG  TYPE RS_T_MSG.
```

DATA: L_S_MSG TYPE RS_S_MSG.

L_S_MSG-MSGTY = I_MSGTY.

L_S_MSG-MSGID = 'DBMAN'.

L_S_MSG-MSGNO = I_MSGNO.

L_S_MSG-MSGV1 = I_MSGV1.

L_S_MSG-MSGV2 = I_MSGV2.

L_S_MSG-MSGV3 = I_MSGV3.

L_S_MSG-MSGV4 = I_MSGV4.

L_S_MSG-DETTLEVEL = '1'. " anzeigen sofort

L_S_MSG-PROBCLASS = '1'. " Nachricht sehr wichtig

APPEND L_S_MSG TO C_T_MSG.

CLEAR: SY-MSGTY, SY-MSGNO, SY-MSGID,
 SY-MSGV1, SY-MSGV2, SY-MSGV3, SY-MSGV4.

IF I_MSGTY = 'E'.

RAISE X_MESSAGE.

ENDIF.

ENDFORM. " MSG_APPEND

&-----

*& Form : CHANGE_QUOTES

*& Purpose : Replace Single Quotes by Double Quotes.

*& Author : Praveen Gudishetty

*& Date : 01 / 01 / 2008

```
FORM CHANGE_QUOTES CHANGING C_STRING TYPE ABAPSOURCE.
```

DATA: L_LEN TYPE I,
 L_SOURCE TYPE I,
 L_TARGET TYPE I,
 L_TSTR TYPE ABAPSOURCE.

FIELD-SYMBOLS: <L_SSTR> TYPE C.

* Quotation marks by double quotation mark replace

```
IF C_STRING CA '''.
  L_LEN    = STRLEN( C_STRING ).
  L_SOURCE = 0.
  L_TARGET = 0.
  WHILE L_SOURCE < L_LEN.
    ASSIGN C_STRING+L_SOURCE(1) TO <L_SSTR>.
    IF <L_SSTR> EQ '''.
      L_TSTR+L_TARGET = '''.
      ADD 1 TO L_TARGET.
    ENDIF.
    L_TSTR+L_TARGET = <L_SSTR>.
    ADD 1 TO L_TARGET.
    ADD 1 TO L_SOURCE.
  ENDWHILE.
  C_STRING = L_TSTR.
ENDIF.
```

ENDFORM. " CHANGE_QUOTES

&-----

```
*& Form      :  DELETE_PSA
*& Purpose   :  Deletes PSA data based on number of days.
*& Author    :  Praveen Gudishetty
*& Date      :  01 / 01 / 2008
```

```
FORM DELETE_PSA    USING    RVP_PSA        TYPE CHECKBOX
                          RVP_BDAYS    TYPE I
                          RVP_CHGLOG    TYPE RSODSNAME
                          RVP_TEST     TYPE CHECKBOX
                          RVP_CHECK     TYPE CHECKBOX
                          RVP_COUNT     TYPE I.
```

```
DATA: DEL_DATE        TYPE SYDATUM,
      L_S_CONDITION   TYPE ABAPSOURCE,
      L_LOW           TYPE ABAPSOURCE,
      L_SEP           TYPE CHAR10,
      RT_SLOG         TYPE RSDRD_T_ABAPSOURCE.
```

* Fetch the date till which the records could be deleted.

```
DEL_DATE = SY-DATUM - RVP_BDAYS.
```

* Fetch the requests currently in the PSA with counts.

```
SELECT DISTINCT REQUEST AS RNR COUNT(*) AS RECORDS FROM (RVP_CHGLOG)
      INTO TABLE T_H_REQUEST
```

```

WHERE REQUEST LIKE C_PSA
GROUP BY REQUEST.

IF SY-SUBRC = 0.
* Fetch the attributes of the requests in PSA.
  SELECT RNR RECORDS DATUM UZEIT TSTATUS RECORDS FROM RSREQDONE
    INTO TABLE T_RSREQDONE
    FOR ALL ENTRIES IN T_H_REQUEST
    WHERE RNR = T_H_REQUEST-RNR.

* Delete the requests which don't accept the criteria.
  LOOP AT T_RSREQDONE ASSIGNING <FS_REQUEST>.
    IF <FS_REQUEST>-DATUM <= DEL_DATE.
      READ TABLE T_H_REQUEST INTO WA_RSREQDONE
        WITH KEY RNR = <FS_REQUEST>-RNR.
      <FS_REQUEST>-RECORDS = WA_RSREQDONE-RECORDS.
    ELSE.
      <FS_REQUEST>-RNR = SY-TABIX.
    ENDIF.
  ENDLOOP.
  REFRESH: T_H_REQUEST.

* Delete all requests which dont fall in the criteria.
  DELETE T_RSREQDONE WHERE RNR NP 'REQU*'.

* Create the Where Clause.
  IF RVP_CHECK = 'X'.
    WRITE:/ C_LINE.
    WRITE:/ 'PSA Requests being deleted are as follows:'.
    WRITE:/ C_LINE.
  ENDIF.
  L_SEP = SPACE.
  LOOP AT T_RSREQDONE ASSIGNING <FS_REQUEST>.
    CONCATENATE L_SEP 'REQUEST'
      INTO L_S_CONDITION SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO RT_SLOG.

    WRITE <FS_REQUEST>-RNR TO L_LOW NO-GROUPING LEFT-JUSTIFIED.
    PERFORM CHANGE_QUOTES
      CHANGING L_LOW.
    IF L_LOW <> SPACE.
      CONCATENATE '''' L_LOW '''' INTO L_LOW.
    ELSE.
      L_LOW = '' '''.
    ENDIF.

