

# Step by Step Approach to Delete and Recreate Secondary Indices in Process Chains



## Applies to:

SAP BW 3.x & SAP BI Net Weaver 2004s. For more information, visit the [Business Intelligence homepage](#).

## Summary

This article explains the step by step approach to delete and recreate secondary indexes of DSO in the process chains.

**Author:** Vikram Srivastava

**Company:** Infosys Technologies Ltd

**Created on:** 16 March 2011

## Author Bio



Vikram Srivastava is working as Technology Analyst with Infosys Technologies Limited. He has got rich experience on various BW Implementation/Support Projects in both SAP BW 3.5 and SAP BW 7.0.

## Table of Contents

Scenario .....	3
Step by Step Approach.....	3
Step 1: .....	3
Step 2: .....	3
Step 3: .....	4
Step 4: .....	4
Step 5: .....	7
Step 6: .....	7
Step 7: .....	7
Using the program .....	8
Disclaimer and Liability Notice.....	9

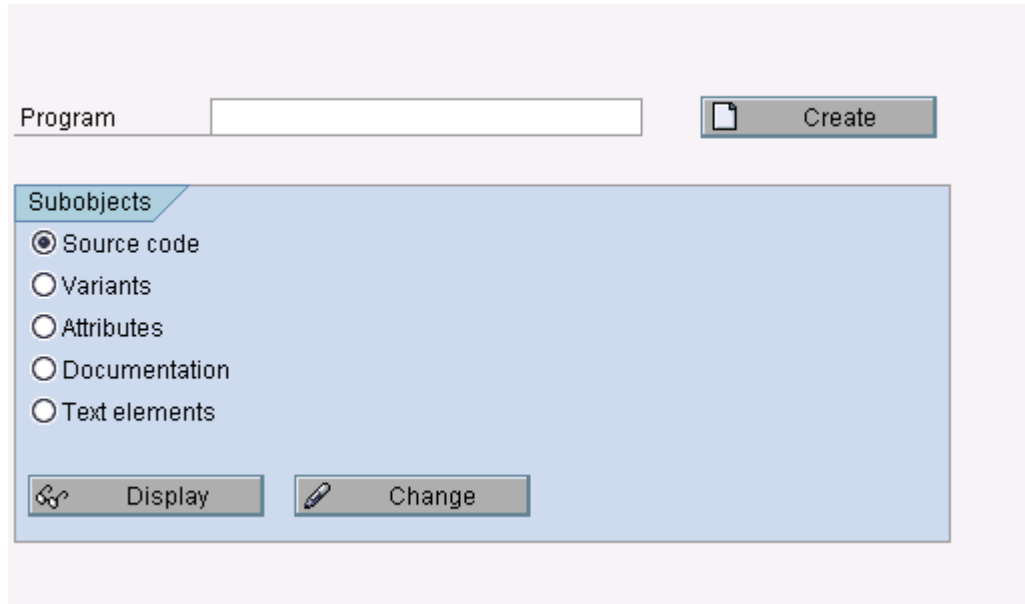
## Scenario

In this article we will understand the step by step approach to delete and recreate indexes of a DSO in process chains and jobs. This becomes very useful and handy while trying to do a constant performance improvement effort.

