SAP BW – Infoprovider Data Display (LISTCUBE) - Improvised

Applies to:

This article is applicable to all the SAP BI consultants who are accustomed with SAP ABAP skills. For more information, visit the EDW HomePage .

Summary

Learn here about methods to display the Infoprovider data without repetitive selection of characteristics and key figures.

Author: Suraj Tigga

Company: Capgemini Consulting India Pvt. Ltd.

Created on: 3 August 2010

Author Bio

Suraj Tigga is a Senior SAP BI / ABAP consultant at Capgemini Consulting, India. Suraj joined Capgemini Consulting in 2008 and has worked on multiple SAP BI implementation and support projects.

© 2010 SAP AG

Table of Contents

Scenario 3
LISTCUBE (Generated ABAP Code) 3
Transaction LISTCUBE 3
Generated ABAP Code 5
Enhancement ABAP Code (RSDD_SHOW_ICUBE) 7
Copy (RSDD_SHOW_ICUBE) 7
Modification (ZRSDD_SHOW_ICUBE) 8
Execution (ZLISTCUBE) 13
Execute the transaction. 13
Execute the transaction. 16
Complete ABAP Code (ZRSDD_SHOW_ICUBE) 16
Related Content 25
Disclaimer and Liability Notice 26

© 2010 SAP AG

Scenario

Transaction LISTCUBE is used to display records from Infoproviders. But there is a restriction of the number of characteristic and Navigational attributes to be displayed (selected). If that particular limit is crossed then every time characteristics and key figures should be selected, whose values should be displayed in the ALV output.

Selection of characteristics and key figures can be avoided using two methods:

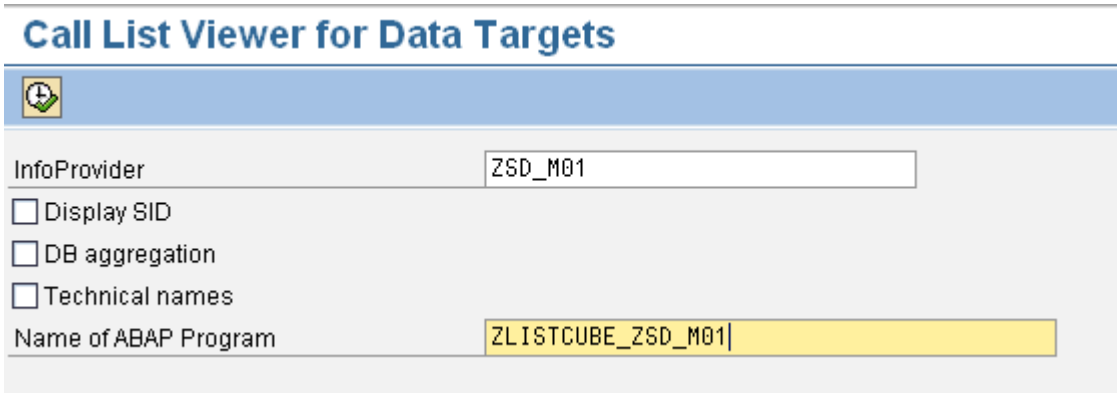
a) **LISTCUBE (Generated ABAP Code)** : Use ABAP Program generated by LISTCUBE(Not preferred in Production System)

b) **Enhancement ABAP Code (RSDD_SHOW_ICUBE):** Make modification to existing code RSDD_SHOW_ICUBE (Also applicable in Production System)

LISTCUBE (Generated ABAP Code)

Transaction LISTCUBE

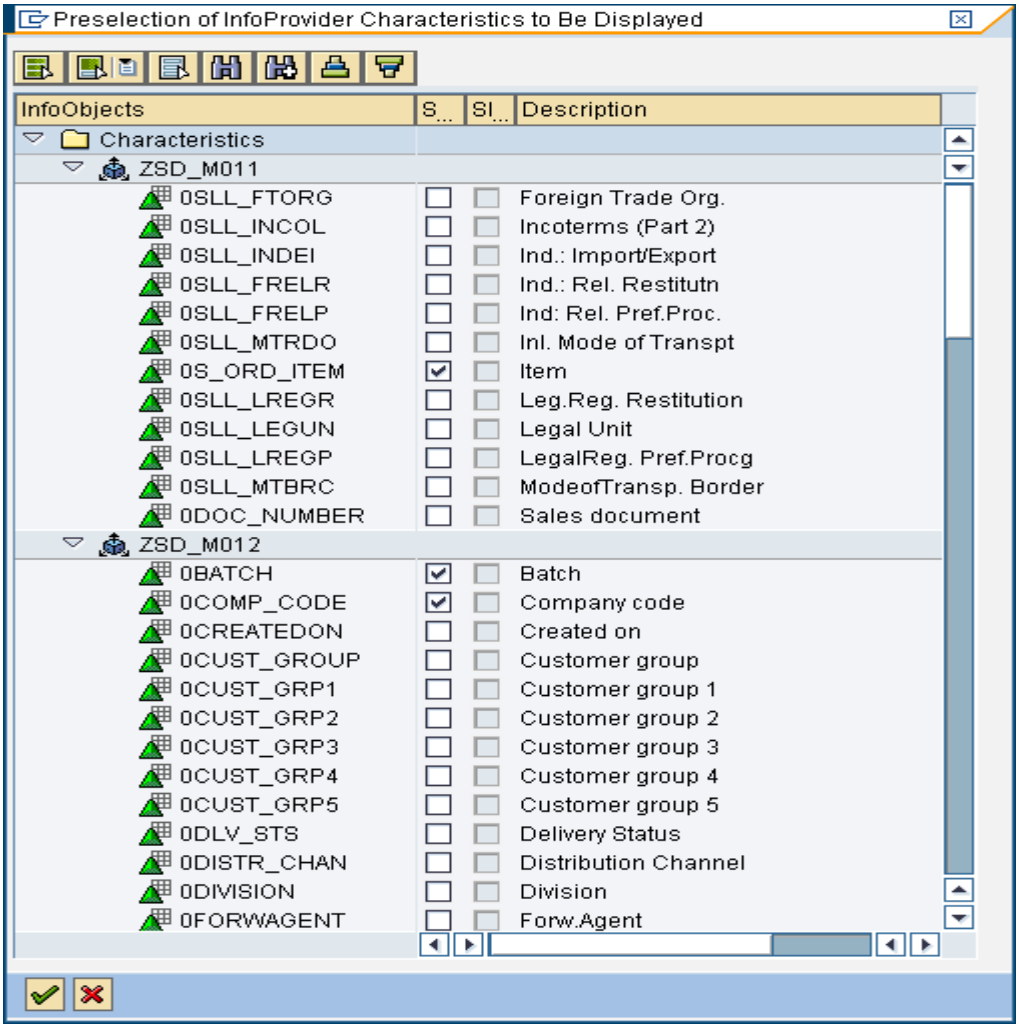
Step1: Go to transaction **LISTCUBE** and enter the Infoprovider 'ZSD_M01' and ABAP Program 'ZLISTCUBE_ZSD_M01'



Execute the transaction.

© 2010 SAP AG

Step2: Select the Characteristics and Key Figures whose data to be displayed from the next Pop up Screen:



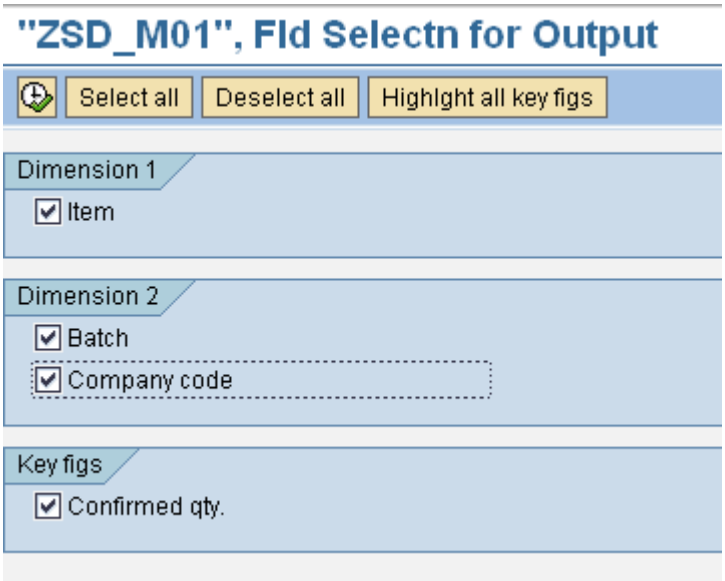
Selected the below mentioned Characteristics and Key Figures:

Key Figure / Characteristic	Technical Name	Description
Characteristic	0S_ORD_ITEM	Item
Characteristic	0BATCH	Batch
Characteristic	0COMP_CODE	Company Code
Key Figure	0CONF_QTY	Confirmed Qty

Click ‘OK’ after the selection is done successfully.