    CONCATENATE '=' L_LOW

```

```

        INTO L_S_CONDITION SEPARATED BY SPACE.
    APPEND L_S_CONDITION TO RT_SLOG.

    L_SEP = 'OR'.
    IF RVP_CHECK = 'X'.
        WRITE:/ <FS_REQUEST>-RNR, SPACE, <FS_REQUEST>-DATUM, SPACE,
            <FS_REQUEST>-UZEIT, SPACE, <FS_REQUEST>-RECORDS.
    ENDIF.
ENDLOOP.
IF RVP_CHECK = 'X'.
    WRITE:/ C_LINE.
ENDIF.

IF RT_SLOG IS NOT INITIAL.
* Delete the contents, only if selection options are there.
    PERFORM DELETE_TABLE USING    RVP_CHGLOG
                                RVP_COUNT
                                RT_SLOG
                                RVP_TEST.

    ENDIF.
ELSE.
* No requests in PSA.
    WRITE:/ C_LINE.
    WRITE:/ 'No requests deleted as PSA is empty.'.
    WRITE:/ C_LINE.
ENDIF.
ENDFORM.                "DELETE_PSA

*&-----*
*& Form      :   GENERATE_TABLE_SEL_SCREEN
*& Purpose   :   Populates Selection Screen and processes the Selection
*&           :   Screen for Table.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 18 / 2008
*-----*
FORM GENERATE_TABLE_SEL_SCREEN
        USING    RP_RUTAB          TYPE CHAR1
                RP_RDTAB          TYPE CHAR1
                RP_UTNAME         TYPE TABNAME
                RV_COUNT          TYPE I
                RV_CHECK          TYPE CHECKBOX
                RV_TEST           TYPE CHECKBOX
                RVC_FORCE         TYPE CHECKBOX
        CHANGING RV_DEL_GEN_RPT_NAME TYPE SY-REPID
                RT_DEL_SUB_PROGRAM TYPE
                                RSDRD_T_ABAPSOURCE

```

```

                                RT_DEL_SELECTION_OPTIONS TYPE
                                RSDRD_T_SELOPT.

DATA: INFO_LINE           LIKE ABAPSOURCE,
      TEXT_QUESTION      TYPE ABAPSOURCE VALUE
                                'Do you still want to continue?',
      ANSWER             TYPE CHAR1.

DATA: INI_SUB_PROGRAM TYPE RSDRD_T_ABAPSOURCE.
DATA: TEMP_EXECUTE TYPE CHAR1 VALUE 'X'.

* Check for the generated report.
SELECT SINGLE * FROM ZDEL_DSO_REPORT
  INTO WA_DSO_RPT
  WHERE DSO_NAME = RP_UTNAME.
IF SY-SUBRC = 0 AND RVC_FORCE = 'X'.
* Force Deletion.
* Update the deletion report and delete the generated report.
  DELETE REPORT WA_DSO_RPT-DELETION_REPORT.

  DELETE FROM ZDEL_DSO_REPORT WHERE DSO_NAME = RP_UTNAME.
  IF SY-SUBRC <> 0.
    MESSAGE ID 'ZBW' TYPE 'E' NUMBER '043'.
    STOP.
  ENDIF.
  COMMIT WORK.
ELSEIF SY-SUBRC = 0 AND RVC_FORCE <> 'X'.
* Execute the report which is already generated.
* Check if the same global parameters are chosen or not.
  IF WA_DSO_RPT-DEL_TABLE = RP_RUTAB AND
    WA_DSO_RPT-UPDATE_TABLE = RP_RDTAB.
    CLEAR: TEMP_EXECUTE.
    SUBMIT (WA_DSO_RPT-DELETION_REPORT)
      VIA SELECTION-SCREEN AND
      RETURN.
  ELSE.
* Same global parameters not used.
    CALL FUNCTION 'POPUP_TO_CONFIRM'
      EXPORTING
        TITLEBAR                = 'Different Global Parameters'
*        DIAGNOSE_OBJECT         = ' '
        TEXT_QUESTION            = TEXT_QUESTION
        TEXT_BUTTON_1           = 'YES'
*        ICON_BUTTON_1          = ' '
        TEXT_BUTTON_2           = 'No'
*        ICON_BUTTON_2         = ' '
*        DEFAULT_BUTTON         = '1'
*        DISPLAY_CANCEL_BUTTON  = 'X'
*        USERDEFINED_F1_HELP    = ' '

```

```

*          START_COLUMN          = 25
*          START_ROW             = 6
*          POPUP_TYPE            =
*          IV_QUICKINFO_BUTTON_1 = ' '
*          IV_QUICKINFO_BUTTON_2 = ' '
IMPORTING
  ANSWER          = ANSWER
* TABLES
* PARAMETER      =
EXCEPTIONS
  TEXT_NOT_FOUND = 1
  OTHERS         = 2
.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

IF ANSWER = '1'.
  CLEAR: TEMP_EXECUTE.
  SUBMIT (WA_DSO_RPT-DELETION_REPORT)
        VIA SELECTION-SCREEN AND
        RETURN.
ELSE.
  CLEAR: TEMP_EXECUTE.
  STOP.
ENDIF.
ENDIF.
ENDIF.

IF TEMP_EXECUTE = 'X'.
* Generate the name of the report.
  PERFORM GEN_TABLE_REPORT_NAME CHANGING RV_DEL_GEN_RPT_NAME.

