## Exercise – Create a Module pool program to display Employee details..The screen have 3 command button for data maintainence.

## 1. Display 2. Cancel 3. Break

This exercise covers creation of a Module pool program which involves defining Screens, Events (such as PAI, PBO, POH etc) for data processing, GUI statuses for the screens and attaching the program to a bespoke Transaction.

Module Pool Programs can be created using transction SE80 or SE51.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Module Pool Programming | Name | Attributes | Description |
|  |  | SAPMZEDET<srno> | Module Pool Program | “Display Employee Details” |
|  |  |  |  |  |
|  |  | In this example, we will create a simple module pool program and learn to   * + Design an Input screen to allow user to enter data   + Create an output screen to display the result of processing the input data entered   + Adding a GUI status to aid in navigation and select different processing options   + Attach the module pool program to a Transaction code.   In this example we shall create a simple transaction with an Input Screen and an Output screen.  The input screen will allow user to enter an Employee ID and press a button to display the details of that employee.  The Employee details shall be displayed on the Output screen in a tabular format (to demonstatre design of a Table control within module pool programming). | | |
|  |  | Creating the Module Pool Program | | |
|  |  | Enter the above program name in SE80 and create a module pool program with title “Display Employee Details”.  Save the empty program and activate it.  *Note - Alternatively SE38 can also be used to create the program. For purpose of this exercise we shall be using SE80.* | | |
|  |  |  | | |
|  |  | Input Screen - 100 | | |
|  |  |  | | |
|  |  | Generate Screens using Screen Painter (Screen 0100) | | |
|  |  | Next step is to design the input / output screens for our Module pool program. The screens can be created / maintained using transaction SE51 or through SE80. | | |
|  |  | **Screen Design (Screen 0100) –**  Design the input screen using transaction SE80 –   1. Right click on the name of the main program created above and select Create à Screen option. 2. Enter the screen number value as “0100” and click on the **Ok** button. 3. Specify the Screen attributes as below and save and activate the object-   Description – Employee Details Input screen  Screen Type – Normal Screen  Next Screen – 0200   1. Press the “Layout” button to enter the Screen Painter and design the input screen layout as shown below. A Palette with different screen elements is present on left hand side of the Screen painter.    1. Select the “Box” icon  from the Pallete. Now position the cursor on right side container denoting the screen and draw a box of desired size by holding down the mouse.   Double clicking on the Box will open up the Screen painter Attributes for the screen element (Box in our case).  Specify the below attributes for the Box –  Name – Box1  Text – Selection Criteria   * 1. Select the “Text Field” icon  from the Pallete. Now position the cursor on right side container within the box created earlier and create a Text Field.   Double click the text field to set the sceen attributes as below –  Name – empidlbl  Text – “Employee ID”   * 1. Now select the “I/O Field” icon  from the Pallete. Position the cursor on right side container within the box created earlier, to the right of the Text field and create the I/O field.   Double click the I/O field to set the sceen attributes as below –  Name – zemployee<srno>-empid  From Dictionary – Should be checked  Input - Required  Program subscreen – Input Field – should be checked and Output field and Output Only options should be unchecked.    This is how the designed screen should look like-     1. Save and activate the screen. Now click on the **Test** button  to see how the designed screen looks like. | | |
|  |  | Building the Flow Logic (Screen 0100) | | |
|  |  | From Screen painter toolbar select the Flow logic button to code the backend processing logic for the screen. | | |
|  |  |  | | |
|  |  | **Data Declaration -**  Add the following data declarations at start of the program SAPMZEDET<srno> or add them to a TOP Include.   * declare internal table of type DDIC table zemployee<srno> * declare work area of type DDIC structure zsemployee<srno> * declare variable to hold the value from system field sy-ucomm containing the user command entered by user on the screen   Reference code –  TABLES: zemployee<srno>. DATA: gt\_emp TYPE STANDARD TABLE OF ztemployee<srno>.  DATA: gwa\_emp TYPE zsemployee<srno>. DATA: ok\_code TYPE sy-ucomm. | | |
