Send Email Using Dynamic Actions

Applies to:

SAP ECC6.0 (Release 700, SP 12). For more information, visit the ABAP homepage.

Summary

In any Organization, HR Management involves many processes to organize or maintain the employee's details. These details are stored as records through infotypes. We may enter into the situation to change or delete the already created data. In between of processing with infotypes and saving it in the database, some things need to be done dynamically based on particular screen value entered. Those types of actions are configured in the table. Using this table, the dynamic actions get triggered for the particular infotype.

Author: Chellamma Devi Chandrasekar

Company: Applexus Technologies

Created on: 26 April 2011

Author Bio

Chellamma Devi Chandrasekar is working as a SAP HR ABAP Technology Consultant in Applexus Software Solutions (P) Ltd.She has completed her Bachelor of Computer Science and Engineering and has 6 months of experience in HCM ABAP.

Table of Contents

Introductions

Dynamic actions are needed when we need to invoke or initiate an action when maintaining an infotype. For example when we try to create/modify/ deleting any infotype data, we need some action to be triggered. i.e. when you try to delete a particular data from an infotype without any authorization, a message should be displayed saying “Not to delete that particular infotype” . These types of actions can be performed using dynamic actions.

HCM example:

Let us try to send the email through dynamic actions to the concerned mail ID as soon as the employee is terminated.

For doing so first we need to know about

V_T588Z: maintenance view to maintain the dynamic action.

To get a better understanding about the maintenance view V_T588Z fields, refer the link given.

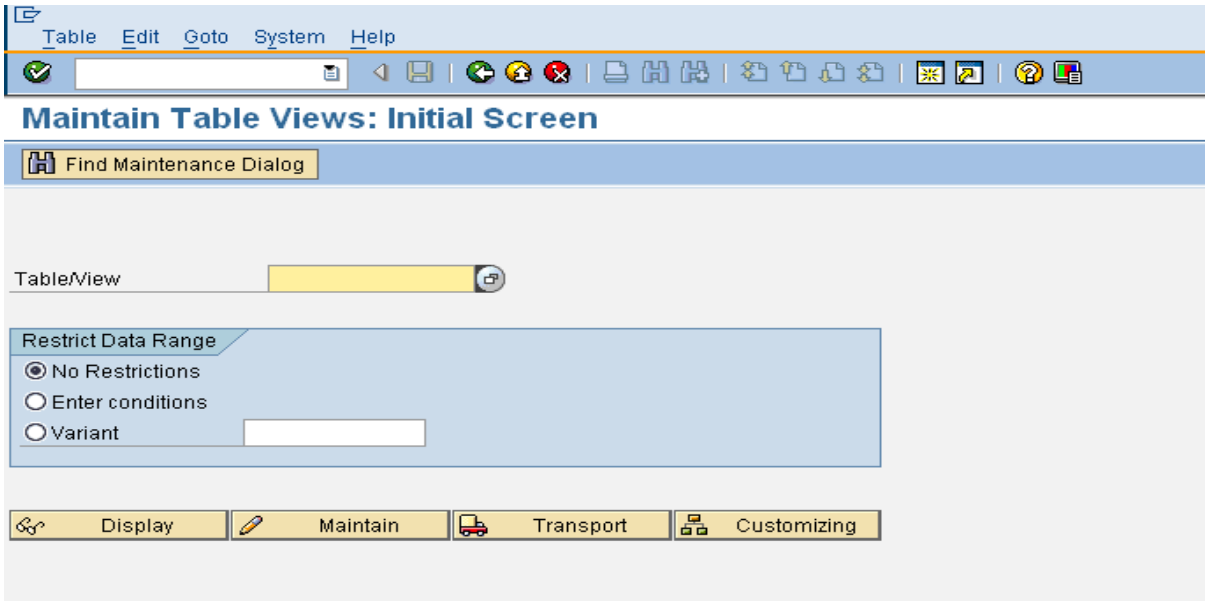
Steps of Creating Dynamic Actions

Step 1:

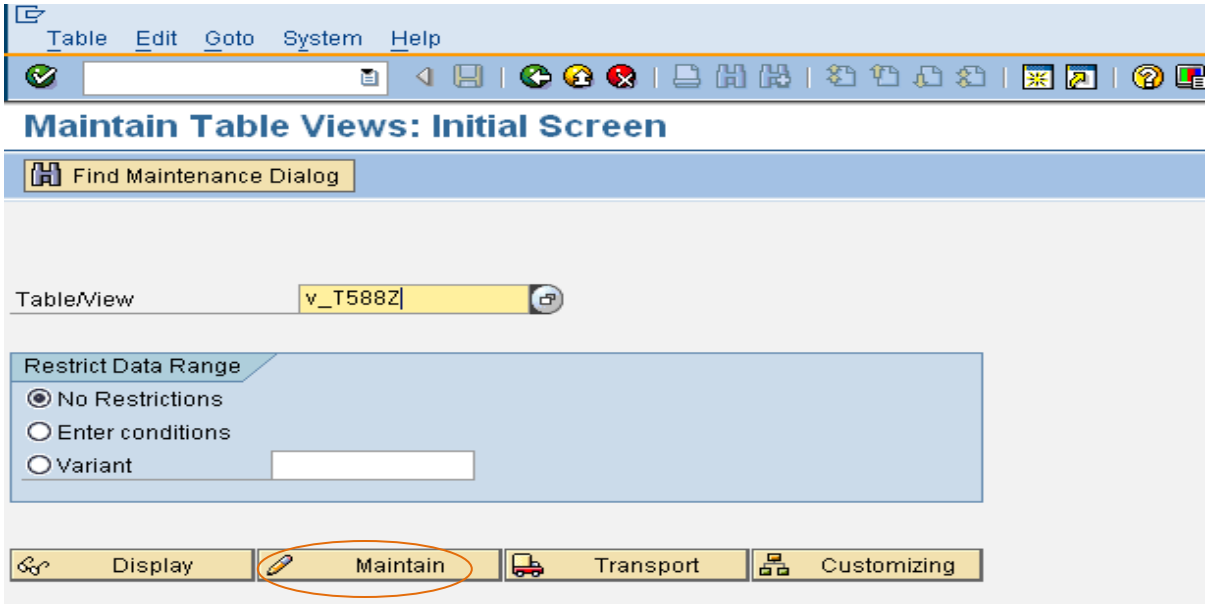
For creating dynamic actions first we need to create an entry in maintenance view V_T588Z based on the requirement.

The transaction SM30 is used to maintain the entries in the views

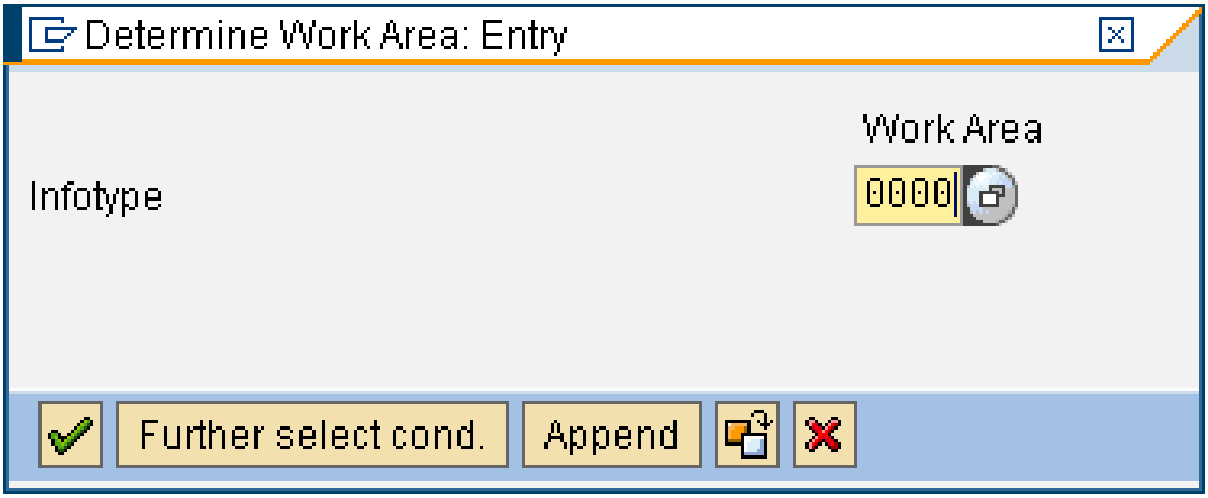
The initial screen of SM30 transaction looks as shown below:



Enter the maintenance view name V_T588Z and press on maintain button

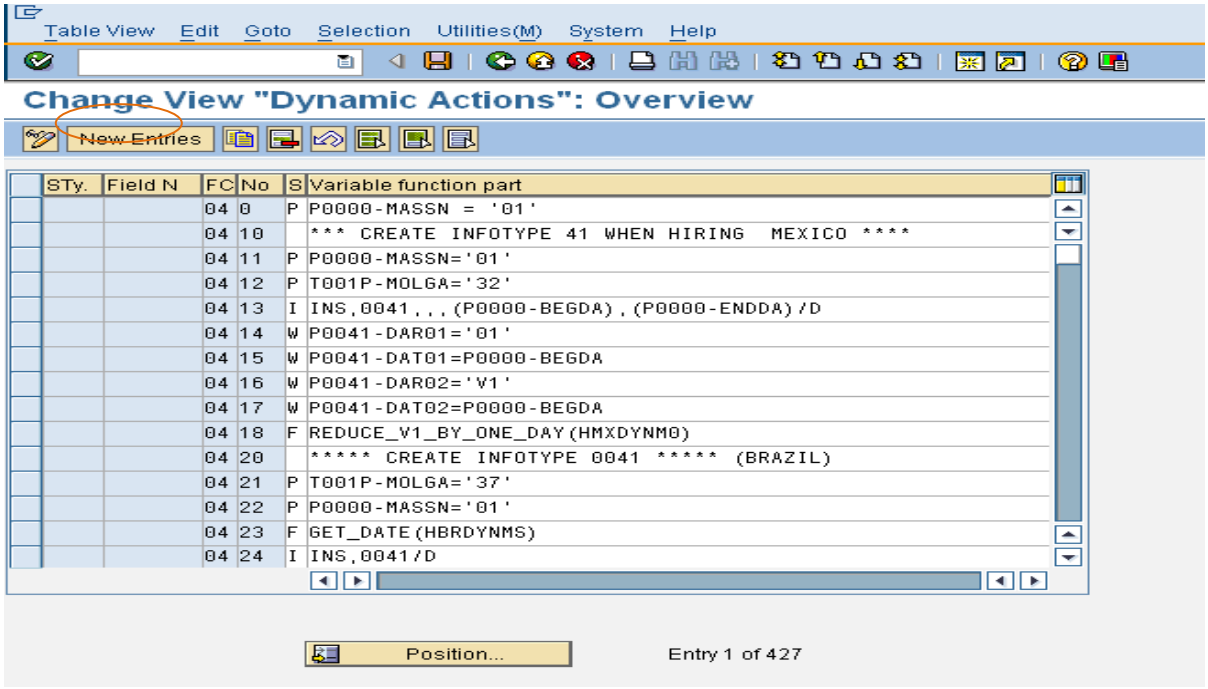


It will ask for a pop up regarding which infotype we are going to trigger. In our case we need to trigger an e-mail when terminating any employee. So info type 0000 is given and entered as shown below,



And press on tick button.

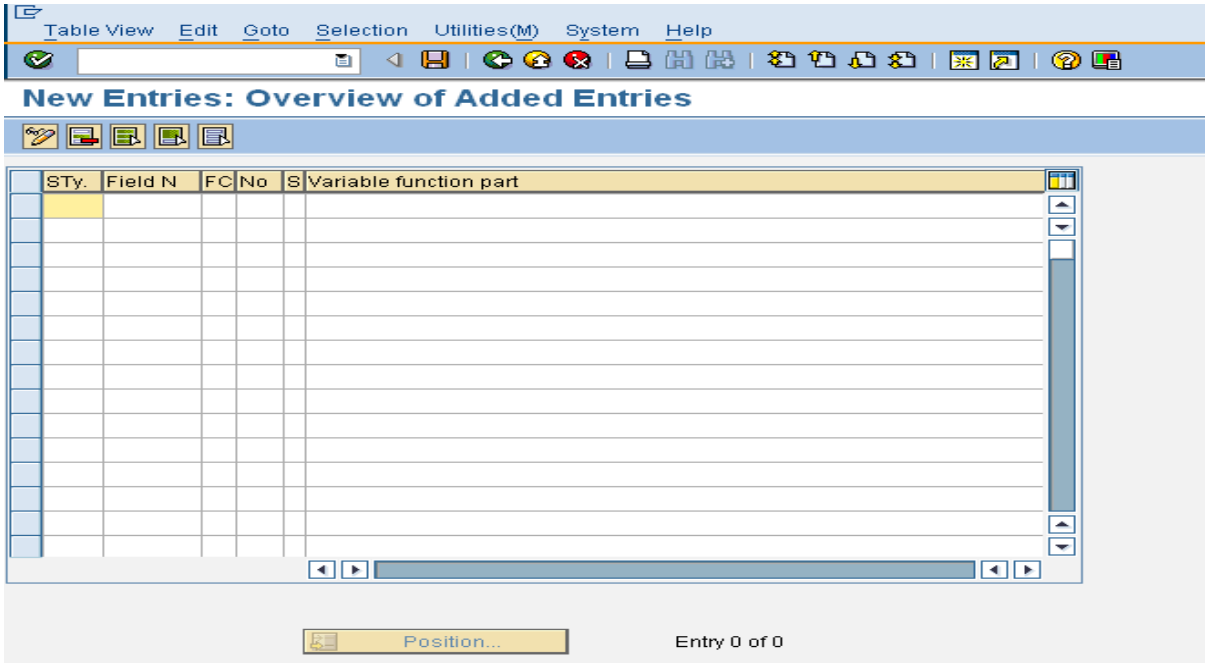
Click on New entries button on the next sreen as shown below.



A blank screen appears as shown below

Step 2:

Configure the table



In the above screen we have 6 fields they are :

Subtype: Which subtype of info type the action has to be triggered.

Field name: To trigger the action, which field value has to be considered?

FC: For each maintenance (create/ delete/ change), there are some function codes assigned.

To have better understanding, refer the Basic s in the given link.

Sequence number: To know on which line your subroutine or check is linked. i.e. the starting line of the coding.

Step: Defines the action to be carried out. (P for checking conditions, F for subroutines, I info type call, W-Defaulting).

Variable function part: This defines the action on fields based on the step code.

In our case we need to trigger an e-mail when terminating any employee. So the entries will be,

Subtype: Is blank because we are not considering any subtype.

Field name: MASSN (Action Type)

FC: 06 for change/create. In this case we are going to create a Termination for an employee.

To have better understanding, refer the Basic s in the given link.

Sequence number: In this case P0000-MASSn = '10' starts at line 301 and SEND_EMAIL_NOTE starts at line 302.