* Header
  INFO_LINE = '* Generated Header'.
  APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
  PERFORM GEN_HEADER USING RV_DEL_GEN_RPT_NAME
        CHANGING RT_DEL_SUB_PROGRAM.

* Type Pool Declaration.
  INFO_LINE = '* Type Pool Declaration.'.
  APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
  PERFORM GEN_TYPE_POOL CHANGING RT_DEL_SUB_PROGRAM.

* Table Declaration.
  INFO_LINE = '* Generated TABLES'.

```



```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
CLEAR: INFO_LINE.
CONCATENATE 'TABLES: ' RP_UTNAME '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .

* DATA/TYPES block
INFO_LINE = '* Generated DATA/TYPES '.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM .
PERFORM GEN_TYPERES_DATA_BLOCK CHANGING RT_DEL_SUB_PROGRAM .

* Selection Screen Option Parameters.
INFO_LINE = '* Generated SELECTION-SCREEN'.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.
PERFORM FETCH_TABLE_SELECTION_OPTIONS
                USING      RP_RUTAB
                        RP_RDTAB
                        RP_UTNAME
                        RV_COUNT
                        RV_CHECK
                        RV_TEST
                        RVC_FORCE
                        RV_DEL_GEN_RPT_NAME
                CHANGING RT_DEL_SUB_PROGRAM
                        INI_SUB_PROGRAM.

APPEND LINES OF INI_SUB_PROGRAM TO RT_DEL_SUB_PROGRAM.

* Initialization Event.
PERFORM GENERATE_TABLE_INITIALIZATION USING      RP_RUTAB
                RP_RDTAB
                CHANGING RT_DEL_SUB_PROGRAM.

*
* Create the Selection Where Clause.
APPEND '* Process the report:' TO RT_DEL_SUB_PROGRAM.
APPEND 'START-OF-SELECTION.'
                TO RT_DEL_SUB_PROGRAM.
LOOP AT T_TYPE_INFOOBJECT INTO WA_TYPE_INFOOBJECT.
CONCATENATE 'IF ' WA_TYPE_INFOOBJECT-L_CHAR
        ' IS NOT INITIAL.'
                INTO INFO_LINE SEPARATED BY SPACE..
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'T_TYPE_DEL_SEL-FIELDNM      = ' SPACE C_QUOTE
        WA_TYPE_INFOOBJECT-FIELDNM C_QUOTE '.'
                INTO INFO_LINE.

```

```

APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'T_TYPE_DEL_SEL-INFOOBJECT = ' SPACE C_QUOTE
  WA_TYPE_INFOOBJECT-INFOOBJECT C_QUOTE ' .'
                                INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'T_TYPE_DEL_SEL-L_CHAR      = ' SPACE C_QUOTE
  WA_TYPE_INFOOBJECT-L_CHAR C_QUOTE ' .'
                                INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'LOOP AT ' WA_TYPE_INFOOBJECT-L_CHAR ' .'
  INTO INFO_LINE SEPARATED BY SPACE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.
APPEND ' CLEAR: L_S_RANGE.' TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-SIGN = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-SIGN.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-OPTION = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-OPTION.' INTO INFO_LINE .
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-LOW = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-LOW.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE ' L_S_RANGE-HIGH = ' WA_TYPE_INFOOBJECT-L_CHAR
  INTO INFO_LINE SEPARATED BY SPACE.
CONCATENATE INFO_LINE '-HIGH.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

APPEND: ' APPEND L_S_RANGE TO T_TYPE_DEL_SEL-L_TYPE_RANGE.'
        TO RT_DEL_SUB_PROGRAM,
        ' ENLOOP.' TO RT_DEL_SUB_PROGRAM,
        ' APPEND T_TYPE_DEL_SEL.' TO RT_DEL_SUB_PROGRAM,
        ' CLEAR: T_TYPE_DEL_SEL.' TO RT_DEL_SUB_PROGRAM,
        ' ENDIF.' TO RT_DEL_SUB_PROGRAM.
ENDLOOP.

```

* Create the WHERE CLAUSE.

```
APPEND: '* Create the WHERE Clause' TO RT_DEL_SUB_PROGRAM,
```

```

' CLEAR: L_SEP, L_T_WHERE.' TO RT_DEL_SUB_PROGRAM,
' ' TO RT_DEL_SUB_PROGRAM,
' LOOP AT T_TYPE_DEL_SEL ASSIGNING <L_SX_CHA>.'
                        TO RT_DEL_SUB_PROGRAM,
' IF NOT L_SEP IS INITIAL.' TO RT_DEL_SUB_PROGRAM,
' APPEND L_SEP TO L_T_WHERE.' TO RT_DEL_SUB_PROGRAM,
' ENDIF.' TO RT_DEL_SUB_PROGRAM,
' ' TO RT_DEL_SUB_PROGRAM,
' CLEAR L_T_CONDITION.' TO RT_DEL_SUB_PROGRAM,
' PERFORM BUILD_CONDITION_ODSTA ' TO RT_DEL_SUB_PROGRAM,
' IN PROGRAM ZBW_DELETE_DSO_CONTENTS'
                        TO RT_DEL_SUB_PROGRAM,
' USING <L_SX_CHA>' TO RT_DEL_SUB_PROGRAM,
' CHANGING L_T_CONDITION' TO RT_DEL_SUB_PROGRAM,
' C_T_MSG.' TO RT_DEL_SUB_PROGRAM,
' ' TO RT_DEL_SUB_PROGRAM.
CONCATENATE 'APPEND' C_QUOTE '(' C_QUOTE 'TO L_T_WHERE.'
            INTO INFO_LINE SEPARATED BY SPACE.
APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM,
' LOOP AT L_T_CONDITION ASSIGNING <L_CONDITION>.'
                        TO RT_DEL_SUB_PROGRAM,
' APPEND <L_CONDITION>-LINE TO L_T_WHERE.'
                        TO RT_DEL_SUB_PROGRAM,
' ENLOOP.' TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'APPEND' C_QUOTE ')' C_QUOTE 'TO L_T_WHERE.'
            INTO INFO_LINE SEPARATED BY SPACE.
APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM.