© 2010 SAP AG

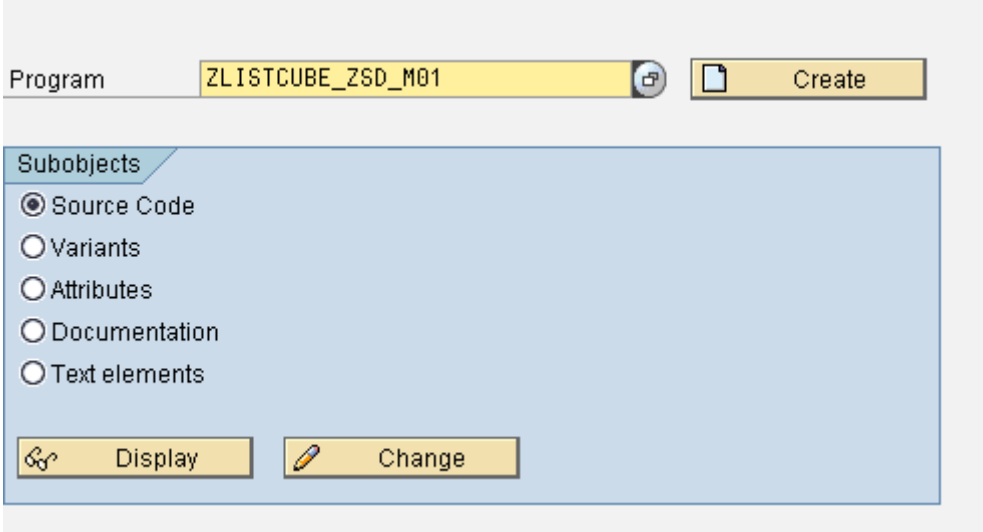
Step3: Check the 'Field Selection for Output'



Item, Batch, Company Code and Confirmed Qty are selected.

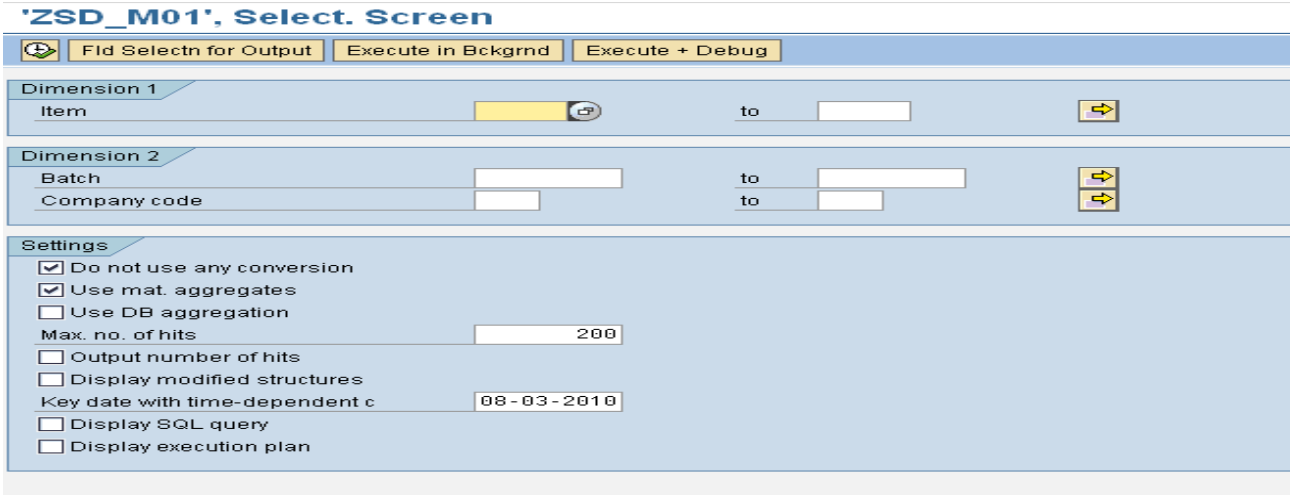
Generated ABAP Code

Step1: Go to transaction SE38 and enter the ABAP Code name 'ZLISTCUBE_ZSD_M01' and execute

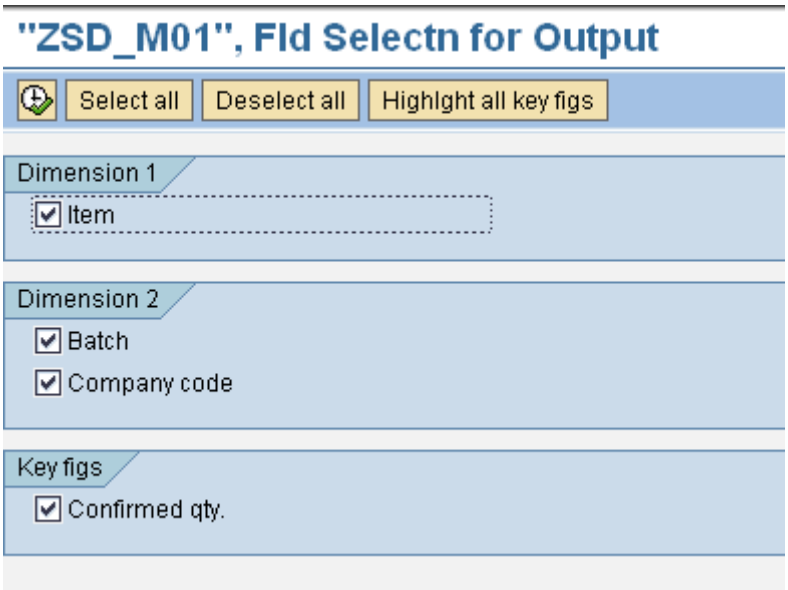


© 2010 SAP AG

Step2: After executing the Code the screen displayed



Check the 'Field Selection for Output'



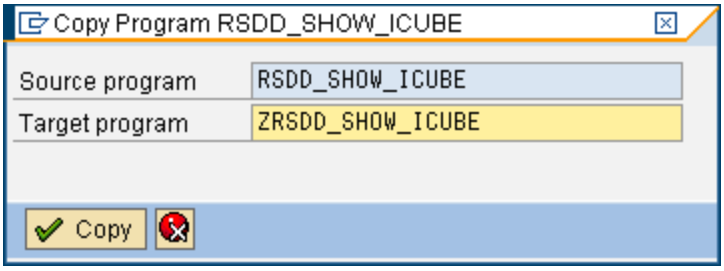
Characteristic and Key Figures are selected properly as in earlier case.