Step: In this case it is P for checking condition wheat her P0000-MASSn = '10' and

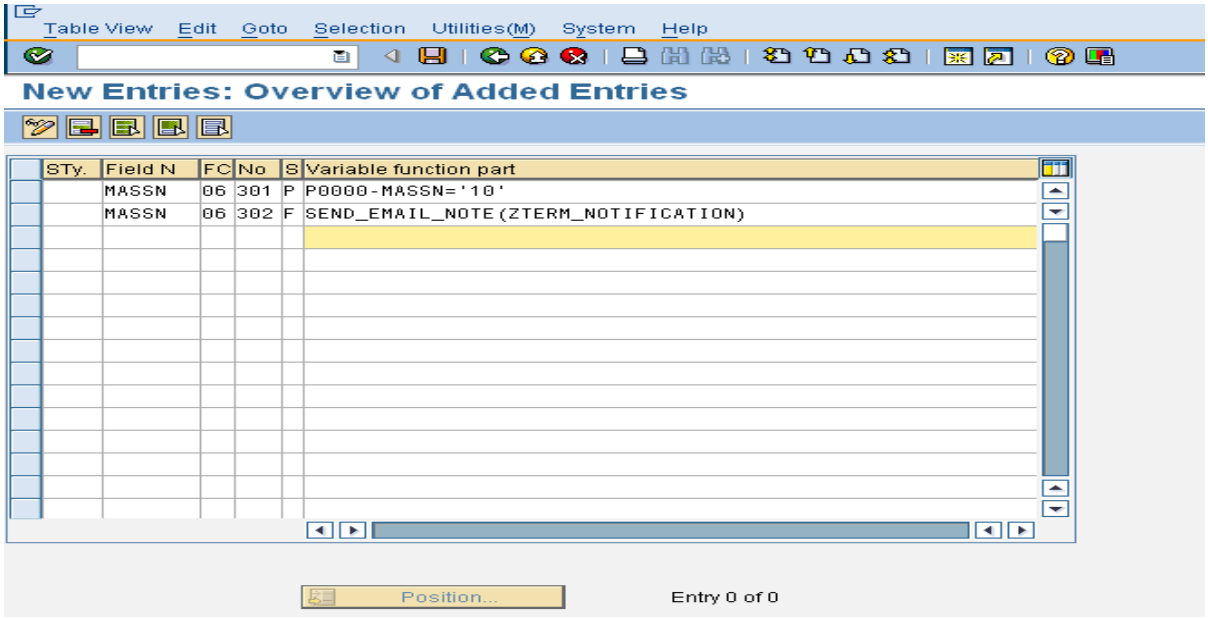
F - for subroutines SEND_EMAIL_NOTE.

Variable function part: in our case it is P0000-MASSn = '10' condition and

SEND_MAIL_NOTE : Sub routine Name

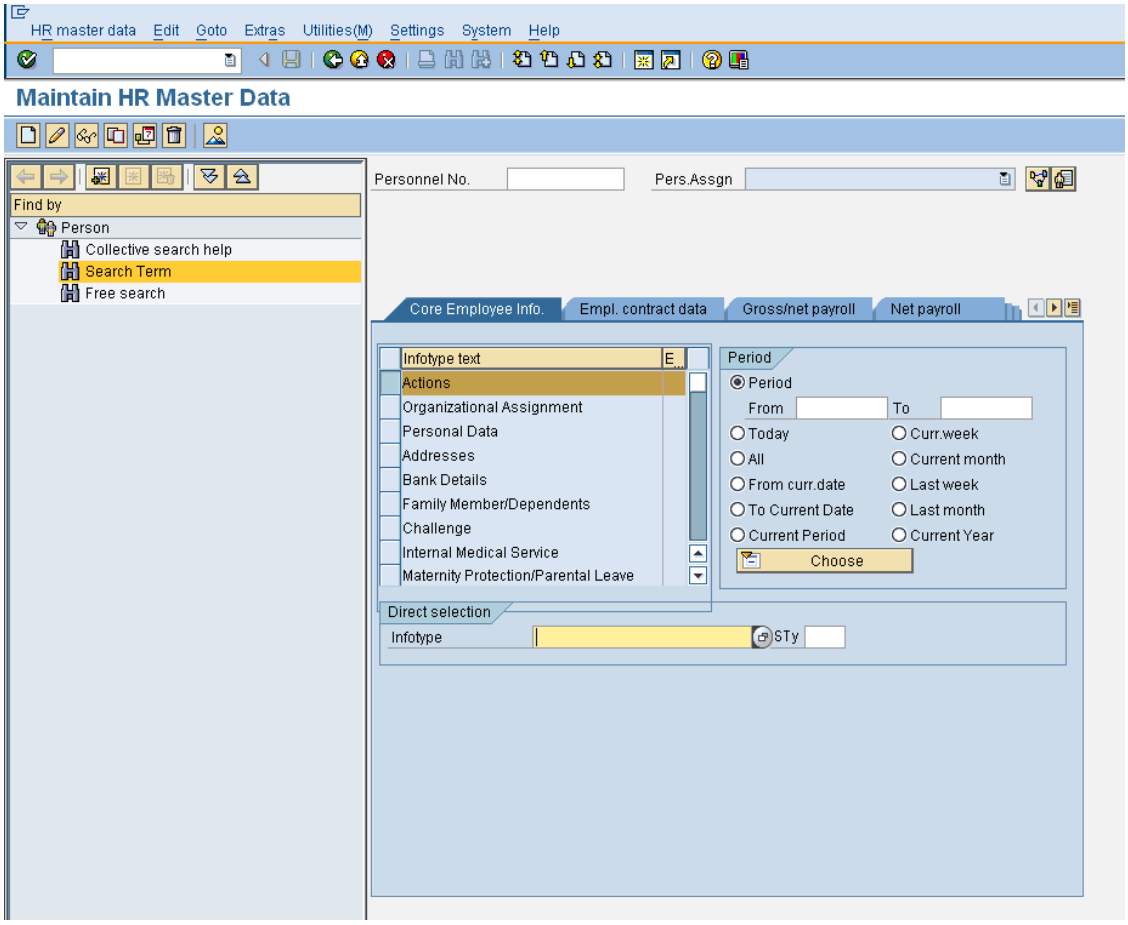
ZTERM_NOTIFICATION: Program name in which subroutine is placed.