CONCATENATE 'L_SEP = ' C_QUOTE 'AND' C_QUOTE '.' INTO INFO_LINE.
APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM,
' ENLOOP.' TO RT_DEL_SUB_PROGRAM.

* Execute only if selection options are given.
APPEND: 'IF L_T_WHERE IS NOT INITIAL.' TO RT_DEL_SUB_PROGRAM.
* Perform the delete based on WHERE Clause.
APPEND '* Perform the delete based on WHERE Clause.'
                        TO RT_DEL_SUB_PROGRAM.

IF RP_RDTAB = 'X'.
* Delete Table.
APPEND:
' PERFORM DELETE_TABLE IN PROGRAM ZBW_DELETE_DSO_CONTENTS'
                        TO RT_DEL_SUB_PROGRAM,
' USING P_UTNAME' TO RT_DEL_SUB_PROGRAM,
' P_COUNT' TO RT_DEL_SUB_PROGRAM,
' L_T_WHERE ' TO RT_DEL_SUB_PROGRAM,

```

```

        ' P_TEST. ' TO RT_DEL_SUB_PROGRAM.

ENDIF.
*
  IF RP_RUTAB = 'X'.
* Update Table.
  APPEND:
    ' PERFORM UPDATE_TABLE IN PROGRAM ZBW_DELETE_DSO_CONTENTS'
      TO RT_DEL_SUB_PROGRAM,
    ' USING P_UTNAME' TO RT_DEL_SUB_PROGRAM,
    ' P_COUNT' TO RT_DEL_SUB_PROGRAM,
    ' L_T_WHERE ' TO RT_DEL_SUB_PROGRAM,
    ' P_RSTR ' TO RT_DEL_SUB_PROGRAM,
    ' P_TEST. ' TO RT_DEL_SUB_PROGRAM.

  ENDIF.
  APPEND: ' ELSE.' TO RT_DEL_SUB_PROGRAM.
* WHERE Clause is NULL then raise exception.
  CONCATENATE ' MESSAGE ID ' SPACE C_QUOTE 'ZBW' C_QUOTE
    SPACE ' TYPE ' C_QUOTE 'E' C_QUOTE SPACE ' NUMBER ' SPACE
    C_QUOTE '030' C_QUOTE '.' INTO INFO_LINE.
  APPEND: INFO_LINE TO RT_DEL_SUB_PROGRAM,
    ' STOP.' TO RT_DEL_SUB_PROGRAM,
  ' ENDIF.' TO RT_DEL_SUB_PROGRAM.

* Insert the deletion report.
  CLEAR: WA_DSO_RPT.
  WA_DSO_RPT-DSO_NAME = RP_UTNAME.
  WA_DSO_RPT-DELETION_REPORT = RV_DEL_GEN_RPT_NAME.
  WA_DSO_RPT-UNAME = SY-UNAME.
  WA_DSO_RPT-DELETION_DATE = SY-DATUM.
  WA_DSO_RPT-DELETION_TIME = SY-UZEIT.
  WA_DSO_RPT-DEL_TABLE = RP_RDTAB.
  WA_DSO_RPT-UPDATE_TABLE = RP_RUTAB.

  INSERT INTO ZDEL_DSO_REPORT VALUES WA_DSO_RPT.
  COMMIT WORK.

* Execute the dynamic report.
  INSERT REPORT RV_DEL_GEN_RPT_NAME FROM RT_DEL_SUB_PROGRAM.
  SUBMIT (RV_DEL_GEN_RPT_NAME)
    VIA SELECTION-SCREEN
    AND RETURN.

ELSE.
* Execute the generated report which exists.

```

```

*
*   INSERT REPORT RV_DEL_GEN_RPT_NAME FROM RT_DEL_SUB_PROGRAM.
*   SUBMIT (RV_DEL_GEN_RPT_NAME)
*       VIA SELECTION-SCREEN
*       AND RETURN.
*   DELETE REPORT RV_DEL_GEN_RPT_NAME.

ENDIF.

ENDFORM.                "GENERATE_TABLE_SEL_SCREEN

*&-----*
*& Form      :   GEN_TABLE_REPORT_NAME
*& Purpose   :   Used to generate the name of the program.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*
FORM GEN_TABLE_REPORT_NAME CHANGING E_REPNM    LIKE SY-REPID.

DATA: L_UNIQUEID TYPE RSSG_UNI_IDC25.

* Get Unique Name
CALL FUNCTION 'RSS_SYSTEM_GET_UNIQUE_ID'
  IMPORTING
    E_UNI_IDC25 = L_UNIQUEID
  EXCEPTIONS
    OTHERS      = 8.
IF SY-SUBRC <> 0.
*   Generation Failed
  PERFORM X_MESSAGE USING RS_C_TRUE 'GEN_REPORT_NAME-1-'.
ENDIF.

CONCATENATE 'TAB_' L_UNIQUEID INTO E_REPNM.