© 2010 SAP AG

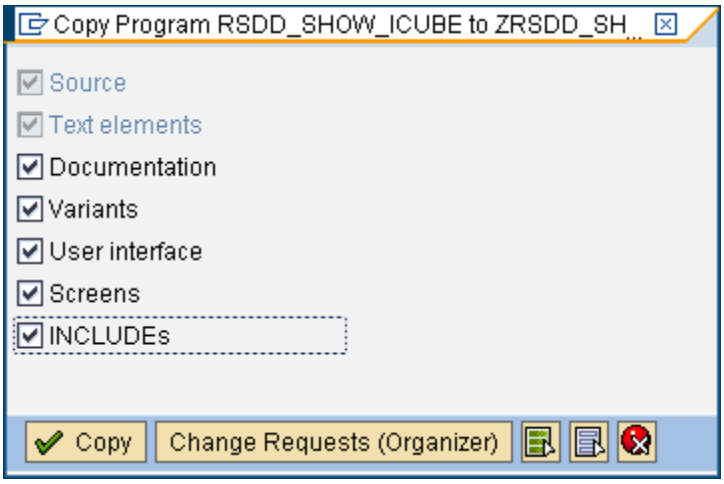
Enhancement ABAP Code (RSDD_SHOW_ICUBE)

Copy (RSDD_SHOW_ICUBE)

Step1: Go to transaction SE38 and Copy the ABAP Code 'RSDD_SHOW_ICUBE' to 'Z RSDD_SHOW_ICUBE'



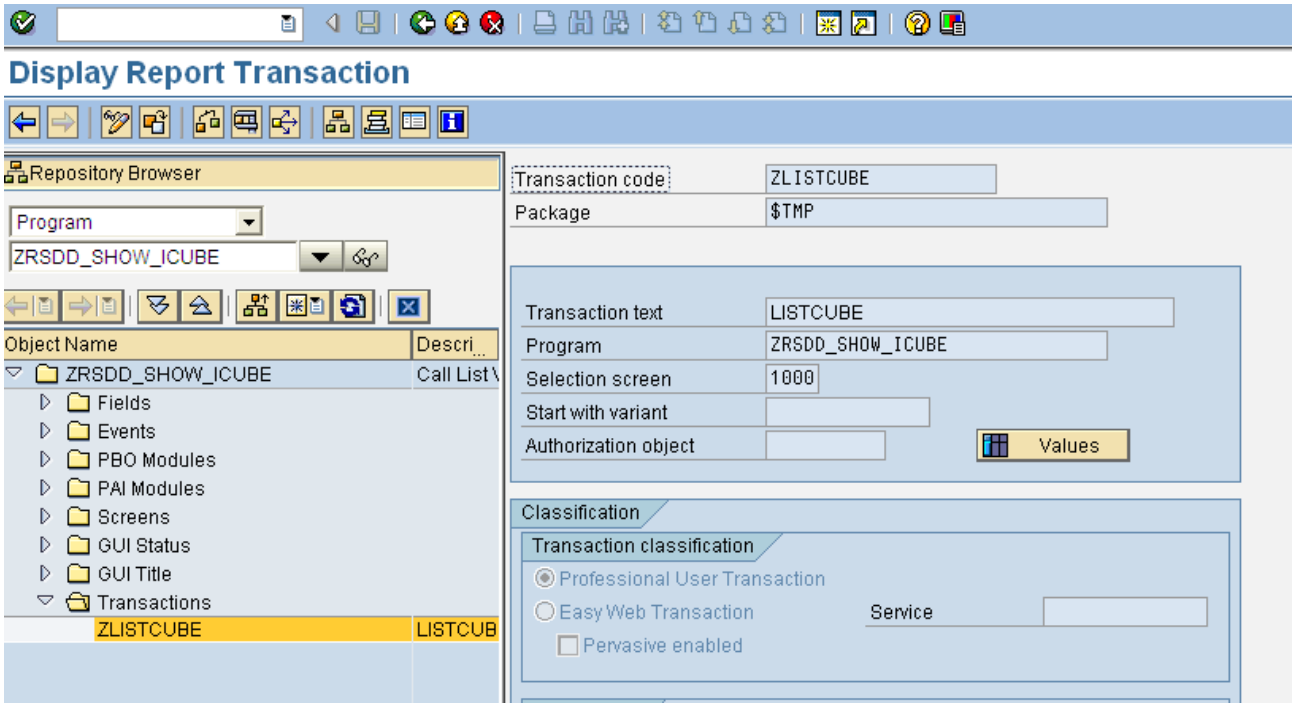
Select all the options (Documentation , Variants , User Interface , Screens , INCLUDEs)



© 2010 SAP AG

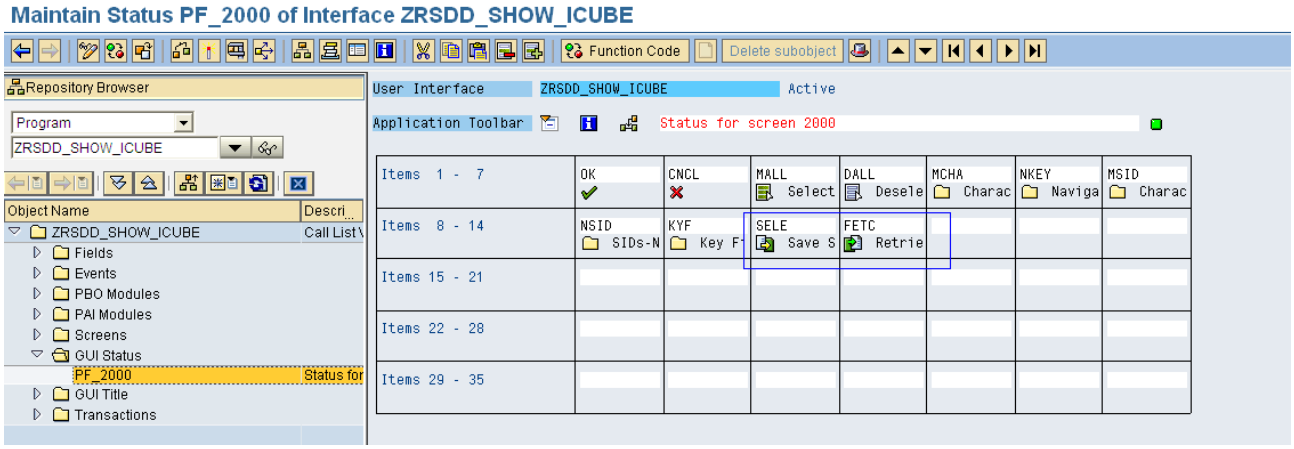
Modification (ZRSDD_SHOW_ICUBE)

Step1: Include transaction code ZLISTCUBE to ABAP code 'ZRSDD_SHOW_ICUBE':



While creating the transaction make it as 'Report Transaction'.

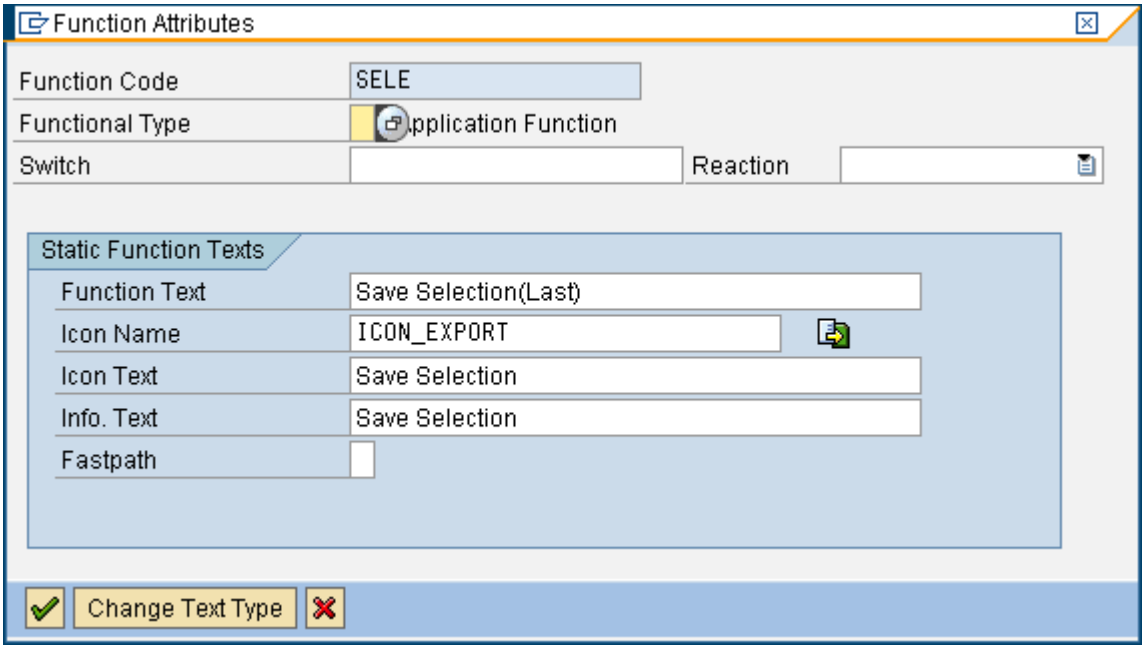
Step2: Modification to GUI Status for Screen 2000