After entering the values the screen looks as shown below,



Step 3:

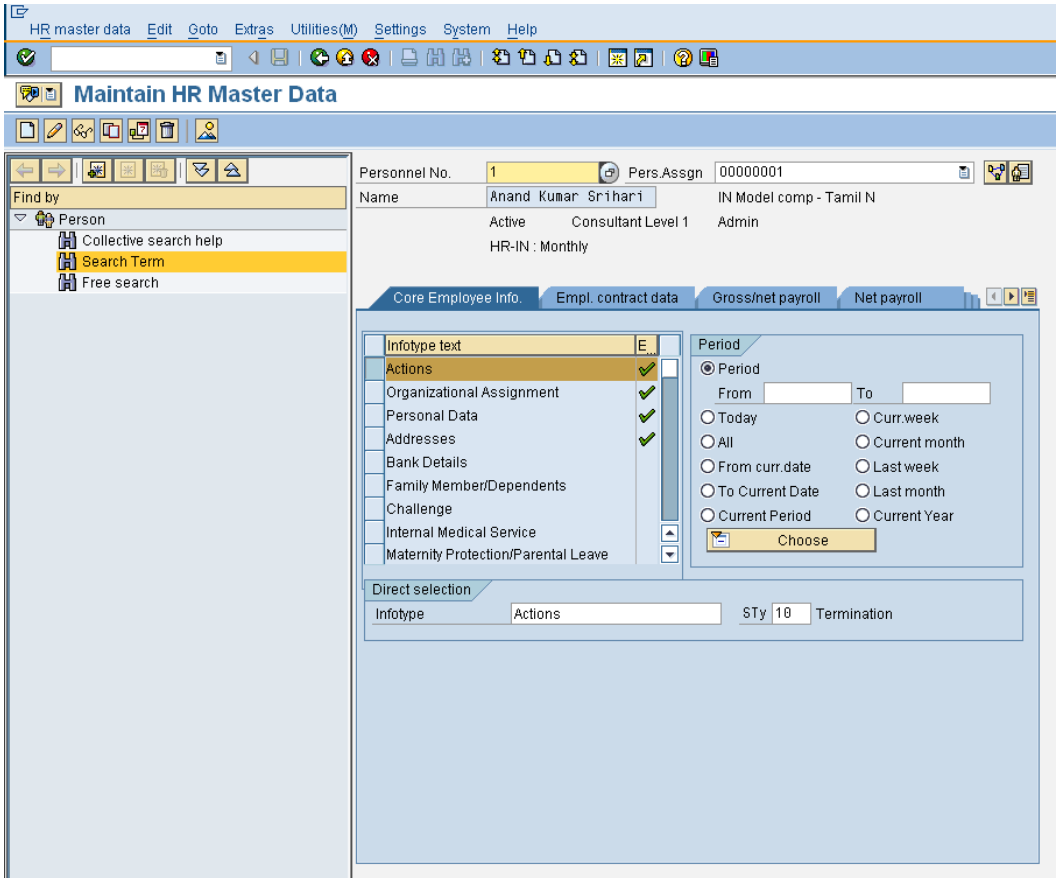
Go to T-Code PA30 (Maintaining HR master data).The initial screen looks as shown,