ENDFORM.                "GEN_TABLE_REPORT_NAME

*&-----*
*& Form      :   FETCH_SELECTION_OPTIONS
*& Purpose   :   Fetches the Selection Option Objects.
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*
FORM FETCH_TABLE_SELECTION_OPTIONS
                                USING    RP_RUTAB          TYPE CHAR1
                                RP_RDTAB          TYPE CHAR1

```

```

RP_UTNAME          TYPE TABNAME
RV_COUNT           TYPE I
RV_CHECK           TYPE CHECKBOX
RV_TEST            TYPE CHECKBOX
RVC_FORCE          TYPE CHECKBOX
RV_DEL_GEN_RPT_NAME TYPE
                  SY-REPID
CHANGING RT_DEL_SUB_PROGRAM TYPE
                  RSDRD_T_ABAPSOURCE
RT_INI_SUB_PROGRAM TYPE
                  RSDRD_T_ABAPSOURCE.

```

```

DATA: INFO_LINE      TYPE ABAPSOURCE,
      INFO_LINE1(100) TYPE C,
      WA_INFOOBJECTS TYPE BAPI6116IO,
      L_CHAR          TYPE CHAR6,
      L_S_SELOPT      TYPE RSDRD_S_SELOPT,
      S_DETAILS       TYPE BAPI6108.

```

```

DATA: INFOOBJECTS    TYPE STANDARD TABLE OF BAPI6116IO,
      E_T_SELSCR      TYPE RSDRD_T_ABAPSOURCE.

```

```

DATA: TEMP_RV_COUNT(10) TYPE N.

```

```

DATA: IFC            TYPE LVC_T_FCAT,
      TABLE_NAME    TYPE RSODSNAME,
      SIFC           TYPE LVC_S_FCAT.

```

```

INFO_LINE = 'AT SELECTION-SCREEN OUTPUT.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

TEMP_RV_COUNT = RV_COUNT.

```

* Fetch the preliminary selections.

```

CONCATENATE '          P_RUTAB AS CHECKBOX DEFAULT ' SPACE
            C_QUOTE RP_RUTAB C_QUOTE ',' INTO INFO_LINE.

```

```

APPEND: 'PARAMETERS: ' TO RT_DEL_SUB_PROGRAM,
        INFO_LINE      TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE '          P_RDTAB AS CHECKBOX DEFAULT '
            SPACE C_QUOTE RP_RDTAB C_QUOTE ',' INTO INFO_LINE.

```

```

APPEND: INFO_LINE      TO RT_DEL_SUB_PROGRAM.

```

```

CONCATENATE '          P_UTNAME TYPE TABNAME DEFAULT ' SPACE
            C_QUOTE RP_UTNAME C_QUOTE ',' INTO INFO_LINE.

```

```

APPEND: INFO_LINE      TO RT_DEL_SUB_PROGRAM.

```

```

IF RP_RUTAB IS NOT INITIAL.

```

* Update String Parameter.

```

CONCATENATE '          P_RSTR TYPE ABAPSOURCE,' SPACE
            INTO INFO_LINE.

```

```

APPEND: INFO_LINE      TO RT_DEL_SUB_PROGRAM.

```

```

ENDIF.
CONCATENATE '          P_COUNT TYPE I DEFAULT' SPACE C_QUOTE
            TEMP_RV_COUNT C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE      TO RT_DEL_SUB_PROGRAM.

CONCATENATE '          P_CHECK AS CHECKBOX DEFAULT' SPACE C_QUOTE
            RV_CHECK C_QUOTE ',' INTO INFO_LINE.
APPEND:  INFO_LINE      TO RT_DEL_SUB_PROGRAM.

CONCATENATE '          P_TEST AS CHECKBOX DEFAULT' SPACE C_QUOTE
            RV_TEST C_QUOTE '.' INTO INFO_LINE.
APPEND:  INFO_LINE      TO RT_DEL_SUB_PROGRAM.

```

* Change Log Declaration.

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXN-NAME = ' C_QUOTE 'P_RUTAB' C_QUOTE '.'
            INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXN-KIND = ' C_QUOTE C_P C_QUOTE '.'
            INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXN-TEXT = ' C_QUOTE 'Update Table Data Records'
            C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTXN TO L_T_SELTXNS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

*

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXN-NAME = ' C_QUOTE 'P_RDTAB' C_QUOTE '.'
            INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXN-KIND = ' C_QUOTE C_P C_QUOTE '.'
            INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTXN-TEXT = ' C_QUOTE 'Delete Table Data Records'
            C_QUOTE '.' INTO INFO_LINE.

```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
*
```

```
CLEAR: INFO_LINE.
```

```
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_UTNAME' C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Table Name:'
  C_QUOTE '.' INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
*
```

```
CLEAR: INFO_LINE.
```

```
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_COUNT' C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE '.'
  INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Delete Record Counter:'
  C_QUOTE '.' INTO INFO_LINE.
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CLEAR: INFO_LINE.
```

```
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'
```

```
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
*
```

```
*
```



```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_CHECK' C_QUOTE ' .'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE ' .'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Print Spool'
  C_QUOTE ' .' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

*

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_RSTR' C_QUOTE ' .'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE ' .'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Update String:'
  C_QUOTE ' .' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

*

*

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE 'P_TEST' C_QUOTE ' .'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

```

