EXERCISE 4

Load Cases Form

Objectives:

- Add a callback for additional analysis information

Exercise 4

Problem Description:

Write a PCL Class called *miracle_loadcase* which presents a form containing a listbox with the list of all loadcases currently defined in the database. In a later exercise we edit the PCL function *miracle_load_aom_data.pcl* to add subordinate forms for additional analysis information. **Selected Loadcases** is one of the forms.

Suggested Exercise Steps:

- Write a new function which displays a form containing a list box and two buttons.
- Compile *miracle_loadcases.pcl*
- Open a database and verify the form displays all the defined loadcases.

Exercise Procedure:

Copy a template file and add the listbox

1. Edit the template function miracle_loadcases.template and complete the missing statements, remove the databox and add an 8 line listbox with widget id **`lcasebx`**. Change the title of the form to something appropriate (like **Selected Load Cases)**.

Add a new function to the class called **refresh().** In the refresh function, load the listbox **`lcasebx`** with all the loadcases in the database using the following lines:

```
ui_item_deleteall( lcasebx )
status = db_get_all_load_case_names()
IF( status != 0 ) THEN
    msg_to_form( status, 4, appcode( status ), 1, 1., "" )
    RETURN status
END IF
WHILE( db_get_next_load_case_name( loadcase_name ) == 0 )
    ui_item_create( lcasebx, "", loadcase_name, TRUE )
END WHILE
```

Call the refresh function from the display function.

```
miracle_loadcases.refresh()
```

Question 1: Why do you think this is necessary?

Add another function which will return the widget id for the listbox for access to the selected Load Cases:

```
FUNCTION get_lc_widget( lbox_id )
/*
 * Return the widget id for the load cases so that
 * they can be used for translation.
 *
 * OUTPUT:
 * lbox_id WIDGET Id of listbox containing selected loadcases
 */

 WIDGET lbox_id

 lbox_id = lcasebx

 END FUNCTION
```

2. Compile *miracle_loadcases.pcl*

Rename the file *miracle_loadcases.template* to **miracle_loadcases.pcl**

Type:

```
>/usr/lib/cpp -P -I/patran/patran3/customization \
  miracle_loadcases.pcl miracle_loadcases.cpp
```

in the xterm. The file miracle_loadcases.cpp should appear as a result.

Type **p3** at the prompt and **<return>.**

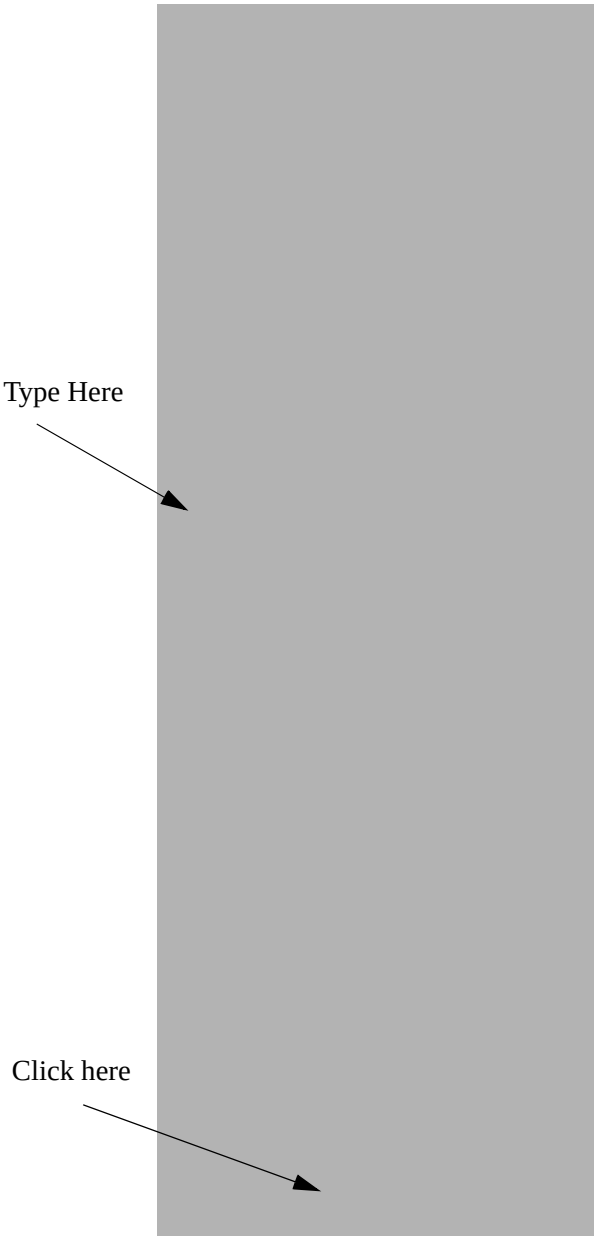
After the main menu and command window appear, type **!!input miracle_loadcases.cpp** in the command line

Resolve any compile errors by editing **miracle_loadcases.pcl** and re-executing the c-preprocessor and the compile command.

3. Verify the function *miracle_loadcases.pcl*.

Open database **exercise_02.db** (from exercise 2). Add three new loadcases called **flight, landing** and **takeoff**.

Select the **Load Cases** radio button on the *Control Panel*. Set the *Action* to **Create**.



In the command line type **ui_exec_function(“miracle_loadcases”,“display”).**
The following should appear:.