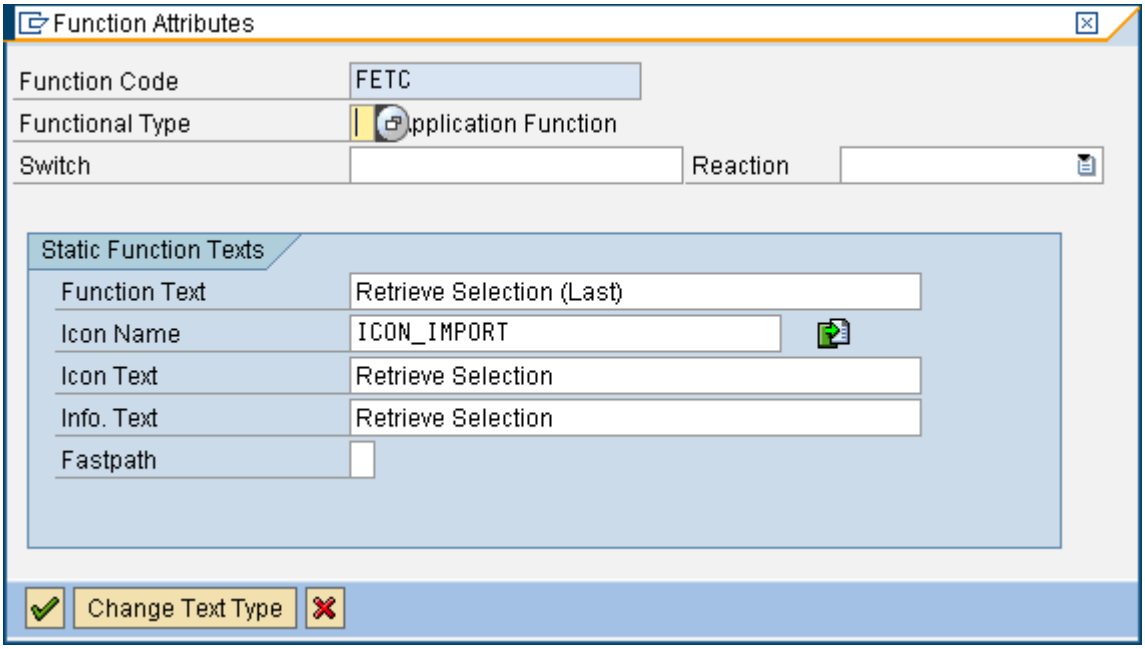
Add two function Codes '*SELE*' (To store selections) and '*FETC*' (To retrieve last selections).

© 2010 SAP AG

Function Attributes (*SELE*):



Function Attributes (*FETC*):



Step3: Go to the report 'ZRSDD_SHOW_ICUBE' and do the following changes:

a) Insert Data Declaration:

```
DATA string_file TYPE string VALUE
  'C:\Documents and Settings\stigga\Desktop\zlistcube.txt'.
DATA w_mark LIKE sy-tabix.
DATA w_loops LIKE sy-loopc.
```

Note: String_file (Used to store the selection conditions to computer local drive)

© 2010 SAP AG

b) If the number of characteristics and navigational attributes selections cross the limit (Below code should be added).Always call the Modified Screen 2000 (Mentioned Earlier), irrespective of whether the number of selection of characteristics and navigational attributes are crossed or not.

```
  LOOP AT g_t_tabl_cntl[]
    ASSIGNING <g_s_tabl_cntl>.

    IF <g_s_tabl_cntl>-tech_name = 'KYF'.
      l_is_kyf = rs_c_true.
    ENDIF.

    IF l_is_kyf = rs_c_false.
      CLEAR <g_s_tabl_cntl>-mark.
    ENDIF.
  ENDLOOP.

*    IF 1 = 1. " sy-uname = 'RUEGER' or
*      " sy-uname = 'DIPPER'.
*      LOOP AT g_t_tabl_cntl WHERE tech_name = 'CHA'
*                              OR tech_name = 'KYF'
*                              OR name      = text-029
*                              OR name      = text-030.
*        DELETE g_t_tabl_cntl.
*      ENDLOOP.
*      CALL FUNCTION 'RSDQ_PRESEL_SCREEN_2010'
*        EXPORTING
*          i_t_ioinf     = g_t_ioinf
*        CHANGING
*          c_t_tabl_cntl = g_t_tabl_cntl[].
*    ELSE.
*     display preselection panel
  CALL SCREEN 2000
      STARTING AT 10 10.
*    ENDIF.
*   retrieve infoobjects to be displayed.
  CALL FUNCTION 'RSDQ_IOINF_FROM_TABLCNTL'
    EXPORTING
      i_t_tabl_cntl = g_t_tabl_cntl[]
    CHANGING
      c_t_ioinf     = g_t_ioinf.

 ELSE.
   EXIT.
 ENDIF.
```

c) Include the statement after the Function Module `'RSDQ_VIEW_INFOPROV'`:

```
* build the infocube-specific listreport and
* execute it.
CALL FUNCTION 'RSDQ_VIEW_INFOPROV'
  EXPORTING
    i_infoprov          = p_dta
    i_s_dta             = g_s_dta
    i_t_dta_dime        = g_t_dta_dime
    i_show_sids         = p_ssids
    i_show_dimids       = p_sdims
```


© 2010 SAP AG

```
    i_tech_nms           = p_tchnm
    i_use_db_aggregation = p_dbagg
    i_repnm              = p_repnm
  CHANGING
    c_t_ioinf            = g_t_ioinf
  EXCEPTIONS
    dta_not_found        = 1
    illegal_input        = 2
    OTHERS               = 4.
IF sy-subrc EQ 1.
  MESSAGE i151(brain) WITH p_dta.
ELSEIF sy-subrc > 1.
  MESSAGE i299(brain) WITH 'ZLISTCUBE' 'RSDQ_VIEW_INFOPROV'.
ENDIF.
```

d) Change the logic for PBO for screen 2000 (Include the logic for Function Codes '*SELE*' and '*FETC*'):

```
MODULE mark INPUT.
* INSERT
  DATA g_id TYPE memory_id.
* INSERT

  FIELD-SYMBOLS:
    <l_s_tabl_cntl>  TYPE rsdq_s_tabl_cntl.

  CASE g_okcode.
    WHEN 'MALL'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        <l_s_tabl_cntl>-mark = rs_c_true.
      ENDLOOP.

    WHEN 'DALL'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        <l_s_tabl_cntl>-mark = rs_c_false.
      ENDLOOP.

    WHEN 'MCHA'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name NP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_false.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'MSID'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name NP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_true.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'NKEY'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name CP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_false.
```


© 2010 SAP AG

```
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'NSID'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name CP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_true.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'KYF'.
      l_is_kyf = rs_c_false.

*   clear the marks except for all key figures ...
      LOOP AT g_t_tabl_cntl[]
        ASSIGNING <l_s_tabl_cntl>.

        IF <l_s_tabl_cntl>-tech_name = 'KYF'.
          l_is_kyf = rs_c_true.
        ENDIF.

        IF l_is_kyf = rs_c_true.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

* INSERT
    WHEN 'SELE'. "Store
      CONCATENATE p_dta sy-uname INTO g_id.
      EXPORT g_t_tabl_cntl TO DATABASE indx(st) ID g_id .
      IF sy-subrc <> 0.
        MESSAGE s368(00) WITH 'Selection Error(Saving)'.
      ELSE.
        CALL FUNCTION 'GUI_DOWNLOAD'
          EXPORTING
            filename                = string_file
            confirm_overwrite       = 'X'
          TABLES
            data_tab                = g_t_tabl_cntl
          EXCEPTIONS
            file_write_error        = 1
            no_batch                = 2
            gui_refuse_filetransfer = 3
            invalid_type            = 4
            no_authority            = 5
            unknown_error           = 6
            header_not_allowed      = 7
            separator_not_allowed   = 8
            filesize_not_allowed    = 9
            header_too_long         = 10
            dp_error_create         = 11
            dp_error_send           = 12
            dp_error_write          = 13
            unknown_dp_error        = 14
```


