SAP Records Management

Integrating a Pushbutton for Navigating to the Record from an Application Transaction

May 4, 2004

Content

1 Introduction

The following tutorial is aimed at consultants who are implementing Records Management. Prerequisites are a working knowledge of Records Management and the ABAP programming language.

Customers often want users to be able to navigate directly from an application transaction to t he appropriate record, without having to open the Records Organizer and find the record first. This tutorial uses a concrete example to show you how to meet this requirement in a user-friendly way.

Take the following scenario: You want the transaction PA20 (*Display HR Master Data*) to include the pushbutton *Display Personnel File* . This pushbutton enables the user to navigate directly to the Records Management personnel file that corresponds to the current personnel number.

This can be realized as follows: We add the pushbutton to the GUI status for transaction PA20 and integrate this function into the screen. This requires us to make a modification.

2 Prerequisites

- You have created a content model for which the unique attribute *Personnel Number* exists. Note: T he standard content model SRM_RCP06 already contains the unique attribute *Personnel Number* (LOIO attribute SRM_PERSONNEL_NO).
- You have created an element type for records and you have entered the new content model for the connection parameter DOCUMENT_CLASS.
- You have created at least one record for this element type and have assigned an existing personnel number to the *Personnel Number* attribute.

3 Extending the GUI Status

Call transaction PA20. To view the program name, screen number, and GUI status of this transaction, choose *System* → *Status*.

Navigate to the GUI status MDIS and add a pushbutton and menu entry.

4 Integrating the Function into the Screen

Navigate to screen 1300 and open the PAI module MODUL FCODE. In the CASE loop for the function code, implement a WHEN query for the function code you assigned to the new pushbutton. Here, call a function module (implemented by you) that executes the function *Display Personnel File* (see below).

4.1 Implementing the Function

Since the user wants to navigate to the pers onal file that matches the current personnel number, use the *Personnel Number* attribute to search for the file. The personnel number is stored in the *rp50g-pernr* field in MODUL FCODE. This results in the importing parameter *i_personnel_no TYPE rp50g -pernr* for the new function module.

First find the correct file with the function module BAPI_RECORD_GETLIST. Use the unique attribute *Personnel Number* as the search attribute. The file that you find is identified uniquely by the *Document Class* and *Object ID* attributes.

To display the file, call the function module SRM_START_FRAMEWORK_RFC. Alternatively, you can use the function module SRM_RECORD_DISPLAY. The function module SRM_START_FRAMEWRORK_RFC

offers you more functions: It is RFC -compliant and, as well as the file, you can also display the Organizer, which offers the user the full range of Records Management features.

You use POIDs to identify the file. You only want to display the current version of the file, which means you can specify 0 for the POID para meters VERSION and VARIANT. Set the POID parameter DOC_ID to the composite string (document class and object ID) that you got when you called BAPI_RECORD_GETLIST. You must specify the RMS ID and SPS ID of the record.

4.2 Example Code

Note: This example has bee n implemented in WebAS 6.20, and is only guaranteed to be valid for this release.

```
FUNCTION zrm_button_display_record.
*"----------------------------------------------------------------------
*"*"Local interface:
*"  IMPORTING
*"     REFERENCE(I_PERSONNEL_NO) TYPE  RP50G-PERNR
*"----------------------------------------------------------------------

  DATA: l_return                TYPE bapiret2,
        lwa_property_selection  TYPE bapipropqy,
        lt_property_selection   TYPE TABLE OF bapipropqy,
        lwa_resulting_list      TYPE bapidoctab,
        lt_resulting_list       TYPE TABLE OF bapidoctab,
        l_lines                 TYPE i,
        l_docid                 TYPE string,
        lt_poid                 TYPE srm_list_poid,
        lwa_poid                TYPE srmpoid.

** set property table
  IF NOT i_personnel_no IS INITIAL.
    lwa_property_selection-propname = 'SRM_PERSONNEL_NO'.
    lwa_property_selection-sign = 'I'.
    lwa_property_selection-option = 'EQ'.
    lwa_property_selection-propval_lo = i_personnel_no.
    APPEND lwa_property_selection TO lt_property_selection.
  ELSE.
    MESSAGE i000(zrm_demo). “Enter a personnel number.
    RETURN.
  ENDIF.

** retrieve correct record
  CALL FUNCTION 'BAPI_RECORD_GETLIST'
    EXPORTING
      rms_id             = 'S_RMS_DEMO'
      sps_id             = 'Z_SRM_SPS_PERSONNELRECORD_DEMO'
    IMPORTING
      return             = l_return
    TABLES
      property_selection = lt_property_selection
      resulting_list     = lt_resulting_list.

** messages when 0 or more than 1 record found
  DESCRIBE TABLE lt_resulting_list LINES l_lines.
  IF l_lines IS INITIAL.
    MESSAGE i001(zrm_demo). “No record found for personnel number.
    RETURN.
  ENDIF.

  IF l_lines > 1.
    MESSAGE i002(zrm_demo). “Multiple records found; first is displayed.
  ENDIF.

** set poid of the record
```

```
  READ TABLE lt_resulting_list INTO lwa_resulting_list INDEX 1.

  CONCATENATE lwa_resulting_list-docclass lwa_resulting_list-objectid
              INTO l_docid
              SEPARATED BY space.

  lwa_poid-id = 'DOC_ID'.
  lwa_poid-value = l_docid.
  INSERT lwa_poid INTO TABLE lt_poid.

  lwa_poid-id = 'VERSION'.
  lwa_poid-value = '0'.
  INSERT lwa_poid INTO TABLE lt_poid.

  lwa_poid-id = 'VARIANT'.
  lwa_poid-value = '0'.
  INSERT lwa_poid INTO TABLE lt_poid.

** display record
  CALL FUNCTION 'SRM_START_FRAMEWORK_RFC'
    EXPORTING
     im_rms_id                     = 'S_RMS_DEMO'
    im_sps_id                     = 'Z_SRM_SPS_PERSONNELRECORD_DEMO'
*     IM_IS_MODEL                    = IF_SRM=>FALSE
     im_sp_poid                    = lt_poid
*     IM_PDIR_ID                     =
      im_activity                   = if_srm_activity_list=>display
     im_option_organizer           = if_srm=>true
     im_option_navigation          = if_srm=>true.
*     im_organizer_sps_id            = 'SRM_SPS_ORGANIZER'
*     im_organizer_rms_id            = 'S_RMS_DEMO'
*     IM_CFW_SPS_ID                  =
*   IMPORTING
*     EX_RESULT_ACTIVITY_STATE       =
*     EX_RESULT_SP_POID              =
*     EX_RESULT_SPS_ID               =
*     EX_RESULT_RMS_ID               =
*     EX_RESULT_IS_MODEL             =

ENDFUNCTION.
```