Sample Solution

```
/*$$ Use of PCL in creating customized forms/widgets
  *
  * Purpose:
  *    Create a form with one listbox containing all the loadcases
  *     in the database, an ok and a cancel button.
  *    Provide a function to get the widget list box id
  *    to get the selected loadcases.
  *
  * Input:
  *    <None>
  *
  * Output:
  *    <None>
  *
  * Log:
  *
  * Notes:
  *   This is exercise 4 of the PAT305 class.
  *
  */

#include "appforms.p"

CLASS miracle_loadcases

   /*   Variable initialization */

   CLASSWIDE widget form_id,                        @
                    lcasebx

   FUNCTION INIT()

      REAL  y_loc, form_x_location

      form_x_location = FORM_X_LOC-FORM_WID_SML -    @
                        FORM_L_BORDER -              @
                        FORM_R_BORDER
      /*
       * Create the form
       */

      form_id=UI_FORM_CREATE(                                          @
                /*callback*/     "",                                    @
                /*    x    */    ************1 *****************      @
                /*    y    */    ************1 ****************       @
                /* position */    "UL",                                  @
                /*  width   */    ************1 ****************      @
                /*  height  */    ************1 ****************      @
                /*  label   */    "Load Cases ",                         @
                /* iconname */    ""                                     )
```

```
y_loc = FORM_T_MARGIN


/*
 * Create the "Load Case List Box"
 */

lcasebx= ui_listbox_create(                                    @
         /*    parent    */  ************** 2 *************   @
         /*   callback   */  "",                              @
         /*      x       */  UNFRAMED_L_MARGIN,               @
         /*      y       */  ************** 2 *************   @
         /*   width      */  LBOX_WID_SINGLE,                 @
         /*   num_rows   */  8,                               @
         /*    label     */  "Selected Load Cases",           @
         /* selection tp */  ************** 2 *************   @
         /*   sort       */  TRUE)                            @

y_loc += LBOX_8L_HGT_LABOVE + @
         2*INTER_WIDGET_SPACE

/*
 * Create the "ok" button
 */

           ui_button_create(                                   @
            /*    parent   */  form_id,                       @
            /*   callback  */ "ok_cb",                        @
            /*      x      */  BUTTON_HALF_X_LOC1,            @
            /*      y      */  y_loc,                         @
            /*    width    */  BUTTON_WID_HALF,               @
            /*    height   */  0.0,                           @
            /*    label    */  "OK",                          @
            /* labelinside */  TRUE,                          @
            /*  highlight  */  TRUE                           )

/*
 * Create the "Reset" button
 */

             ui_button_create(                                 @
            /*    parent   */  ****************** 3 ********** @
            /*   callback  */  ****************** 3 ********** @
            /*      x      */  BUTTON_HALF_X_LOC2,            @
            /*      y      */  ****************** 3 ********** @
            /*    width    */  BUTTON_WID_HALF,               @
            /*    height   */  ****************** 3 ********** @
            /*    label    */  "RESET",                       @
            /* labelinside */  ****************** 3 ********** @
            /*  highlight  */  ****************** 3 ********** )
```

```
   y_loc  += ********************** 4 ************************

   ui_wid_set( form_id, "HEIGHT", ********** 5 ************ )


END FUNCTION

FUNCTION DISPLAY()

  miracle_loadcases.refresh()
  ui_form_display( "miracle_loadcases")

END FUNCTION

FUNCTION EXIT()

  ui_form_hide( ******** 6 ***************)

END FUNCTION

FUNCTION REFRESH()

  STRING loadcase_name[80]
  INTEGER status

  ui_item_deleteall( lcasebx )
  status = db_get_all_load_case_names()
  IF( status != 0 ) THEN
     msg_to_form( status, 4, appcode( status ), 1, 1., "" )
     RETURN status
  END IF

  WHILE( db_get_next_load_case_name( loadcase_name ) == 0 )
     ui_item_create( lcasebx, "", loadcase_name, TRUE )
  END WHILE

END FUNCTION

FUNCTION get_lc_widget( lbox_id )

   /*
    * Return the widget id for the load cases so that
    * they can be used for translation.
    *
    * OUTPUT:
    *   lbox_id WIDGET  Id of listbox containing selected loadcases
    */

   WIDGET lbox_id

   lbox_id = lcasebx
```

```
   END FUNCTION

   FUNCTION ok_cb()

      ui_form_hide( "miracle_loadcases" )

   END FUNCTION    /* ok_cb end function */

   FUNCTION reset_cb()

     UI_WID_RESTORE (***************** 7 ****************** )
     analysis_main.button_class_closed( "miracle_loadcases" )

   END FUNCTION   /* restore_cb end function */

END CLASS      /* miracle_loadcases class */
```

Solutions

Fill in the blanks:

```
1)/* x */ form_x_location, @
   /* y */ FORM_Y_LOC , @
   /* position */ "UL", @
   /* width */ FORM_WID_SML, @
   /* height */ FORM_HGT_FULL, @

2)/* parent */ form_id, @
   /* callback */ "", @
   /* x */ UNFRAMED_L_MARGIN, @
   /* y */ y_loc, @
   /* width */ LBOX_WID_SINGLE, @
   /* num_rows */ 8, @
   /* label */ "Selected Load Cases", @
   /* selection tp */ "MULTIPLE", @

3)/* parent */ form_id, @
   /* callback */ "ok_cb", @
   /* x */ BUTTON_HALF_X_LOC2, @
   /* y */ y_loc, @
   /* width */ BUTTON_WID_HALF, @
   /* height */ 0.0, @
   /* label */ "RESET", @
   /* labelinside */ TRUE, @
   /* highlight */ FALSE )

4)y_loc += BUTTON_DEFAULT_HGT + FORM_B_MARGIN

5)ui_wid_set( form_id, "HEIGHT", y_loc )

6)ui_form_hide("miracle loadcases")

7)UI_WID_RESTORE ("miracle_loadcases")
```

Answer 1: Placing the refresh function in the display function insures that when the form is displayed, its information will be up-to-date.