## Step by Step Approach

### Step 1:

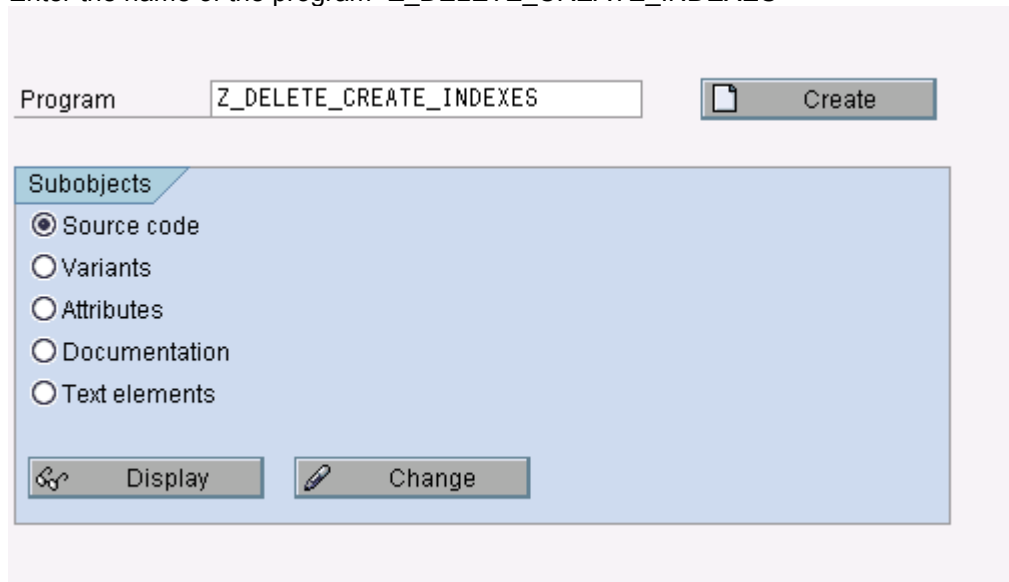
Go to TCode SE38



The screenshot shows the SAP SE38 TCode interface. At the top, there is a 'Program' field which is currently empty, followed by a 'Create' button with a document icon. Below this is a 'Subobjects' section with a light blue background. It contains five radio button options: 'Source code' (which is selected), 'Variants', 'Attributes', 'Documentation', and 'Text elements'. At the bottom of the 'Subobjects' section, there are two buttons: 'Display' with a magnifying glass icon and 'Change' with a pencil icon.

### Step 2:

Enter the name of the program "Z\_DELETE\_CREATE\_INDEXES"



The screenshot shows the SAP SE38 TCode interface after the program name has been entered. The 'Program' field now contains the text 'Z\_DELETE\_CREATE\_INDEXES', and the 'Create' button remains visible. The 'Subobjects' section is identical to the previous screenshot, with 'Source code' selected and the 'Display' and 'Change' buttons at the bottom.

**Step 3:**

Click on Create and specify the attributes as shown below and Click on Save  
Below Screen appears:

ABAP: Program Attributes Z\_DELETE\_CREATE\_INDEXES Change

Title **Program to Delete and Create Secondary Indices of a DSO**

Original language **EN** English

Created [Redacted] [Redacted]

Last changed by [Redacted] [Redacted]

Status **New(Revised)**

**Attributes**

Type **Executable program**

Status **Test Program**

Application **Business Information Warehouse**

Authorization Group [Redacted]

Logical database [Redacted]

Selection screen [Redacted]

Editor lock  Fixed point arithmetic

Unicode checks active  Start using variant

Save Undo Redo [Redacted] [Redacted]

**Step 4:**

Write the program as mentioned below:

```
*&-----*
*& Report  Z_DELETE_CREATE_INDEXES
*&
*&-----*
*&
*&
*&-----*
```

```
REPORT  Z_DELETE_CREATE_INDEXES.
```

```
TYPE-POOLS: rsdu.                "DB utilities
```

```
data: ODSIndex          type rsdu_s_index,
      Function(3)       type c,
      rc                like sy-subrc,
      current_schema    type db6schema,
      tabschema         type db6schema,
      table_name        type db6tabnam.
```

```
Parameters: iODS(30)      type c,
             iIdx(3)       type c,
             iFunctn(6)    type c.
```

```

ODSIndex-indexname = iIdx.
ODSIndex-tabname   = iODS.
Function           = iFunctn.

* Get the current DB schema
perform get_schema in program sdb1fdb6
      changing current_schema.

* check existance of table
exec sql.
      select tabschema from syscat.tables into :tabschema
      where tabschema = :current_schema
      and tabname = :ODSIndex-tabname
endexec.

if sy-subrc <> 0.
  message id 'RSM' type 'E' number '799'
  with 'Finding the table ' ODSIndex-tabname
  ' in the current schema,' current_schema.
endif.

case Function.
when 'DELETE'.
  perform DropODSIndex using ODSIndex rc.
when 'CREATE'.
  perform CreateODSIndex using ODSIndex rc.
  table_name = ODSIndex-tabname.
  perform CreateStatistics using table_name tabschema.

when others.
  message id 'RSM' type 'E' number '799'
  with 'Only CREATE (Create Index) and DELETE (Delete Index)'
  ' are valid for the Function'.
endcase.

message id 'RSM1' type 'S' number '799'
with Function ' successful'.

*-----*
*      Form DropODSIndex
*-----*
*      text
*-----*
*      -->P_ODSTABLENAME text
*      -->P_INDEXID text
*-----*
FORM DropODSIndex USING P_ODSIndex type rsdu_s_index
P_RC.

CALL FUNCTION 'RSDU_DROP_INDEX'
EXPORTING
  I_S_INDEX           = p_ODSIndex
IMPORTING
  E_SUBRC            = p_rc
EXCEPTIONS
  DROPPING_ERROR     = 1
  OTHERS              = 3.