© 2010 SAP AG

```
            access_denied           = 15
            dp_out_of_memory        = 16
            disk_full               = 17
            dp_timeout              = 18
            file_not_found          = 19
            dataprovider_exception  = 20
            control_flush_error     = 21
            OTHERS                   = 22.
          MESSAGE s368(00) WITH 'Last Selection Saved Successfully'.
        ENDIF.
      WHEN 'FETC'. "Retrieve
        CONCATENATE p_dta sy-uname INTO g_id.
        IMPORT g_t_tabl_cntl FROM DATABASE indx(st) ID g_id .
        IF sy-subrc <> 0.
          MESSAGE s368(00) WITH 'No Selection Found'.
        ELSE.
          MESSAGE s368(00) WITH 'Selection Retrieved'.
        ENDIF.

  * INSERT
      WHEN OTHERS.
    ENDCASE.
  ENDMODULE.
```

When OK Code (Equals To):

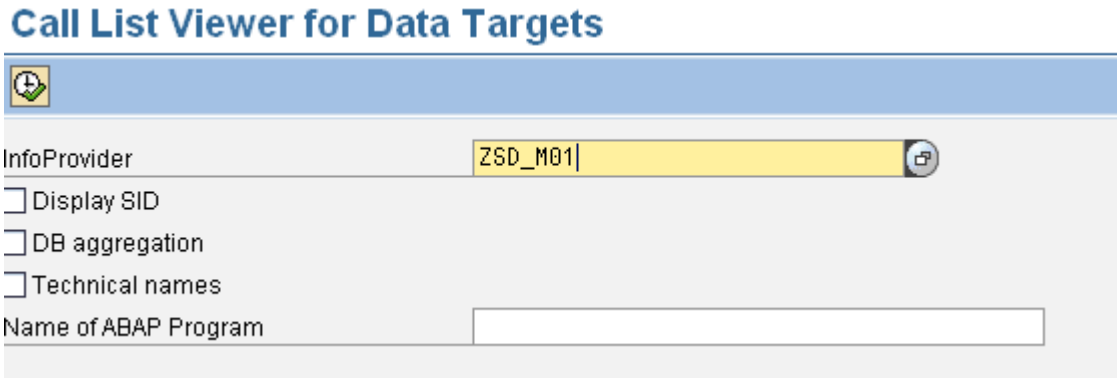
1)'*SELE*' (Saving the Selection Conditions): It saves the characteristics and key figures chose to Database Table INDX.Also it saves a flat file which contains the list of characteristics and key figures which were selected in the last iteration.

2) '*FETC*' (Retrieve the Selection Conditions): It retrieves the characteristics and key figures , which were chose in the last display from database table INDX.

Execution (ZLISTCUBE)

SAVE SELECTION

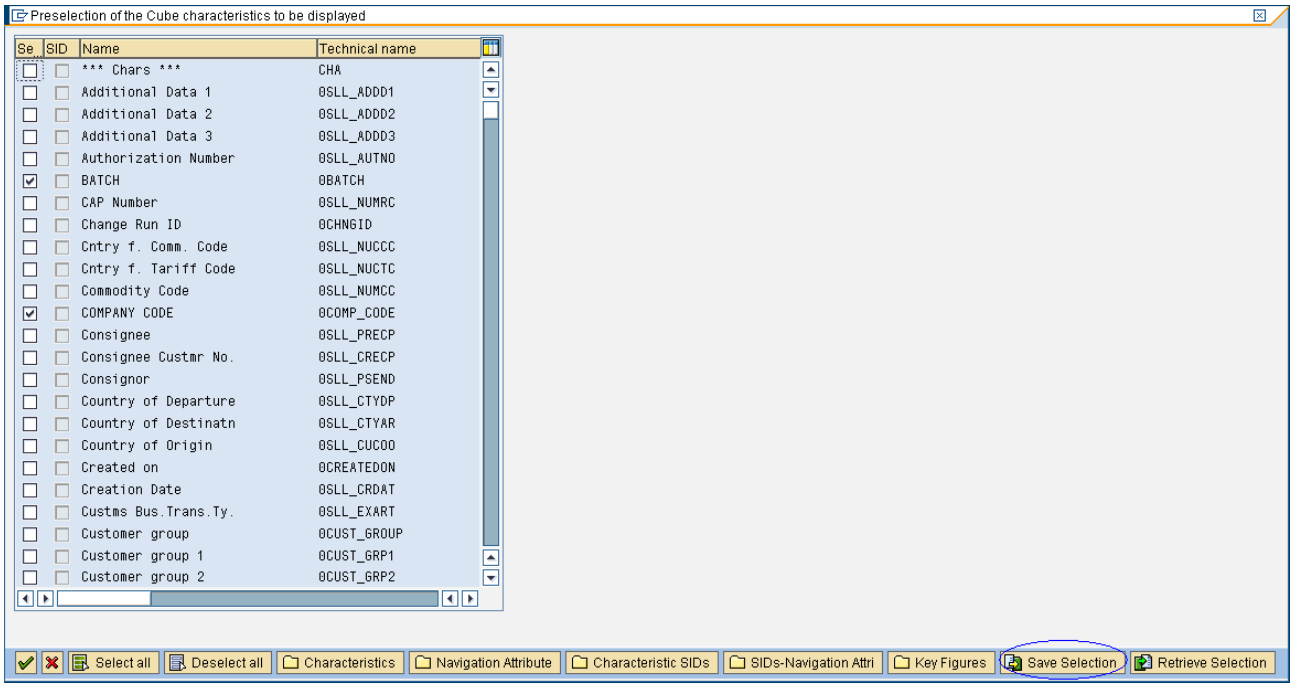
Step1: Go to transaction ZLISTCUBE and enter the Infoprovider 'ZSD_M01'.



Execute the transaction.

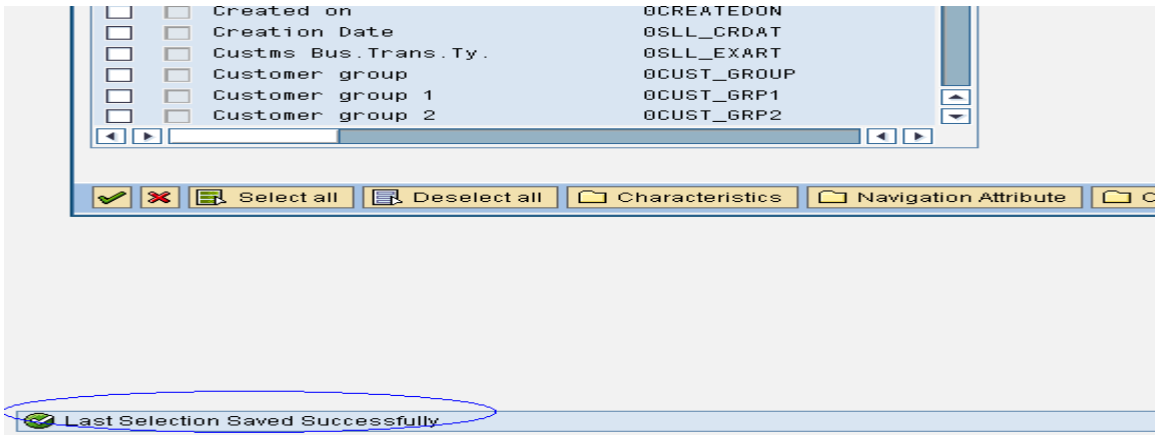
© 2010 SAP AG

Step2: Select the same set of Characteristics and Key Figures as selected earlier section:



Characteristics: 0BATCH, 0COMP_CODE, 0S_ORD_ITEM and 0CNF_QTY.

After the characteristics and Key Figures are selected, click '**Save Selection**'.