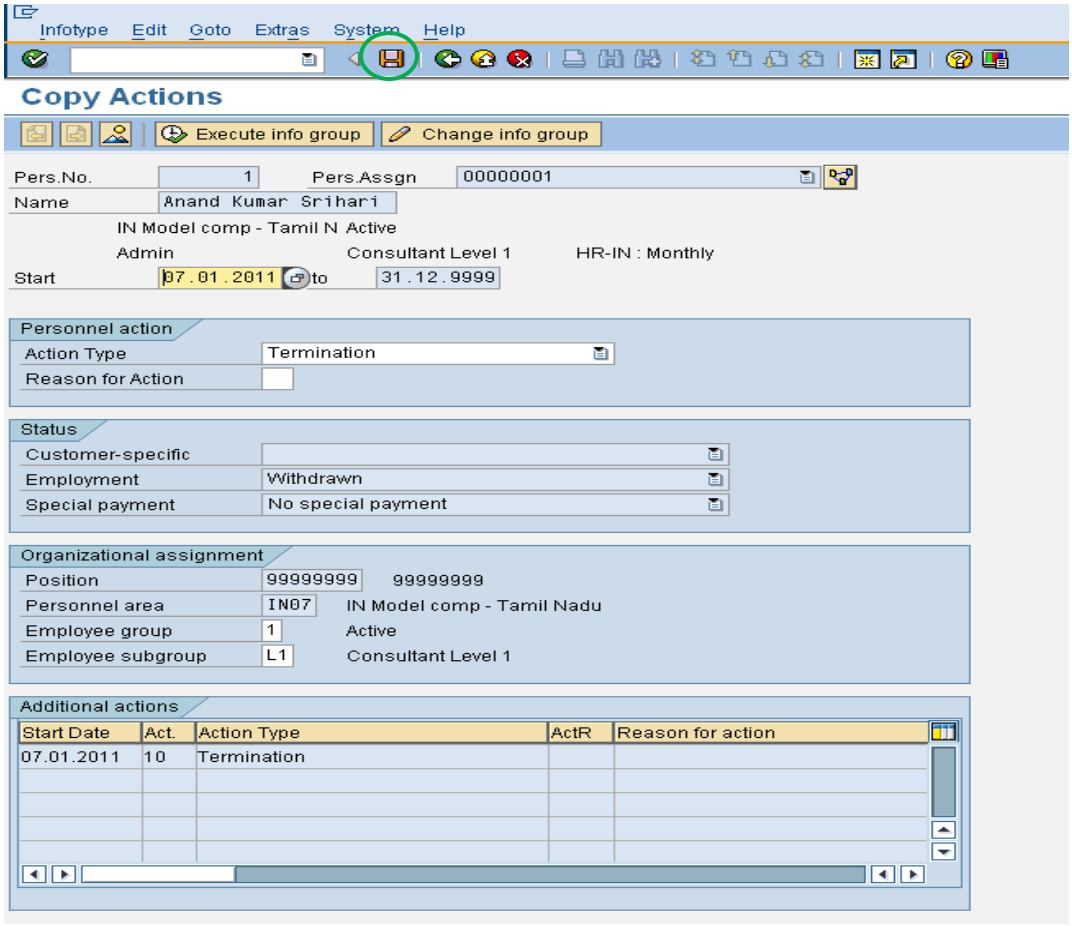
Enter the Personnel Number of an employee. In our case we can consider an employee with personnel number 1.

Select info type 0000(actions),

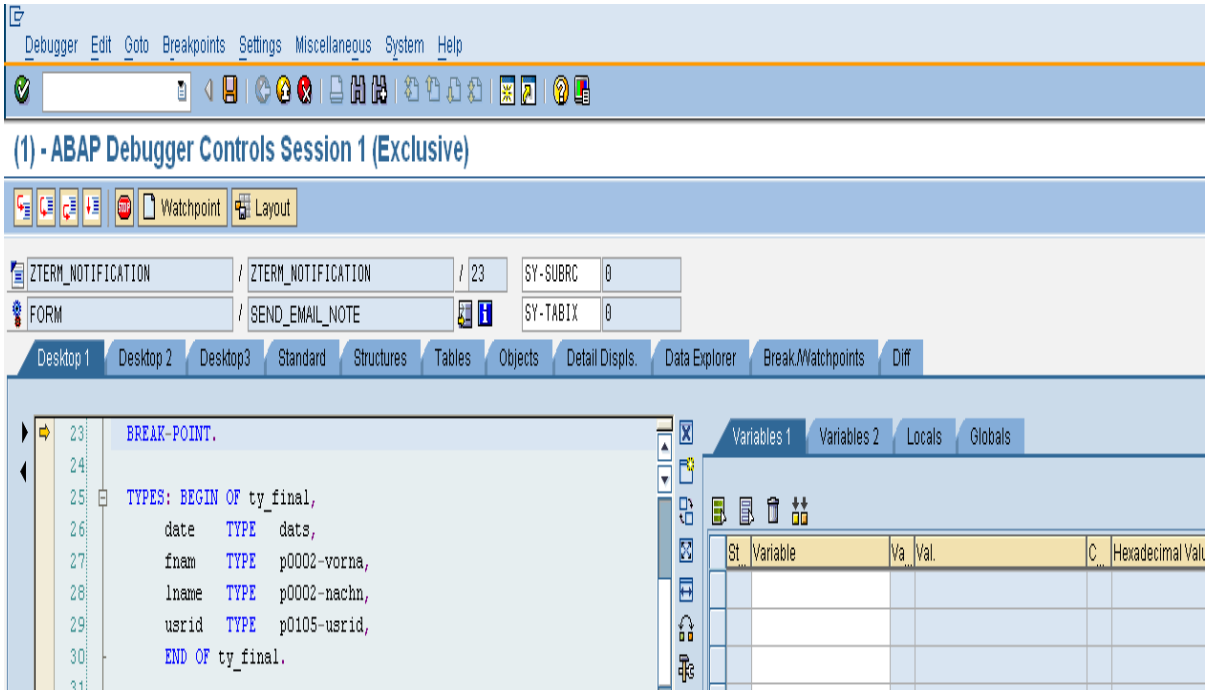
Subtype(10) for termination. As shown below.