```

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_P C_QUOTE ' .'

```

```

      INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE 'Test Run'
           C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELXTS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
*
* Get the structure of the Dynamic Table
CALL FUNCTION 'DDIF_FIELDINFO_GET'
  EXPORTING
    TABNAME           = RP_UTNAME
*   FIELDNAME         = ' '
*   LANGU             = SY-LANGU
*   LFIELDNAME       = ' '
*   ALL_TYPES        = ' '
*   GROUP_NAMES      = ' '
*   UCLEN            =
* IMPORTING
*   X030L_WA         =
*   DDOBJTYPE        =
*   DFIES_WA         =
*   LINES_DESCR      =
  TABLES
    DFIES_TAB        = T_DFIES_TAB
*   FIXED_VALUES     =
  EXCEPTIONS
    NOT_FOUND        = 1
    INTERNAL_ERROR   = 2
    OTHERS           = 3
.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

LOOP AT T_DFIES_TAB INTO WA_DFIES WHERE SIGN IS INITIAL.
  CLEAR: L_CHAR.
  L_CHAR      = SY-TABIX.
  SHIFT L_CHAR LEFT DELETING LEADING SPACE.
  CONCATENATE RSDQ_C_VARNM-CHAR L_CHAR INTO L_CHAR.
  CLEAR: INFO_LINE.

```

```

CLEAR: INFO_LINE.
CONCATENATE RP_UTNAME '-' WA_DFIES-FIELDNAME INTO INFO_LINE1.
CONCATENATE 'SELECT-OPTIONS: ' L_CHAR ' FOR '
  INFO_LINE1 '.' INTO INFO_LINE SEPARATED BY SPACE.
APPEND INFO_LINE TO RT_DEL_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-NAME = ' C_QUOTE L_CHAR C_QUOTE '.'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-KIND = ' C_QUOTE C_S C_QUOTE '.'
  INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE 'L_S_SELTX-TEXT = ' C_QUOTE WA_DFIES-SCRTEXT_L
  C_QUOTE '.' INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
INFO_LINE = 'APPEND L_S_SELTX TO L_T_SELTXS.'.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

CLEAR: WA_TYPE_INFOOBJECT.
*   WA_TYPE_INFOOBJECT-INFOOBJECT = WA_INFOOBJECTS-INFOOBJECT.
    WA_TYPE_INFOOBJECT-FIELDNM = WA_DFIES-FIELDNAME .
    WA_TYPE_INFOOBJECT-L_CHAR = L_CHAR.
    APPEND WA_TYPE_INFOOBJECT TO T_TYPE_INFOOBJECT.

ENDLOOP.
* Function to super impose the field titles.
APPEND: ' CALL FUNCTION ''SELECTION_TEXTS_MODIFY''' TO
  RT_INI_SUB_PROGRAM,
  ' EXPORTING ' TO RT_INI_SUB_PROGRAM.

CLEAR: INFO_LINE.
CONCATENATE '          PROGRAM          = ' C_QUOTE
  RV_DEL_GEN_RPT_NAME C_QUOTE INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.

APPEND: ' TABLES ' TO RT_INI_SUB_PROGRAM,
  '          SELTEXTS          = L_T_SELTXS' TO
  RT_INI_SUB_PROGRAM,
  ' EXCEPTIONS ' TO RT_INI_SUB_PROGRAM,
  '          PROGRAM_NOT_FOUND = 1 ' TO

```

```

                                RT_INI_SUB_PROGRAM,
PROGRAM_CANNOT_BE_GENERATED = 2 ' TO
                                RT_INI_SUB_PROGRAM,
OTHERS                          = 3. ' TO
                                RT_INI_SUB_PROGRAM.

```

```
APPEND: ' LOOP AT SCREEN.' TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE ' IF SCREEN-NAME = ' C_QUOTE 'P_UTNAME' C_QUOTE ' OR
SCREEN-NAME = ' C_QUOTE 'P_RUTAB' C_QUOTE 'OR' INTO INFO_LINE.
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE 'SCREEN-NAME = ' C_QUOTE 'P_CHECK' C_QUOTE ' OR '
INTO INFO_LINE.
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE 'SCREEN-NAME = ' C_QUOTE 'P_RDTAB' C_QUOTE
'.' INTO INFO_LINE.
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE ' SCREEN-INPUT = ' C_QUOTE '0' C_QUOTE ' .'
INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE ' SCREEN-OUTPUT = ' C_QUOTE '1' C_QUOTE ' .'
INTO INFO_LINE.
APPEND INFO_LINE TO RT_INI_SUB_PROGRAM.
```

```
CONCATENATE ' SCREEN-ACTIVE = ' C_QUOTE '1' C_QUOTE ' .'
INTO INFO_LINE.
APPEND: INFO_LINE TO RT_INI_SUB_PROGRAM,
' MODIFY SCREEN. ' TO RT_INI_SUB_PROGRAM,
' ENDIF.' TO RT_INI_SUB_PROGRAM,
' ENDLOOP.' TO RT_INI_SUB_PROGRAM.
```

```
ENDFORM. "FETCH_TABLE_SELECTION_OPTIONS
```

```

*&-----*
*& Form      : GENERATE_TABLE_INITIALIZATION
*& Purpose   : Generates the Initialization Section of the report..
*& Author    : Praveen Gudishetty
*& Date      : 01 / 01 / 2008
*-----*
FORM GENERATE_TABLE_INITIALIZATION USING RP_RUTAB TYPE CHECKBOX
                                         RP_RDTAB TYPE CHECKBOX
                                         CHANGING RT_DEL_SUB_PROGRAM TYPE

```

```

RSDRD_T_ABAPSOURCE.