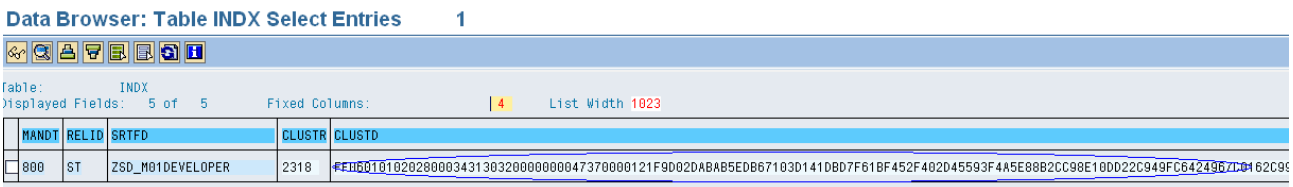


© 2010 SAP AG

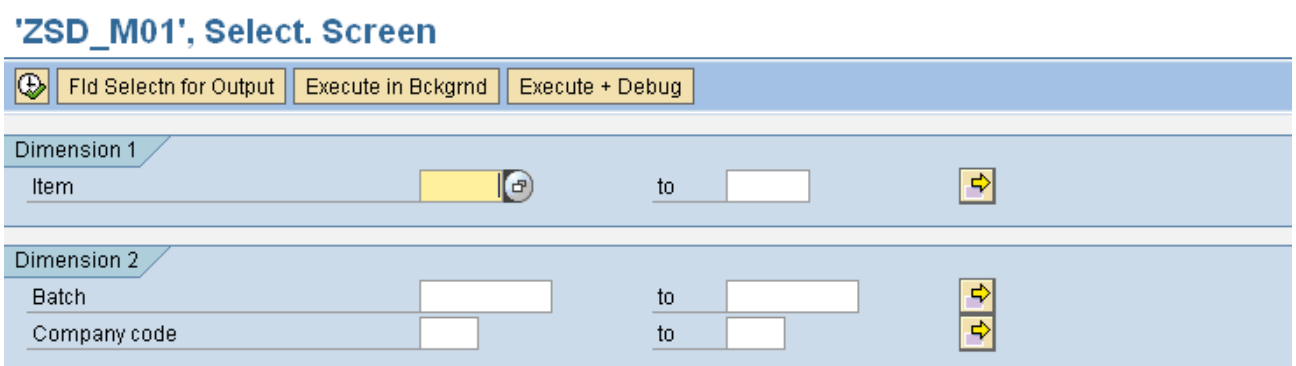
It gives a successful message '**Last Selection Saved Successfully**'. Also a file 'ZLISTCUBE' would be created in local drive (Location as mentioned in the variable 'String_file'), which contains the list of Characteristics and Key Figures (Flagged).

```
  *** Chars ***                  CHA
  Additional Data 1              0SLL_ADDD1
  Additional Data 2              0SLL_ADDD2
  Additional Data 3              0SLL_ADDD3
  Authorization Number           0SLL_AUTNO
X BATCH                          0BATCH
  CAP Number                     0SLL_NUMRC
  Change Run ID                  0CHNGID
  Cntry f. Comm. Code            0SLL_NUCCC
  Cntry f. Tariff Code           0SLL_NUCTC
  Commodity Code                 0SLL_NUMCC
X COMPANY CODE                   0COMP_CODE
  Consignee                      0SLL_PRECP
  Consignee Custmr No.           0SLL_CRECP
  Consignor                      0SLL_PSEND
  Country of Departure           0SLL_CTYDP
```

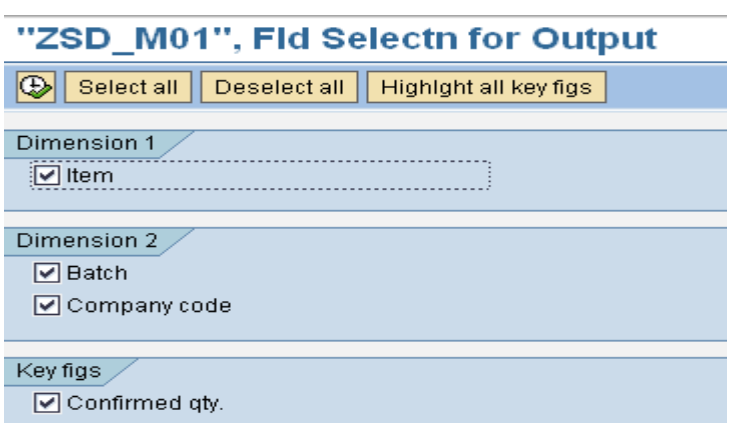
- The Selection Screen Values would be exported to Database Table INDX for the last selection(Creating particular Cluster ID) :



After this is done, Click 'OK' to go to next screen.



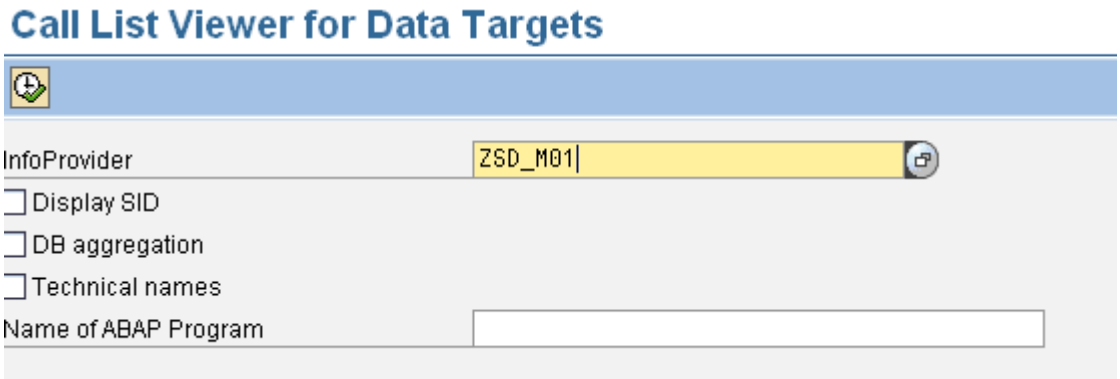
Check the 'Field Selection for Output'



© 2010 SAP AG

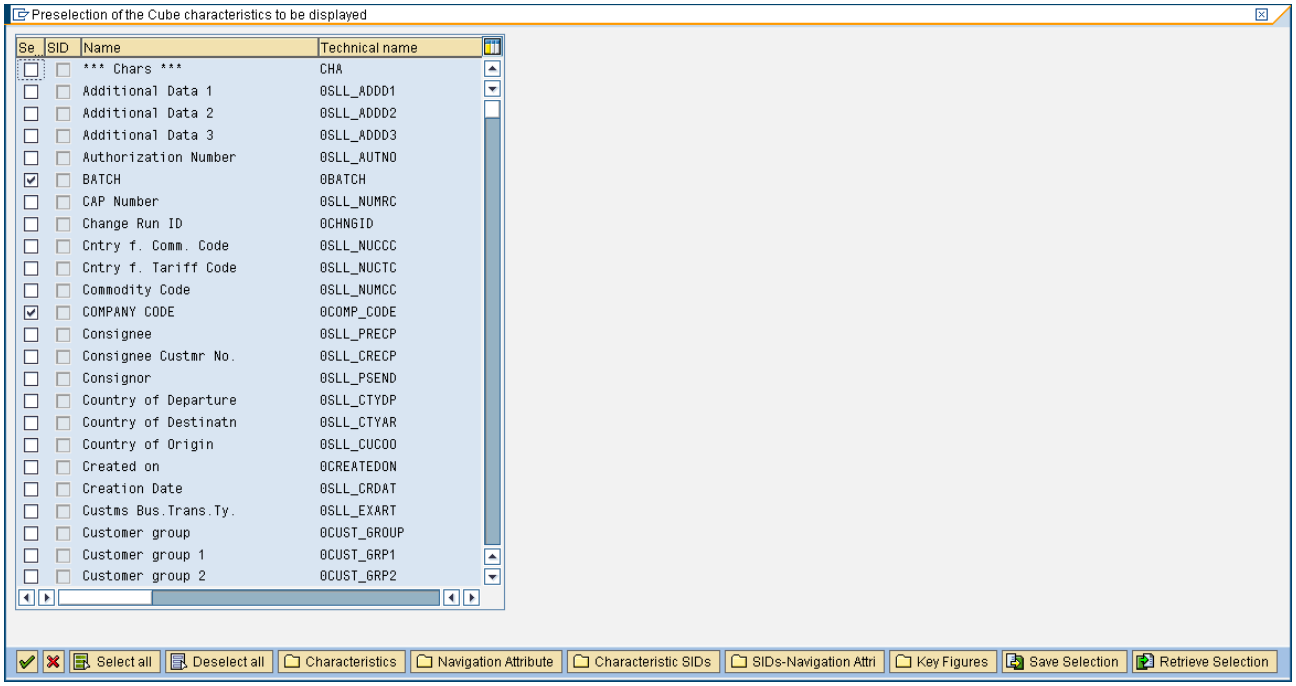
RETRIEVE SELECTION

Step1: Go to transaction ZLISTCUBE and enter the Infoprovider 'ZSD_M01'.