|  |  |  | | |
|  |  | **Process After Input (PAI) module (Screen 0100)-**   1. Uncomment the “MODULE USER\_COMMAND\_0100.” Statement under “PROCESS AFTER INPUT”. 2. Double click on the USER\_COMMAND\_0100 to create the module. Create module in the main Module pool program itself.   *The USER\_COMMAND\_0100 module will be triggered when user enters values on the screen 0100 and performs some actions such as click a button, select a processing option from the menubar, double click on some field etc. Here the next process step to be performed will be defined e.g. navigate to a different screen, fetch data from database, perform some calculations etc.*  Enter the below piece of code in this Module. The code checks for the action selected by user (menu option or button clicked) and alters program flow accordingly. When user has clicked on the Display button (DISP), the program will call subroutine to fetch employee details and navigate to the screen 200. If the user selects BACK or CANCEL button from menu then the program leaves current screen and takes control back to the initial screen.  CASE ok\_code.  WHEN ‘DISP’.  perform read\_data.  Call screen 200.  CLEAR ok\_code.  WHEN 'CANC'.  CLEAR ok\_code.  SET SCREEN 0. LEAVE SCREEN.  WHEN 'BACK'.  CLEAR ok\_code.  SET SCREEN 0. LEAVE SCREEN.  ENDCASE.   1. Double click on the “read\_data” present in above code and create a new subroutine in the Main program.   Write a Select query to fetch the employee details from table zemployee<srno> into the work area gwa\_emp for the Employee ID which will be entered by user on screen 100. If a record is found, append this into the internal table gt\_emp.  SELECT SINGLE \* FROM zemployee<srno> INTO CORRESPONDING FIELDS OF gwa\_emp  WHERE empid EQ zemployee<srno>-empid.  IF sy-subrc EQ 0.  APPEND gwa\_emp TO gt\_emp.  ENDIF. | | |
|  |  |  | | |
|  |  |  | | |
|  |  | **Process Before Output (PBO) module (Screen 0100)-**   1. Uncomment the “MODULE STATUS\_0100.” Statement under “PROCESS BEFORE OUTPUT”. 2. Double click on the MODULE STATUS\_0100 to create the module. Create module in the main Module pool program itself.   *The* status 100 *module will be triggered when displaying screen 0100 and is used to set the PF status for that screen.*  *Enter below code in this module and save the changes.*  SET PF-STATUS '100'.  SET TITLEBAR '100'. | | |
|  |  | Creating a GUI Status for the screen 100 | | |
|  |  | GUI status is required to enable navigation between screens and to specify different actions that can be performed by user.   1. Right click on the name of the main program in SE80 created above and select Create à GUI Status option. 2. Enter the GUI status value as “100” and click on the **Ok** button. 3. Design the GUI status. Create a custom button to trigger display of the Employee details.   Menubar – create menu by name Display. Code – DISP / TEXT – View Details  Application Toolbar – create item DISP  Function Keys – Specify Function Code DISP against Function key F5. Double click to add an Icon and Icon text to this Function code.   1. Under Function Keys à Standard Toolbar add the Function codes for the standard buttons as shown below –      1. Save and activate the GUI status | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |
|  |  | Output Screen - 200 | | |
|  |  |  | | |
|  |  | Generate Screens using Screen Painter (Screen 0200) | | |
|  |  | **Screen Design (Screen 0200) –**  Design the input screen using transaction SE80 –   1. Right click on the name of the main program created above and select Create à Screen option. 2. Enter the screen number value as “0200” and click on the **Ok** button. 3. Specify the Screen attributes as below and save and activate the object-   Description – Employee Details Input screen  Screen Type – Normal Screen  Next Screen – ‘ 200‘   1. Press the “Layout” button to enter the Screen Painter and design the input screen layout as shown below. A Palette with different screen elements is present on left hand side of the Screen painter.    1. Select the “Text Field” icon  from the Pallete. Now position the cursor on right side container and create a Text Field.   Double click the text field to set the sceen attributes as below –  Name – empdetlbl  Text – “Employee Details”   * 1. Now select the “Table Control” icon  from the Pallete. Position the cursor on right side container, under the Text field and create the Table control.   