DATA: L_STRING          TYPE ABAPSOURCE.
L_STRING = 'INITIALIZATION.'.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.
CLEAR: L_STRING.
CONCATENATE 'SET TITLEBAR ' C_QUOTE 'SELECTION' C_QUOTE
  ' OF PROGRAM ' C_QUOTE 'SAPLRSDRD' C_QUOTE ' WITH ' INTO L_STRING.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.

IF RP_RDTAB IS NOT INITIAL.
CONCATENATE C_QUOTE 'Selection Options for deletion' C_QUOTE '.' INTO
  L_STRING SEPARATED BY SPACE.
ELSEIF RP_RUTAB IS NOT INITIAL.
  CONCATENATE C_QUOTE 'Selection Options for Update' C_QUOTE '.' INTO
    L_STRING SEPARATED BY SPACE.
ENDIF.
APPEND L_STRING TO RT_DEL_SUB_PROGRAM.
ENDFORM.          "GENERATE_TABLE_INITIALIZATION

*&-----*
*& Form      :   UPDATE_TABLE
*& Purpose   :   Updates the table with the parameters stated..
*& Author    :   Praveen Gudishetty
*& Date      :   01 / 01 / 2008
*-----*
FORM UPDATE_TABLE USING      RVP_UTNAME   TYPE RSODSNAME
                             RVP_COUNT   TYPE I
                             RV_SLOG     TYPE RSDRD_T_ABAPSOURCE
                             RV_SETSTRING TYPE ABAPSOURCE
                             RVP_TEST    TYPE CHECKBOX.

DATA: TABLE_NAME          TYPE RSODSNAME,
      V_COUNTER            TYPE I,
      TOTAL_COUNT(20)     TYPE N.
FIELD-SYMBOLS: <FS_SLOG>          TYPE ABAPSOURCE.

* Check if the Update String is NULL.
IF RV_SETSTRING IS INITIAL.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '047'.
  STOP.
ENDIF.

* Check if it is a Test Run or not.
IF RVP_TEST <> 'X'.

* Do not allow this report to be run in foreground
IF SY-BATCH = SPACE.
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '031'.
  STOP.

```

```

ENDIF.
ENDIF.

WRITE:/ 'Update in ', RVP_UTNAME, ' using selection option:'.
* Print the Where Clause:
LOOP AT RV_SLOG ASSIGNING <FS_SLOG>.
  WRITE:/ <FS_SLOG>.
ENDLOOP.

* Print the Set String
WRITE:/ 'Update parameter: '.
WRITE: RV_SETSTRING.

* Fetch the total record count to be updated:
SELECT COUNT(*) FROM (RVP_UTNAME)
      INTO TOTAL_COUNT
      WHERE (RV_SLOG).

* Check if it is a Test Run or not.
IF RVP_TEST <> 'X' AND SY-SUBRC = 0.
* Update Table
  UPDATE (RVP_UTNAME)
    SET (RV_SETSTRING)
    WHERE (RV_SLOG).
  IF SY-SUBRC <> 0.
    EXIT.
  ENDIF.
  CALL FUNCTION 'DB_COMMIT'.
ELSEIF SY-SUBRC <> 0 AND RVP_TEST <> 'X' .
  MESSAGE ID 'ZBW' TYPE 'E' NUMBER '042'.
ENDIF.
* Print the total records deleted.
ULINE.
WRITE:/ 'Total record count being updated: ', TOTAL_COUNT.
ULINE.