Execute the transaction.

Step2: Click 'Retrieve **Selection'** in the next screen



(Values would be fetched from the Last Cluster ID Created in Table INDX)

All the Characteristics and Key Figures would be selected as per last iteration.

Complete ABAP Code (ZRSDD_SHOW_ICUBE)

All the Characteristics and Key Figures would be selected as per last iteration.

```
*&---------------------------------------------------------------------*
*& Report  ZRSDD_SHOW_ICUBE                                            *
*&                                                                     *
*&---------------------------------------------------------------------*
*& This report generates a infocube-specific listreport with a
*& selection screen to be able to select data from the infocube.
*& Listcube does not have any OLAP-Logic, just displays the plain
*& infocube content.
*&---------------------------------------------------------------------*
```


© 2010 SAP AG

```
REPORT  zrsdd_show_icube              .

TYPE-POOLS: rsd, rsdq.
DATA:
  g_r_dta          TYPE REF TO cl_rsd_dta,
  g_obj_overflow   TYPE rs_bool,
  g_okcode(4)      TYPE c,
  g_tlogo          TYPE rs_tlogo,
  g_t_tlogo        TYPE rs_t_tlogo,
  l_is_kyf         TYPE rs_bool,
  g_s_dta          TYPE rsd_s_dta,
  g_t_ioinf        TYPE rsdq_t_iobj_info,
  g_t_dta_dime     TYPE rsd_t_dta_dime,
  g_t_tabl_cntl    TYPE rsdq_t_tabl_cntl WITH HEADER LINE,
  g_t_dynp         LIKE dynpread OCCURS 0 WITH HEADER LINE.

* INSERT

DATA string_file TYPE string VALUE
  'C:\Documents and Settings\stigga\Desktop\zlistcube.txt'.
DATA w_mark LIKE sy-tabix.
DATA w_loops LIKE sy-loopc.

* INSERT


CONTROLS:
  table_cntl       TYPE TABLEVIEW USING SCREEN 2000.

FIELD-SYMBOLS:
  <g_s_tabl_cntl>  TYPE rsdq_s_tabl_cntl.

PARAMETER:
  p_tlogo          TYPE rs_tlogo          NO-DISPLAY
                                          MEMORY ID /bic/rsdq/cubetype,
  p_dta            TYPE rsinfoprov        OBLIGATORY
                                          MEMORY ID /bic/rsdq/infoprov,
  p_ssids          TYPE dbman_show_sid    AS CHECKBOX
                                          MEMORY ID /bic/rsdq/showsids,
  p_sdims          TYPE dbman_show_dim    "AS  CHECKBOX
                                          "MEMORY ID /bic/rsdq/showdims
                                          NO-DISPLAY,
  p_dbagg          TYPE dbman_db_aggregation
                                          AS CHECKBOX
                                          MEMORY ID /bic/rsdq/db_aggr,
  p_tchnm          TYPE dbman_technm      AS CHECKBOX
                                          MEMORY ID /bic/rsdq/tech_nms,
  p_repnm          TYPE rsdr0_repnm.

SET PARAMETER ID '/BIC/RSDQ/INFOPROV'   FIELD p_dta.
SET PARAMETER ID '/BIC/RSDQ/SHOWSIDS'   FIELD p_ssids.
SET PARAMETER ID '/BIC/RSDQ/SHOWDIMS'   FIELD p_sdims.
SET PARAMETER ID '/BIC/RSDQ/DB_AGGR'    FIELD p_dbagg.
SET PARAMETER ID '/BIC/RSDQ/TECH_NMS'   FIELD p_tchnm.
SET PARAMETER ID '/BIC/RSDQ/REPNM'      FIELD p_repnm.
```


© 2010 SAP AG

```
* Displaying DIMIDS is not implemented yet
p_sdims = rs_c_false.

* retrieve information about the infocube infoobjects
CALL FUNCTION 'RSDQ_GET_DTA_INFO'
  EXPORTING
    i_infoprov   = p_dta
    i_show_sids  = p_ssids
  IMPORTING
    e_s_dta      = g_s_dta
    e_t_ioinf    = g_t_ioinf
    e_t_dta_dime = g_t_dta_dime
  EXCEPTIONS
    OTHERS       = 1.
IF sy-subrc <> 0.
  MESSAGE e203(dbman) WITH p_dta.
ENDIF.

SET PARAMETER ID '/BIC/RSDQ/CUBETYPE'   FIELD g_s_dta-tlogo.

DO.
* check whether all infoobjects fit into the
* report to be generated
  CALL FUNCTION 'RSDQ_TABLCNTL_FROM_IOINF'
    EXPORTING
      i_t_ioinf      = g_t_ioinf
      i_called_from  = 'L'         "Called by Listcube
    IMPORTING
      e_t_tabl_cntl  = g_t_tabl_cntl[]
      e_obj_overflow = g_obj_overflow.

  IF g_obj_overflow EQ rs_c_true.
*   customer has to preselect infoobjects to be
*   displayed in selection screen of report
    MESSAGE i201(rsdd).

    l_is_kyf = rs_c_false.
*   In preselection panel clear the marks for all
*   infoobjects except for the key figures ...
    LOOP AT g_t_tabl_cntl[]
      ASSIGNING <g_s_tabl_cntl>.

      IF <g_s_tabl_cntl>-tech_name = 'KYF'.
        l_is_kyf = rs_c_true.
      ENDIF.

      IF l_is_kyf = rs_c_false.
        CLEAR <g_s_tabl_cntl>-mark.
      ENDIF.
    ENDLOOP.

*     IF 1 = 1. " sy-uname = 'RUEGER' or
*       " sy-uname = 'DIPPER'.
*       LOOP AT g_t_tabl_cntl WHERE tech_name = 'CHA'
*                                   OR tech_name = 'KYF'
*                                   OR name      = text-029
```


© 2010 SAP AG

```
*                                OR name       = text-030.
*         DELETE g_t_tabl_cntl.
*       ENDLOOP.
*       CALL FUNCTION 'RSDQ_PRESEL_SCREEN_2010'
*         EXPORTING
*           i_t_ioinf      = g_t_ioinf
*         CHANGING
*           c_t_tabl_cntl = g_t_tabl_cntl[].
*     ELSE.
*      display preselection panel
    CALL SCREEN 2000
        STARTING AT 10 10.
*     ENDIF.
*    retrieve infoobjects to be displayed.
    CALL FUNCTION 'RSDQ_IOINF_FROM_TABLCNTL'
      EXPORTING
        i_t_tabl_cntl = g_t_tabl_cntl[]
      CHANGING
        c_t_ioinf     = g_t_ioinf.

  ELSE.
    EXIT.
  ENDIF.
ENDDO.

* build the infocube-specific listreport and
* execute it.
CALL FUNCTION 'RSDQ_VIEW_INFOPROV'
  EXPORTING
    i_infoprov           = p_dta
    i_s_dta              = g_s_dta
    i_t_dta_dime         = g_t_dta_dime
    i_show_sids          = p_ssids
    i_show_dimids        = p_sdims
    i_tech_nms           = p_tchnm
    i_use_db_aggregation = p_dbagg
    i_repnm              = p_repnm
  CHANGING
    c_t_ioinf            = g_t_ioinf
  EXCEPTIONS
    dta_not_found        = 1
    illegal_input        = 2
    OTHERS               = 4.
IF sy-subrc EQ 1.
  MESSAGE i151(brain) WITH p_dta.
ELSEIF sy-subrc > 1.
  MESSAGE i299(brain) WITH 'ZLISTCUBE' 'RSDQ_VIEW_INFOPROV'.
ENDIF.

*******************************
* end of report
*******************************

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_dta.

* g_t_dynp-fieldname = 'P_CUBET'.
```