Press on create button next screen appears as shown below,



The corresponding routine is created as report. The break point is set inside the code. When we click the save button, control goes to the code as shown below. Dynamic action gets triggered only when the save button is clicked.



The mail is prepared using the routine as follows.

Reference variables are declared.

```
lr_send_note       TYPE REF TO   cl_bcs,           " Send request
lr_document        TYPE REF TO   cl_document_bcs,  " Email attachment
lr_sender          TYPE REF TO   cl_sapuser_bcs,   " Sender Address
lr_receiver        TYPE REF TO   if_recipient_bcs, " Recipient Address
lr_bcs_exception   TYPE REF TO   cx_document_bcs,  " BCS Exception
lr_addr_exception  TYPE REF TO   cx_address_bcs,   " Address Exception
lr_send_exception  TYPE REF TO   cx_send_req_bcs,  " Send Exception
lv_result          TYPE           sy-binpt.
```

Message body is created with subject.

Document is prepared using the following method.

```
lr_document = cl_document_bcs=>create_document   “subject, Message body
```

Sender and receiver address are set using following method.

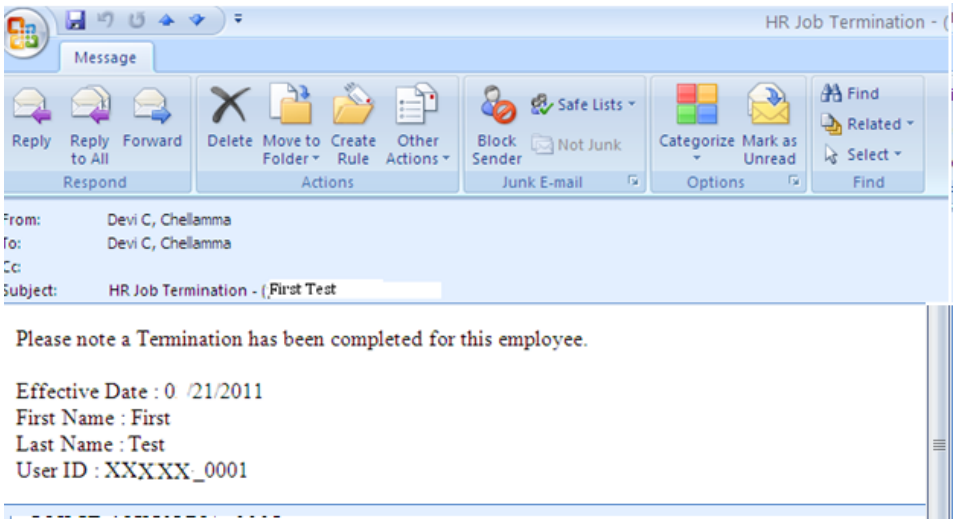
```
lr_sender = cl_sapuser_bcs=>create( sy-uname ).  “Sender

CALL METHOD lr_send_note->set_sender
lr_receiver = cl_cam_address_bcs=>create_internet_address( lv_mail ). “Receiver
CALL METHOD lr_send_note->add_recipient
```

Mail is sent through the method.

```
CALL METHOD lr_send_note->send
```

For Preparing and better understanding of mail process, refer the link given.

Below is the mail which has been sent through dynamic action.