Double click the Table control to set the sceen attributes as below –  Name – emptab  Line – 5  Column – This should be same as number of fields in the zemployee<srno> table   * 1. Now we need to create the column heading and columns in this Table control. The no of columns and column headings should match the number of fields of the database table zemployee<srno>. Specify the attributes for each of the columns and column headings.      1. To create column headings select the “text field” icon  from the Pallete and create the headings in the header row of table control.      2. To create columns select the “text field” icon  from the Pallete and create the columns in the bosy of table control.     This is how the designed table control should look like-     1. Save and activate the screen. Now click on the **Test** button  to see how the designed screen looks like. | | |
|  |  |  | | |
|  |  | Building the Flow Logic (Screen 0200) | | |
|  |  | From Screen painter toolbar select the Flow logic button to code the backend processing logic for the screen. | | |
|  |  |  | | |
|  |  | **Data Declaration -**  Add the following data declarations at start of the program SAPMZEDET<srno> or add them to a TOP Include.   * Add declaration for the Table contol to be designed on the screen 0200   Reference code -  CONTROLS emptab TYPE TABLEVIEW USING SCREEN '0200'. | | |
|  |  |  | | |
|  |  |  | | |
|  |  | **Process After Input (PAI) module (Screen 0200)-**   1. Uncomment the “MODULE USER\_COMMAND\_0200.” Statement under “PROCESS AFTER INPUT”. 2. Double click on the USER\_COMMAND\_0200 to create the module in the main module pool program.   *The USER\_COMMAND\_0200 module will be triggered when the screen 0200 is displayed which will display the Employee details of the Employee ID selected by user on screen 0100.*  Enter the below piece of code in this Module. The code checks for the action selected by user (menu option or button clicked) and alters program flow accordingly. If the user selects BACK or CANCEL button from menu then the program leaves current screen and takes control back to the initial screen.  CASE ok\_code.  WHEN 'CANC'.  CLEAR ok\_code.  LEAVE TO SCREEN 100.   WHEN 'BACK'.  CLEAR ok\_code.  LEAVE TO SCREEN 100.  ENDCASE.   1. Add an empty LOOP/ENDLOOP on the internal table gt\_emp under the PAI event of flow logic for 200 screen. (The LOOP/ENDLOOP statement is mandatory)   LOOP AT gt\_emp.  ENDLOOP. | | |
|  |  |  | | |
|  |  | **Process Before Output (PBO) module (Screen 0200)-**   1. Uncomment the “MODULE STATUS\_0200.” Statement under “PROCESS BEFORE OUTPUT”. 2. Double click on the MODULE STATUS\_0200 to create the module. Create module in the main Module pool program itself.   *The* status 200 *module will be triggered when displaying screen 0200 and is used to set the PF status for that screen.*  *Enter below code in this module and save the changes.*  SET PF-STATUS '200'.  SET TITLEBAR '200'.   1. Write logic to display the Employee details in a tabular format on the screen 0200 using the Table control created during screen design.   Loop at the internal table gt\_emp to read the employee details and move these into the work area defined (gwa\_emp) specifying the name of table control and specifying the cursor position within the table control.  Reference code -  LOOP AT gt\_emp  INTO gwa\_emp  WITH CONTROL emptab  CURSOR emptab-current\_line.  ENDLOOP. | | |
|  |  | Creating a GUI Status for the screen 200 | | |
|  |  | GUI status is required to enable navigation between screens and to specify different actions that can be performed by user.   1. Right click on the name of the main program in SE80 created above and select Create à GUI Status option. 2. Enter the GUI status value as “200” and click on the **Ok** button. 3. Under Function Keys à Standard Toolbar add the Function codes for the standard buttons as shown below –      1. Save and activate the GUI status | | |
|  |  |  | | |
|  |  | Creating a Transaction code and attaching the module pool program to the Transaction | | |
|  |  | 1. Go to Treansaction SE93 2. Enter Transaction Code as “ZEDET<srno>’ and click on **Create** button. 3. Enter any meaningful description for the transaction and select the option “Program and Screen (dialog transaction) and press **Ok.** 4. Enter the module pool program name **SAPMZEDET<srno>** in Program and careen number as **0100**. 5. Select GUI Support as SAP GUI for Windows, Java and HTML 6. Save the Transaction. | | |
|  |  |  | | |
|  |  | Now execute the transaction “ZEDET<srno>” in the same manner as you would run any standard SAP transaction. | | |