© 2010 SAP AG

```
*  g_t_dynp-fieldname = 'P_TLOGO'. "UK 998954
*  APPEND g_t_dynp.
  g_t_dynp-fieldname = 'P_DTA'.
  APPEND g_t_dynp.

  CALL FUNCTION 'DYNP_VALUES_READ'
    EXPORTING
      dyname              = 'RSDD_SHOW_ICUBE'
      dynumb              = '1000'
      translate_to_upper  = 'X'
    TABLES
      dynpfields          = g_t_dynp
    EXCEPTIONS
      invalid_abapworkarea = 1
      invalid_dynprofield  = 2
      invalid_dynproname   = 3
      invalid_dynpronummer = 4
      invalid_request      = 5
      no_fielddescription  = 6
      invalid_parameter    = 7
      undefind_error       = 8
      OTHERS               = 9.

*  CHECK sy-subrc = 0.

  READ TABLE g_t_dynp
       WITH KEY fieldname = 'P_DTA'.
  p_dta = g_t_dynp-fieldvalue.

  CLEAR g_t_tlogo.
  READ TABLE g_t_dynp
       WITH KEY fieldname = 'P_TLOGO'.

  p_tlogo = g_t_dynp-fieldvalue.
  IF p_tlogo IS INITIAL OR
     ( p_tlogo <> rs_c_tlogo-infocube      AND
       p_tlogo <> rs_c_tlogo-aggregate     AND
       p_tlogo <> rs_c_tlogo-multiprovider AND
       p_tlogo <> rs_c_tlogo-ods_object    AND
       p_tlogo <> rs_c_tlogo-infoobject    AND
       p_tlogo <> rs_c_tlogo-infoset ).
    CLEAR p_tlogo.
    CLEAR g_t_tlogo.
    APPEND rs_c_tlogo-infocube      TO g_t_tlogo.
    APPEND rs_c_tlogo-aggregate     TO g_t_tlogo.
    APPEND rs_c_tlogo-multiprovider TO g_t_tlogo.
    APPEND rs_c_tlogo-ods_object    TO g_t_tlogo.
    APPEND rs_c_tlogo-infoobject    TO g_t_tlogo.
    APPEND rs_c_tlogo-aggrlevel     TO g_t_tlogo.
    APPEND rs_c_tlogo-infoset       TO g_t_tlogo.
  ENDIF.

  CALL METHOD cl_rsd_dta=>f4
    EXPORTING
      i_tlogo    = p_tlogo
      i_t_tlogo  = g_t_tlogo
```


© 2010 SAP AG

```
    CHANGING
      c_infoprov = p_dta.


*&---------------------------------------------------------------------*
*&      Module  PROCESS_SELECTION  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE process_selection INPUT.

  MODIFY g_t_tabl_cntl INDEX table_cntl-current_line.

ENDMODULE.                           " PROCESS_SELECTION  INPUT


*&---------------------------------------------------------------------*
*&      Module  STATUS_2000  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE status_2000 OUTPUT.
  SET PF-STATUS 'PF_2000'.
*  SET TITLEBAR '200'.
* INSERT
  CLEAR w_mark.
  LOOP AT g_t_tabl_cntl WHERE mark = 'X'.
    w_mark = w_mark + 1.
  ENDLOOP.
   SET TITLEBAR '200'.
* INSERT
  CLEAR g_okcode.
ENDMODULE.                           " STATUS_2000  OUTPUT


*&---------------------------------------------------------------------*
*&      Module  EXIT  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE exit INPUT.
  IF g_okcode EQ 'CNCL'.
    LEAVE PROGRAM.
  ENDIF.
ENDMODULE.                           " EXIT  INPUT


*&---------------------------------------------------------------------*
*&      Module  MARK  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE mark INPUT.
* INSERT
  DATA g_id TYPE memory_id.
* INSERT
```


© 2010 SAP AG

```
  FIELD-SYMBOLS:
    <l_s_tabl_cntl>  TYPE rsdq_s_tabl_cntl.

  CASE g_okcode.
    WHEN 'MALL'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        <l_s_tabl_cntl>-mark = rs_c_true.
      ENDLOOP.

    WHEN 'DALL'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        <l_s_tabl_cntl>-mark = rs_c_false.
      ENDLOOP.

    WHEN 'MCHA'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name NP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_false.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'MSID'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name NP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_true.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'NKEY'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name CP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_false.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'NSID'.
      LOOP AT g_t_tabl_cntl ASSIGNING <l_s_tabl_cntl>.
        IF <l_s_tabl_cntl>-tech_name CP '*__*'
          AND <l_s_tabl_cntl>-sid EQ rs_c_true.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

    WHEN 'KYF'.
      l_is_kyf = rs_c_false.

*   clear the marks except for all key figures ...
      LOOP AT g_t_tabl_cntl[]
        ASSIGNING <l_s_tabl_cntl>.

        IF <l_s_tabl_cntl>-tech_name = 'KYF'.
          l_is_kyf = rs_c_true.
```


© 2010 SAP AG

```
        ENDIF.

        IF l_is_kyf = rs_c_true.
          <l_s_tabl_cntl>-mark = rs_c_true.
        ENDIF.
      ENDLOOP.

* INSERT
    WHEN 'SELE'. "Store
      CONCATENATE p_dta sy-uname INTO g_id.
      EXPORT g_t_tabl_cntl TO DATABASE indx(st) ID g_id .
      IF sy-subrc <> 0.
        MESSAGE s368(00) WITH 'Selection Error(Saving)'.
      ELSE.
        CALL FUNCTION 'GUI_DOWNLOAD'
          EXPORTING
            filename                = string_file
            confirm_overwrite       = 'X'
          TABLES
            data_tab                = g_t_tabl_cntl
          EXCEPTIONS
            file_write_error        = 1
            no_batch                = 2
            gui_refuse_filetransfer = 3
            invalid_type            = 4
            no_authority            = 5
            unknown_error           = 6
            header_not_allowed      = 7
            separator_not_allowed   = 8
            filesize_not_allowed    = 9
            header_too_long         = 10
            dp_error_create         = 11
            dp_error_send           = 12
            dp_error_write          = 13
            unknown_dp_error        = 14
            access_denied           = 15
            dp_out_of_memory        = 16
            disk_full               = 17
            dp_timeout              = 18
            file_not_found          = 19
            dataprovider_exception  = 20
            control_flush_error     = 21
            OTHERS                   = 22.
        MESSAGE s368(00) WITH 'Last Selection Saved Successfully'.
      ENDIF.
    WHEN 'FETC'. "Retrieve
      CONCATENATE p_dta sy-uname INTO g_id.
      IMPORT g_t_tabl_cntl FROM DATABASE indx(st) ID g_id .
      IF sy-subrc <> 0.
        MESSAGE s368(00) WITH 'No Selection Found'.
      ELSE.
        MESSAGE s368(00) WITH 'Selection Retrieved'.
      ENDIF.

* INSERT
    WHEN OTHERS.
```


© 2010 SAP AG

```
  ENDCASE.
ENDMODULE.                             " MARK  INPUT


*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_2000  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE user_command_2000 INPUT.
  IF g_okcode EQ 'OK'.
    READ TABLE g_t_tabl_cntl
      WITH KEY mark = rs_c_true
      TRANSPORTING NO FIELDS.

    IF sy-subrc <> 0.
      MESSAGE i203(rsdd).
      RETURN.
    ENDIF.

    SET SCREEN 0.
    LEAVE SCREEN.
  ENDIF.
ENDMODULE.                             " USER_COMMAND_2000  INPUT

* INSERT
*&---------------------------------------------------------------------*
*& Module set_table OUTPUT
*&---------------------------------------------------------------------*
MODULE set_table OUTPUT.
  w_loops = sy-loopc.
ENDMODULE. " set_table OUTPUT
* INSERT
```


© 2010 SAP AG

Related Content

For more information, visit the EDW HomePage .

© 2010 SAP AG

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.

© 2010 SAP AG