The entire report program of this dynamic action is as follows,

Sample Code

```
REPORT  ZTERM_NOTIFICATION.

tables: p0000,p0001,p0002.

form SEND_EMAIL_NOTE .

  BREAK-POINT.

  TYPES: BEGIN OF ty_final,
       date    TYPE   dats,
       fnam    TYPE   p0002-vorna,
       lname   TYPE   p0002-nachn,
       usrid   TYPE   p0105-usrid,
       END OF ty_final.

data: lx_p0105  type p0105,
      lx_term   type ty_final.

data: lt_p0105 type table of p0105,
      lt_term  type table of ty_final.

CONSTANTS : lc_raw     TYPE   so_obj_tp   VALUE  'RAW',     " Document type
           lc_htab     TYPE   abap_char1  VALUE  cl_abap_char_utilities=>hori
zontal_tab,
           lc_space    type   abap_char1  value ' ',
           lc_open     type   abap_char1  value '(',
           lc_close    type   abap_char1  value ')',
           lc_x        type   boole_d     value 'X'.

 DATA :     lv_date      TYPE    char10,                      " Date
            lv_sub       TYPE    so_obj_des,                  " Subject
            lv_subject   type    so_obj_des,
            lt_msg_text  TYPE    soli_tab,
            lv_message   TYPE    string,
            lv_count     TYPE    so_obj_tp,
            lv_mail      TYPE    ad_smtpadr,
            lv_fnam_lnam type    PAD_CNAME,
            lv_admin     type    t526-sachn,
            lv_credcard  type    char1,
            v_pernr1     type    P_PERNR.


data:  v_pernr  type PERNR_D.

DATA: lr_send_note       TYPE REF TO   cl_bcs,            " Send request
      lr_document        TYPE REF TO   cl_document_bcs,   " Email attachment
      lr_sender          TYPE REF TO   cl_sapuser_bcs,    " Sender Address
      lr_receiver        TYPE REF TO   if_recipient_bcs,  " Recipient Address
      lr_bcs_exception   TYPE REF TO   cx_document_bcs,   " BCS Exception
      lr_addr_exception  TYPE REF TO   cx_address_bcs,    " Address Exception
      lr_send_exception  TYPE REF TO   cx_send_req_bcs,   " Send Exception
      lv_result          TYPE sybinpt.                   " Batch Input


v_pernr  = p0000-pernr.

lx_term-date  = Sy-datum.
lx_term-fnam  = p0002-vorna.
lx_term-lname = p0002-nachn.
**** For user ID ********************
CALL FUNCTION 'HR_READ_INFOTYPE'
  EXPORTING
```

```
    TCLAS                 = 'A'
    pernr                 = v_pernr
    infty                 = '0105'
   BEGDA                 = P0000-BEGDA
   ENDDA                 = P0000-ENDDA
  TABLES
    infty_tab             = lt_p0105
 EXCEPTIONS
   INFTY_NOT_FOUND       = 1
   OTHERS                = 2
        .
IF sy-subrc <> 0.
MESSAGE s071(zchellama).
ENDIF.

** Subtype Based filteration ******

 Read TABLE lt_P0105 INTO lx_P0105 WITH KEY PERNR = v_pernr
                                            SUBTY  = '0001'.
  if sy-subrc eq 0.
   lx_term-usrid  = lx_p0105-usrid.
  endif.

*******8 final Table
  APPEND lx_term TO lt_term.

**** Mail Preparation ***
if lt_term is not INITIAL.

   LOOP AT lt_term INTO lx_term.

  TRY.
  lr_send_note = cl_bcs=>create_persistent( ).

******************Subject for Mail*************************************
CONCATENATE text-007 lv_fnam_lnam INTO lv_subject
                                            SEPARATED BY lc_space.

 " Text-007 is Terminated Employee Details

CONCATENATE text-009  lc_space lv_date  INTO lv_sub
                                            SEPARATED BY lc_htab.
    APPEND lv_sub TO lt_msg_text.


**** First Name : xxxxxxx **********
    CONCATENATE text-010  lc_space lx_term-fnam  INTO lv_sub
                                            SEPARATED BY lc_htab.
                                        "text-010 is 'First name :' Label
    APPEND lv_sub TO lt_msg_text.

****Last Name  : xxxxx **************
    CONCATENATE text-011  lc_space lx_term-lname  INTO lv_sub
                                            SEPARATED BY lc_htab.
                                       "Text-011 is 'Last Name: ' Label
    APPEND lv_sub TO lt_msg_text.

********User ID : xxxxxx **********
  CONCATENATE text-012  lc_space lx_term-usrid  INTO lv_sub
                                         SEPARATED BY lc_htab.
                                          " Text-012 is 'User ID :  ' Label

    APPEND lv_sub TO lt_msg_text.
```

```
************ Creating Document(Message body) *************************
    lr_document = cl_document_bcs=>create_document(
                        i_type        = lc_raw
                        i_text        = lt_msg_text
                        i_subject     = lv_subject )
                       .
******* Document is added to send ************************************
    CALL METHOD lr_send_note->set_document
      EXPORTING
        i_document = lr_document.

*****Getting From address ***************************************************
        lr_sender = cl_sapuser_bcs=>create( sy-uname ).

**************Set the mail with from address*******************************
        CALL METHOD lr_send_note->set_sender
          EXPORTING
            i_sender = lr_sender.

******* Getting Recipient address ***************************************


        lv_mail =  'MAIL ID'.
        lr_receiver = cl_cam_address_bcs=>create_internet_address( lv_mail ).

        CALL METHOD lr_send_note->add_recipient
          EXPORTING
            i_recipient  = lr_receiver.

********** to send Immediately ****************************************
 lr_send_note->set_send_immediately( lc_x ).

********** Calling a method for sending ********************************
        CALL METHOD lr_send_note->send
          EXPORTING
            i_with_error_screen = lc_x
          RECEIVING
            result              = lv_result.


        COMMIT WORK.
**** Exceptions***************************************************************
        CATCH cx_document_bcs INTO lr_bcs_exception.
        lv_message = lr_bcs_exception->get_text( ).
        MESSAGE lv_message TYPE 'S'.

      CATCH cx_send_req_bcs INTO lr_send_exception.
        lv_message = lr_send_exception->get_text( ).
        MESSAGE lv_message TYPE 'S'.
      CATCH cx_address_bcs  INTO lr_addr_exception.
        lv_message = lr_addr_exception->get_text( ).
        MESSAGE lv_message TYPE 'S'.
endtry.

clear: lx_term,
       lv_fnam_lnam,
       lt_msg_text,
       lv_subject,
       lv_sub.
endloop.

else.
 Message s055(zchellam).
endif.
```

Related Content

Common Mistakes need to be taken care on dynamic actions

Basics of dynamic action

Dynamic Action – Procedure.

Basics for preparing mail

For more information, visit the ABAP homepage

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.