ENDFORM.          "update_table

```

Parameter Attributes



Text Symbols			
Symbol	Symbol Text	dLen	mLen
001	Change Log Selection Options.	29	29
002	Active Table Selection Options.	31	31
003	PSA Selection Options.	22	22
004	Update Table Options.	21	25
005	Global Parameters.	18	18
006	DSO Maintenance Parameters.	27	28
007	Table Update Parameters.	24	50

Selection Text	
Parameter Name	Parameter Text
P_ACT	Delete Active Table Data.
P_ATNAME	Active Table Tech. Name.
P_BDAYS	Delete Any PSA Before Days.
P_CHECK	Printing Spool Data.
P_CHGLOG	Change Log Tech. Name.
P_COUNT	Commit Record Counter.
P_FORCE	Force Generate Report.
P_LOG	Delete Change Log Data.
P_ODS	ODS Tech. Name.
P_PSA	Delete PSA Table Data.
P_RDTAB	Delete Table Data Records
P_RNONE	No Update / Delete.
P_RUTAB	Update Table Data Records.
P_TEST	Test Run.
P_UTNAME	Table Name.

Execution Snapshots: Deletion from Change Log

1. Snapshot for "Deletion" of data records from DSO Change Log:

Report to delete Active Table / Change Log / PSA corresponding to ODS



DSO Maintenance Parameters.
ODS Tech. Name.
Change Log Selection Options.
Change Log Tech. Name.
 Delete Change Log Data.
Active Table Selection Options.
Active Table Tech. Name.
 Delete Active Table Data.
PSA Selection Options.
 Delete PSA Table Data.
Delete Any PSA Before Days.

Table Update Parameters.
Update Table Options.
 Update Table Data Records.
 Delete Table Data Records
 No Update / Delete.
Table Name.

Global Parameters.
 Force Generate Report.
Commit Record Counter.
 Printing Spool Data.
 Test Run.

2. Execution of the above report generates a report for deletion of data from the change log of the DSO. Snapshot as follows:

Selection Options for deletion

Delete Change Log

Change Log Table Name:

Delete Active Table

Active Table Name:

Delete Record Counter:

Print Spool

Test Run

Schedule Line Number	<input type="text"/>	to	<input type="text"/>	↕
Default Del. Block	<input type="text"/>	to	<input type="text"/>	↕
Plan. GI Date of SL	<input type="text"/>	to	<input type="text"/>	↕
Conf. Delivery Date	<input type="text"/>	to	<input type="text"/>	↕
Sched. Line Deleted	<input type="text"/>	to	<input type="text"/>	↕
EAN/UPC	<input type="text"/>	to	<input type="text"/>	↕
Goods Issue Date	<input type="text"/>	to	<input type="text"/>	↕
Grid Value	<input type="text"/>	to	<input type="text"/>	↕
AFS Theme	<input type="text"/>	to	<input type="text"/>	↕
AFS Collection	<input type="text"/>	to	<input type="text"/>	↕
Item	<input type="text"/>	to	<input type="text"/>	↕
Item bill.data	<input type="text"/>	to	<input type="text"/>	↕
Pricing	<input type="text"/>	to	<input type="text"/>	↕
Rejection status	<input type="text"/>	to	<input type="text"/>	↕
Reason for Rejection	<input type="text"/>	to	<input type="text"/>	↕
Sched.line category	<input type="text"/>	to	<input type="text"/>	↕
Changed on	<input type="text"/>	to	<input type="text"/>	↕
Reason for rejection	<input type="text"/>	to	<input type="text"/>	↕
Billing Block Item	<input type="text"/>	to	<input type="text"/>	↕
Storage Location	<input type="text"/>	to	<input type="text"/>	↕
Item category	<input type="text"/>	to	<input type="text"/>	↕

Value Help for InfoObject OSCHED_

Sched. Lin	Short Description
#	Not assigned

✓ 🔍 ✚ 🏠 ✖

Execution Snapshot: Update a Database Table

1. Snapshot of the report to update the table records:

Report to delete Active Table / Change Log / PSA corresponding to ODS

DSO Maintenance Parameters.

ODS Tech. Name.

Change Log Selection Options.

Change Log Tech. Name.

Delete Change Log Data.

Active Table Selection Options.

Active Table Tech. Name.

Delete Active Table Data.

PSA Selection Options.

Delete PSA Table Data.

Delete Any PSA Before Days.

Table Update Parameters.

Update Table Options.

Update Table Data Records.

Delete Table Data Records

No Update / Delete.

Table Name.

Global Parameters.

Force Generate Report.

Commit Record Counter.

Printing Spool Data.

Test Run.

- Snapshot of the report to execute Update of the table:

Selection Options for Update

Selection Options for Update

Update Table Data Records

Delete Table Data Records

Table Name:

Update String:

Delete Record Counter:

Print Spool

Test Run

Sales document	<input type="text" value="5742874"/>	to	<input type="text"/>	
Item	<input type="text"/>	to	<input type="text"/>	
Delivery Status	<input type="text"/>	to	<input type="text"/>	
Overall Del. Status	<input type="text"/>	to	<input type="text"/>	
Created on	<input type="text"/>	to	<input type="text"/>	

Execution Snapshot: Deletion from Change Log and Active Table


1. Execution of the report to delete data from Active Table and Change Log:

Report to delete Active Table / Change Log / PSA corresponding to ODS

DSO Maintenance Parameters.	
ODS Tech. Name.	ZSD_S05
Change Log Selection Options.	
Change Log Tech. Name.	/BIC/B00000550000
<input checked="" type="checkbox"/> Delete Change Log Data.	
Active Table Selection Options.	
Active Table Tech. Name.	/BIC/AZSD_S0500
<input checked="" type="checkbox"/> Delete Active Table Data.	
PSA Selection Options.	
<input type="checkbox"/> Delete PSA Table Data.	
Delete Any PSA Before Days.	
Table Update Parameters.	
Update Table Options.	
<input type="radio"/> Update Table Data Records.	
<input type="radio"/> Delete Table Data Records	
<input checked="" type="radio"/> No Update / Delete.	
Table Name.	/BIC/AZSD_S0100
Global Parameters.	
<input checked="" type="checkbox"/> Force Generate Report.	
Commit Record Counter.	1,000
<input checked="" type="checkbox"/> Printing Spool Data.	
<input type="checkbox"/> Test Run.	

- Update the selections based on which the data has to be deleted:

Selection Options for deletion




















Delete Change Log
 Change Log Table Name:

Delete Active Table
 Active Table Name:

Delete Record Counter:

Print Spool
 Test Run

Schedule Line Number	<input type="text"/>	to	<input type="text"/>	
Default Del. Block	<input type="text" value="25"/>	to	<input type="text"/>	
Plan. GI Date of SL	<input type="text"/>	to	<input type="text"/>	
Conf. Delivery Date	<input type="text" value="2/25/2007"/> 	to	<input type="text"/>	
Sched. Line Deleted	<input type="text" value="X"/>	to	<input type="text"/>	
EAN/UPC	<input type="text"/>	to	<input type="text"/>	
Goods Issue Date	<input type="text"/>	to	<input type="text"/>	
Grid Value	<input type="text"/>	to	<input type="text"/>	
AFS Theme	<input type="text"/>	to	<input type="text"/>	
AFS Collection	<input type="text"/>	to	<input type="text"/>	
Item	<input type="text"/>	to	<input type="text"/>	
Item bill.data	<input type="text"/>	to	<input type="text"/>	
Pricing	<input type="text"/>	to	<input type="text"/>	
Rejection status	<input type="text"/>	to	<input type="text"/>	
Reason for Rejection	<input type="text" value="J1"/>	to	<input type="text"/>	
Sched.line category	<input type="text"/>	to	<input type="text"/>	

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.