```

```

IF SY-SUBRC <> 0 or p_rc <> 0.
  message id 'RSM' type 'E' number '799'
  with P_ODSIndex-tabname P_ODSIndex-indexname
  ' Failed to drop ODS index with RC = ' rc.
ENDIF.

ENDFORM.                  " DropODSIndex

*-----*
*      Form CreateODSIndex
*-----*
*      text
*-----*
*      -->P_ODSINDEX text
*      -->P_RC text
*-----*
FORM CreateODSIndex USING P_ODSINDEX type rsdu_s_index
                        P_RC.

CALL FUNCTION 'DD_DATABASE_UTILITY'
  EXPORTING
    FCT                = 'CRE'
    ID_NAME            = p_ODSIndex-indexname
    OBJ_NAME          = p_ODSIndex-tabname
    OBJ_TYPE          = 'INDX'
  IMPORTING
    SUBRC              = p_rc
  EXCEPTIONS
    UNEXPECTED_ERROR  = 1
    UNSUPPORTED_FUNCTION = 2
    UNSUPPORTED_OBJ_TYPE = 3
    TABLE_IS_LOCKED_BY_TCNV = 4
    AUTHORITY_CHECK_FAILED = 5
    ABORT_FUNCTION = 6
    CONVERSION_ERROR = 7
    OTHERS = 8.

IF SY-SUBRC <> 0 or p_rc <> 0.
  message id 'RSM' type 'E' number '799'
  with P_ODSIndex-tabname P_ODSIndex-indexname
  ' Failed to create ODS index with RC = ' sy-subrc.
ENDIF.

ENDFORM.                  " CreateODSIndex

*-----*
*      Form CreateStatistics
*-----*
*      text
*-----*
*      -->P_TABLENAME text
*      -->P_TABSCHEMA text
*-----*
FORM CreateStatistics USING P_TABLE_NAME
                        P_TABSCHEMA.

CALL FUNCTION 'DB6_PM_RUNSTATS_REORGCHK'

```

```

EXPORTING
  TNAME                = p_table_name
  TOWNER               = p_tabschema
  DETAIL               = 'Y'
* SUPPRESS_DBALOG     = 0
EXCEPTIONS
  C_CALL_FAILED       = 1
  INVALID_PARAMETER_SET = 2
  OTHERS              = 3.

IF SY-SUBRC <> 0.
  message id 'RSM' type 'E' number '799'
  with p_table_name
    ' Failed to create DB Statistics with RC = ' sy-subrc.
ENDIF.

ENDFORM.                    " CreateStatistics

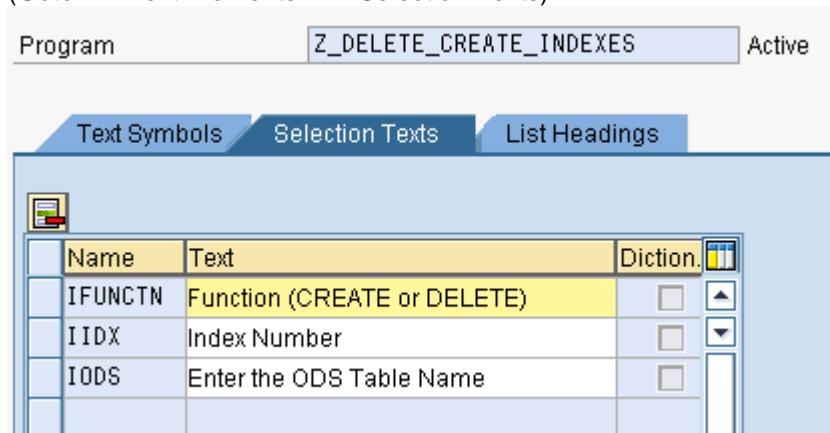
```

**Step 5:**

Save and Activate

**Step 6:**

Go to Text Elements and enter the description as shown below  
(Goto → Text Elements → Selection Texts)

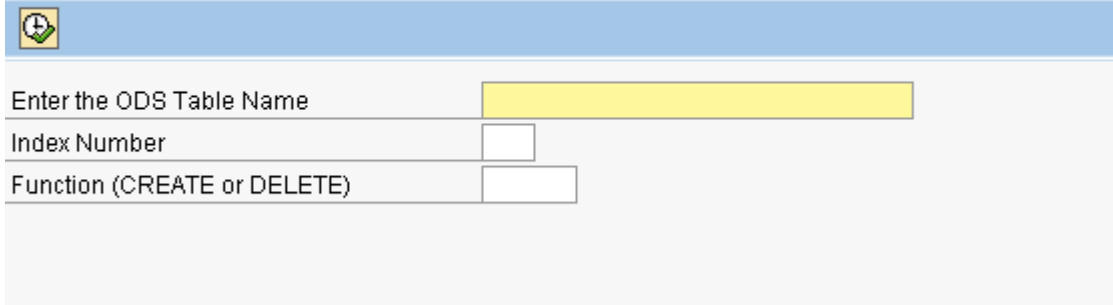
**Step 7:**

Save and Activate. The program is ready.

## Using the program

Go to TCode SE38 and execute the program "Z\_DELETE\_CREATE\_INDEXES"

### Program to Delete and Create Secondary Indices of a DSO



Enter the ODS Table Name	<input type="text"/>
Index Number	<input type="text"/>
Function (CREATE or DELETE)	<input type="text"/>

This program can now be used in process chains by creating the variants.



## **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.